Final Report

for

ADL Registry Client (ADL-RC)

Version 1.0

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

1 March 2007

Table of Contents

Table of Contents ii
Revision History iii
1. Introduction 1
1.1 Executive Summary 1
1.2 Background 2
1.3 Purpose 3
2. Tasks & Products 4
2.1 Task Breakdown 4
2.2 Deliverables Summary 5
2.3 Presentations 5
2.4 Published Papers 5
3. Research & Development 6
3.1 Research Strategy 6
3.2 Design Strategy 6
3.3 Major Accomplishments 8
3.4 Major Obstacles 9
3.5 Technical Shortfalls 10
3.6 Lessons Learned 10
3.7 Process Feedback 11
3.8 Assumptions & Recommendations 11
4. Next Steps 13
4.1 Transition Prototype to User Community 13
4.2 Expand Version Control Initiatives 13
Acronyms 14
References 15

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Executive Summary

The Advanced Distributed Learning (ADL) Initiative and its Shareable Content Object Reference Model (SCORM) offer the education and training community tremendous potential for content object reuse and concomitant cost savings. Of equal importance is the potential for SCORM to allow for leveraging relevant expertise from multiple sources. Achieving this potential has been an evolutionary process as more content and learning management systems (LMS) become SCORM conformant. Implementing Department of Defense Instruction (DoDI) 1322.26 "Development, Management, and Delivery of Distributed Learning" is the next step; as part of this instruction, DoD organizations are expected to comply with SCORM, register content, and store content in repositories. This mandate promotes the sharing of standardized learning content among Services and other contributors, in an attempt to reap the economic benefits brought by using content that is accessible, adaptable, affordable, durable, interoperable, and reusable. Performing simple searches within the ADL Registry (ADL-R) provide users a means to discover appropriate content that can be used as is, or repurposed (customized to fit the new audience) – either way, reducing the costs and effort incurred by creating the content object from scratch.

Although users can now search and discover information about reusable content, the ADL-R itself (in its current form) has no way of addressing the long-term sustainment of courseware. For instance, users cannot yet log into the Registry to monitor content they are interested in and check for updates to specific learning content. On the other hand, we also can't expect all developers and/or content owners to monitor the use of their content and push changes to those in the community that might be interested. Ideally there needs to be an automated process that alerts users to changes in that content to support individual organizations' long-term requirements. Without such notification and readily available updates, shared content may soon become outdated.

Since November 2005, Northrop Grumman and the Joint Forces Staff College (JFSC) have taken an active role to solve these issues by developing the ADL Registry Client (ADL-RC) prototype application with support from the Joint ADL Co-Laboratory. The ADL-RC automates the long-term maintenance of registered learning objects. When a change to shared content is noted by the system, ADL Registry users running the ADL-RC are notified that the newer version of the learning content has been registered in the ADL Registry (ADL-R). Users may then retrieve the new learning content as required. This tool ensures users are aware of the most current versions of the learning content they discover using the Registry.

The ADL-RC offers a means to keep shared learning content both current and accurate. The initial prototype of the ADL-RC now runs at JFSC proving the concept's validity. We are now at the stage where it could be implemented at other organizations in the ADL community with minor configuration adjustments. Additionally, the open source code can be shared and expanded to maximize usability, promote reuse, and help realize the vision of automating version control among the ADL user community.

1.2 Background

JFSC's Advanced Joint Professional Military Education (AJPME) blended learning course contains roughly 50 online lessons. The AJPME distance learning instruction was the first SCORM-conformant joint professional military education (JPME). It was developed by Northrop Grumman in 2003 and is since maintained by two key personnel on the ADL-RC project, Mrs. Angela Lindsey and Mr. Greg Frederick. Both Angela and Greg work onsite at JFSC in its Distance Learning Division (DLD) under the direction of Dr. Ken Pisel, dean of the Joint Continuing and Distance Education School (JCDES) and principal of the Distance Learning Coordination Council (DLCC) (a subset of the Military Education Coordination Committee). Together we have combined experience in the multiple facets of the distance education field, largely analysis, design, courseware development, implementation, and management. We fully support the ADL initiatives and strive to leverage SCORM and promote sharing amongst the military education schools to realize its potential.

Prior to proposing development of the ADL-RC prototype, our Northrop Grumman team worked on supporting development initiatives that included creating a searchable AJPME Learning Object Repository and a Thesaurus of Joint Indexing Terms. We routinely debated SCORM and ADL initiatives, looking specifically into how to engage the community to the point where the Services are sharing content and not just talking about sharing content. We identified some challenges that can potentially stifle this vision. Specifically, we have ongoing concerns about overcoming taxonomy and currency issues.

Currently, terminology used in registered metadata must match the term searched in order for it to be discovered. Ideally, the Registry should determine if synonyms or similar terms are searched and discover additional related objects that may meet the searchers' needs. Another suggestion is to allow registered content to be categorized and searched by broader topic and to allow users to subscribe to alerts that would push data about relative subject-matter as it is registered. This remains a hot issue and we are proposing a similar "subscription service" in our FY07 R&D proposal to the Joint ADL Co-Lab.

As for the currency issue, the Registry primarily supports the front-end effort of searching and discovering information about reusable content and doesn't address changes to shared content necessary for the sustainment of content over the courses' lifecycles - such as how a user can keep tabs on borrowed content that's updated periodically. Although the Registry is still in its infancy, it is vital to add versioning controls and automate update notifications in the near term. If an institution has collected registered content from various organizations (either unchanged or as the foundation for a new content package) and is employing it as a link or within its own LMS or SCORM run-time environment (RTE), there must be a mechanism that alerts the borrowing institution to a change. Without an automated process, the institution will most likely deploy the out-of-date content when newer content is available, or it risks unknowingly linking to content that no longer meets its need. If this happens, we risk the ADL vision failing as institutions rely solely on developing their customized content without the intent to share amongst the community. To offset this risk, our team envisioned creating the ADL Registry Client (ADL-RC), a tool that periodically queries the ADL Registry database to determine whether any content drawn from other repositories has changed.

Backed by JADL, our Northrop Grumman team at JFSC led the effort to design and develop an open source prototype of the ADL-RC to be shared within the ADL community. Although the initially released application of the ADL-RC was designed to run on the Joint Forces Staff College (JFSC) server, the code is open source and can be customized to work at various institutions running other server platforms.

The resulting prototype compares metadata of learning content that is both formally registered with ADL and collected locally at JFSC. When the system notes a version change to any of the selected content items, the ADL-RC sends an email notification to alert its user that a newer version of the learning content is available. Once notified, the organization then has the option to retrieve the new learning content if desired.

The ADL-RC prototype was developed using object oriented methodologies; it uses cutting-edge, cross-platform technology that will make version control seamless and uncomplicated. Additionally, the ADL-RC source code will be open source and available for all to download, modify, and enhance. With this approach, we expect widespread usage across the ADL community.

1.3 Purpose

The purpose of this project was to ensure that Registry users have access to the most current and up-to-date versions of the learning content they discover through the ADL Registry. The ADL-RC prototype is one means to provide users this much needed visibility of version control changes. It is the single tool designed to support the long-term maintenance of SCORM content packages within the ADL community.

2. Tasks & Products

2.1 Task Breakdown

The ADL Registry Client project was divided into six primary tasks. The first five tasks were funded initially by the Joint ADL Co-Lab as part of its FY05 prototype program. Based on results of the prototype, task six may be executed down the line to transform the working prototype into a customizable application and enhance the version control capabilities for the larger ADL community. The task breakdown is depicted in Table 1.

TABLE 1. TASK SUMMARIES

Task	Title	Description
1	Research and Analysis	The key players met to initiate planning and establish communication. Afterwards, the development team began research to fully understand the interworkings of the ADL Registry and its supporting concepts and applications. As part of this task, the team provided management/staffing plans, progress reports, a milestone chart, and requirements documentation. This documentation captured the shared vision of the ADL-RC prototype and laid out expectations (or requirements) the prototype would employ.
2	System Design	The ADL-RC team used the requirements documentation as a baseline to create a detailed software design document of the prototype. Once approved, development commenced.
3	Prototype Development	The prototype was programmed to run at JFSC according to the agreed-upon design documents. The developed prototype: 1) parsed through local metadata; 2) connects to ADL Registry and pulls out data; 3) compares data; and, 4) creates notifications client side.
4	Quality Assurance	Third-party testing and quality assurance (QA) measures were performed by LSI to systematically assess the ADL-RC prototype application to make sure it functions between the ADL Registry and the JFSC repository. Errors and issues were noted on test incident reports (TIR) and fixed/closed prior to prototype delivery.
5	Prototype Delivery	The team demoed the prototype at the Joint ADL Co-Lab and presented JADL with a prototype user manual describing configuration and operations specifications, as well as the working application code.
6	System Enhancements	If opted, the prototype will be expanded and enhanced to the envisioned ADL-RC application based on feedback from task 5. Users throughout the ADL community will be able to download it online to maximize utilization.

2.2 Deliverables Summary

Besides the project schedule and monthly progress reports, the key deliverables provided to JADL as part of the ADL-RC research and development project included:

- System Requirements Document
- Software Design Document
- Project Final Report
- User Manual
- ADL-RC Prototype & Code
- Prototype Presentation & Demo

2.3 Presentations

The ADL-RC team briefed members of the JADL Co-Lab and the Registry user community to explain version control issues brought on by sharing content and the ADL-RC solution. In addition to three briefs to the JADL Co-Lab, we presented at the Joint ADL Implementation Fest 2006 and the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) 2006.

2.4 Published Papers

Lindsey, A., Pisel, K., & Pike, W. (2006). *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), Vol. 2006*. The ADL Registry Client: Solving the Currency Issue of Content Reuse. Retrieved February 13, 2007 from http://ntsa.metapress.com/link.asp?id=4pj8jbt535f08l06

3. Research & Development

3.1 Research Strategy

Since the Registry was not operational when we began this project, our initial approach was to learn technical details about the upcoming Registry and to get as many technical opinions as possible. Our research began mid 2005, when the Director of the Alexandria ADL Co-Lab, Mr. Paul Jesukiewicz, and the ADL Chief Architect, Mr. Philip Dodds, invited JFSC to join the ADL Registry Pilot group. This pilot group consisted of various stakeholders from the ADL Co-Laboratories, Defense Technical Information Center (DTIC), Corporation for National Research Initiatives (CNRI), Carnegie Mellon University (CMU), Knowledge Media Inc. (KMI) and other military training and education institutions.

The pilot group was initially led by Mr. Dodds and most early decisions were approved by him. The pilot group email distribution list and weekly phone conferences helped us understand the architecture of the ADL-Registry and kept us up to date on the progress and changes to the Registry.

During the pilot phase, multiple versions of the Registry were tested prior to its public release. Based on its analysis, the JFSC team submitted two key requests to help distinguish version changes; this included adding the capabilities to capture change description information and timestamps within submitted transaction files. The timestamp was added to the transaction file but the change description was pushed to the “B list” and has since been denied due resistance in deviating from the Learning Object Metadata (LOM) specification.

After the ADL Registry was deployed officially at DTIC, the pilot team members formed the new ADL-RUG, or ADL Registry Users Group. We have continued to be active members and have since kept track of impending changes to the Registry that impact how third-party tools such as the ADL-RC will interact with the Registry.

Although we were specifically concerned with automating ADL Registry searches to pinpoint content object updates, we also performed research to address interoperability issues and solutions across platforms, applications, and programming languages. Additional analysis was conducted to determine how the required metadata in the registration process supports version control efforts and what information to compare to determine changes have been made to shared content.

3.2 Design Strategy

The ADL-RC runs a data comparison process that reviews downloaded transaction data at the client site and data in the ADL Registry database. It will periodically execute the comparison on a set schedule (hourly, daily, etc.) set by the user through its Task Scheduler. When run, version information from a local copy of the XML is compared to corresponding data in the ADL Registry database.

When a content package is registered in the ADL Registry, it will be given a unique ID. This ID will be the primary key for lookup in the database. When the ADL-RC detects differences between the local data and Registry data, a notification is created and emailed to the user and information about the run is logged locally in a separate file. Figure 1 below illustrates the design flow for this process.

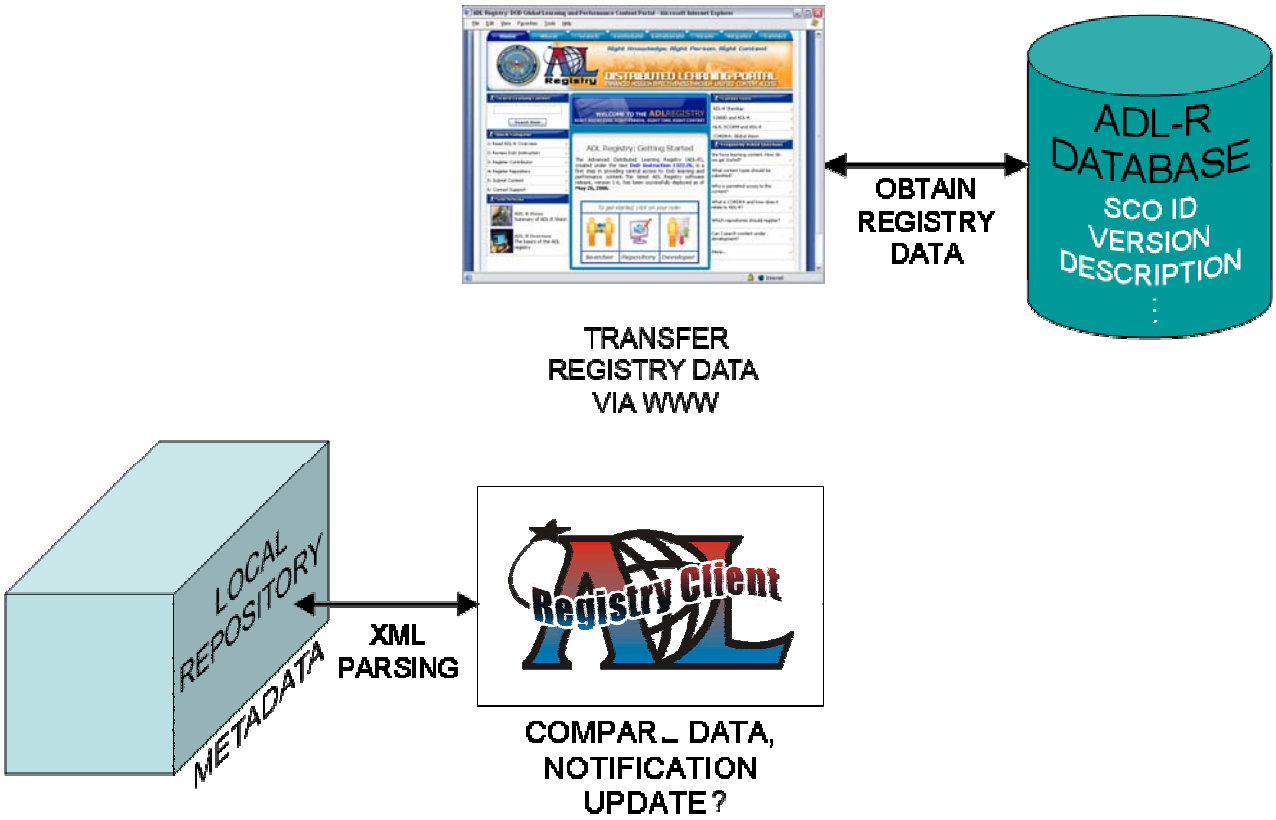

Figure 1. Information Design Flow

The ADL-RC was developed as open source in the Java programming language. The program needs to be installed on a personal computer or server running Windows with the Java Runtime Environment (JRE) 1.5.0_06 installed. The machine also needs access to the ADL Registry through an Internet connection and access to the local repository.

Some basic considerations that impacted the application's design included: system requirements, design methodology and pattern, and network restrictions or security risks. The ADL-RC uses object-oriented methodology to promote reusability and maintainability. This methodology allows the application to be more reliable, robust, and extensible. The design pattern for the ADL-RC application is the model-view-controller; this pattern offers a unique benefit by separating the presentation code (GUI) from the business logic code (application logic), and from the data access code (Value Objects).

The ADL-RC consists of three main packages – the Installation Utility, Configuration Change Utility, and Core Utility. They each import a common package of components (classes) used by one or more of the utilities to prevent duplicate coding. A summary of these packages is included below. For detailed design information, see the ADL-RC Software Design Document (SDD).

The ADL-RC Installation Utility runs from an executable file in order to guide users through the ADL-RC installation process. It is an InstallShield®-type program designed to step users through the various configuration options and set up the Task Scheduler.

The ADL-RC Configuration Change Utility allows users to modify properties set during the installation. It will be accessible through Windows main menus (Start → Programs → ADL-RC).

The ADL-RC Core Utility performs the key operations; it connects to the ADL Registry, compares data for shared content packages, determines if updates exist, and produces email notifications when newer content packages are discovered. It will also create a log file with details from each run of the program. The Core Utility automatically runs based on user selections within the task scheduler component. It also has an option to run manually in a Windows environment through a typical menu selection process or via a desktop shortcut.

3.3 Major Accomplishments

As a part of the ADL Registry pilot group, we were able to help with testing and provide user feedback before the Registry launch. We submitted content packages to several versions of the Registry and connected to them using the HTTP post interfaces. This not only helped us develop the prototype, but it helped develop the Registry itself. Because there was no way of telling when the submission had taken place, we requested to have a timestamp added to submission XML files; this was implemented as part of v.1.5.1. “Rolling with the punches” of the different versions, hosting sites, and overall changes proved challenging and kept us on our toes. Additionally, our exhaustive testing of the HTTP post interface helped recognize the need for a new Registry Interface Mechanism (RIM). The RIM-Lite version, and eventually, the ADL-RIM, will allow third-party tools to securely automate interaction with the Registry.

We have met our deliverable schedule and are now able to successfully run the ADL-RC prototype from JFSC in accordance to our design requirements and specifications. It currently connects to the Practice Registry as suggested by Mr. Bill Redeen, the Registry User Support manager. When it checks for updates (either as scheduled or when run manually), the program compares Registry data online with the local XML and notifies the user via email and text log to changed metadata. It does not need access to the actual content because it is instead reviewing metadata describing the registered item. If no changes are discovered, the user will not receive an email notification but the results are still written to the text log. Also, the executable install feature and configuration settings can be set by the user. Some changes may be required to run the program as configured at another organization, but the prototype concepts were proven valid and the code can be changed as necessary.

3.4 Major Obstacles

There were several major obstacles that impacted this project, including: design changes, schedule delays, connection issues and post interface limitations, and employee/POC turnovers.

When designing the ADL-RC we were using some early documentation of the Registry. This was changed sometimes without us knowing and with no chance to provide input. For example when we started writing the design document, the Registry would return the original Registry Transaction (Reg-T) XML file in the search results. This was changed in one of the version updates and it forced us to redesign our comparison process. Our initial design had called for comparing a local submission file with the submission file contained in the Registry database. This would have been an easy one-to-one comparison to check for updates. Unfortunately, the technical pilot group containing the ADL-R staff and developers (which we were excluded from) made these decisions and we weren't always kept in the loop on changes that affected our prototype. We were told that this group primarily talked about scheduling and deploying the Registry. When we learned of the submission change, we decided to focus our comparison on the "Version" portion of the registered LOM.

Automating the connection with the Registry was challenging. Limited flexibility with the post interface began as an obstacle but led the ADL team to a major accomplishment by creating the ADL RIM-Lite. The moving of the Registry from KMI to DTIC also changed the post interface we are currently using. This was a last minute update because of the delay transitioning the Registry to DTIC. Additional changes down the line to the RIM and Registry are expected to affect the Registry connection on future versions of the ADL Registry Client.

The National Defense University (NDU) performed major network maintenance in December 2006 that disconnected the .mil domain from the university. Because the Joint Forces Staff College is a subset of NDU, JFSC lost .mil access. This prevented us from reaching the adlregisty.dtic.mil site during this period, which did not stop our work but did cause minor delays.

Unfortunately, employee turnover was another hurdle that impacted our project. When we received the initial award, two of the key Northrop Grumman personnel we had proposed for the work were no longer available. It took several weeks to find both a qualified system architect and Java programmer. Although this delayed our start on the prototype effort, corresponding delays in the DODI signing and Registry development would have also slowed progress so the timing actually worked out.

Another unforeseen obstacle was the sickness and leave of absence of Mr. Phil Dodds. He was the go to guy for all answers and most discussions we had with technical staff at CNRI and KMI ended with "let's run this by Phil". Mr. Avron Barr later took over of the pilot group and eventually the ADL-RUG. We did not find out he had taken Phil's place until several months after. Avron has been very helpful since he joined the team.

Finally at the end of October 2006, Ms. Kathryn Johnson (our Java programmer) resigned and it was nearly impossible finding a qualified replacement so far into the contract without guaranteed follow-on work. Although she had done a majority of the required programming, we needed a hand to tie up any loose ends and be available to troubleshoot and fix any issues raised during the testing

phase. After several weeks of accepting resumes and not getting any worthy of interviewing, we had to complete the work given our current resources. We were fortunate to arrange the necessary Java support through LSI, our subcontractor. They were an integral part of completing the effort as contracted and on schedule.

3.5 Technical Shortfalls

Security is always a technical shortfall these days and we came across some security issues. Getting a program approved by any Service installations is very hard and time consuming. The Navy Marine Corps Intranet (NMCI) would most likely ban the installation of the ADL-RC. Users machines are locked to software installs and only approved software on its gold disk are loaded. We attempted to research NMCI's requirements for approved software but they were very secretive. The response we got was that they send any software to their testing lab in San Diego, but because of security reasons they could not tell us what they test.

Another technical shortfall dealt with both delays and access to *current* documentation. For instance, the original HTTP post interface document was delayed which caused us to delay coding. Additionally, we could not code the ADL-RC to use the new RIM-Lite because the specs were not available when needed. The delay in moving the Registry to DTIC also caused a last minute change to our HTTP post interface. This also delayed testing, and therefore, the testing phase was shorter than planned.

Network restrictions may exist when the ADL-RC acts to send email notifications. Most email is sent through port 25; unfortunately due to massive spamming in recent years, many computers are configured to block mass emails to port 25. This could potentially hamper attempts by the application to email change notifications. Users may bypass email restrictions by going through their local network administrators or checking ADL-RC log file(s) manually. Results of the application "run" will be written and saved to a log file that resides on the local machine. The log will contain data such as when the comparison occurred and what content changes (if any) were discovered. Additionally, any errors or problems executing the application will be written to the file.

In order to overcome these network restrictions and maximize usability within the ADL community, we decided to use a commercial email server. We created a Google mail (g-mail) account and hard-coded it into the program. Without modifications to the prototype, email change notifications will be sent from adl.registry@gmail.com. This could be changed to a local email server with minor changes to the code.

3.6 Lessons Learned

Ideally, we wouldn't have begun the prototype design until the Registry was complete or at least much closer to public release. We initially began our research during ADL-R version 1.2 and it was recently updated to 1.6.1 as of late January 2007. Although there are some advantages to getting an early start, we were constantly trying to stay on top of the changes and trying to hit a moving target.

The current ADL Registry version 1.6.1 allows registered users (content developers, repository managers, etc.) to register reusable learning content by uploading transaction files containing required metadata elements to describe the content. Users are first expected to register objects within the "Practice Registry" environment and then graduate to use the "Operational Registry" or ADL-R when they have successfully mastered the registration process. We had fully expected to use the actual ADL-R, but were told to connect to the Practice Registry for demo purposes. This allows us the opportunity to submit mock transaction files solely for testing purposes.

Another lesson learned is that the ADL-RC application won't be a solution for the entire ADL community due to security restrictions. Also, further changes to the HTTP post interface specification/RIM may require further changes to the ADL-RC. To bypass these limitations, we recommend enhancing the Registry itself by adding a web-based version control system that includes a subscription service allowing users to subscribe to alerts to watch content objects or content categories.

3.7 Process Feedback

The prototype process in and of itself was extremely organized and we didn't require much direction past the kickoff meeting and contract to understand the Joint ADL Co-Laboratory's expectations of Northrop Grumman.

In the future we recommend that JADL make formal announcements to the other ADL Co-labs and subsidiaries to advise interested parties on what prototypes were awarded and request that parties provide technical expertise/assistance as necessary. It was imperative to work closely with members of the Joint ADL Co-Lab, the Alexandria ADL Co-Lab, and its subsidiaries to complete the prototype so it would be compatible with the ADL Registry. Frequently over the course of the project, we needed to ask the right people questions and get answers and/or documentation to progress. Unfortunately, getting questions to the right people and getting the timely resolutions was not always easy. Several people we encountered did not know what the prototype program was nor did they understand we were working for JADL. Perhaps getting the word out early to affected parties would have helped us get the support we needed.

3.8 Assumptions & Recommendations

Organizations that wish to use the ADL-RC should adhere to DoDI 1322.26 requirements. To do so, they will be expected to conform to set standards, as well as register their sharable content and store that content in searchable repositories. Repository managers or those registering the content must understand XML to create the Reg-T files, and required procedures to upload to the ADL Registry.

In order for the ADL-RC to be a success, it is imperative that organizations (or content owners) make a conscious effort to employ meaningful metadata in their submissions to the Registry. Additionally, organizations will have to define the version initially and update the metadata record when the content changes over its lifecycle. In other words, it's not enough to register your objects once and forget about them. If content has been updated locally, push the version change to the Registry so the ADL-RC can recognize the update to notify its user.

Communities of practice (COPs) that expect to share content amongst similar organizations (for instance, the professional military institutions forming the DLCC) should form their own guidelines to promote sharing and version control efforts. Ideally each institution would employ the ADL-RC in a similar fashion and consistently push any version changes to the Registry. Along the same lines, COPs should consider whether they want to add community level metadata, or define custom tags to further explain connections between related data and changes from one version to the next.

4. Next Steps

4.1 Transition Prototype to User Community

Remember that the ADL-RC is still in a prototype stage and requires minor customization to make it work for other organizations. When the RIM is complete and the ADL-RC has been updated accordingly, integration of the ADL Registry Client into the ADL community will be almost effortless. There are no modifications needed to existing SCORM packages. No expensive hardware is required to run the application and no additional software needs to be purchased. All of the information required by the ADL-RC is available from the existing metadata submitted to the ADL Registry. Additionally, the ADL Registry will supply the packages with unique identifiers and the required transaction files are all the ADL-RC needs to carry out its processes.

The ADL Registry Client prototype can be easily distributed through the Internet. It can be a placed in the third party tools section of the ADL website along with RELOAD and other value-added tools where it can be easily downloaded. All supporting documentation can also be made available through the Internet.

Although this is a prototype designed to run at the Joint Forces Staff College, the ADL Registry Client source code will be open source and available for everyone to download, modify, and enhance. This approach offers several key benefits. Custom tailoring would allow the ADL-RC to work with a specific client's software architecture; and finally, new features and functionality may be added to enhance application performance.

The source package was compiled to an executable file using Inno Setup v. 5.1.10. This freeware can be downloaded from http://www.jrsoftware.org/isinfo.php. The source file is contained in config folder of the package.

4.2 Expand Version Control Initiatives

The ADL Registry Client prototype is the first step to combating the content currency issue that threatens the Registry as it is currently designed. However, it is not a solution for the entire community if other DoD organizations will be unable to use the ADL-RC due to network security restrictions (ex/NMCI). Therefore, we recommend that ADL continue to support initiatives that expand the ADL Registry to provide version control web services.

Northrop Grumman has recently proposed a partnership with KMI to analyze, design and develop a web-based solution that bypasses security limitations by offering advanced content update notifications and version control functionality through the Registry portal itself. We will use the lessons learned from the ADL-RC prototype project to better serve the ADL community. The ADL-RC may still be available for download but the web-based version control system would better suit the secure networks of DoD.

Acronyms

ADL-R – Advanced Distributed Learning-Registry
ADL-RC – Advanced Distributed Learning-Registry Client
ADL-RIM – Advanced Distributed Learning-Registry Interface Mechanism
ADL-RUG – Advanced Distributed Learning-Registry Users Group
AJPME - Advanced Joint Professional Military Education
CMU – Carnegie Mellon University
CNRI – Corporation for National Research Initiatives
COP – Community of Practice
CORDRA – Content Object Repository Discovery and Registration/Resolution Architecture
DLCC – Distance Learning Coordination Council
DLD – Distance Learning Division
DoD – Department of Defense
DoDI – Department of Defense Instruction
DTIC – Defense Technical Information Center
GUI – Graphical User Interface
HTTP – Hypertext Transfer Protocol
JADL – Joint Advanced Distributed Learning
JCDES – Joint Continuing and Distance Education School
JFSC – Joint Forces Staff College
JRE – Java Runtime Environment
KMI – Knowledge Media Inc.
LMS – Learning Management System
LOM – Learning Object Metadata
MECC – Military Education Coordination Council
NMCI – Navy Marine Corps Intranet
QA – Quality Assurance
Reg-T – Registry Transaction
RSS – Really Simple Syndication
RTE – Run-Time Environment
SCORM – Sharable Content Object Reference Model
SDD – Software Design Document
TIR – Test Incident Reports
XML – Extensible Markup Language

References

Related information about ADL initiatives is available from the following sources.

Advanced Distributed Learning	*http://www.adlnet.gov/*
ADL-R DoD Portal	*https://adlregistry.dtic.mil*
ADL-R Community Resources	*http://adlcommunity.net/course/view.php?id=7*
ADL-R Document Library	*https://adlregistry.dtic.mil/5/index1.htm*
CORDRA™ documents	*http://cordra.net/*
DoD Instruction 1322.26	*http://www.dtic.mil/whs/directives/corres/html/132226.htm*
Joint ADL Co-Laboratory Research & Development	*http://www.jointadlcolab.org/research/*