
	<h2>Learning Technology</h2> <p>publication of</p> <p>IEEE Computer Society</p> <p>Technical Committee on Learning Technology (TCLT)</p>	
---	--	---

Volume 8 Issue 1/2	ISSN 1438-0625	January/April 2006
--------------------	----------------	--------------------

Special Issue on ADL/SCORM	2
Revisiting the -ilities: Adjusting the Distributed Learning Marketplace, Again?.....	3
SCORM Frameworker: An Approach to Supporting Implementation of Standards-based Distance Learning.....	5
Identifying and Embracing the Trade-offs in Creating SCORM-Conformant SCOs	7
The JADL 2012 Integrated Prototype Architecture.....	9
Integrating the Delta 3D Engine with a SCORM-Managed Learning Environment.....	11
DARWARS: An Architecture for Delivering Experiential Training.....	13
Intelligent Delivery of Sharable Content Objects – An Integrated Solution to Enhance Learning	15
Authoring Tools for ITS with consideration of SCORM Standard	17
NLTM: A Mark-up Language for Transformation-based Question Generation	20
<i>Regular article: Podcasting in Law School: Technical Considerations in Building a Podcast for Use in Higher Education.....</i>	23

Special Issue on ADL/SCORM

The Advanced Distributed Learning Initiative held its second Workshop of SCORM Sequencing and Navigation at the University of Memphis (Memphis, Tennessee) on 11 November 2005. This workshop was held in conjunction with two sequential conferences on Authoring Tools for Advanced Learning Environments and Industry Solutions. (Details available at <http://www.elearningsummit.org>.) The papers presented here are from that Workshop.

The preponderance of military technical training takes the form of traditional, didactic instruction in declarative and procedural knowledge. The use cases chosen to focus the development of SCORM naturally focused on that type of instruction. Other instructional forms have begun to generate considerable interest since the early days of SCORM development and the Workshop sought to focus on these issues.

The first paper, by Avron Barr, takes a broad look at the distance-learning horizon and finds challenges to be faced. The Haynes (et al) paper proposes a solution to what many find the daunting complexity of SCORM for traditional instructional designers and developers. Bill Blackmon, from the Learning Systems Architecture Lab at Carnegie-Mellon University, brings to bear several years of experience teaching instructional developers how to modify their practices to build SCORM-conforming content and suggests there is not a single, correct method but a continuum of possibilities. Chris Bray, from the Joint ADL Co-Lab in Orlando, FL, offers up one notion of an architectural possibility for integrating various types of instructional delivery options. The Smith-McDowell paper describes very successful work integrating an open-source game engine with SCORM-conforming Learning Management Systems. Bruce Roberts describes experience with building DoD-sponsored programs of experiential learning. The final three papers all come from the host of the Workshop, the University of Memphis, specifically the ADL Workforce Co-Laboratory and the Institute for Intelligent Systems. Each of these papers address different aspects of developing intelligent systems and having them communicate with SCORM-conforming systems.

Eric J. Roberts, Ph. D
Chief Scientist for Learning
Advanced Distributed Learning Initiative

Revisiting the -ilities: Adjusting the Distributed Learning Marketplace, Again?

The SCORM Reference Model, developed and evangelized by the Advanced Distributed Learning Initiative, has had a profound impact on a large part of the eLearning marketplace. In particular, for managed, browser-based training, SCORM's establishment of a set of packaging and runtime standards created an open market for training content. Courses were no longer tied to a proprietary learning management system (LMS). Instruction management and instructional design could evolve independently.

SCORM also created the possibility of sharing content and creating new content by assembling existing elements. Eventually, this vision could dramatically reduce the cost of training development. The ADL is just now rolling out another foundational element of this vision of reuse: a content registration system that will allow teachers and content developers to find and preview content that might be useful to them. This central library for learning materials might reside across dozens or even hundreds of repositories.

SCORM's rapid acceptance has been surprising. It has been adopted by dozens of LMS vendors, and will be mandated as a requirement for all DL content created by and for the US DoD. Moreover, it has already found broad adoption by major corporations, and even by some entire countries as their training content standard. Key to SCORM's success is the realization that customers and vendors both win if market inefficiencies and fragmentation are removed. From all indications, the impact of SCORM will be global and long-lasting. But standards must adjust to the times, and there are some problems on the horizon.

SCORM Clouds

There are many issues that the SCORM community is dealing with and that limit or interfere with broader adoption of the reference model. (SCORM is a collection of standards, which are in part based on standards established by other organizations.) Here's a list of some of the well-known issues.

Success

The biggest problem SCORM faces at this point in time is its own success. Escalating global demand for support, better tools for a wider and wider range of content developers, vendor certification, training, and so on have outstripped available resources, especially as the ADL has focused on developing and deploying their ADL-R content registry – the next critical element to their vision for plug-and-play courseware assembly. The search is on for an appropriate steward organization for SCORM, one which can involve the international SCORM community in the continued maintenance and evolution of the standards.

Reuse

When fully developed, the ADL-R content registry will make it possible for students, teachers, and instructional designers to find content that might be relevant to their purposes. That doesn't mean anyone will be motivated to do so. Current "time and materials" funding of DoD contractors removes the financial incentive, and it is unclear whether the DoD's Instruction 1322, which mandates reuse, will have any means of enforcement. Reuse requires organizational change. Content must be authored and maintained with the intent of it being reused. Incentives must be changed. This kind of change will take time, and may take root first in other countries, or in corporations, where incentives are more easily changed.

S1000D

One of the current activities of the ADL is an exploration of the possibility of integrating the S1000D standard for technical manuals with the SCORM reference model for training materials. On-the-job performance support has always been a part of the ADL's vision for distributed learning, and in many kinds of jobs, performance support takes the form of technical manuals. But technical manuals and training materials have very different life cycles, different user motivations, and different expectations of interactivity (training requires assessment and

feedback). There are technical issues too, including the fact that S1000D content is not currently read in a browser but rather using proprietary reader software.

Changing Training Requirements in the DoD

Computer savvy recruits expect a rather more sophisticated type of interaction than is typically delivered in browser-based training. The training audience itself is now composed not only of active duty personnel, but Reserves, National Guard, coalition partners, other government agencies, and even civilian contractors. The missions are more diverse and often require extensive training in material unrelated to combat. And the life cycle of relevant knowledge continues to accelerate. As one general put it, Al Qaida is the fastest learning organization in the world. These changes require innovation in training, which in turn puts pressure on SCORM. No doubt, requirements in other major SCORM communities are also evolving.

Intelligent Tutoring Systems

Despite a long history of experiments that never find commercial application, the use of AI technologies in the delivery of on-line training still holds great promise. AI components can coach, remediate and review. They converse with the student and keep an elaborate model of the student's history, knowledge, and learning styles. Unfortunately, these experimental systems are typically monolithic, with all student activity, keystroke by keystroke, monitored by the AI components. To some extent, this is antithetical to SCORM's model of SCOs (which reflect discrete learning objectives) launched individually by the LMS. It is critical to automate the instructor's role in online training, in order to reduce the cost of high-quality training. Experiments continue, but it might be that these monolithic systems find their first deployment in non-managed instructional settings.

Experiential Learning

A similar situation exists with one of the most important new categories of computer-based training, the use of "virtual training environments" to give students relevant practice and feedback. Simulators have been in widespread use in a variety of military and civilian training applications. But new technology, including PC-based simulations and games, on-line games, multiplayer games, and persistent virtual worlds offer great promise for high-quality training in a variety of domains. The question is, how will these inexpensive, distributed virtual training environments, integrate with managed instructional environments? Multiple students, with different learning objectives, who take different roles in a scenario, and who might together have team performance goals, present one of the most difficult problems. Also, these systems are not browser-based. There's a lot of active investigation of this area, including DARPA's DARWARS project, several prototype projects at the Joint ADL Co-Lab, and a joint effort by SISO and the IEEE Learning Technology Standard Committee.

In addition to these pedagogy-related developments, software, network, and computing technologies continue to evolve. Wireless devices, streaming media, service-oriented architecture, semantic mediation, on-line collaboration and community, intelligent agents, and speech-based computing all have great potential, and will impact the managed instruction framework in ways we can't predict. The question is, does the ADL, and the DoD as the world's biggest training organization, again have a role in shaping the marketplace. And if it does, is it possible to continue to find solutions as brilliant and successful as SCORM.

Avron Barr
Aldo Ventures, Inc.

SCORM Frameworker: An Approach to Supporting Implementation of Standards-based Distance Learning

SCORM Frameworker (SFW) is a tool developed by Intelligent Automation, Inc. under a contract with the Joint ADL Co-lab, to support course designers and developers in the process of building courseware conforming to the SCORM specification for distributed distance learning. Specifically, SFW supports intelligent course assembly and creation of metadata. Without SFW, the process currently requires individuals with programming expertise to perform these tasks. However, it is our view that course assembly and describing the instructional properties of the courseware in metadata could best be performed by individuals with instructional, rather than programming expertise. Since most commonly individuals are trained in one domain or the other, it is our purpose in developing SFW to enable development of SCORM conforming training that is both instructionally sound and well developed technically.

SFW accomplishes the goal described above by using case-based reasoning (CBR) to match the metadata requirements of a courseware developer's current task to one or more 'cases' of similar courseware developed previously, and use that information to recommend the metadata that needs to be provided for the current instance. The specific form of CBR used for this purpose is 'conversational,' in that the user interacts with the CBR engine by both posing and answering questions, where the user's responses result in an increasingly focused set of recommendations or 'tips' for metadata creation. For example, the courseware developer is asked questions about the course profile, the content types, the target learning management system(s), the target repository, and relevant business rules for use and re-use of the content, and other items that will help in determining an efficient but complete set of metadata for the user's goal. This set of metadata will be optimized for search and discovery of the content, according to applicable business rules.

The distributed distance learning enterprise envisioned using SCORM, provides metadata that can support this effort in a number of ways, including the following:

1. storing and finding appropriate content using repository and/or search engines where relevant metadata might include items such as title, description, keywords;
2. improved instructor support through the content profile, which might include items that specify suitability for different types of students or prerequisites;
3. improved support for business use of content, such as content maintenance or third-party use, where metadata might consist of items such as copyright, author, and version information;
4. improved efficiency in identifying potential content for specific settings where technical requirements for content delivery may be either a limiting or enabling factor;
5. support for learning program management through use of outcome data, where metadata might describe the author's intent for providing credit or certification upon successful completion of the content.

To support the design of SFW to promote its 'fit' in the real world of designing and developing distance learning courseware, IAI conducted a series of focused interviews with individuals who are currently employed as instructional system architects, designers and developers. The interviewees work in a variety of settings, and do not know one another. The implication of these interviews as a whole provide insight into the current state of implementation of standards-based distributed distance learning and how tools like SFW can provide the most support to the implementation effort. The following section summarizes the findings of these interviews.

1. We found that most instances of SCORM implementation were superficial, and done to fulfill a requirement. As such, minimum effort was put into content aggregation and creation of metadata that performed any functions other than allowing the content to run in the SCORM RTE (run time environment) and communicate with a specific learning management system. As such SCORM was fulfilled in letter rather than in spirit.
2. There are specific elements of SCORM that designers/developers find extremely difficult and time consuming.
 - a) Writing instruction as SCO's takes more time than writing instruction in the usual way.
 - b) At the current time, searching for, finding and reviewing existing SCO's takes more time than creating new ones.
 - c) The rules for Simple Sequencing and Navigation are exceedingly complex, difficult to understand, and impossible to implement for a non-programmer.
 - d)

4. Instructional designers and developers believe that tools to support SCORM implementation should do the following:
 - a) Reduce work and errors in metadata creation;
 - b) Take information they already consider, and put that information into SCORM;
 - c) Identify the needed set of metadata attributes, and automate the process of filling it in to the fullest extent possible;
 - d) Provide the rationale and explanations for the SCORM requirements.
 - e)
4. SCORM is important now to a limited audience. Within DoD, there is an inconsistency between what is viewed as important at the program level and at the level of the specific services' programs. At the highest levels, there is strong and growing support for a standards-based approach, including requirements for conformance to SCORM. Nevertheless, at the level of program acquisitions, SCORM is viewed as an additional burden, as an unfunded requirement, unnecessary, and a problem causing price increases and delays.
5. DoD continues its policy of making most awards for content development to the lowest bidder, without regard for the quality of implementation of SCORM, if it is required at all. This policy is viewed by vendors as indicating DoD's lack of commitment to SCORM and a relatively short "life expectancy" for the SCORM requirement. The vendors are therefore unwilling to put their resources into developing expertise, quality tools, and development procedures that lead to improved effectiveness and efficiency in providing SCORM-conforming courseware to their customers.
6. Courseware is largely developed by individuals with little or no training in instruction, learning, or pedagogy. The majority of the work is performed by programmers, subject matter experts, and junior-level "developers" who are trained in using a given set of tools. In fact, more money is paid to artists, videographers and animators, than to most content developers. Individuals with expertise and experience in instructional system design, psychology, or pedagogy are mostly limited to program management functions, with little direct input into the content of instruction.

Based on the information obtained in the interviews, SFW is designed to address these issues by working with a range of metadata standards including SCORM, GEM, Dublin Core, S1000D, and others. Using SFW should improve task efficiency and usability for the developer, allow for cross-team development of content and metadata as well as organizational customization of metadata, and support direct upload to LMSs and CORDRA repositories. SFW also provides a method to verify (both for the developer and the customer) that the resulting content is discoverable within its repository.

While SCORM Frameworker is a tool to support metadata creation, it is our view that it is really an attempt to provide technical support for the ADL enterprise, in ways that may assist the efforts implementation on a larger scale. Some of the features being designed for SFW were not envisioned initially as a part of the tool, but are now considered as crucial to achieving the purpose of this class of tools, based on our interview data. Further development and use of SFW will show whether these objectives can or will be achieved.

Jacqueline A. Haynes, Ph. D.

Intelligent Automation, Incorporated
Rockville, Maryland, USA

Daniel Fowler

Intelligent Automation, Incorporated
Rockville, Maryland, USA

David Ryan-Jones, Ph. D.

Naval Air Warfare Center Training Systems Division
Orlando, Florida, USA

Identifying and Embracing the Trade-offs in Creating SCORM-Conformant SCOs

“Is my content SCORM-conformant?” is the most common question I hear as an instructor of SCORM workshops. Although it would appear that this is a yes or no question, the critical question to ask is actually, “How can I use SCORM to attain the level of interoperability I require for my learners?”

Through a series of workshops on SCORM taught at the Learning Systems Architecture Lab, we have found that there is conflict between what people want to do with their content and what they think the label “SCORM-conformant” requires. For the first two years of the workshop, many times when a student asked “Can I make my content SCORM-conformant and do ___?”, the instructor with the instructional design background would say, “No,” while the instructor with the computer engineering background would say, “Yes, but...”

It took years of negotiation for the two instructors to figure out a single answer to give the students. The answer involves asking the student a now-obvious question: “What is it you’re trying to do?” Or at a more technical level, “SCORM enables interoperability, but does not guarantee it. How generic do you want your content to be and still meet your goals?”

Instructional designers are familiar with the trade-offs between contextualization and reusability – the more context you add to an object, the less reusable it is. For example, an eye wash procedure can be created about the general principles of safety and how to safely wash your eyes (high on reusability; low on context), or it can be created to include where the eye wash stations are in the learner’s particular building (high on context; low on reusability).

Likewise, SCOs must be engineered with trade-offs between contextualization and reusability. SCOs can be SCORM-conformant and yet work in only one LMS or they can work in every LMS. This is possible because SCORM is not (and should not be) an all-encompassing specification. SCORM enables interoperability by specifying how engineers create SCOs to communicate with the LMS, but SCORM cannot prevent other types of interactions and dependencies from happening between the SCO and LMS.

What this means to the instructional designer is that there isn’t necessarily a simplistic answer to the question, “Is what I want to do SCORM-conformant?”

Together, instructional designers and engineers must analyze the needs of the learner and deploy SCOs that have the right mix of contextualization and reusability, both from an instructional perspective and a technical perspective. It simply makes no sense to make a SCO work in every LMS if it fails to meet the needs of your learners.

Customizing SCORM content

One question that often comes up is, “I thought SCORM guaranteed interoperability. Why is it possible that SCORM content be customized so that it isn’t interoperable?”

SCORM enables interoperability of content by providing a standard API and data model for a SCO to communicate with an LMS. This enables the SCO to ask the LMS for the learner’s name or to report the learner’s score. A SCO that uses the API and the data model will work in any LMS.

However, there are many ways that a SCO could be engineered to not be interoperable. For example:

a SCO could be written for Flash 8 and deployed on computers that do not have Flash 8 installed or that are on networks that prevent Flash objects from being delivered.

A SCO could be written that requires a specific server plug-in to operate. For example, a SCO that includes Quicktime streaming media will only work on LMSs that have the Quicktime streaming server.

Any of these SCOs can be SCORM-conformant, but may not work across all LMSs.

The Customization continuum

A SCO can fall anywhere along a continuum between complete customization and complete generalization. At the far end of customization, a SCO could be developed that only works on one installation of an LMS. At the other end, a SCO could be developed to work on any LMS and run on any web browser on any operating system; at this end, SCOs use only the most generic web technologies (simple HTML and images) and the safest use of the data model elements. Most SCOs will fall in between the two extremes.

The SCORM testability continuum

A SCO can fall anywhere along a continuum that has three specific ranges of SCORM-conformance:

Range 1: not testable because it falls outside the realm of what SCO specifies; these include any SCO that does not use the API, including PDFs, plain HTML files, stand-alone Flash movies, etc.

Range 2: fully testable by the SCORM Test Suite and it follows all the implicit assumptions of the SCORM specification;

Range 3: not testable because it uses the API and data model in ways that violate the spirit of SCORM.

When engineering SCOs, special conditions may require a SCO to fall in Range 3 to meet the needs of the learner, but SCOs that fall in Range 2 are more likely to withstand any differences between LMSs. Many SCOs may not require any communication with the API and fall within Range 1 and this is still SCORM-conformant; many instructional designers fear that they must create SCOs that use the API, but this is not true.

The term “SCO” is used throughout this paper as any content launched by an LMS. Technically, SCORM defines a “SCO” as content that communicates with an LMS through the SCORM API and data model, while an “asset” is any other content that is launched but does not communicate with the LMS.

William H Blackmon
Learning Systems Architecture Lab
Carnegie Mellon University
lsal@lsal.org

The JADL 2012 Integrated Prototype Architecture

Following the completion of several successful prototypes, the Joint ADL Co-Lab has identified the integration of simulation-based learning experiences and SCORM environments as a primary area of research. The results of our past prototypes have been used to prepare this phase of investigation, which will take a more integrated, less piecemeal approach.

We have developed the “JADL 2012 Integrated Prototype Architecture” (IPA). The IPA is a straw-man model to help guide our investigation; it is *NOT* a representation of what future policies or revisions to SCORM might require of SCORM-conformant tools, systems, and content. The JADL 2012 IPA is simply a representation of *one possible* solution to *one possible* problem space. We intend to use the IPA as a communication tool and sounding board when conducting research. The IPA addresses simulation integration by proposing a system-of-systems approach consisting of small set of infrastructure services and specifications.

The primary component of the IPA is a “Distributed Training Event Coordination Service” (DTECS). The DTECS component works in conjunction with an LMS which has been modified to recognize a new type of SCORM object called a “Lightweight Scenario Format” (LSF) file. LSF is *not* intended to be a common configuration file format with the ability to initialize all simulations; an LSF file cannot be used by itself to initialize *any* simulation.

LSF files are represented using an XML-based syntax and can easily be created, modified, and reused within and between courses. LSFs can be designed independently of any particular training system, registered in the ADL Registry (the ADL-R), and stored in globally-accessible content repositories.

LSFs contain high-level information that can describe key elements of a scenario that will be common to all training systems which are capable of executing the scenario. These elements are similar to those which are described by the DARWARS OCM model – Objectives, Conditions, and Measures. Objectives can be correlated with IMS Simple Sequencing objectives. Conditions can describe startup factors like equipment types, locations, environmental factors, etc. Measures describe assessable factors which can be used to provide values to objectives. The LSF file also includes scenario roles and indicates which roles are required, which are optional, and which can be filled by intelligent agents if available.

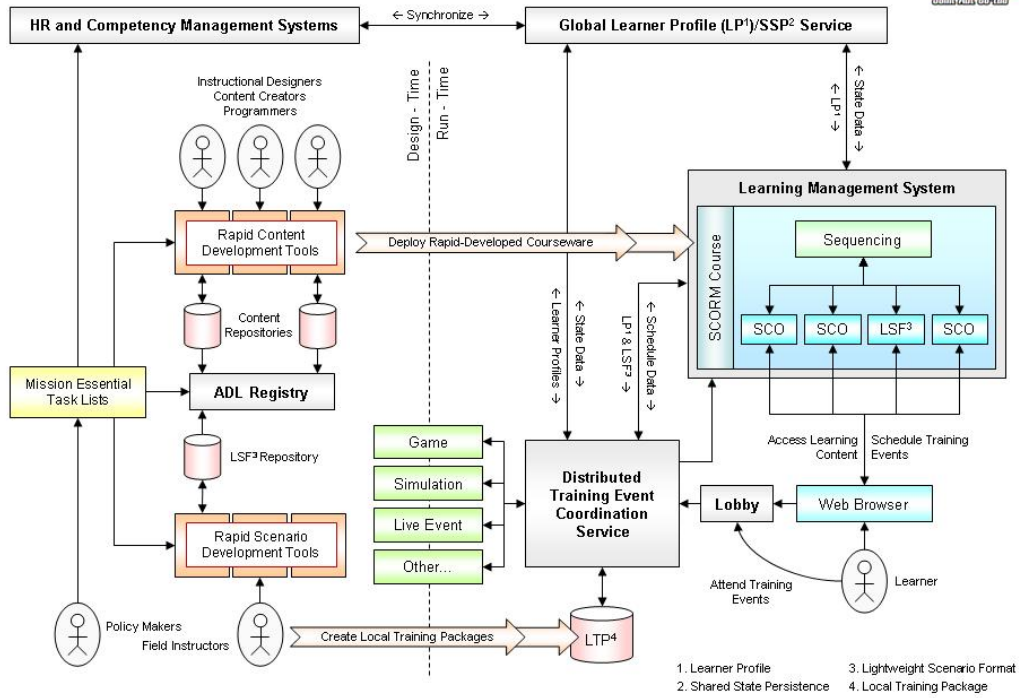
To initialize a training system, some proprietary configuration information is almost always required. In IPA terminology, this is called a Local Training Package (LTP). It is the LTP file (*not* LSF) which contains configuration data for a specific training system. LTPs are built from LSFs, and specify how a particular training system needs to be configured to deliver a scenario that supports the objectives, conditions, and measures that are specified in the LSF.

In a fully implemented IPA instance, a new type of LMS (called a “Brokering LMS”) has been defined which assists users in the process of locating and initializing the best, most accessible training system that is available to them at any given moment. This process starts with embedding an LSF file within a piece of courseware. When a brokering LMS encounters an LSF file, it locates and searches one or more DTECS implementations to see if each contains an LTP file matching the LSF file. After merging the results, the LMS will auto-generate a web page which lists the learner’s options. A learner may be able to execute the simulation immediately, or may be required to schedule a training event if additional roles are needed.

When the user is ready to begin executing a simulation, the LMS transfers them to a DTECS-provided “lobby.” This is essentially a web page which offers a brief idle period during which simulation participants can gather. The lobby provides learners with a smooth transition from an asynchronous to a synchronous learning environment. When all roles have been filled, the DTECS initializes the simulation and waits for measures to be reported. These measures are passed back to the originating LMS where they are used assign values to Simple Sequencing Objectives.

Overall, the use of LSF and LTP files in conjunction with Brokering Learning Management Systems and any number of standards-based DTECS implementations will allow for a model which has excellent characteristics for supporting the integration of simulation-based learning experiences with SCORM while adhering to the ADL “ilities.”

JADL 2012 Integrated Prototype Architecture (IPA) – Baseline Model



Chris Bray
 Joint ADL Co-Lab
<http://www.jointadlcolab.org>

Integrating the Delta 3D Engine with a SCORM-Managed Learning Environment

The United States Navy is engaged in an enterprise-wide transformation of how it trains¹. One key component of this transformation is the development of the Navy's Integrated Learning Environment (ILE). In December of 2002, the Naval Education and Training Command (NETC) established the ILE to help launch, track and manage almost 4000 E-Learning courses for approximately 1.2 million active-duty Sailors, Marines, Department of the Navy civilians, Reservists, retirees and family members enrolled in the Defense Enrollment Eligibility Reporting System (DEERS)². This initiative uses a variety of instructional development strategies to meet the diverse requirements of the Navy's workforce and assures content which is relevant, current accurate and engaging. It combines support tools for developing and distributing electronic course materials, and managing student and curriculum records, with standards for classifying content, formatting files, and interoperability among other systems.

Another key component of this transformation is the development of the Open Source Delta3D engine. Decades of cognitive science research have shown that people perform better after instruction when they have learned information in the context of doing³. This project, also developed through NETC, is an Open Source gaming and simulation engine that has great potential to support adaptive learning by placing the learner in a "real-world" environment and allowing the student to learn in context. It is this contextual experience that theoretically enables the learner to create their own constructs that can be applied to new unfamiliar situations. While simulations offer the opportunity to undergo informative interactive experiences, they do not, by themselves, constitute training or instruction. Assessment is an important element in the teaching and learning process.

This article will take a broad look at the underlying technology that is currently being developed to allow the Navy's ILE to launch, track and manage training simulations that have been developed using the Delta3D engine. In particular, it will discuss the architecture of an application being developed to demonstrate this capability. In order for the Delta3D engine to meet these requirements, technology needs to be developed to allow in-depth assessments of student performance against training objectives to identify student deficiencies and provide feedback to students and instructors. The challenge, therefore, is to find an efficient way to:

- Enable the Delta3D engine to deliver interactive, multi-media instruction through the Navy's Integrated Learning Environment,
- Provide practical feedback and hands-on experience in situations that can not easily be practiced using real scenarios, and
- Deliver complex information to a diverse, geographically dispersed audience in a short period of time when the information itself will be constantly changing.

The SCORM Run-Time Environment specification handles the launch of learning content, communication between content and an LMS, data transfer, and error handling. The RTE was designed to work with web pages and as such, it has no inherent capabilities to directly launch or communicate with a native application session running on the client machine. However, recent "Smart Client" technologies have made it possible to blur the distinction between pure web-based applications and traditional, locally installed applications. Two competing technologies that enable the development of smart clients are "Click-Once" from Microsoft and "Java Web Start" from Sun. Both of these technologies allow application components to be stored on a web server and delivered to users on demand.

Due to the cross-platform requirements of the NETC Delta3D project, the demonstration application uses Java Web Start technology. The launch page is a standard html document that handles the installation, launch and communications with the Delta3D simulation running on the client machine. This page also contains various scripts and a java applet that handle tasks such as communication to the LMS and detection of the Java Runtime Environment on the client machine.

Once the simulation application has been launched on the client, there must be a mechanism to pass data to and from the LMS via the RTE's standard API functions; this data may represent launch parameters going to the simulation and/or assessment data regarding learner performance going from the simulation to the LMS. Ultimately, this communication pipeline needs to be developed using lower-level socket communication protocols provided by the languages that are used (i.e. C#, Java). The proof-of-concept application mentioned earlier demonstrates the following method of creating this message pipeline:

- The native simulation code is wrapped by a Java application and sends simulation “event messages” to it via the Java Native Interface (JNI).
- The Java application processes these events into relevant assessment data
- The Java application communicates it via a TCP/IP socket to a Java applet running on the web page
- The Java applet calls ECMAScript (JavaScript) functions on the web page in order to communicate with the LMS.

In order for student performance to be tracked inside a simulation, the data communicated to the LMS must be confined to the assessment data model that SCORM provides. Generally speaking, each SCO can contain a number of “objectives”, and learner progress toward each objective can be tracked according to basic data values such as completion status, success status, and score. (For a more detailed discussion of the assessment data model defined by SCORM, refer to the Run-Time Environment documentation at <http://www.adlnet.org/downloads/70.cfm>). Thus, while a simulation scenario 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.

In summary, the assessment process is broken into logical components. The simulation raises simulation events as the user interacts with it. The assessment module listens to these simulation events, and uses them to track a learner’s progress toward completion of defined tasks. An assessment data model is used to combine events and tasks in a hierarchical manner to form more complex tasks, which ultimately culminate in one or more objectives. As a learner completes the training objectives, the assessment component communicates this information to the LMS through an embedded Java applet.

References

Maskell, R. “Taking Learning to the Next Level”, Published June 11, 2003 in Military Training Technology, Volume 8, Issue: 2

Integrated Learning Environment Website, Overview, Navy Personnel Development Command, Jan 25, 2005

Kelly, H., Van Blackwood, Roper, M., Higgins, G., Klein, G., Tyler, J., Fletcher, D., Jenkins, H., Chisolm, A. & Squire, K. “Training Technology against Terror: Using Advanced Technology to Prepare America’s Emergency Medical Personnel and First Responders for a Weapon of Mass Destruction.

Brent Smith

Engineering and computer Simulations
Orlando, Florida, USA
<http://www.ecsorl.com>

Perry McDowell

U.S. Naval Postgraduate School

DARWARS: An Architecture for Delivering Experiential Training

Initiated in 2003, the DARWARS Project is part of DARPA's *Training Superiority Program*. The program's goal is to transform military training by introducing a new kind of cognitive training experience for units and individuals. These low-cost, mobile, on-line, simulation-based training systems take advantage of the ubiquitous presence of the PC and of new software technologies, including multi-player games, virtual worlds, off-the-shelf PC simulations, intelligent agents, and on-line communities. *Experiential* lightweight simulation systems create immersive training environments for a wide range of domains, from language training to battlefield strategy, and often offer automated feedback and assessment for each participant.

BBN Technologies, along with Aptima, Inc and MÄK Technologies, is developing an architectural framework, the *DARWARS Core*, which includes a broad set of web services, tools, and system interface definitions to facilitate the development and delivery of experiential training. *Training Packages* encapsulate the information needed to coordinate and launch distributed training events involving multiple trainees interacting with multiple training systems. The DARWARS Core provides services for scheduling training events, assigning participants to particular roles, marshalling resources needed by a training system, checking in participants, and remotely launching training systems and other required computational resources (e.g., servers or databases). It also provides services for recording and accessing trainee profiles and other data collected during a training session.

Training Packages are linked in an explicit representation of objectives, conditions and measures, which also provides the structure for recording and reviewing trainee performance:

Objectives are what the trainee will need to be able to do or know after participating in a training event. An objective may be the description of a task or activity, and can be hierarchical. For example, the objective: "Operate a vehicle in a convoy" may have nested sub-objectives, such as "Observe, assess, prepare for an ambush" and "React to contact." A particular training event may have multiple objectives. Trainees can be matched to an appropriate training experience (e.g., taking on a particular role in a specific scenario) based on these objectives.

Conditions describe the specific configuration of parameters within a training system that have implications for learning. For instance, within a flight simulator, the weather conditions and number of players may vary. Conditions also include the fixed parameters of a training system, such as the air platform (F-18, helicopter, commercial aircraft). These fixed and changing parameters constrain the types of objectives that can be addressed, and are consequently fundamental to training. For example, pilots must acquire knowledge of flying under a variety of conditions, such as adverse weather and low visibility, before they become fully competent.

Measures are behaviorally anchored, observable actions within the training system that can be calculated and linked to a particular objective (or sub-objective) to demonstrate mastery of a task (or competency, knowledge, skill) in a particular set of conditions. In this way, the objective is operationalized; the objective being met is defined by the measures employed. Thus, the choice of measures is important to both instructors and trainees. For a training system to be maximally effective, the measures should clearly relate actions to objectives.

In addition to providing a technological architecture for integrating training systems into an instructionally coherent and navigable training landscape, DARWARS provides a general pedagogical framework for simulation-based instruction – linking training objectives to students' experiences in simulations and virtual worlds. DARWARS *trainees* are able to find what training experiences are available to them; choose ones that match their needs; review their performance in specific training experiences; track their progress over time; and estimate their readiness for the next training experience or real-world event. DARWARS *trainers* will schedule training sessions based on the training objectives of individuals and teams. They will have access to performance information to enable the development of new curricula; tools to provide individualized real-time coaching; and applications that will allow them to conduct effective After Action Reviews (AAR). DARWARS supports all of these features in its single, unifying framework of specified objectives, conditions, and measures.

The DARWARS Core can be viewed as an Learning Management System (LMS) for experience-based training, a type of training that demands more from a LMS than is currently covered by the existing SCORM standard or its many realizations. An experience in a shared virtual world is hard to characterize as a learning object and user interactions with the virtual world can be very rich. Several learners may be involved in the same exercise, each

with his or her own role, learning objectives and view of the situation. Teams, not just individuals, have objectives, which must be monitored and tracked. It is impossible to accurately predict or control what will happen during a training session, especially one involving teams: unexpected events and actions will occur; some expected events will not. Moreover, the actual delivery of the training involves considerable machinery—in addition to the simulation itself—to manage the scheduling, lobbying, launching, monitoring and AAR, which is a key component of this type of training environment.

To accommodate these many demands, the DARWARS Core is envisioned as a companion to existing LMSs, not a replacement. Rather than attempting to embed an experiential training event within a SCO, the SCO might refer instead to an objective, a training package, or a scheduled event, and delegate responsibility for managing this element to a local instance of the DARWARS Core. This linkage—another web service provided by the Core—would also mediate the exchange of information between the LMS and the unfolding event.

A goal of DARWARS is to hasten the move from isolated expert-dependent training systems with no clear stated objectives to scaleable simulation-based training systems characterized and linked through explicit objectives. A structure of inter-linked objectives, conditions, and measures facilitates this move, and makes possible a host of activities that should improve training effectiveness and readiness. At the same time, it offers a mechanism for extending the types of training offered within a SCORM compliant environment to include richer game-based training systems.

Bruce Roberts
BBN Technologies
Cambridge, Massachusetts, USA

Intelligent Delivery of Sharable Content Objects – An Integrated Solution to Enhance Learning

Standards in eLearning

The Sharable Content Object Reference Model (SCORM), was developed to address the need for interoperability of learning objects between learning management systems. Only the most recent version of SCORM (SCORM 2004) is capable of delivering dynamic content that is based on learner performance. It has been argued that SCORM 2004 can be used to mimic the performance of some ITSs that have achieved learning gains that go beyond web-based training.

Our goal in the paper is to outline a method that makes use of existing flat structured page turning content (typical SCORM 1.2 or earlier contents) and enhance it to achieve better learning for students.

Intelligent delivery of SCOs to create effective learning systems

We believe the prior content of early and present day Sharable Content Objects (SCOs) can be leveraged in smarter learning management systems that enhance the delivery of this content and improve the amount of knowledge that is acquired. Utilizing existing SCOs in an intelligent system can enhance learning much like an ITS. There are some basic requirements for this proposed approach:

1. **No SCOs Conversion.** Instead of converting SCOs in SCORM 1.2 courses to SCOs that are compatible with SCORM 2004 courses (where possible adaptive learning can be achieved), we propose to use SCOs in their original form and deliver them in an enhanced environment.
2. **No new LMS will be created.** We do not propose the building of a new LMS that will have the enhanced delivery features. We instead propose to provide utilities that can be used by existing SCORM compatible LMSs.
3. **No new information will be added to SCOs.** We will only propose the use of existing metadata for the SCOs and available text information (content) in the raw data of a SCO.

We believe that the most constructive approach that satisfies the above three basic requirements is to 1) make the best use of existing metadata and raw data of SCOs and 2) use computational linguistics tools to help the learner understand content of the SCO. We term this process ‘content enhancement.’ We propose that the implementation of such enhancements be based on learning theories developed in cognitive psychology.

Our proposed enhancements include:

1. The learner will have immediate access to supplementary material links to previously learned content, schemas, related topics.
2. The learner will be asked questions for testing their learning and for revising the material, and answers to questions about the material that they may have.
3. The learner will be provided summaries of current content based on metadata and available text information of the SCO.
- 4.

Such content enhancement would mean that for one page of content, the learner will spend significantly more time reading, thinking, and cognitively digesting the content hopefully resulting in significant learning gains – much like a human tutor inspires cognitive disequilibrium requiring more thought and more digestion.

Making the most out of metadata

The advantage of developing learning delivery systems that operate on SCORM objects is that SCORM has become the most widely used international standard for eLearning. Every SCORM object is packaged in metadata that provides information about the learning object and its content, such as topic area, subject, and so on. This metadata provides a rich source of information for generating questions, summaries, related material, and further learning. The system that we propose would take full advantage of the metadata to provide rich additional content to enhance learning.

In the SCORM 2004 release, the content aggregation model has an XML schema binding for Learning Object Metadata, Content Structure and Packaging, and Sequencing and Navigation information. This SCORM metadata describes the different components of SCORM Content Model which includes components like SCOs. Metadata is a form of labeling that enhances search and discovery of components. This metadata provides significant possibilities for enhancing and scaffolding the learning process in real time and for providing intelligent, tailored delivery. Until now, these possibilities have not been realized.

Metadata can be collected in catalogs, as well as directly packaged with the learning resource it describes. Learning resources that are described with metadata can be systematically searched for and retrieved for use especially for enhanced delivery.

Advantages over existing approaches

Such an approach has advantages over existing approaches to eLearning. There are currently several ITSs that have been developed and tested in real educational settings. ANDES for example teaches physics, AutoTutor teaches computer literacy and physics, while CIRCSIM teaches medical students about the circulatory system. ITSs such as these are often developed as research projects, intended to explore new methods for delivering automated tutoring. They are typically developed from the ground up, each with its own unique teaching style, information structure, and delivery method. This has created a situation where the field is destined to produce long strings of prototypes that never connect to one another, and that are subsequently in danger of leaving no legacy. In comparison to ITS, content enhancement is a more modest approach to providing intelligent delivery that does not approach the sophistication of such systems, but has the advantage that it can use existing content that is not currently intelligently delivered.

Conclusions

A system that enhances the content of learning material provides the opportunity to rapidly convert one-dimensional learning objects into rich, interactive learning experiences without time-consuming intervention of an instructional designer. By providing a deeper, richer template of information, such a system would also provide the first step in a bridge between current standards in learning objects and the ‘holy grail’ of eLearning, standardized intelligent delivery. We believe that the technology exists to begin such an enterprise. SCORM provides the ideal platform for content enhancement, because 1) of its growing status as *the* standard in eLearning, and 2) because it provides useful information through metadata that can be exploited to create additional relevant material.

An enhanced content system is not only an achievable goal for the eLearning community, it is an important step if eLearning is to make the leap from flat, page-turning software to automated intelligent delivery.

David F. Dufty

Advanced Distributed Learning Workforce Co-Lab,
University of Memphis, Memphis, TN 38152

Xiangen Hu

Advanced Distributed Learning Workforce Co-Lab,
University of Memphis, Memphis, TN 38152

Eric C. Mathews

Advanced Distributed Learning Workforce Co-Lab,
University of Memphis, Memphis, TN 38152

Suresh Susarla

Advanced Distributed Learning Workforce Co-Lab,
University of Memphis, Memphis, TN 38152

Authoring Tools for ITS with consideration of SCORM Standard

Introduction

AutoTutor is a computer tutor that teaches students by having a conversation simulating the discourse patterns and pedagogical strategies of human tutors. To create AutoTutor in different domains, subject matter experts (SMEs) need to create Curriculum Scripts, loosely ordered but well defined collections of concepts, correct answers and question-answer units. To make this process of creating these curriculum scripts easier we have created an authoring tool, AutoTutor Script Authoring Tool (ASAT). It provides automatic question generation and sentence similarity checking to help domain expert create scripts quickly and error free. The tool can save scripts as SCORM objects, thus we can use standard authoring tools to modify the scripts and import them into ASAT. Architecture of AutoTutor.

Architecture of AutoTutor

AutoTutor 3D is a distributed client-server application on the Internet, capable of “scaling out” to multiple computers as necessary to handle greater load. The latest version of AutoTutor has a central hub which acts like a messenger between all the different modules. There are five modules connected to the central hub (fig. 1) 1) Client module 2) Speech Act Classifier (SAC) module 3) Assessments module 4) Dialog module and 5) Log module. In addition to these modules, there are four supporting utilities: the Latent Semantic Analysis, Parser, Question Answering, and the Curriculum Script Utilities which are used by the modules for some specific purpose.

Curriculum scripts

AutoTutor has a curriculum script that organizes the topics and content of the tutorial dialog. The script includes didactic descriptions, tutor-posed questions, example problems, figures, and diagrams.

The first version of the tool was created by a group of instructional designers using Authorware in order to ease the creation of curriculum script by SMEs [3]. It was a simple tool which gave a graphical user interface to the user to add new scripts to the system and gave a big boost to the process of creating scripts easily and quickly.

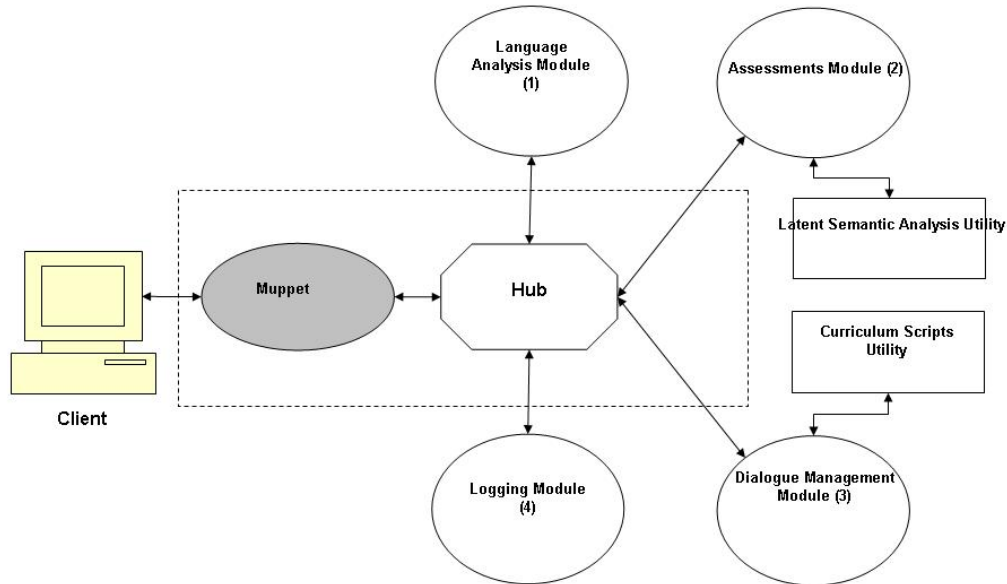
The tool gives an abstract structure of the script to understand the system thus spares the author of understanding the cryptic syntax of the scripts. It provides multiple screens with instructions to write the script in fill-in-the-blanks manner and it will be automatically converted to the proper AutoTutor format. The tool eliminates the requirement of having a computer programmer involved in the process of creating the scripts since the domain expert could directly interact with ASAT to create scripts in required format. Our studies showed that using this interface reduced the time required for create a script from months to about 90 minutes.

The process follows a paradigm of Rule-Example-Practice which has proved to be very efficient in instructional design. This process ensures that the user understands the significance of the data in context with AutoTutor and the required format in which the data is to be provided. For each part of data to be given by the user, the user is presented with a screen in 3 parts. The first part, Rule defines what the data is and how it will be used. The second screen gives an example of how the data is to be given. It explains what a valid form of data is and how to avoid giving unnecessary data. In the third screen the user is provided space for inputting the data.

New version

The latest version of ASAT is a web version, that can be accessed easily from any computer with an internet connection and there is no requirement of installation of any software on the client side.

One of the important features that this new version implemented was to allow a script to be created in multiple sessions and by multiple authors. Multiple users could edit the script multiple times before submitting it to the AutoTutor system.



New utilities to improve scripts creation

As discussed in previous sections, the curriculum script consisted of a question and multiple expected answers, called expectations that a student has to cover before moving to the next question. These expectations have to cover the entire answer but each expectation should address different parts of the answer and should thus be different than others. If two expectations are too similar, then AutoTutor may consider the student covered both the expectations instead of only one. LSA sentence similarity check was implemented to avoid this. Each expectation is compared with all other expectations using LSA cosine function to check for similarity. If this cosine value is more than 0.7, then the user is warned. The user can modify the expectations until the warning is removed or can choose to ignore the warning and continue writing the scripts.

We conducted a study to test ASAT by asking 15 instructors to generate one ASAT script. They were capable of completing a curriculum script on a question within approximately an hour, mean = 43.5 minutes, SD = 29.0. It was observed that the experts had difficulty segmenting an ideal answer into individual expectations. They had difficulty generating a reasonable number of hints and prompts for each expectation. They provided about 2 hints and 2 prompts for each expectation, on the average, whereas 3-6 would be needed to effectively handle an expectation. The hints and prompts covered a very low percentage (11%) of the constituent concepts that experts believed were important to answer to the main questions. We have developed an automatic question generator for ASAT. The question generator uses NLGML, Natural Language Generation Markup Language, to create questions. The users of the system can use the suggested questions in the script as hints and prompts, modify the suggested questions to suit the hints or prompts or can create new hints and prompts. Any choice made by the user will be saved by the system to train the question generator to generate better questions.

Script Maintenance

The web version of the tool allows users to maintain the scripts easily in different modes. In the simplest edit mode the script is broken into multiple modules and each module is shown in a different page with the Rule-Example-Practice paradigm. This mode is more suitable for beginners who use instructions while creating the scripts and takes more time to create the script. Review mode displays the entire script in one page view and is used by authors to review the script quickly in rapid prototyping. The script can also be exported in XML files,

used by programmers to modify the script quickly. After modification the script can be imported back into the database.

Once the script has been created, it can be transferred to the AutoTutor database by just a couple of clicks. This feature facilitates the rapid prototyping of the script by easily allowing scripts to be exported to AutoTutor for testing.

Using Standards

Adapting standards is a natural sign of maturity in any field. This is no different to the field of Intelligent Tutoring. In this paper we have discussed about development of one such system called AutoTutor and its authoring tool. While developing this authoring tool we have realized that there are varieties of different contents that require different authoring styles. Furthermore, there are different content materials may require different organizational structures. We propose a concept of a “Lesson Planner” that has two levels of authoring capabilities: Expert Level Authoring and Novice Level Authoring. Expert Level Authoring is a meta authoring tool that configures the application for Novice Level Authoring. In the Expert Level Authoring, users can specify structures of the scripts so Novice Level Authoring simply follows the steps.

Pavan Karnam

Institute of Intelligent Systems
The University of Memphis
Memphis, TN 38152
pkarnam@memphis.edu

Suresh Susarla

Institute of Intelligent Systems
The University of Memphis
Memphis, TN 38152
ssusarla@memphis.edu

Xianguan Hu

Institute of Intelligent Systems
The University of Memphis
Memphis, TN 38152
xhu@memphis.edu

NLTML: A Mark-up Language for Transformation-based Question Generation

AutoTutor Questions

AutoTutor is a dialog-based tutoring system that has been used by colleges to teach conceptual physics and computer literacy. ASAT is an authoring tool created to facilitate the authoring of curriculum scripts so that AutoTutor can teach other knowledge to learners. AutoTutor teaches by helping learners to solve problems. In the curriculum script, the ideal answer of a problem is split into single sentence elements, called “expectations”. The system “expects” a learner to speak out all the expectations. The system forms some questions for each expectation and asks a learner these questions until the learner covers the expectation. These questions are called “hints” and prompts. The following example shows an expectation and the hints and prompts associated with it in the physics tutor curriculum:

Expectation: There are no horizontal forces on the packet after release.

Hint 1: What can you say about the horizontal forces on the packet.

Hint 2: After release, in which direction is there zero force on the packet?

Prompt 1: After release, the packet is not affected by any force that is _____?

Prompt 2: After release, there are zero horizontal forces on the _____?

Prompt 3: There are zero horizontal forces on the packet after _____?

Prompt 4: After release, the horizontal force on the packet is _____?

From this example, we see that the hints and prompts are so similar to the expectation that simple rules may be created to map the expectation to the hints and prompts. For example, *Hint 1* is a typical AutoTutor question “What can you say about X?” This can certainly be a question template. To find what “X” is, we need to identify a noun phrase from the source sentence. In our example expectation, we have three noun phrases “horizontal forces”, “the packet” and “horizontal forces on the packet”. The following rule will then create 3 questions:

Rule: *If “X” is a noun phrase in the source sentence, then ask “What can you say about X?”*

Input: *There are no horizontal forces on the packet after release.*

Output 1: *What can you say about horizontal forces?*

Output 2: *What can you say about the packet?*

Output 3: *What can you say about the horizontal forces on the packet?*

Our goal is to create a mark-up language that is simple but capable of describing such rules.

The Mark-up Language

This work is highly stimulated by the need of a question generator for AutoTutor and the performance of AIML(Artificial Intelligent Mark-up Language). AIML enables people to define stimulus-response transformations for chat-bot. While it is greatly successful for common chat, it is limited in casual dialogs and simple question answering categories. AutoTutor is more like an answer-questioning system – given the answer stored in the curriculum, generate adequate questions as hints and prompts to help the learner to learn. This needs the identification of syntactic and semantic properties of phrases in a given text. NLTML(Natural Language Transformation Mark-up Language) is designed to perform such complex tasks.

Syntactic Structure

Before we proceed, let us take a look of the NLTML script for the example rule in the above section:

```
<category>
  <pattern>
    <NP> _np_ </NP>
  </pattern>
  <template>
    What can you say about _np_?
  </template>
</category>
```

The category-pattern-template structure is borrowed from AIML. A category indicates a rule. A pattern specifies a branch of the syntax tree for the source sentence to match. And, a template describes a text to generate. The tag <NP> is from the tag set many parsers (such as Apple Pie parser, Charniak parser, etc.) use. It refers to “Noun Phrase”. `_np_` is used as a variable to save the words in the noun phrase. The difficult part of writing a category is to form a pattern, which needs a parser to help. A pattern is considered as a simplified syntax tree, following the rules below:

A pattern is formed from any sub-tree of a syntax tree.

All sub-trees of a tree node can be removed, with a variable to denote content text of the sub-trees.

The tag <star> is used to ignore some of the sub-trees of a tree node without remembering the content text.

A variable is a string of symbols starting and ending with a “_”.

Semantic features

Consider a simple sentence: “The boy went to school.” The parsed tree is as follows:

```
<SI><S><NP><DT>The</DT><NN>boy</NN></NP><VP><VBD>went</VBD><PP><TO>to</TO>
<NP><NN>school</NN></NP></PP></VP><.></.></S></SI>
```

The following pattern describes “somebody went to somewhere”

```
<pattern>
<S>
  <NP person="true">_person_</NP>
  <VP><VBD>went</VBD>
  <PP><TO>to</TO>
  <NP location="true">_place_</NP>
  </PP>
  </VP>
  <star />
</S>
</pattern>
```

This pattern can be associated with templates such as:

```
<template>Where did _person_ go?</template>
<template>Who went to _place_?</template>
<template>Why did _person_ go to _place_?</template>
```

The attributes specified in the phrases (person= “true”, location= “true”) are called semantic features. When “person” is “true”, a “who” question might be asked, otherwise a “what” question might be asked. When “location” is “true” a “where” question can be composed. Another important feature is “time”, it can be used to compose a “when” question.

Functions

Some functions are designed for making necessary word transformations, such as getting the lemma form of a word. A function is a string of symbols starting with a “_” and ended by a pair of parenthesis containing a variable defined in the pattern. Functions are used in templates only. The following example shows the use of functions.

```
<category>
  <pattern>
    <S>
      <NP name="false">_subj_</NP>
```

```

    <VP>
      <VBD>_vbd_</VBD>
      <PP>_pp_</PP>
      <NP time="true"></NP>
    </VP>
    <star />
  </S>
</pattern>
<template>
  When did _lowerFirst(_subj_)_getLemma(_vbd_)_pp_?
</template>
</category>

```

If the source sentence is “The boy went to school yesterday”, the above category will generate the question “When did the boy go to school?” The function `_lowerFirst` changes “The boy” to “the boy” and the function `_getLemma` changes “went” to “go”. The “name” feature is used to avoid changing the first letter of a named entity to lower case.

Interpreter of the mark-up language

An interpreter of the mark-up language needs to integrate a proper parser and some other computational linguistic modules, such as named entity identifier, time-location expression labeler, etc. We created a web tool (<http://141.225.13.163/nlgml/>) that can be used to create categories and generate questions. The syntax parser for this tool is Charniak parser and the major semantic component is WordNet.

Conclusion

Although question generation has long been a difficult problem, the mark-up language we designed provides a possible solution. The mark-up language is now used by AutoTutor Script Authoring Tool for hints and prompts generation. However, the language is capable of describing other types of text transformation, including question answering.

Zhiqiang Cai

The University of Memphis
zcaai@memphis.edu

Xiangen Hu

The University of Memphis
xhu@memphis.edu

Arthur C. Graesser

The University of Memphis
graesser@memphis.edu

Regular article: Podcasting in Law School: Technical Considerations in Building a Podcast for Use in Higher Education

Introduction

A month ago the papers on Harvard Law's Podcast came across my desk as they were floating about the Educational Technology office suite. Having already implemented my own brand of multi-media documentary syndication, coined *Quantumedia*, on my personal site, <http://www.gregory-otoole.com>, and recently completing the Advanced Coldfusion MX 7 courses, I jumped at the chance to build something new and exciting – and highly worthwhile – for the University of Denver's Sturm College of Law where, at the time, I was the Web Assistant

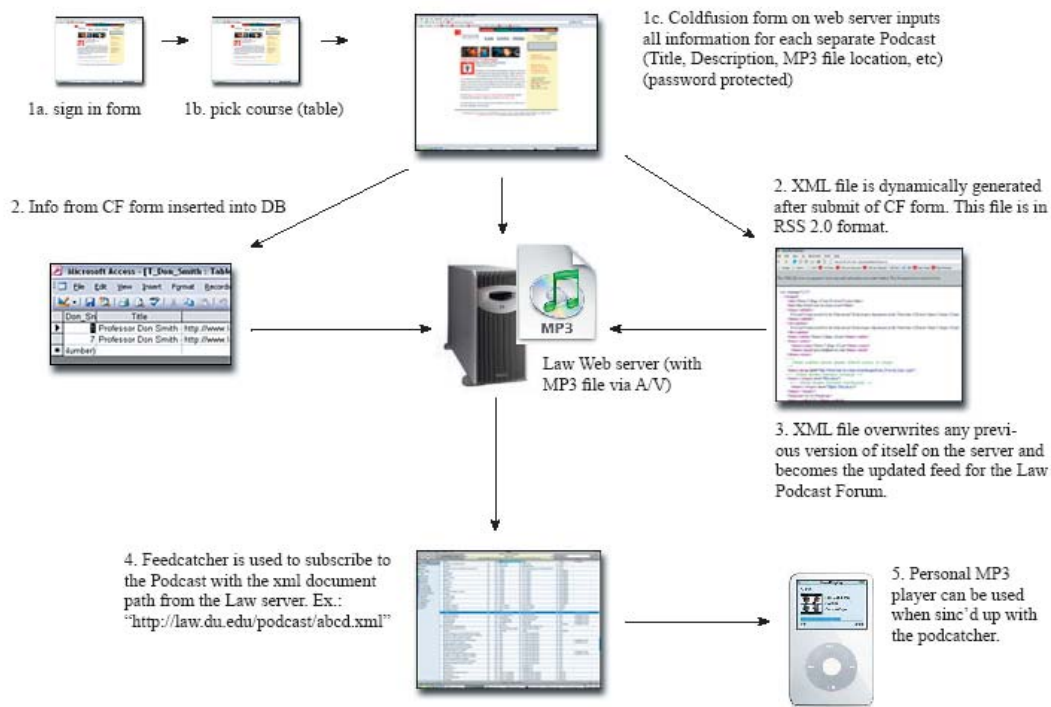
At first we only had in mind to create a space that would make available to all of the faculty an xml document, essentially, and a location where students could use that document's URL to subscribe to that particular course's Podcast. The files were to be all audio as in what is "conventionally" done in a Podcast. However, like technology itself, that idea changed very rapidly. The simple audio Podcast was quickly summoned to allow for video capabilities, as well as a slew of other customizable features. It still evolves today.

The remainder of this paper will be technically oriented. I will make available the process (and code used) to construct what has since been named the Law Media & Podcast Forum 1.0. The home for this online application is http://law.du.edu/podcast/podcast_signin.cfm. The student, or user, side of the application is <http://www.law.du.edu/podcast/>. Here the students can find and subscribe to their specific needs, as well as find other information to get them started.

Audio Podcast

In order to subscribe to something, there needs to be content, so I'll start with the Law Media & Podcast Forum Manager 1.0. This is the interface that faculty and staff have available to upload files. It is password protected for obvious reasons, but it works as described below. First, a visual aid.

Law Podcast Forum Map <http://law.du.edu/podcast/>



1a. A Coldfusion form is used to sign in to the Law Media & Podcast Forum Manager 1.0. 1b. A drop-down menu is made available that is populated by the names of all database tables that can be used with the Manager. One (Microsoft Access Database) table has been previously assigned to each professor's course or admin for the program. 1c. A Coldfusion form is used to collect information for each Podcast (Title, Description, Length, File, Type, etc.) that will be used to populate the xml page. (See example below for proper iTunes RSS 2.0 formats. Others available online.) 2. Data from the form collects into the designated database table; the mp3 file is uploaded to the server; the xml file is dynamically generated in RSS 2.0 format for use in all updated "feed catchers", especially, the popular iTunes brand. This xml file is written to the server in the proper directory after being generated. This allows for the updated Podcast to be always available. 4. Shown here is a thumbnail screen capture of iTunes being used to subscribe to the Law Media & Podcast 1.0. 5. Personal mp3 players synced up with your "feed catcher" allow for convenience and portability.

Video-Cast

Once success was had with the audio-based syndication, naturally the desire arose for video. DU Law Professor David Thomson played a major role in motivating this particular functionality. In the end, it was decided, that sticking with the original format was best suited for all those involved; the Manager was easy to use and worked well, and the subscription site was functioning properly, and was convenient for the user. One major alteration was made when we ran into an issue with file size (of the videos). The solution is shown below.

LAW MEDIA & PODCAST FORUM MANAGER 1.0™
STURM COLLEGE OF LAW - UNIVERSITY OF DENVER | law.du.edu

UNIVERSITY OF DENVER
STURM COLLEGE OF LAW

Current Administrator: gre gre | Sign Out

Managing T_David_Thomson

Use the following form to enter media & Podcast files. **NOTICE!** For Audio mp3s use the BROWSE field to insert your file. For Video files you must use the "Video Podcast Path" field to paste in the URL path to the video file on the server. Video files are too big and will not upload at this time. Please Email Greg or Wayne if you have questions.

Title (OneWordOnly)

Description

Audio File Upload Not for video podcast files!

Video File Path Paste in FULL URL to file, por favor!

Length

Type

The University of Denver College of Law • 2255 East Evans Avenue • Denver, Colorado 80208 • (303) 871-6000 • FAX: (303) 871-6378
Contact Us | University of Denver | Copyright 2003 University of Denver College of Law

The Audio/Visual Department here has always handled recordings of lectures and class material. This, eventually, was found to be the answer to our file size dilemma. Since AV had the original video files for the Video-cast, they did not necessarily need to upload the files, as much as tell the xml document where those files were located. Problem solved. The admin, then, formatting a new Video-cast would simply use the "Video File Path" field I inserted here to enter the URL path to that particular file. And, since Wayne Rust, the AV Manager already had the files on the web server, and had converted them into the proper format to play on a Video iPod (many commercial applications available on the market), it was very easy for him to copy and paste that URL.

The Code

I have made available the HTML and Coldfusion code used on all of the pages to make the Law Media & Podcast Forum Manager 1.0 work. After you have your xml documents created, it's a simple task of copy-and-pasting that URL into your iTunes or other feed catcher. There are plenty of sites out there to help you with that, so I would not include those steps here. Please see "HTML and Coldfusion code" for the commented code for this application: http://lutf.ieee.org/learn_tech/issues/january2006/podcasting_code_otoole.doc. Please note that some unnecessary text has been omitted for sake of space.

Gregory O'Toole, M.A.
Web Manager
Educational Technologies
Sturm College of Law
University of Denver
Denver, CO 80208

Tel: +1 303-871-6164
gotoole@law.du.edu