The Advanced Distributed Learning Registry (ADL-R): Best Practices for Updating Registered Content

Version 2.0

Northrop Grumman Technical Services
Training and Simulation Group
12150 Monument Drive, Suite 800
Fairfax, VA 22033
http://www.ts.northropgrumman.com

12 October 2007

Table of Contents

Table of Contents ..... **ii**

Revision History ..... **iii**

1 INTRODUCTION **1**

1.1 Background 1

1.2 Project Summary 1

2 BEST PRACTICES **3**

2.1 Follow Relevant Guidance and Documentation 3

2.2 Establish Working Groups within Communities of Practice (COPs) 3

2.3 Utilize a Life Cycle Content Management (CM) Approach 4

2.4 Implement a Versioning Approach 5

2.5 Register Content Object Metadata 6

2.6 Register Version Updates and History 7

2.7 Use the Relation Metadata Element to Describe Content Relationships 10

2.8 Set Up Version Notifications for Relevant Content 12

3 REFERENCES **13**

4 Appendices **15**

4.1 Additional Resources 15

4.2 Acronyms 15

Revision History

Name	Date	Reason For Changes	Version
Angela Lindsey	11/9/07	Updated with changes from KMI and JADL.	2.0

1 INTRODUCTION

1.1 Background

The Sharable Content Object Reference Model (SCORM) is a Department of Defense (DoD) standard for developing and assembling Web-based learning courseware. It is a collection of specifications adapted from multiple sources to provide a comprehensive suite of e-learning capabilities. The Advanced Distributed Learning (ADL) initiative designed SCORM to enable accessibility, adaptability, affordability, durability, interoperability, and reusability of Web-based learning content.

Courseware developed using SCORM standards can be registered, discovered, accessed, dissected into shareable content objects (SCOs), adapted, repackaged, and reused in multiple learning management systems. Savings are realized when this reuse occurs, regardless of whether only a portion of the content is edited or "repurposed" or if the content object is used entirely by another organization or audience outside of the original intent. As budgets are slashed and just-in-time training is necessary, the potential savings in development costs and time mean a great deal.

In order to maximize savings due to sharing amongst the Services and other governmental organizations, Department of Defense Instruction (DoDI) 1322.26 documents the procedures that DoD Components are expected to follow concerning front-end analysis (FEA), SCORM conformance, repositories, registration, intellectual property, royalties, and licensing of distributed learning. Most importantly, the instruction mandates that online distributed learning content shall conform to SCORM, be registered in the ADL Registry (ADL-R) (available at http://adlregistry.adlnet.gov/), and shall be maintained in "searchable and accessible" repositories (DoDI 1322.26 is available at http://www.dtic.mil/whs/directives/corres/html/132226.htm).

As of November 2007, the ADL Registry has migrated to http://adlregistry.adlnet.gov and is currently undergoing major design changes to make it more user friendly for the ADL community. During the transition some of the reference material noted in this paper may not be available. When complete, the ADL-R help section (http://adlregistry.adlnet.gov/help) should link to any related documentation and DoD guidance to get users started with the Registry.

1.2 Project Summary

Even though the potential savings are there, there has yet to be a wide adoption of the ADL Registry system as it is still in its infancy. Version 1.6.1 of the ADL-R software was deployed in January 2007, giving users a stable platform to perform insert operations in which metadata describing learning content is added to the Registry catalog, and providing an open search engine to discover metadata describing available content. As more DoD Components register content, the likelihood of developers finding preexisting reusable content increases. However, buyer beware – how does one know if the most recent version of another's content is what's being

used? Whose responsibility is it to track version history and changes, and notify other potential users? What if borrowed content is updated or deactivated down the line?

As part of the Joint ADL Co-Laboratory (JADL) prototype program, the Northrop Grumman team at the Joint Forces Staff College (JFSC) has set out to answer these questions and more. Under this effort, the team created a prototype application called the ADL-Registry Client (ADL-RC) that compares data in the Registry with local transaction files of borrowed content and emails users change notifications when version updates are detected in the metadata (see http://www.jointadlcolab.org/downloads/research/2005/adlr_client/adlrc_final_report.pdf for details). The next step is to merge the application functionality to the Registry portal; this will provide users enhanced options to check for updates, either manually or via scheduled alerts that run periodically as set by individual users.

Prior to designing the version notification enhancements, the effort required extensive research and analysis into the methods users currently take to version content and update courseware. Additionally, the target audience was asked about their preferences for proposed enhancements to the Registry in regards to version control. A short questionnaire was emailed to various user groups including: the Distance Learning Coordination Committee (DLCC), a consortium of Joint Professional Military Education (JPME) schools; ADL-R registered users and repository owners; the ADL-RUG (Registry Users Group); as well as attendees from past JADL Implementation Fest conferences. Developers within the DLCC community of practice (COP) and a few seasoned Registry users also met with the prototype team to discuss content updates in relation to the Registry. Results from the research study are available at http://blackboard.jfsc.ndu.edu/docs/research_papers/ADL-R_Research_Results.rtf.

In this paper, suggestions and best practices to update registered content are summarized to promote sharing within different COPs at all stages of the content lifecycle. The proposed best practices related to version control will allow for better tracking of content relationships within the Registry framework. Where applicable, resources are linked to offer added details and examples are provided for clarification.

2 BEST PRACTICES

2.1 Follow Relevant Guidance and Documentation

Organizations should identify and follow the policies and procedures on learning content applicable to their organizations. For instance, DoD Components are directed to meet requirements in multiple DoD publications, such as DoDI 1322.26 and DoD Directive 1322.18. See http://www.jointadlcolab.org/support/policy/index.aspx for links to related DoD Directives and Instructions. Additionally, those submitting metadata to the ADL Registry should follow guidance outlined in the Getting Started documentation (see http://adlregistry.adlnet.gov/help). These resources contain specifics on the content registration process including metadata schema and a sample transaction file.

Generally, the Services have additional detailed guidance documented for those developing online learning content. This guidance can be in the form of standard operating procedures (SOPs), style guides, best practices, business rules, and more. The Army has its business rules and best practices available online at http://www.atsc.army.mil/itsd/imi/bus_rules.asp. The Navy Integrated Learning Environment (ILE) produced a *Content Developer's Handbook* that lays out its policy and guidance to "develop, deliver, manage, maintain, and evaluate content" (see https://ile-help.nko.navy.mil/ile/content/policy/handbook.aspx). The Air Force created an Advanced Distributed Learning Hub to provide USAF-specific guidance; it's available online at https://adlhub.golearn.csd.disa.mil. The *United States Marine Corps Interactive Multimedia Instruction (IMI) Style Guide* is available at http://www.tecom.usmc.mil/cce/references/design/IMI_Style_Guide_v_2_1.doc and the USMC College of Continuing Education has its own technical manual for courseware development at https://www.marinenet.usmc.mil/marinenet/links/files/MarineNet Content Technical Guidelines Version 3 _2_.pdf (requires CAC). Another resource for those developing for the Joint Knowledge Online (JKO) portal is the *Joint Knowledge Development and Distribution Capability (JKDDC) Knowledge Management Content Design and Development Guidelines* available online at http://www.dtic.mil/doctrine/education/dlcc0407_jkddckm.pdf.

2.2 Establish Working Groups within Communities of Practice (COPs)

Across the spectrum of education and training, there are communities of practice (COPs) where the potential for sharing content is significantly greater. These COPs, such as the Distance Learning Coordination Committee (DLCC), medical associations, specific training or university communities, or consortia, can maximize this potential by determining and adopting some common business practices to better define the finer points for developing and sharing content.

The DLCC is a community of practice with common interests and complimentary curricula; it is made up of professional military education institutions that teach (at least in part) via distance learning. As such, they meet regularly to discuss sharing content and related issues. Additionally, a working group of developers within the DLCC has begun addressing a range of technical issues, including: SCORM data calls and metadata requirements; granularity of SCOs and

packaged/registered content; naming schema; community metadata requirements (i.e., version history, content relationships, and change descriptions); graphical user interface (GUI) and style guidelines; repository accessibility and strategies to share content; and processes to maintain version control. Together, the COP will continue to refine a set of practices that will enable the schools to achieve the cost efficiencies through sharing. Details on the community's initial business rules and sharing strategy are outlined in the SCORM + ID conference paper, *Leveraging SCORM across a Community of Practice*, located at http://blackboard.jfsc.ndu.edu/docs/DLCC_Supporting_Docs/ID+SCORM_Pisel-Lindsey-5-16-07.doc.

2.3 Utilize a Life Cycle Content Management (CM) Approach

Think long-term – it is important to have a plan in place and the tools necessary to maintain digital content over the entire life cycle. Ideally, institutions should utilize a content management system (CMS) in conjunction with version control processes to help track and manage multiple versions of digital content as it evolves. A CMS can offer repository features giving users the ability to search and retrieve content on demand. Whether or not an organization utilizes a CMS, user roles and responsibilities should be documented to direct an organization to efficiently and effectively create, update, publish, archive and retire its content using a systematic approach.

As a prime example, the Navy ILE team has incorporated a comprehensive CM approach and uses a system called the Navy ILE Content Management and Administration Desktop (CMAD). This tool helps manage the distribution of Navy ILE content from inception to delivery; it provides access to submission forms, search and retrieval capabilities, and other CM functionality. When a content object is first introduced to CMAD, it is assigned a unique content identifier (ContentID) and a version number (VersionID) of "1.0". The status of the content object is also tracked (new, final, etc.). Once the content object is finalized and made publically searchable, the owner of the content aka "content sponsor" may have a need to update the content at some point. He/she would then submit a change request that would use the same ContentID, but assign a new VersionID and status. For example, a change request to fix a grammatical error would require an update to the VersionID (making it version "1.1"), but still use the same ContentID. After the new version of the content object is validated and ingested into CMAD, the ILE cyber-librarian (cybrarian) will "finalize" it, which will automatically push the new submission to the ADL Registry. Unless the content sponsors deactivate one of the submissions to disable its visibility, both versions will be available to be searched.

Key to this approach is the ILE cybrarian team. The ILE cybrarian utilizes planned submission, change request, checkout, review support, and distribution processes over the content life cycle. For more information about CMAD, visit https://ile-help.nko.navy.mil/ile/content/supportapps/cmad.aspx. The Cybrarian function is documented at https://ile-help.nko.navy.mil/ile/contentItems/Navy ILE Cybrarian SOP_20070410.pdf.

The requirement for version control management and change notification is critical to any content management system. Changes can occur for a variety of reasons, including scheduled technical updates, content updates, policy changes, system changes, content reuse, mission changes, application context changes or learning objective changes. Users also may encounter times when it is best to keep using an older version or go back to a previous version as necessary.

The best practices for metadata need to support the easy identification of changes related to both the underlying content and metadata. This is especially true when the content is formally registered and available to be reused by others.

2.4 Implement a Versioning Approach

Every organization that plans to maintain and share its content, should implement effective procedures for distinguishing between various versions and revisions to its registered content objects. Besides it just being a good practice to implement version control and naming methods, it is crucial to know what version of what content is available and how said content relates to other registered content in order to provide useful update notifications through the ADL-R.

Based on the research results in this area, most organizations employ a numbering system for versioning content that is typical to software configuration management practices to indicate changes to content. A best practice for maintaining version control (or revision control) of content requires that data be kept on changes made to published content and identified by incrementing an associated 2-digit number for each published version or revision.

In addition, there are numbering standards for other content that may be registered, such as S-1000D Standard Numbering System (SNS). This numbering system can include content version, hierarchy, relationship information, learning objective and task performance information.

Besides being maintained locally, it is recommended that the latest version number of the registered content is included in the ADL-R transaction file under “version” of the Learning Object Metadata (LOM) Lifecycle element. Athough the version element can contain up to 50 alpha numeric characters, this version number should be consistently indicated in a numerical format with at least 2 digits (example 1.2, where the first number “1” is a major revision and the number “2” reflects a minor but material revision).

Key to this approach is defining and distinguishing between what constitutes a revision versus a new version. Research suggests that the term “revision” should be used to reflect any minor updates that do not influence the intent or instructional value of the content, such as typo fixes or font changes. In contrast, versions should denote material changes that are more significant such as procedural changes or accepted terminology, to name a few. The criteria for a material change is a change impacting more than 10% of the original content, or a significant change to the structure, objective, policy or procedures of the content. It is not spelling or grammatical edits.

In general, minor changes to the style sheets, graphical user interface (GUI), metadata or SCORM objects should be considered revisions, and although they should be tracked locally, they don’t need to be registered (unless the owner deems it necessary to inform the user community). The idea is that change notifications will be sent to users wanting to know about major version updates but not each individual revision.

2.5 Register Content Object Metadata

As previously mentioned, DoD Components are expected to register any newly developed SCORM content with the ADL-R. Along with this expectation is the responsibility for those organizations to keep their registered metadata current. If metadata isn't updated, over time more and more data will become irrelevant and users won't have a means to learn about version updates that exist. In theory, searching the Registry should generate any relevant metadata records describing reusable content that is available now within any entity's repository. To find information on relevant content, Registry submissions must include well-formed and valid metadata. This metadata should also be comprehensive and descriptive in nature, containing enough detail for a searcher to have a general idea what the instruction covers, whether or not it would work for their institution, and how to retrieve the content if it is not linked to directly.

The issue of metadata has become more significant with the most recent changes to SCORM. The SCORM 2004, 3rd Edition, Impact Summary (2006) explains that metadata requirements were removed in the latest version of SCORM. Instead of requiring certain metadata to be part of the SCORM content package itself, the ADL Registry now imposes the metadata requirements for SCORM packages. Metadata created for a content object should be stored with that content object's package and registered with the ADL Registry.

Certain Learning Object Metadata (LOM) elements are required in Registry submissions; these include: title, keywords, description, version, author, format, etc. Additionally, submission metadata is required to allow the ADL Registry to process the submission. For example, the submission metadata includes any information on the submitter, type of transaction, content and repository identifiers, and their locations. Users making Registry submissions should review the documentation and follow the guidelines and technical details provided online at http://adlregistry.adlnet.gov/about/developers. Although this documentation provides sufficient instruction to register content, some elements are addressed here for further clarification.

Providing a resolvable location within the <contentLocation> element of the XML transaction file is extremely important. Locations provided are linked to directly from the Registry search results and are intended to be the means for users to access desired content objects. Ideally, this location resolves to the actual content object in a repository. However since the ADL-R is publically available for searching and local repository policies may require authentication and authorization, it is not always be feasible to link directly to the content object itself. As an alternative, ADL suggests that the location "be resolvable to some system or process that will allow the searcher to determine the steps required to obtain the content object" (http://adlregistry.adlnet.gov/about/developers). Linking to a portal or CMS that requires a login does not help the average user unless there are specific directions to obtain access. In instances preventing direct access, organizations should either link to an online content request form or directions to formally request content access. This allows organizations the flexibility to control access (if needed) and maintain a list of potential users outside of its organization. ADL is developing interim best practices for the information to be displayed at the resolvable location. Basic metadata and a prefilled content request form is considered best.

Although unclassified, content may occasionally be taken out of context (such as when part of a scenario) or not considered appropriate for all audiences. For this reason, the Joint Forces Staff

College links to an online content request form (available at http://blackboard.jfsc.ndu.edu/asp/repoform/repoform.asp) to manage access to its registered content. This form requires its users to input contact information, select the desired content items, and describe their need. Submitted requests are then pushed to a database and emailed to a generic email address for the JFSC repository manager (this could easily be changed to a cybrarian, curriculum department, etc.). Once approval is granted, the repository manager can issue access to the object(s) through its repository. As changes are made to the object down the road, JFSC then has the ability to push change notifications to its users as appropriate (although coming Registry enhancements will provide automatic version notification functionality).

Too often users limit themselves to completing the required metadata and don't consider what additions would make it easier for searchers in the long run. If possible, communities of practice should determine whether the required LOM elements are sufficient or if community-level custom metadata should be added to their XML transaction files. For example, the DLCC COP has decided to make the relation element mandatory for linking related content, change descriptions and versions, as well as add metadata describing the JPME level, learning areas, and learning objectives.

Although the granularity of metadata incorporated and the size of registered content objects will vary by nature, it is important to consider alternate target audiences and try to maximize reusability and discoverability of content. Therefore as a best practice, organizations registering content should try to write for a generic audience, that is, as long as it doesn't compromise the integrity of the instruction. Also, they should package and register content at the learning objective level where research suggests there is greater potential for reuse. If the originators are consistently being asked about the assets and/or SCOs that compose the registered package or the package isn't reusable without repurposing the content, it may be worthwhile to package and register at a more granular level.

2.6 Register Version Updates and History

As part of their obligation to keep registered metadata current, owning organizations need to push information about content changes as new versions are made available. In addition to the required metadata, organizations should include details on version history and content relationships in their ADL-R transaction records and update that metadata as version changes are made to registered content. This can be done using a variety of ADL-R operations; best practices on using these operations in regards to version control include:

- INSERT should be used to push a new content record when new content or new version of content (requires a material change in existing content) is introduced. Insert creates a new content metadata instance and allows for the creation of a series of content metadata instances.

- UPDATE should be used for indicating revisions (minor changes to existing content, such as in the case of corrected spelling or grammar). It should not be used for material changes to content metadata whereby more than 10% of the original content has changed;

insert should be used instead. Update is used to push changes to the revision number (e.g. changing 1.0 to 1.1) and overwrite other metadata as necessary.

- DELETE removes an existing instance of registered metadata for the content object. It should almost never be used since it eliminates potentially useful version control data. Fortunately, the delete transaction can only be run by the submitter of that specific metadata instance.

- WITHDRAW should only be used in extreme cases where the content object owner does not want a registered content object to be discoverable. The withdraw transaction removes all of the metadata instances associated with a content object regardless of who submitted the metadata instance.

- DEACTIVATE should be used to temporarily hide an existing metadata instance from searches. This is useful when the metadata instance is no longer relevant for a certain timeframe but it is still required to support the continuity of content or reuse relationships. Use of deactivate prevents the object from being used again but maintains existing references. Using deactivate only hides the identified metadata instance from searches; the handles will still resolve and any other metadata instances describing that content object remain discoverable.

- ACTIVATE should be used to activate a previously deactivated content instance.

- MOVE should only be used when the content object location changes. An example would be if the URL changes because the content is moved to a new repository.

When expired content is superseded by another content item, it should be withdrawn from use, archived and retired. To do so effectively, use INSERT to create a new content metadata instance for the new version and relate it to the expired content (this will be explained in detail in the section 2.7). Next, UPDATE the lifecycle status on the expired content to out of service or program terminated (whichever fits the organization's reasoning) and then use the WITHDRAW operation to remove the expired content from search results.

Each Registry submission contains useful information related to version control. The key metadata elements currently available to support version control are described below. The relevance to version control is described and, where relevant, the feasibility and ease of future data migration is addressed.

- <contentIdentifier> uniquely identifies the content object being registered. This identifier shall be defined in handle syntax. This can include a version reference after the handle if desired by the repository. Each time a submission is made with a new content identifier a new content instance is created. This allows for the search and identification of all updates related to a content item.

- <Timestamp> indicates the date and time that the Registry transaction was created. This value is a non-negative timestamp. This should not be used for the version control content creation date but may be used for metadata ordering.

- <LOM:Title> describes the title for the learning object being registered. This can include a version reference or an equipment series or mod number, which can provide quick version control reference.

- <LOM:Description> provides a narrative describing the object being registered. This important element should include a specific description of the change from the last version.

- <LOM:Lifecycle:Version> identifies the version number of the object being registered. This version number should be consistently expressed as a numerical format with at least two digits (Example: 1.2, where 1 is a major revision and 2 reflects a minor but material revision).

- <LOM:Lifecycle:Status> explains the status of the object being registered. The restricted values for this element are: query only, proposed development, under development, development or acquisition completed, program revision, out of service, program terminated, other. This can help identify the evolution of content versions and can be used for a custom version control report by status phase.

- <LOM:DateTime> describes the date the content was created. This should consistently be the date of completion of the content instance for the lifecycle status declared.

- <LOM:ContentType> identifies the type of content object being registered for a transaction. The Entry element is the means for describing the type of object. This element's value holds the value that identifies the content object being registered. This element's value shall be one of the following character strings (these are language independent tokens): asset, SCO, or aggregation. This element is required to establish the hierarchical relationship of the content and can also be used to identify specific changes within a SCO or aggregation.

- <LOM:Relation> identifies content relationships. The LOM metadata element <relation> can be used to define the content identifier to which the new content item is related. It may define either hierarchical relationships (example: asset to SCO) or reuse relationship (example: SCO to SCO). This is important when combined with other elements. This is accomplished by defining the type of relationship, for example as "**Relation**.ispartof", "**Relation**.haspart" or. "**Relation**.isversionof". These are described in detail in the next section.

For further details about the ADL-R LOM profile (including additional elements and descriptions), refer to the ADL-R LOM Cardinality document located within the technical reference section of the Registry (http://adlregistry.adlnet.gov/about/developers), or download it from http://www.adlnet.gov/downloads/DownloadPage.aspx?ID=307.

Keep in mind that any relevant version information should be included in a content object's metadata, such as associations to related content and prior versions. Specifically, organizations should incorporate data describing version history (version numbers, dates, and change descriptions), content genealogy and content hierarchy.

Content genealogy defines the reuse relationship between content. For example, a particular content object may have been reused, referenced or modified numerous times. In this case, any new metadata instances should indicate where the content object originated.

Content genealogy helps define a community of interest for the content and version control alerts by tracking the owners and authors of each reuse instance. Additionally, conveying genealogical relationships can help content developers identify case examples and opportunities for shared funding of common requirements.

One feature that is desirable for the version notification system is the ability to view content according to reuse ancestry/genealogy. This allows a search to display all copies and references to a given content instance, and then view the owners of that reused content.

Content hierarchy is important to version control because it defines the relationship of content, such as aggregation to SCOs and SCOs to assets.

Any content component may require a version change and thus may impact related hierarchical content. In addition, reuse of a specific content component requires that owners and users of that particular component be notified of a change. The result of the change notification may be changes to related components or the context of reuse.

One issue that presents a challenge to the ADL-R user community is inconsistency in the content hierarchy taxonomy. While the ADL-R version 1.6.1 schema defines aggregation, SCO and asset as the recommended vocabulary describing the content type, several repositories use different vocabularies. The Army Training Support Center (ATSC) for example, uses course, module, topic and asset. The S-1000D also uses different hierarchy descriptors. Therefore, ADL-R version notification features should allow for hierarchical display according to the applicable model.

2.7 Use the Relation Metadata Element to Describe Content Relationships

The ADL-Registry transaction file accepts the Institute of Electrical and Electronics Engineers (IEEE) LOM standard for submitting data about a content object. Certain LOM elements are required, but any of the LOM elements can be used.

The *relation* element of IEEE LOM is used to describe relationships between the content object and other content. These could be versions of the content object, uses parts of, etc. As a best practice, use this element in all registered content objects to keep track of where the content originated, what version is the most recent, and what has changed in the latest version. The relation element contains three parts: kind, identifier, and description. The *kind* element is used to identify what type of relation is being referred to. There are two child elements under the kind element: source and value. The *source* element contains the current version of the LOM (i.e. LOMv1.0). The *value* element describes the type of relation. Acceptable entries (vocabulary tokens defined by the LOM) and definitions are:

Value	Definition
ispartof	The content object is part of another object or lesson
haspart	The content object has part of another object or lesson
isversionof	The content object is a version of anther object
hasversion	The content object has a version of another
isformatof	Is a format of another content object
hasformat	Has format of another content object
references	References another
isreferencedby	Is referenced by
isbasedon	Is based on
isbasisfor	Is basis for
requires	Requires
isrequiredby	Is required by

The *identifier* element informs the user where the content object came from (e.g. whether it is referenced, based on, etc.). The identifier element contains two child elements catalog and entry. The *catalog* element describes the type of entry (Content Identifier, URI, ISBN, etc.). The *entry* element contains the Content Identifier number, website or ISBN number.

Lastly, the *description* element holds the string element. Although anything can be placed in this field, at a minimum it should contain a description of what has changed within the content object and the corresponding version numbers. Ideally, this element should also provide content relationship data to enable those monitoring changed content with enough information to decide whether to access the new content version.

The following code is an example of using the relation element:

```
<relation>
  <kind>
    <source> LOMv1.0 </source>
    <value> isversionof </value>
  </kind>
  <resource>
    <identifier>
      <catalog> Content Identifier </catalog>
      <entry> 100.50.06.02.01/S001_ADL </entry>
    </identifier>
    <description>
      <string> 2007/10/10-- a description of the target learning object -- </string>
    </description>
  </resource>
</relation>
```

There are no restrictions to the number of relation elements used within a XML transaction file submitted to the Registry. As a rule, metadata should include separate relation elements describing each content object relationship.

2.8 Set Up Version Notifications for Relevant Content

It is critical that a COP have visibility of any changes made to content objects that are directly applicable to their curricula. Aside from the obvious need to track versions of a given object, there is a cost efficiency to consider. Versions are especially important if somebody else is using the object. Proper tracking of objects will enable the originator to see whether an object is being used or not and will empower them to decide whether they need to register the change as a new object or to make changes without modifying the Registry data.

Communities can be dynamically defined based on an analysis of metadata and reuse within the Registry. Authors, repository managers, owners of reused or referenced content can all be grouped as a COP related to a particular content instance, collection, content identifier, mission area and/or taxonomic expressed interest during registration, alert creation and search.

The ADL-Registry Client (ADL-RC) is a means to ensure that adopted or adapted content remains current. SCORM users that adopt or adapt content objects from other sources will add Registry transaction data to a locally resident database. The program then performs periodic comparisons of versions registered in the ADL-Registry with data in the local database. If a more current version is registered, the system notifies the ADL-RC user. Another effort is underway to enhance the ADL-R portal with version notification functionality. When available, users will be able to sign into the Registry, subscribe to email alerts, and maintain visibility of changes according to their set preferences.

Version control and automated notification is critical to the success of sharing content via the ADL-R. Any time a user adopts or adapts a piece of content from an external source there is a degree of risk assumed. If the content of that object becomes dated and inaccurate and the new user has no means to see that a change has occurred, it is a failure. Users will only allow a limited number of failures before abandoning shared content for the assurance of creating and maintaining their own content. If the ADL initiative is to succeed, the risk must be mitigated.

3 REFERENCES

Advanced Distributed Learning (ADL). (n.d.). *ADL-R: The basics*. Retrieved August 10, 2007, from http://www.knowledge-media.com/synergy/contentexport/651/6/index2.htm

ADL. (n.d.). *ADL Registry submission operations.* Retrieved August 10, 2007, from http://www.knowledge-media.com/synergy/contentexport/651/5/asset/adl registry submission operations.doc

ADL. (2006). *SCORM 2004, 3rd Edition: Impact summary.* Retrieved March 1, 2007, from http://www.adlnet.gov/downloads/DownloadPage.aspx?ID=161

Department of Defense (2006). *DoDI 1322.26: Development, management, and delivery of distributed learning*. Retrieved March 2, 2007, from http://www.dtic.mil/whs/directives/corres/rtf/132226x.rtf

Fogel, K. (2007). *Producing open source software: How to run a successful free software project.* Retrieved October 4, 2007, from http://producingoss.com/html-chunk/vc.html

Hickey, J. (2006). Version control vs. configuration management, *CM Journal*. Retrieved September 28, 2007, from http://www.cmcrossroads.com/articles/cm-journal/version-control-vs.-configuration-management.html

Institute of Electrical and Electronics Engineers (IEEE). (2002). *Draft Standard for Learning Object Metadata.* Retrieved September 4, 2007, from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Intelligent Automation. (2006). *Use case: Finding content in the ADL-R.* Retrieved Oct 1, 2007, from http://www.s1000dscorm.org/usecases/Finding Content in ADL Registry.asp

Navy ILE. (2007). *Content developer's handbook.* Retrieved September 28, 2007, from https://ile-help.nko.navy.mil/ile/content/policy/handbook.aspx

Navy ILE. (n.d.). *Content Management and Administration Desktop (CMAD).* Retrieved September 28, 2007, from https://ile-help.nko.navy.mil/ile/content/supportapps/cmad.aspx

Navy ILE. (2007). *Standard operating procedures (SOP) for the Cybrarian function.* Retrieved September 28, 2007, from https://ile-help.nko.navy.mil/ile/contentItems/Navy ILE Cybrarian SOP_20070410.pdf

Northrop Grumman. (2007). *Final report for ADL Registry Client*. Retrieved Sept 10, 2007, from http://www.jointadlcolab.org/downloads/research/2005/adlr_client/adlrc_final_report.pdf

Northrop Grumman. (2007). *Research results for the ADL Registry version control system (VCS) requirements gathering questionnaire*. Retrieved September 10, 2007, from http://blackboard.jfsc.ndu.edu/docs/research_papers/research_results_9-10-07.doc

Pisel, K., & Lindsey, A. (2007). *Leveraging SCORM across a community of practice*. Retrieved September 4, 2007, from http://blackboard.jfsc.ndu.edu/docs/DLCC_Supporting_Docs/ID+SCORM_Pisel-Lindsey-5-16-07.doc

Powell, A., Johnson, P., Campbell, L., & Barker, P. (2005). *Guidelines for using resource identifiers in Dublin Core metadata and IEEE LOM.* Retrieved September 28, 2007, from http://www.ukoln.ac.uk/metadata/dcmi-ieee/identifiers/

Washington State Library. (n.d.). *Digital best practices: Indexing & metadata for digital objects.* http://digitalwa.statelib.wa.gov/newsite/collection/indexing.htm

Wikipedia. (2007). *Content management system.* Retrieved September 28, 2007, from http://en.wikipedia.org/wiki/Content_management_system

Wikipedia. (2007). *Revision control.* Retrieved September 28, 2007, from http://en.wikipedia.org/wiki/Revision_control

4 Appendices

4.1 Additional Resources

ADL	http://www.adlnet.gov
ADL-Registry	http://adlregistry.adlnet.gov
ADL-R Community Resources	http://adlcommunity.net/course/view.php?id=7
ADL-R Technical Reference	http://adlregistry.adlnet.gov/about/developers
ADL-R Help	http://adlregistry.adlnet.gov/help
ADL Policy & Guidance	http://www.jointadlcolab.org/support/policy/index.aspx
Joint ADL Co-Laboratory Research & Development	http://www.jointadlcolab.org/research
Business Rules, Best Practices & Examples for Army SCORM Compliant Courseware	http://www.atsc.army.mil/itsd/imi/bus_rules.asp
Navy ILE Standards, Specifications, Policy & Guidance	https://ile-help.nko.navy.mil/ile/content/policy.aspx
Air Force Advanced Distributed Learning Hub	https://adlhub.golearn.csd.disa.mil
JKDDC Knowledge Management Content Design and Development Guidelines	http://www.dtic.mil/doctrine/education/dlcc0407_jkddckm.pdf

4.2 Acronyms

ADL – Advanced Distributed Learning

ADL-R – ADL Registry

ADL-RC – ADL-Registry Client

ADL-RUG – Registry Users Group

ATSC – Army Training Support Center

COP – community of practice

CM – content management

CMAD – Content Management and Administration Desktop

CMS – content management system

DoD – Department of Defense

DoDI – Department of Defense Instruction

DLCC – Distance Learning Coordination Committee

FEA – front-end analysis

GUI – graphical user interface

IEEE – Institute of Electrical and Electronics Engineers

ILE – Integrated Learning Environment

ISBN – International Standard Book Number

JADL – Joint ADL Co-Laboratory

JFSC – Joint Forces Staff College

JKDDC – Joint Knowledge Development and Distribution Capability

JKO – Joint Knowledge Online

JPME – Joint Professional Military Education

LOM – Learning Object Metadata

SCO – shareable content object

SCORM – Sharable Content Object Reference Model

SOP – standard operating procedures

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

XML – Extensible Markup Language