February 18
2003

THE eLEARNING DEVELOPERS' JOURNAL™

Strategies and Techniques for Designers, Developers, and Managers of eLearning

THIS WEEK — MANAGEMENT STRATEGIES

SCORM Deployment Issues in an Enterprise Distributed Learning Architecture

BY JEFFREY C. ENGELBRECHT

Enterprise distributed learning systems are required to support many technologies, subsystems, and specifications that were initially designed for a campus environment. However many of these include functionality that prevents them from being expanded to support an enterprise architecture. Delivering content to distant users located in dispersed networks, separated by firewalls and different Web domains, requires extensive customization and integration.

Military systems are particularly difficult since military environments often consist of conflicting and overlapping firewall restrictions, and multiple local and wide area networks. In addition, Department of Defense (DoD) and command-level security policies require systems to comply with specifications for interoperability and standardization. This article outlines some of the problems of implementing the Sharable Content Object Reference Model (SCORM) in the Marine Corps Distance Learning System (MarineNet) and extends the discussion to other educational institutions as well as to the corporate sector.

SCORM is a content interface specification that is becoming widely accepted by DoD and the commercial learning technology industry world-wide. While SCORM provides important benefits for interoperability, advanced dynamic tracking and sequencing of student progress, the specification does not adequately support deployment in enterprise architectures. This article reviews the impacts of the DoD Mobile Code Policy and the inherent restrictions of Web programming that require distributed learning content to be located near the user and prevents the content from com-

Continued on next page

SCORM incompatibility across multiple Web domains is not a frequently-discussed problem, but it exists, and in some organizations presents a major obstacle to enterprise-wide distributed learning. In this article, you will learn how this may affect your organization, and some strategies that may help you work around it.

A publication of

THE eLEARNING GUILD™

municating to a centralized Learning Management System (LMS) via the SCORM specification.

An introduction to courseware interface specifications

The goals of all courseware interface specifications are to promote interoperability, accessibility, durability, and reusability. The four organizations that are leading e-Learning developers toward standards are:

- The Aviation Industry Computer-Based Training Committee (AICC),
- The Instructional Management System Global Learning Consortium, Inc. (IMS),
- The Institute of Electrical and Electronic Engineers (IEEE), and
- The Advanced Distributed Learning (ADL) initiative.

(Editor's Note: Find more information about these bodies in the Guild *Resource Directory, under "Design & Development Standards" category.)*

All of the standards bodies have undergone considerable evolution since their creation and many of them now share the same guidelines.

The IMS is an international standards body that promotes many of the core data formatting and packaging specifications that are used in both the AICC and ADL specifications. Both the AICC and ADL groups assist user communities in implementing the core IMS specifications. The AICC group focuses on the aviation industry and the ADL group focuses on the DoD. Both the AICC and ADL communities expand on the IMS specifications, providing additional guidance on how courseware should exchange the information with a LMS as well as for the sharing of content and information between other programs and systems.

The AICC specification offers two web-based approaches for courseware to interface with an AICC-compliant LMS, the Application Programming Interface (API) implementation and the HyperText Transport Protocol (HTTP)-based Computer Managed Instruction (CMI) Protocol (HACP) implementation. *(Editor's Note: See the Glossary at the end of this article for definitions of terms that may be new to you.)* While both implementations operate on the same underlying IMS data structures and utilize similar function calls, the mechanisms providing the communication link are quite different.

The API implementation is a JavaScript approach, while the HACP implementation is a traditional HyperText Markup Language (HTML) approach. This is an important difference that must be understood by every distributed learning architect because there are several limitations with both approaches. This article, however, primarily discusses the API implementation in detail.

The SCORM specification developed through the ADL initiative is based directly on AICC's API implementation and thus shares its advantages as well as inherent limitations. The SCORM specification does not provide a HTTP-based implementation as does AICC; however the API incorporates the same course launch construct as well as the information exchange functions specified by the AICC API specification.

While the differences between an API and a HTTP implementation might seem trivial, the impact on a distributed learning architecture is significant. Programming technologies, their respective network protocols, and inherent security constructs produce very different outcomes and potential problems and limitations for enterprise architectures.

The SCORM and AICC API interfaces

The SCORM and AICC API specifications operate in the same manner. The key advantage of the API interface is the reduced complexity (compared to HACP) for the courseware developer. Complexity is reduced since most of the direct interfacing with the LMS database is abstracted to a set of JavaScript functions. Additionally, the inclusion of the JavaScript functions is simplified by using a script within the content to locate the API functions within the LMS browser windows.

Figure 1 illustrates the process of locating and including the API functions and identifies the sources and responsibilities of each of the code components. Upon initial launch of an API-enabled course, whether SCORM or AICC, the first step is to locate and include the API functions provided by the LMS so that the content can initiate communication with the LMS. The assignable unit (AU), which is called a sharable content object (SCO) in SCORM, runs a JavaScript `FindAPI()` function that searches all of the available web browser frames and windows opened on the user's workstation. Once the search locates the API object, the function returns the object so the SCO can execute the contained functions.

Three types of functions, specified in the API Adapter, are executed by the content: execution state functions, state management functions, and data transfer functions. There are two execution state functions that are responsible for initiating and terminating session connections with the LMS. There are also three state management functions that are responsible for handling errors in the communications with the LMS. Lastly, three data transfer functions exist that are responsible for transfer-

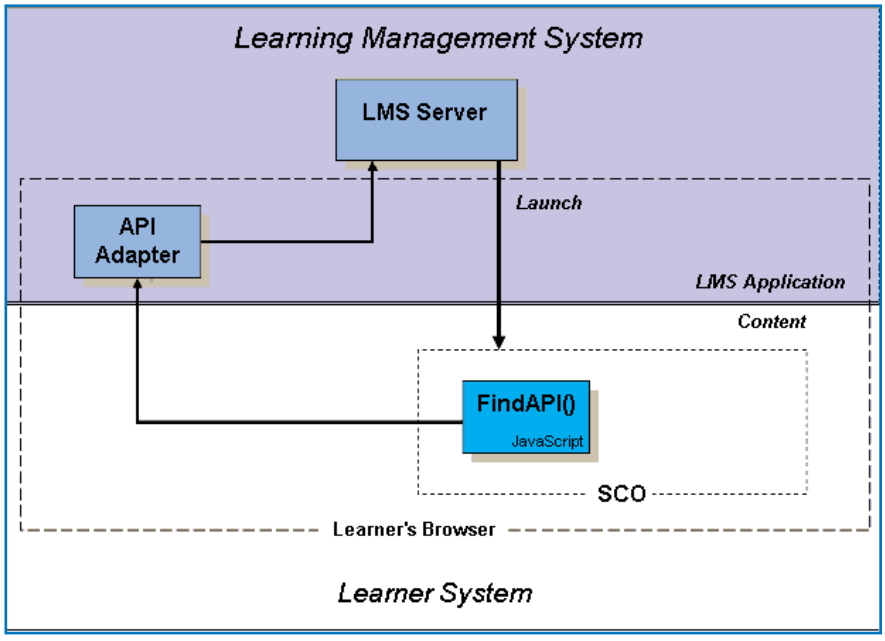

FIGURE 1 *Locating and including the API functions is the first step after launching a course.*

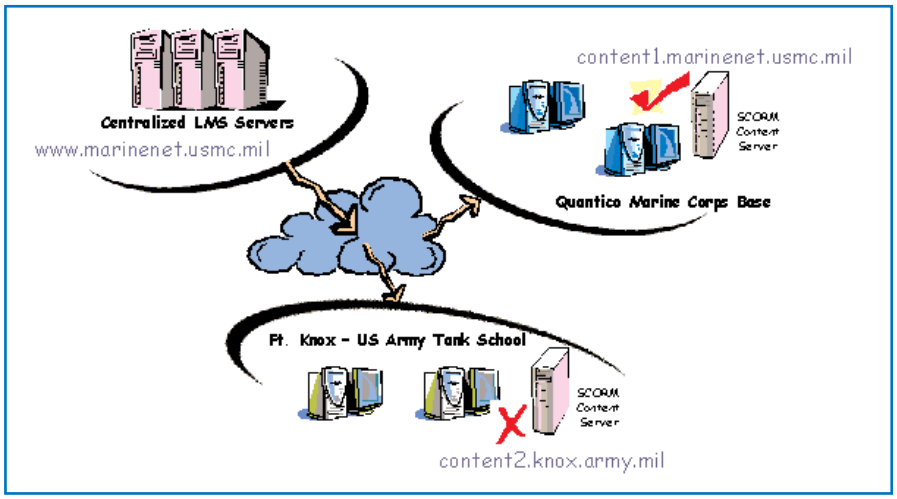

FIGURE 2 *SCORM incompatibility in an enterprise environment is illustrated by the current Marine Corps scenario.*

ring data to and from the LMS. These eight functions are all that comprise the API Adapter from the content's perspective. The only other function that is part of the API specification is the `FindAPI()` function, which is programmed by the content developer and contained within the content itself. The ADL web site located at www.adlnet.org provides detailed information about each of the functions. Other than the `FindAPI()` function, these details are not central to the theme of this article.

One important point to understand about the construction of the API implementation is the relative owner or source of the code components. The `FindAPI()` function and the execution of the individual functions contained within the API are the responsibility of the individual SCOs, however the source of the API is the LMS. When content is launched from the LMS, the LMS must maintain a web browser window or frame that contains the API and the SCO must be capable of locating and including the API functions as if the functions were an integral part of the content.

The problem with the API implementation in an enterprise architecture

As described in the previous section, when SCORM-conformant content is launched, the first action that must take place is for the JavaScript code contained in the content to locate the SCORM API that is provided by the LMS. The SCORM API Adapter is typically located in a different web browser window or frame than the content. This implementation is feasible as long as the content and the SCORM API Adapter are both in the same web domain (For example, both are in the *marinenet.usmc.mil* web domain). However, if the LMS window is in a different web domain than that of the content window, then security rules will prohibit the code within the content window or frame from accessing the SCORM API Adapter contained within the LMS window or frame because it is also in that other domain.

Figure 2 illustrates how the current Marine Corps environment impedes the use of SCORM. In this scenario, Marine students are located at two geographically separated training locations, some within the Marine Corps domain and some within the Army domain. This scenario is described because large populations of Marine students attend resident training at other service locations. Figure 2 illustrates the example of providing skill training to tank crew members at both an Army and Marine Corps installation. The course on basic tank operation and maintenance is applicable to Marine students attending resident training at the Army's Tank School at Ft. Knox, Kentucky, as well as the Marines receiving refresher training at Quantico, Virginia. Marine students at each location would interface with the central Marine Corps LMS to initiate the training session, but would be redirected to a local content server at the respective location to receive the high-bandwidth content.

The Marine Corps operates centralized LMS servers located within the *marinenet.usmc.mil* web domain. Since DoD mobile code policies make it problematic to centralize content hosting, content servers

THE eLEARNING DEVELOPERS' JOURNAL

Publisher David Holcombe

Editorial Director Heidi Fisk

Editor Bill Brandon

Copy Editor Charles Holcombe

Design Director Nancy Marland

The eLearning Guild™ Advisory Board
Ruth Clark, Conrad Gottfredson, John Hartnett, Bill Horton, Kevin Moore, Eric Parks, Brenda Pfaus Marc Rosenberg, Allison Rossett

The eLearning Developers' Journal is published weekly by ***The eLearning Guild***, 525 College Avenue, Suite 215, Santa Rosa, CA 95404. Phone: 707.566.8990. ***The eLearning Guild*** is an operating unit of Focuszone Media, Inc., 1030 Beatrice Street, Eagan, MN 55121.

The Journal is included as part of ***Guild*** membership. To join the ***Guild*** go to www.eLearningGuild.com.

The eLearning Developers' Journal™ is designed to serve the industry as a catalyst for innovation and as a vehicle for the dissemination of new and practical strategies and techniques for e-Learning designers, developers and managers. The **Journal** is not intended to be the definitive authority. Rather, it is intended to be a medium through which e-Learning practitioners can share their knowledge, expertise and experience with others for the general betterment of the industry.

As in any profession, there are many different perspectives about the best strategies, techniques and tools one can employ to accomplish a specific objective. This **Journal** will share these different perspectives and does not position any one as "the right way," but rather we position each article as "one of the right ways" for accomplishing a goal. We assume that readers will evaluate the merits of each article and use the ideas they contain in a manner appropriate for their specific situation. We encourage discussion and debate about articles and provide an Online Discussion board for each article.

The articles contained in the **Journal** are all written by people who are actively engaged in this profession at one level or another — not by paid journalists or writers. Submissions are always welcome at any time, as are suggestions for articles and future topics. To learn more about how to submit articles and/or ideas, please refer to the directions in the sidebar on page 6 or visit www.eLearningGuild.com.

must be located within the security enclave of each base network. The impact of the mobile code policy is further explained later in this article. These distributed content servers can operate within various Web domains as illustrated in Figure 2 on page 3.

Content servers located on Marine Corps Base (MCB) Quantico are located within the *marinenet.usmc.mil* Web domain, so that SCORM-enabled content can interact with the SCORM API Adapter located within a LMS window or frame. However, content servers located on other bases such as Ft. Knox, which operate within the *knox.army.mil* Web domain, cannot operate SCORM-enabled content since the content code cannot interact with the SCORM API Adapter located in a different Web domain.

A review of web security

The reason that SCORM-enabled content cannot interact with a SCORM API Adapter served by a different Web domain is attributable to the core security restrictions of Web coding. Web browsers, including Netscape Navigator and Microsoft Internet Explorer, allow potentially destructive code to be embedded in a Web page, downloaded across the net, and executed on a local machine even without the user's knowledge. The two most popular approaches to protect users from malicious code are sandboxing and code signing. Sandboxing, the approach implemented by Java, protects the user by restricting the untrusted code from operating outside the scope of the program space and thus restricting it from interfering with other applications or processes. Code signing, the approach implemented by ActiveX, protects the user by allowing the user to determine the authenticity of the code prior to it running, thus providing an authentication model.

Both sandboxing and code signing are widely accepted and implemented security approaches, but each has its detractors. ActiveX does not have an inherent security model but rather uses a trust model. Thus, ActiveX requires that the user understand the purpose of the code and have some amount of trust of the developer that signed the code. If the user understands the purpose of the code and trusts the developer, then he or she can acknowledge the trust and allow the code to function. One thing to note is that once the user

SCORM-developed courseware expects to have access to the SCORM API functions provided by the LMS, and traditionally the API is contained within the code build of the LMS. However, if the API functions were co-located with the courseware, thus abstracted from the LMS, the courseware would be able to access the functions even if the content was located in a different Web domain.

explicitly trusts the code, it may operate with the same system rights as other programs that were installed by the user. (Whether the user can adequately determine if the developer's signature is authentic is not a concern for this article.)

Sandboxing does not require the user to explicitly trust the code, rather all sandbox code is considered to be inherently harmful. Thus, to prevent the code from damaging the user's system the code is prevented from interacting with other applications and processes on the user's machine. The largest scope of the sandbox is restricted to the Web domain under which the code is executed, but this scope can also be further restricted by the applet or the user's Web browser.

A SCORM-conformant implementation specifies that the LMS software implement the SCORM API Adapter. The API Adapter must also be coded in Java or JavaScript. Therefore the use of the ActiveX trust model, as a possible solution, is not supported. Thus if SCORM is an enterprise requirement, then the architecture must deal with the limitations of distributing code that operates within a single sandbox. This means that the sandbox is restricted to the Web domain where the LMS and the SCORM API is located. If the courseware that is calling the SCORM API is located in a different Web domain, thus a different sandbox, then it will not be able to execute or locate the API.

The illustration used in Figure 1 is modified in Figure 3 to identify where the domain conflict is located.

As illustrated in Figure 3, the `FindAPI()` function is unable to locate the SCORM API Adapter since the `FindAPI()` function is located in a different Web domain than the API Adapter. The result of the `FindAPI()` function would return a null value and the content would report that the LMS is unavailable or otherwise could not connect.

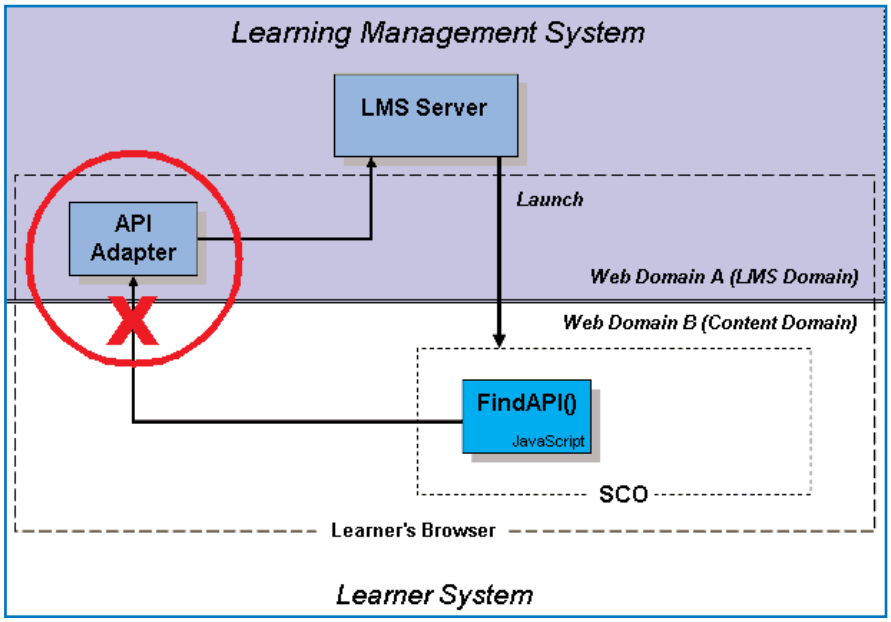

FIGURE 3 *Cross-domain problems in SCORM arise when the content and the LMS are in different domains.*

Mobile code policies — pushing content to the edge

Many distributed learning applications are not forced to confront the problems associated with operating in multiple Web domains; however the DoD has strict information security policies that prevent courseware from traversing security boundaries. The policy that impacts centralized content hosting is the DoD Mobile Code Policy (http://www.c3i.osd.mil/org/cio/doc/mobile-code11-7-00.pdf).

The DoD Mobile Code Policy restricts the use of web technologies such as Java, JavaScript, ActiveX, Macromedia Shockwave, Macromedia Flash and most other client-side programming technologies. Each mobile code technology is assigned to one of three categories:

- Category 1, which includes ActiveX, is fully authorized to operate.
- Category 2, which includes Java applets and other Java mobile code, is authorized only if the code is obtained from a trusted source.

While category 1 and 2 technologies possess the necessary tools to allow courseware to effectively communicate to the LMS, courseware without user interactivity is not likely to create a worthwhile learning experience for the user.

- Category 3 contains the bulk of the technologies that allow for interactive and stimulating learning experiences. Category 3 technologies include JavaScript, VBScript, Macromedia Shockwave, Macromedia Flash, and even Adobe portable document format (PDF).

The DoD Mobile Code Policy allows for Category 3 technologies to be used on DoD information systems, however many military firewall policies, including the Marine Corps', prohibit some of these technologies from breaching network boundaries. Most distributed learning content, including those employed by the Marine Corps, leverage heavily on Flash and Shockwave for instructional purposes; however the use of Category 1 content, such as Java, is also critical to content.

With courseware relying heavily on Category 3 technologies, DoD distributed learning programs must find ways to deliver content to distributed users without violating policies. The typical solution is to host the content at each of the military bases, thus behind the firewall boundaries. Since it is too costly to operate an LMS at each user site, a centralized LMS is forced to interact with numerous content servers, each located behind a firewall. Courseware being delivered to the user from within the firewall boundary can operate all mobile code technologies; however the SCORM interface between the courseware and the central LMS server must now deal with the problem of multiple Web domains, which is the motivation of this article.

DoD is not the only organization affected

The military example described in this paper may be the most convenient way to highlight the problems with SCORM in enterprise architectures; however it is not the only example. Large geographically dispersed organizations must also confront this problem. Moreover, organizations that are consolidated in a single location may have similar problems if they attempt to use content provided by third-party organizations.

Many commercial companies have con-

solidated many of the information technology (IT) services to centralized or regional headquarters to reduce costs and consolidate resources. The benefits of operating a single centralized LMS are easy to justify and the justification to extend the storage of high-bandwidth content to the outer edges of the network is also justifiable. The cost of purchasing, administering, and maintaining simple web-enabled content servers at each company location is considerably lower than the cost of the bandwidth consumption of multiple users trying to pull identical content from a single location.

Additionally, many organizations see the value of accessing content from third-party locations such as the Library of Congress, Carnegie Mellon's Artificial Intelligence Repository, the National Archives, or simply between each other. In order to enjoy economies of scale, eliminating duplication of effort through the consolidation of similar functions may be necessary. Therefore having the content owner also host the information relieves each subscriber organization from duplicating the same infrastructure and associated costs.

Allowing other organizations to re-host third-party content creates the potential for legal as well as lifecycle management issues. Beyond just the economic benefits of consolidating resources, copyright and other legal restrictions can prevent organizations from re-hosting content in their own domain. Furthermore, once an organization re-hosts the content, the link between the content owner and the user increases the probability of out-of-date or incorrect information.

The solutions in sight

There are several potential alternatives for overcoming this issue. These solutions include:

1. Modify the SCORM API Adapter enabling it to be hosted on the distributed SCORM content servers.
2. Modify the Web domain structure of the enterprise system to bring the LMS and all distributed SCORM content servers under a single Web domain.
3. Continue using other protocols such as the AICC HACP specification until SCORM is updated to be compatible with enterprise systems.

Solution 1 would entail recoding the LMS's SCORM API Adapter, enabling it to be hosted on remote content servers instead of the LMS servers. By porting the Adapter to the content server, both the content code and the Adapter code would be within the same sandbox. The resulting complication is that the LMS vendors would have to modify the back-end communication to the LMS database or middleware software to accept, store, and retrieve the information for the distributed SCORM API Adapter that is located on a remote content server. Furthermore, the communications between the Adapter and the LMS would have to be developed to operate using acceptable network protocols to avoid firewall security restrictions.

Solution 2 does not require reprogramming, however it does require the authority

to manipulate third-party networks and to potentially mislead users. While the Web domain model was initially developed to preclude users from manually entering IP addresses, the domain model has developed into an identification of ownership and accountability. Hosting systems on other domains is permitted by some network owners based on the desire to consolidate resources; however the foster systems do not typically fall under the jurisdiction of the hosting network. Thus, if the foster system malfunctions, violates security policies, or simply has to be upgraded, then the responsibility falls on the system owner rather than the network owner who is hosting the system.

Hosting all of the content servers within a single Web domain operated by a single distributed learning provider might simplify this problem, but this scenario requires that hosting networks allow the LMS system to publicize IP addresses as being owned by the application provider rather than the hosting network. For example, the Navy might allow the Marine Corps to advertise a Navy-owned IP address as a Marine Corps component (thus, *www.marinenet.usmc.mil* would resolve to the Navy-owned IP address of 205.70.115.15), but current policies prohibit advertising external commercial IP addresses as being a Marine Corps component (*www.marinenet.usmc.mil* resolving to the commercial IP address of 198.175.96.33).

Both solutions 1 and 2 are attractive in that they do not deviate from the API specification used by both AICC and SCORM, however these solutions may not be compatible with all enterprise distributed architectures. Solution 1 is completely dependent on the LMS since the API Adapter is a software component of the LMS. If LMS vendors could develop a capability of porting the API to a remote content server, then that LMS might be closer to meeting an enterprise LMS requirement as previously described. However, to date, this feature is not an advertised capability of any commercial LMS product.

Distributed learning programs that implement solution 2 also benefit by not deviating from the API specification, however they must incorporate network management into their responsibilities. Managing domain naming servers or even elaborate virtual private network systems may exceed the capabilities and resources at the disposal of many organizations. Commercial companies and educational institutions may have the ability to alter the network and employ such network technologies, however this does not guarantee a solution since the ownership and hosting of some content may still be restricted. As long as content is hosted by a third party, such as the National Archives, and policies restrict the re-advertising of such resources, the ability for distributed learning systems to offer these resources to their users will be restricted.

From the Marine Corps perspective, neither solution 1 nor 2 has proven to be viable since they require customizations either to the network or the LMS software. While it may be feasible to customize the LMS, these customizations would inherent-

GLOSSARY

API adapter: A Java applet provided by the LMS and running in the learner's browser. The API adapter makes it possible for the content in the SCO to communicate with the LMS.

Application Programming Interface (API): A set of routines, protocols, and tools for building software applications. A good API makes it easier to develop a program by providing all the building blocks, and all programs using a common API will have similar interfaces.

Assignable Unit (AU): The Assignable Unit refers to the actual content level in hierarchies defined by ADL/SCORM and by AICC. It is the smallest element of a course that can be assigned to a learner by a Computer Managed Instruction (CMI) system. One or more Assignable Units constitute a lesson, or nested instructional block. The highest level hierarchy is the Course level.

Code components: The elements (specifically, the API Adapter and the `FindAPI()` function) that make it possible for the Learning Management System (LMS) to find, launch, and communicate with lessons in courses.

Function: In programming and scripting languages, functions are commands that perform specific tasks (routines) and (usually) return a value. A function call in a script or program invokes the function. Calling a function consists of specifying the function name and, optionally, parameters (values that are needed in order for the function to carry out its task). In using an LMS, `FindAPI()` is a function that searches all of the available web browser frames and windows to find the API and permit an Assignable Unit or lesson to run.

HyperText Transport Protocol (HTTP): The underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

JavaScript: A scripting language developed to enable Web authors to design interactive sites. Although it shares many of the features and structures of the full Java language, it was developed independently. Javascript can interact with HTML source code, making it possible for sites to deliver dynamic content. JavaScript is an open language that anyone can use without purchasing a license.

LMS browser window (or frame): The window or frame within the browser in which the API Adapter is located.

LMS server: The physical server where the Learning Management System is running.

LMS specification: A standard that details how a Learning Management System launches a lesson, and how data is communicated between the LMS and the lesson

Web browser frame: A feature supported by most modern Web browsers that divides the browser display area into two or more sections (frames). The contents of each frame are taken from a different source; in the case of a Learning Management System, the API Adapter is typically located in a different frame or window than the content. As the article points out, code in the content window may not be able to access the API Adapter in the LMS window or frame.

(Editor's Note: Additional definitions may be found online at www.webopedia.com and at www.whatis.com.)

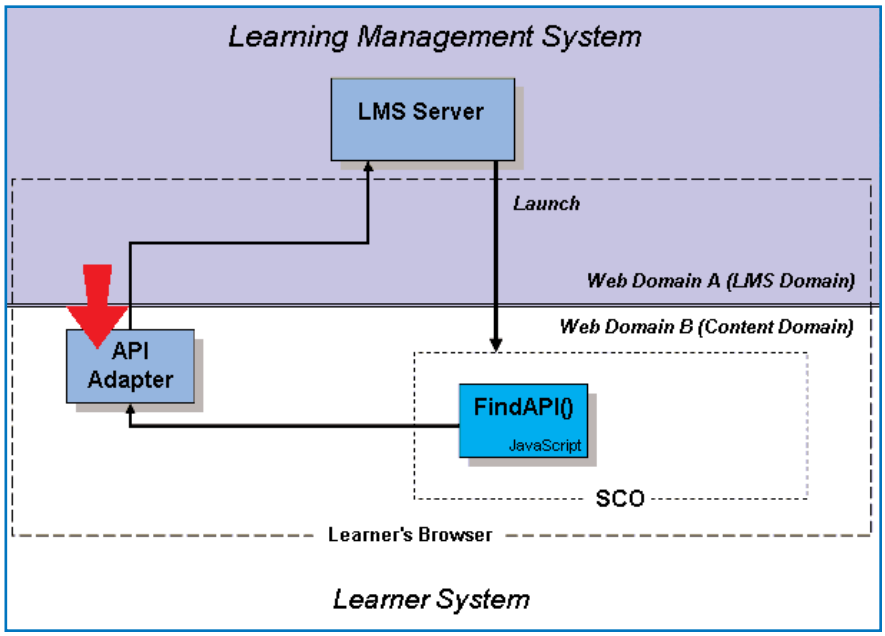

FIGURE 4 *Porting the API Adapter to the remote content server allows courses to access the API.*

ly be tied to that specific LMS build, thus limiting or even preventing future upgrades and replacement without additional customization. Furthermore, when dealing with content hosted by other organizations, the new portable API Adapter would have to be contained within, or provided with, the courseware.

Additionally, the Marine Corps Distance Learning Program does not own any network infrastructure. In fact, distributed learning applications use the Marine Corps Enterprise Network and therefore must accept the policies and service levels set forth by government and commercial network service providers. Custom configuration of such an infrastructure to accommodate specific distributed learning needs is not feasible in this environment. Therefore, the Marine Corps Distance Learning Program is pursuing solution 3 while continuing to investigate other potential sub-optimal options such as the integration of a HACP solution with the SCORM specification.

The case for integrating SCORM and AICC

There is one rather straightforward approach to bridging the gap between distributed courseware and a central LMS without customizing courseware or the LMS. SCORM-developed courseware expects to have access to the SCORM API functions provided by the LMS, and traditionally the API is contained within the code build of the LMS. However, if the API functions were co-located with the courseware, thus abstracted from the LMS, the courseware would be able to access the functions even if the content was located in a different Web domain. Figure 4 illustrates this potential solution.

The problems with this approach are that the back-end communications to the LMS would have to be able to communicate with the centralized LMS database and the entire functionality of the interchange between courseware and the LMS would be limited to the capabilities of the back-end communication.

Most LMS vendors would require a direct database connection from the API functions to the LMS database in order to reuse the existing courseware interface modules contained in the API Adapter. All of the distributed content servers would also have to establish Open Database Connectivity (ODBC) connections. This would impact both the content servers since most would not already have database capabilities. It would also impact networks since the database transmissions would occur over database ports instead of traditional Web HTTP ports. This approach may work for some organizations, however database communications across unprotected networks create several security concerns and within the DoD environment are strictly limited.

A potential solution is to have the API functions operate the basic SCORM functions for the courseware, and then translate the requests into AICC HACP interactions with the LMS. This approach would require that the LMS be AICC-compliant, but that is the only requirement. The API functions would allow the hosting of both AICC and SCORM courseware yet the LMS would only know of AICC courseware. The major limitation is that the LMS would now only be able to track AICC information, thus preventing upgrades to support future SCORM capabilities (such as sequencing) unless the AICC specification implemented by the LMS also supports the identical specification.

While this solution would not satisfy the leading-edge distributed learning applications that are focused on advancing the student's experience through the use of sequencing and other advancements, it would allow for enterprise distributed learning systems to deploy both SCORM and AICC courseware to distributed users.

The advantages of AICC HACP

The advantages of the AICC HACP specification to enterprise architectures are based on the underlying design of the specification. The AICC HACP specification does not require the use of either JavaScript or ActiveX and thus does not have either of their security limitations. The AICC HACP specification uses conventional HTML request and response messages to exchange AICC IMS-based information.

Since the interactions between the content and the LMS are HTML-based there are additional requirements placed on the content. In the AICC HACP specification the content is required to manage more session variables, extract information from the client's universal resource locator (URL) line, encode and decode URL encoded strings, and most importantly, know the address of the LMS server. The complexity of implementing the HACP specification increases when the architecture includes remote courseware servers located in different domains and security policies restricting plaintext transmissions via HTTP.

Conclusion

One conclusion that can be derived from this article is the acknowledgement that owners of enterprise distributed learning systems are forced to adopt technologies originally developed for the typical campus

learning environment and manipulate them to meet their unique organizational goals. The developers of the SCORM specification and the managers of the network security policies employed by DoD appear to be moving along diverging paths. Until the ADL community recognizes the current limitations of implementing SCORM in enterprise environments, distributed learning providers will be forced to find solutions outside of SCORM or continue to use sub-optimal distributed learning management architectures which are undesirable cost drivers for the training organization.

AUTHOR CONTACT

Jeffrey C. Engelbrecht is a Senior Information Technology Engineer at the MITRE Corporation and has been a key architect for distributed learning programs since 1997. He holds a Bachelors and a Masters degree in Computer Science from Virginia Tech. Jeffrey has functioned as one of the key architects for the Marine Corps Distance Learning Program (MarineNet) since its creation in 1997. For the MarineNet program, he has been involved in virtually all aspects of distributed learning with particular focus on Learning Management Systems (LMS) and courseware integration and fielding. He was the lead developer of the Marine Corps' first LMS and he has played a key role in the evaluation and selection of all successors. He was also the primary author of the Marine Corps technical courseware development manual. Currently at MITRE, Jeff leads the research on identifying the solution for implementing SCORM in the MarineNet architecture. Jeff can be reached by email at jengelbrecht@mitre.org.

ONLINE DISCUSSIONS

Extend your learning beyond the printed page! If you are looking for more information on this topic, if you have questions about an article, or if you disagree with a viewpoint stated in this article, then join the online discussions and extend your learning.

Follow these easy steps to participate:

1. Go to www.eLearningGuild.com and sign in.
2. Click on the Online Discussion button on the main menu.
3. Using the pull down menu, select the Online Discussion: Journal Topics
4. Select this article from the Subject list.
5. Click on ADD A NEW MESSAGE.
6. Enter your message. It will be posted as soon as you hit the Submit button on the form.

Additional information on the topics covered in this article is also listed in the* Guild *Resource directory.

About the Guild

The eLearning Guild™ is a Community of Practice for designers, developers, and managers of e-Learning. Through this member-driven community, we provide high-quality learning opportunities, networking services, resources, and publications. Community members represent a diverse group of instructional designers, content developers, web developers, project managers, contractors, consultants, and managers and directors of training and learning services — all of whom share a common interest in e-Learning design, development, and management.

The eLearning Developers' Journal™

The Guild publishes the only online "e-Journal" in the e-Learning industry that is focused on delivering real world "how to make it happen in your organization" information. The Journal is published weekly and features articles written by both industry experts and members who work every day in environments just like yours. As an active member, you will have unlimited access to the Journal archive.

People Connecting With People

The Guild provides a variety of online member networking tools including online discussion boards, and the Needs & Leads™ bulletin board. These services enable members to discuss topics of importance, to ask others to help them find information they need, and to provide leads to other members.

Resources, Resources, Resources

The Guild hosts the e-Learning industry's most comprehensive resource knowledge database. Currently there are over 2,300 resources available. Members have access to all of these resources and they can also post resources at any time!

Guild Research

The Guild has an ongoing industry research service that conducts surveys on 20 topics each year. These topics are identified by the Research Advisory Committee. The data collected is available for all members.

It's About Leadership

The Guild draws leadership from an amazing Advisory Board made up of individuals who provide insight and guidance to help ensure that the Guild serves its constituency well. We are honored to have their active engagement and participation. The Guild has also established three committees made up of active members who help steer its editorial, events program and research efforts.

Discounts, Discounts, Discounts

Guild members receive discounts on all Guild conferences and on other selected products and services. Your Guild membership will save you 20% off the list price of Guild events!

Join today at www.eLearningGuild.com!

Become a member today! Join online at www.eLearningGuild.com.

THANK YOU TO THESE GUILD ENTERPRISE SPONSORS

CLARK Training & Consulting (CTC) is a global leader in instructional design offering both training and consulting services. Our award-winning seminars are based on the latest research in instructional psychology and human performance improvement.

www.clarktraining.com
Contact: Kimberly Perkins
602-230-9190

Cyclone Interactive is an interactive media and web development firm creating online, CD and presentation solutions for a wide range of clients and industries.

www.cycloneinteractive.com
Contact: Earl Dimaculangan
earl@cycloneinteractive.com
617.350.8834

Spectra Interactive Learning is a unique, full service, e-Learning consulting company — growing and expanding in North America and Europe to meet the growing need for expertise in e-Learning strategy development, instructional design and program implementation. www.spectrainteractive.com

Contact: Brenda Pfaus, President
bpfaus@spectrainteractive.com
Ottawa, Canada (613) 230-9978

Because the most dramatic learning happens through first-hand experience, everything we do is focused on creating experiences — engaging, challenging and yes, fun, experiences — that people will take back to work and use every day to improve their performances.

www.alleninteractions.com
Contact: Jackie McMillan
jmcmillan@alleni.com
800.204.2635

To learn how to become a ***Guild*** Enterprise Sponsor, please contact David Holcombe at dh@eLearningGuild.com or call 707.566.8990.