ADL 3D Repository: A Blueprint for Discovery and Access

Damon Regan
Booz Allen Hamilton, in support of the Advanced Distributed Learning Initiative
Orlando, FL
damon.regan.ctr@adlnet.gov

Robert Chadwick
Katmai, in support of the Advanced Distributed Learning Initiative
Orlando, FL
robert.chadwick.ctr@adlnet.gov

ABSTRACT

The effective, efficient, and appealing learning content that future education and training depends on can take significant time and money to produce. This is especially true for highly interactive and 3D content used to create adaptive and immersive learning environments. The ability to discover and access existing, reusable content can significantly reduce costs and development time. Past approaches to enable this discovery and access have been focused on the Advanced Distributed Learning (ADL) Registry (a searchable catalog of metadata describing content and its location). The ADL 3D Repository is a technology that reflects an expanded focus to include repositories that provide immediate value to users. While this repository technology is focused on sharing 3D models, it provides a blueprint for enabling the discovery and access of learning content. It makes uploading, finding, viewing, converting, and downloading content easy. It integrates directly into immersive environments and editors where 3D models are used. This paper will describe the 3D Repository technology and how it addresses the challenges reported in a 2009 RAND study investigating the prospects for reuse of training content. Efforts to share and federate this repository platform with partner organizations will be discussed.

ABOUT THE AUTHORS

Damon Regan is an Associate with Booz Allen Hamilton supporting the Advanced Distributed Learning Initiative as a Technical Team Lead. He is currently focused on enabling the discovery and access of learning content. Damon is a Ph.D. candidate in Instructional Technology at the University of Central Florida. He has an MBA from Rollins College and a B.S. in Computer Science from the University of Central Florida.

Rob Chadwick is a 3D programmer and artist who has been working in the 3D content field for more than ten years. In that time, he has designed custom game engines, managed render farms, and written productivity tools. He has taught entry level video and photo editing, and offered courses on advanced 3D animation. He has designed simulations for the FBI and the US Navy using custom and off-the-shelf game software, and animated virtual spaces for architecture and interior design companies. While primarily interested in rendering technologies, Rob has worked for the last year with the Advanced Distributed Learning Initiative to foster interoperability of 3D assets. He holds a Bachelor of Computer Science degree from Virginia Tech.

ADL 3D Repository: A Blueprint for Discovery and Access

Damon Regan
Booz Allen Hamilton, in support of the Advanced Distributed Learning Initiative
Orlando, FL
damon.regan.ctr@adlnet.gov

Robert Chadwick
Katmai, in support of the Advanced Distributed Learning Initiative
Orlando, FL
robert.chadwick.ctr@adlnet.gov

INTRODUCTION

The effective, efficient, and appealing learning content that future education and training depends on takes significant time and money to produce. This is especially true for highly interactive and 3D content used to create adaptive and immersive learning environments. 3D models available through popular commercial 3D exchanges like TurboSquid[1] can each be worth several hundred dollars or more. For example, an advanced Russian Destroyer model was found on sale for $1,250. The ability to find existing content that can be reused enables significant reduction in costs and development times.

This paper reviews the challenges associated with discovering and accessing training content and describes the ADL 3D Repository, which is a prototype system for sharing 3D models. We believe that this technology is a key starting point for realizing the larger vision of a DoD or Federal discovery marketplace that enables the right content to be found and used at the right time.

Review of Previous Work

DoD has previously undertaken efforts to share and reuse learning content. The Defense Audio Visual Information System (DAVIS), the Defense Information Technology Information System (DITIS), and the ADL Registry (ADL-R) have each attempted to foster the reuse of training content.

In 2009, the RAND Corporation published a report analyzing the prospects for increasing the reuse of digital training content within the DoD, in which it reviewed these and other sharing systems. Key negative characteristics of the DoD systems were: difficulties with access, entries of inconsistent or low quality, and limited number of entries. The RAND report also reviewed implementation obstacles identified by training and development organizations. Primary obstacles were metadata or repository-related issues, legal or security issues, and cultural issues blocking implementation. These implementation obstacles that are still present today are reviewed below with suggestions for avoiding the negative characteristics of previous DoD systems.

CURRENT CHALLENGES

Metadata and Repositories

Nearly two-thirds of training and development organizations told RAND that the costs of creating metadata and the lack of useful repositories were obstacles to implementing a reuse strategy (p. 69). If we look to successful commercial sharing sites, we can see the strategies to overcome this obstacle. These strategies include 1) ensuring that an easy to use platform exists, 2) that the user experience is worthwhile, 3) that metadata doesn't get in the way of an easy to use upload mechanism, and 4) that content found can be easily used.

Let us consider successful commercial sharing sites for personal news, videos, pictures, slideshows, and 3D models (i.e., Facebook[2], YouTube[3], Flickr[4], and SlideShare[5], Google 3D Warehouse[6], respectively). The first strategy we see with these sites is that they ensure an easy to use platform exists. Users of these sites are able recognize and remember the identity and location of these individual platforms. A user simply points his or her Web browser to one of these sites and begins using the site. All the complexity of the site is hidden from the user. There is no installation required, software that needs to be written, or IT department that has to be consulted.

[1] http://www.turbosquid.com

[2] http://www.facebook.com
[3] http://www.youtube.com
[4] http://www.flickr.com
[5] http://www.slideshare.net
[6] http://sketchup.google.com/3dwarehouse/

The second strategy we observe is that when users go to these sites, the user experience is worthwhile. Users upload 35 hours of video to YouTube every minute[7]. More than 30 billion pieces of content are shared on Facebook each month[8]. What compels users to share so much content on these sites? YouTube and Facebook encourage information producers to post content in part through the creation of community feedback loops. If the content a user posts is commented on by others, that user is more likely to post again. This effect has been found in the context of sharing educational materials by researchers at the University of Southampton (McSweeney et al., 2011). Leveraging social mechanics is critical to creating the worthwhile experience users seek.

When a user uploads content to one of these commercial sites, the site stays out of the way. This is the third common strategy. Users are not expected to write extensive metadata; a title or description is usually all that is required beyond the content itself. If a user wants to associate a logo with the content, he or she is free, but not forced, to do so.

The fourth and final strategy is that these sites make it easy to use content that is found. There are two aspects to this strategy. The first is the nature of how content in the system can be used. The second is the nature of the content itself.

Most of the successful sharing sites enable content to be shared whereas only a couple of the successful sharing sites enable content to be downloaded. Consider YouTube or SlideShare. These sites allow users to easily upload, find, and share videos or presentations. The sharing is limited to viewing, embedding, or emailing a link to the content. Now consider the Google 3D Warehouse. This site allows users to upload, find, and share content, but it also allows the user to download the content in a form that best suits his or her needs. This ability to download in the format preferred by the user is necessary to support most cases of reuse.

In the case of sites that allow the user to download the content, it is important that the form of the content can be used. Flickr, for example, allows users to download images in various sizes (e.g., small, medium, large, and original) if the uploader chooses to make this available. The file format of the downloaded image is JPEG, which is used as both a common interchange format and run-time format. The Google 3D Warehouse similarly allows users to download 3D models in various formats. Even in the case of sites that do not allow download, support for common file types is used for upload (e.g., YouTube). Common file types and interchange file types in particular are important for reuse. As is typically the case, clear and understandable choices are appreciated.

[7] http://www.youtube.com/t/press_statistics

[8] http://www.facebook.com/press/info.php?statistics

Legal and Security

Half of the training development organizations that RAND surveyed reported legal and security challenges as an obstacle to implementing a reuse strategy. These challenges are complex and situation-based. There are many policies and guidelines at various levels that impact the right to share training content. In order for DoD training content to be shared and reused through a DoD sharing system, it needs to have been produced through contracts that obtained the necessary data in the optimal formats and with the appropriate rights for distribution. The content should be part of a process that ensures that digital rights are clearly understood and that they adhere to the law and DoD regulations. Establishing the processes necessary to clarify legal and security issues is not easy and requires up-front strategic planning.

Cultural

Half of the training development organizations that RAND surveyed also reported cultural issues for the adoption of reuse. RAND identified three categories of cultural issues. The first is a resistance to change practices. The second is a "not invented here" attitude which leads an organizational culture to avoid using existing products because of their external origins. The third category was a tendency to see material, even from nearly identical subject areas, as "unique" and not a candidate for reuse. These common attitudes can likely be changed through training, leadership, and role models.

Another cultural issue that is difficult to address is an individual's unwillingness to share work, so-called information hoarding. There is a natural personal investment that we make with our work. We seek to protect it from competition. This is especially true given the potentially unknown or unclear scopes of sharing beyond an organization's boundary. This issue is naturally entangled with legitimate legal and security issues previously discussed. Up-front strategic planning that clarifies legal and security issues can also help address such cultural issues.

3D REPOSITORY PROTOTYPE

Various content types will have different use cases, and therefore different challenges will be faced in building services specific to each content type. The ADL 3D Repository (3DR) hosts 3D model files, and supplies value added services to enable the reuse of 3D meshes. We hope that our work will serve as an example of how a content repository can foster interoperability, although we understand that some of the challenges for other content types will be unique.

3D Sharing Landscape

Several commercial systems exist that attempt to meet the search and discovery needs of 3D artists and designers. These systems have various benefits and drawbacks and differing operating philosophies. Here, we will review some of the capabilities and methodologies of several existing systems.

TurboSquid is a commercial 3D content sharing system that was founded in 2000. It currently claims to host more than 200,000 models and has marketplaces for many other types of content. It is widely used in the entertainment industry. Content hosted on TurboSquid is heterogeneous, meaning that an author can upload any file and sell it as a "model." While users can search based on the type of file, this data is self-identified by the author and could be unintentionally or maliciously inaccurate. Likewise, much of the content is uploaded in authoring tool-specific formats, such as Autodesk 3DSMax's .max. Much of the content is of varying quality, and may be authored in ways that cannot readily be exported to interchange formats. Extensive art time is often required to make use of TurboSquid content. While this serves the needs of the "3D artist community" as stated in the TurboSquid corporate philosophy, it is seldom useful for government program managers, who have neither the software nor the expertise to use one of these formats in a training environment.

The Google 3D warehouse is another system that allows for the archiving, discovery of, and access to 3D content. This system has several advantages over TurboSquid and one major drawback. The Google 3D warehouse is composed of homogeneous content. Content hosted by Google is encoded in either Google's custom 3D file format, Sketchup (.skp), or its archive format .kmz. KMZ files are archives of textures and a data type called KML. KML is an XML format based on COLLADA[9]. However, it only implements a small subset of COLLADA features. Google controls the authoring tool that creates this data – one cannot directly translate COLLADA into KML. This greatly increases the usefulness of the format, in that any system that can accept an SKP or KMZ file should accept ANY SKP or KMZ file. Because Google has control over the hosting and the encoding of the content, it can ensure that the content always works. However, Google's SKP and KML formats only support a small subset of the features that many artists require, and make assumptions that might not apply to legacy content. In fact - because Google exercises such tight control over the formats, it is difficult to move legacy content into the Google 3D ecosystem. If an artist has content authored in the legacy STL format[10], he or she must find a tool that can convert this into OBJ or 3DS (common interchange formats). Then, the artist must buy or gain access to Sketchup Pro, which can be used to translate these interchange formats into the Google formats.

3D Repository Overview

The ADL 3D Repository (3DR) is a research and development project funded by the DoD that seeks to combine the benefits of both previously discussed approaches. The 3DR allows for search and discovery of content through a variety of means. Users can search based on user reviews, ratings, and manual or automatically generated metadata. The system includes fields for data describing the author, sponsor, and developer of a model, as well as the number of vertexes, the file format of the content, and user-supplied descriptions. Because we believe that it is imperative that content appear to the user as it is described by the author, the 3DR allows for a real-time, web-based 3D view of the content. This allows consumers to be confident that the content the author has described is actually the content that the consumer receives. To further this goal, the 3DR can automatically generate thumbnails for models to be displayed where the user has opted not to, or cannot, view the 3D content.

The ADL 3D Repository differs from previous systems in several important ways. While other content repositories are simple storage and retrieval tools, the

[9] COLLADA is an interchange file format for interactive 3D applications http://en.wikipedia.org/wiki/COLLADA

[10] STL is a file format native to the stereolithography CAD software created by 3D Systems http://en.wikipedia.org/wiki/STL_(file_format)

3DR is closely tied to the content it hosts without requiring a specific tool or format. When content is uploaded to the repository, the 3DR parses, optimizes and re-encodes the uploaded content. While a user can always request to download the original, the design of the system encourages users to consume the "sanitized" version, which is available in several common formats.

The 3DR provides download and search services through a REST API. This allows software developers to use the 3DR from within an application. For instance, a developer may leverage the 3DR as an asset store for a game engine. The 3DR is also an open source project. It is built on open standards and an open technology stack so that anyone may download and improve the source code. Local installations of the 3DR at different organizations can be registered at an ADL controlled website to participate in a federated network of 3D repositories.

While the 3DR is focused on sharing 3D models, it provides an example and a blueprint for the sharing of other learning resources. It differs from previous learning resource registries by hosting the content, rather than registering the content's location. This allows organizations that do not have the IT infrastructure to host content to upload their content to a 3DR instance run by another organization, possibly by the original instance at ADL. Also, because the content is curated by the cybrarian assigned to the 3DR instance, dead, broken or missing content can be quickly removed. The 3DR also makes possible new asset workflows by translating file formats on download. Users can upload an FBX file from their authoring tool, and import it directly into a tool that only accepts OBJ files. Additionally, because the 3DR's functionality is exposed through a REST service, software developers can write import functionality into their tools that will fetch data in their desired format directly from the repository. Using this workflow, the end user need not know what formats a tool supports. Instead, the user of a tool would only need to know that the tool can import content from the 3DR.

The ability to manage your own collection of content is an important security and management requirement. Some organizations may want to keep control of their intellectual property, while still allowing Internet users to access the content. To address this requirement, ADL will host a federated 3D repository. Instances of the 3DR platform can be hosted by a partner organization, and registered with the ADL 3DR federation. Users can search for content from all federated repositories, and access that content through the federated interface. This will allow organizations to host their own content while still enabling discovery and reuse of that content across the partner organizations. This is possible because the 3DR server is open source software. Partner organizations may freely modify their local instances, and maintain compatibility with the federation by implementing the defined interfaces. Partner organizations can add access control rules, new file formats, or change data storage strategies without impacting the effectiveness of the system. These changes could be optionally submitted back to the community for inclusion in future 3DR releases.

The ADL 3D Repository addresses several of the problems with other 3D sharing systems. Like the Google 3D warehouse, it contains homogeneous content, and therefore the content hosted in the repository has a high level of reliability. The system overcomes the difficulties of including legacy content by allowing for the direct import of several interoperability formats. Also, by allowing for on-demand conversion, the system assures that content is not trapped in a specific development pipeline. This contrasts with vertical silo systems, such as the asset stores built into popular game editors like Unity or VBS2. Like TurboSquid, the 3DR allows users to submit thumbnails, logos and other supporting documents, but ensures that content matches the submitter's description by showing consumers a real time rendering of the models. This avoids the common pitfall of large non-curated commercial repositories, which frequently contain content that does not appear as it is described. The commercial sector has realized the same needs. KataSpace's OurBricks content sharing system is an emerging platform that addresses similar issues. Like YouTube or Facebook, this system is a closed source platform. The 3DR is open source software developed by ADL that enables the DoD to respond to continuously changing requirements[11].

DETAILED DISCUSSION OF 3D REPOSITORY COMPONENTS

Metadata and Search

The 3DR allows for a variety of metadata (see Tables 1 and 2 below) to be optionally associated with each model. Some of the metadata is gathered automatically by the content conditioning system (see Table 2).

---

[11] Please see the DoD CIO memorandum, "Clarifying Guidance Regarding Open Source Software (OSS)" http://cio-nii.defense.gov/sites/oss/2009OSS.pdf

Table 1. User-supplied metadata

Metadata Parameter	Description	Optional
Title	A title for the model	No
Description	A text description of the model and its contents	Yes
Tags	A list of values	Yes
Submitter	The email address of the user who uploaded the content	No
More Info URL	A hyperlink intended to link to additional description data	No
Developer Name	The name of the developer, if different than the submitter	Yes
Sponsor Name	The name of the sponsoring organization	Yes
Artist Name	The name of the artist, if different than the submitter	Yes
Developer Logo	An image file for the Developer	Yes
Sponsor Logo	An image file for the Sponsor	Yes

Table 2. System-generated Metadata

Metadata Parameter	Description
Unit Scale	This is parsed from the asset data. The ratio of one unit in the models coordinate system to a meter.
Up Axis	This is parsed from the asset data. The direction the model considers “up”.
Format	The original format of the uploaded 3D model file
Number of Polygons	The number of polygons in the model. This number is counted by the conditioning step.
Number of Textures	The number of texture files referenced by this model
Missing Textures	The textures referenced that are not included in the package
PID	A unique identifier for the content
Number of Downloads	The number of times the 3D model has been downloaded
Number of Views	The number of times the model has been viewed by a browser

Model Validation

One of the key features of the 3D Repository is the ability for submitters and consumers to see the actual content that the repository contains. This is important to ensure that the consumer of content will receive the assets they expect based on the metadata and to ensure that the model conditioning process has correctly understood the content. When a user uploads a model, he or she is presented with a real-time preview of the conditioned model. If this preview does not appear as the user expects it to, the user knows that the content he or she is attempting to submit contains features that are not supported, errors, or is simply not the correct content. When a model has been viewed by the submitter and accepted, it is added to the repository. This ensures that content in the repository will be usable for the largest possible user base. Content that cannot be understood by the 3DR is unlikely to be understood by many other systems and therefore is not a good candidate for sharing.

While this process ensures the highest level of quality for the data in the repository, it does require users to submit content to the 3DR that it can understand. This limitation frequently prevents the use of large libraries of legacy content. Often, an organization has a library of content that they would like to share, but which is not authored to common best practices and cannot be understood by the 3DR parsers. In many cases this content will require human intervention. Neither ADL nor the contributing organization has the resources to manually condition large amounts of data, making it difficult to reuse. In the future, we will work to develop separate paths for content that cannot be parsed and manipulated.

The 3DR relies on open source tools to parse and manipulate 3D content. While ADL has made every effort to create as robust a system as possible, the ability of the model conditioning process to read files is limited by the state of development of these open source efforts. Specifically, the 3DR uses OpenSceneGraph (OSG)[12] to load and save files. OSG is a graphics library with support for many file formats. ADL has worked extensively with several of the format

12 http://www.openscenegraph.org

parsers and officially supports five of the available format readers. Models can be uploaded in five different formats (see Tables 3 and 5) and downloaded in six (see Tables 3 and 4). OpenSceneGraph provides a plug-in architecture that allows additional model types to be loaded without changes to the 3DR core.

Table 3. Supported Formats for Upload and Download

Format	Description
.dae	An open standard XML format
.fbx	Works with 3D Studio, Maya, and Unity
.3ds	An older format that is widely supported
.obj	Another widely supported legacy format

Table 4. Supported Formats for Download Only

Format	Description
.json	An OSGJS library representation
.o3dtgz	Google O3D format

Table 5. Supported Formats for Upload Only

Format	Description
.skp	Google Sketchup tool format

OpenSceneGraph, and therefore the 3DR, has limited support for other file types. These include FLT, STL, OSG, IVE, DXF, DirectX (.x) and many others[13].

The 3DR stores models internally in COLLADA. All models are saved as COLLADA in the database, though the native format should be transparent to users. COLLADA is a feature rich, widely supported XML-based representation for 3D models.

Model Conditioning

The 3DR attempts to make the content it stores as widely reusable as possible. Therefore, content that is submitted to the repository is modified before being stored. This step makes it possible to ensure that content can be loaded by many different systems. First and foremost, models are "triangulated" during the conditioning process. Most 3D file formats can store faces in several different ways; as lists of polygons, lists of triangles, lists of quadrangles, triangle strips and triangle fans. The 3DR decomposes all these types of geometry into simple lists of triangles. Likewise, the content is "deindexed," which removes some complexity from the vertex data at the expense of repeating some entries. The increase in portability is well worth the slight increase in necessary storage space.

A scene is usually composed as lists of geometry located at different positions in space. Sometimes, some position data is grouped and associated with position data for the entire group. This data structure is referred to as a scene graph. Wherever possible, the 3DR flattens the scene graph to remove redundant nodes, which sometimes are added by modeling tools as part of the export process. The 3DR also changes names of objects to remove non-standard characters and prevent name collisions. Some systems will not accept files in which an object name contains nonstandard characters or where objects have duplicate names.

Finally, the 3DR will convert all textures into Portable Network Graphic (PNG) files. The model will be updated to reference these new files instead of the original file names. Absolute paths to files will be stripped to the raw file name and the PNG textures will be stored in the ZIP archive without any sub-folders. All texture names are translated to lowercase and all texture references are updated to match. This ensures that modeling tools can find the proper textures associated with a model when the ZIP archive is extracted. Textures that cannot be found in the zip file are noted in the database, so that the author can add these textures later.

Real-time Model Previews

The 3DR requires that users be able to see a real time preview of their content at several points. Content submitters are presented with a validation view and downloaders are presented with a preview. Both of these requirements call for a solution for viewing 3D content on the web. Because the ability to display content in a browser is cutting edge technology, the 3DR uses a tiered strategy to deliver this content to a browser. If none of the preview strategies are available, the user may still use the developer-submitted thumbnail as a reference for the content of the model. The three different technologies used by the 3DR are WebGL, Google's O3D, and Adobe's Flash. All viewer techniques allow the user to rotate the model, view the textures, see the wire frame, and, when uploading, capture an image to use as a thumbnail. Each viewer has specific abilities particular to only itself.

WebGL is an emerging standard for presenting 3D content natively in a web browser. Our system uses a

[13] See http://www.openscenegraph.org/projects/osg/wiki/Support/UserGuides/Plugins for details.

supporting library called OSG.js to facilitate rendering and model loading. It is our preferred mode for display, as it is high performance and runs without the user installing a browser plug-in. This has benefits in organizations that do not allow users to extend their browsers or install plug-in software. In addition to the standard features listed above, the WebGL viewer allows users to animate the rotation of the model, view real time generated shadows, and examine the scene graph hierarchy. The WebGL view is appropriate for large models up to several hundred thousand vertexes. Unfortunately, WebGL is not yet a mature technology, and is only supported on recent browsers such as FireFox 4 or Google Chrome 12.

Google's O3D 3D viewer uses a browser plug-in to display 3D content. Because the content is rendered directly on the user's video card by the plug-in, it has the highest performance of any of the available solutions. It is suitable for content up to millions of polygons (depending on client system capabilities) and can render with smoothed edges and high quality texture filtering. However, the O3D browser plug-in is no longer actively maintained by Google, leading to fears that browser evolution will make the last release unusable in time. Already, we have experienced performance problems with newer browsers. In addition to this concern, the user must have the administrative rights to install the plug-in on their machine.

The 3DR's final mode for 3D display runs on Adobe's Flash platform. Using a software renderer called Away3D, the COLLADA files can be visualized directly on a Flash object in the webpage. Because Flash enjoys significant market penetration, this viewer is widely supported even though it requires the Flash browser plug-in. Flash currently does not have support for hardware-accelerated 3D and the Away3D library must directly compute the color of each pixel using Flash's internal programming language. This severely limits the performance of the Away3D viewer. We therefore limit its display of content to models with less than two thousand polygons to avoid an unacceptable level of responsiveness, although the performance will vary based on the power of the user's computer. In addition, Away3D does not use a Z-Buffer for rendering. This is a technique used to sort the pixels of a render to display intersecting and overlapping triangles properly. Because Away3D lacks this feature, there are certain patterns of triangles that will always display rendering artifacts.

3D on the web is an active area of research with several different important projects underway. In particular, we are paying close attention to the continued adoption of WebGL. Additionally, future versions of Flash will support hardware 3D. We are eagerly waiting for a time when we can rewrite our Flash support to take advantage of this powerful feature. We will continue to revise our real time viewer strategy as these projects develop.

REST API and Integration

Future technologies will make use of constant and ubiquitous Internet connectivity to access data and services in the cloud. The ADL 3DR is positioned to enable this important use case by exposing its services as a REST webservice API. Users of the API can access content from the 3DR to integrate it into their own systems. For instance, a developer could write an asset store for a game engine by calling the search and retrieval services. Users could pull models into a game directly without a web browser.

Currently, to develop an application against the API, an application developer must request an application key from ADL via the main 3DR website. This key will be used to keep usage statistics and enforce rate limits. Model retrieval from the API is limited to publicly accessible models and upload is currently disabled. For a full description of the REST API, please visit the ADL 3DR Blog[14]. Here, you can find technical details on the available functions and demonstrations of integrations between the 3DR and several popular platforms. In the future, we will enable upload via the API and add features to support complete non-browser based interaction with the content.

The public API will also form the basis for federation between different instances of the 3DR application running at different organizations. ADL will create a common API instance, which will aggregate search results from several registered federated repositories, and will serve as a single endpoint for the retrieval of content. Managers of federated repositories will be able to choose to allow downloading, metadata retrieval, or only search for their system through the federation.

LESSONS LEARNED AND FUTURE WORK

Lessons Learned

We have captured many lessons learned from our 3D Repository prototyping efforts that we will continue to

[14] http://3dr.adlnet.gov/blog/

leverage in our future work and that we desire to share so that others may benefit from our experiences. The challenges represented in these lessons learned are generally not technical. The three primary areas of our lessons learned are content formats, legal and security challenges, and community. These areas have been previously discussed and found in other reviews. We offer some new perspectives on these challenges from our experiences.

We discovered that simply having access to source materials is not sufficient in the case of 3D models. Some of the content we received was provided with source materials, but was unable to be parsed and viewed in the web-based platform. This is because proprietary run-time environment plug-ins prematurely optimized source materials for their associated runtime environments. While it may seem appropriate to discount the value of the web-based parsing and viewing, there are other ramifications, which potentially impact the organization's ability to use the content in the future, as well as impact partner organizations that might reuse the content. We see two general approaches to address this issue. The first is to consolidate and streamline the content pipeline to use a single platform and common set of tools. This will mitigate content format issues. The second is to focus on the use of clean, interoperable formats in addition to source and run-time formats. It is important to note that neither of these approaches addresses the volumes of legacy content that exist without associated interoperable formats or currently used run-time environments. If source materials exist, manual manipulation of the content is likely the only option to make this content usable in new contexts.

We also discovered that despite the introduction of Creative Commons licenses, legal and security issues remain complex and likely limit sharing. We provided all the available Creative Commons licenses for selection by content submitters. Some commercial organizations said they could not share information using a Creative Commons license as they required all rights to be reserved through traditional copyright. Many government contributors did not provide guidance on the selection of a Creative Commons license. While the system that facilitates the sharing of content could attempt to impose a particular license to clarify complexity, that approach likely does not address the issue. Copyright guidance for government works and contracted government works is complex and typically requires particular language to be embedded in contracts. If contracts do not make licensing explicit (i.e., transferring copyright to the U.S. Government), the ambiguity will persist and lead to complexity. Guidance and best practices tailored to particular communities around particular types of content would clarify how to operate in these complex situations.

Even with explicit licensing terms in contracts, there still remains the challenge of determining how content should be released. Simply because copyright law does not apply to government produced works does not mean those works are public domain. Not all content free of legal rights is publicly available. Processes for determining the release of information are likely considered a laborious burden and non-essential task. We've had conversations with many potential contributors about adding access control features to the 3D Repository to assist organizations in providing some controls over the visibility of information. However, unless such access control systems are based on DoD-approved identity and authentication mechanisms, they are unlikely to be useful for controlling information beyond privacy settings.

The final lesson we are still in the process of learning concerns the creation of healthy communities around content sharing. We increasingly observe social mechanics changing the way we use the Web. These social mechanics are critical to creating incentives to return to Web sites and share information. When someone comments on content a user shared, that user is probably more likely to share additional content. That is the observed practice on commercial Web sites. However, it is not clear how to create such forms of Web-based communications on professional, government, and military sites. While we have 150 registered users who have been given the ability to rate models and create reviews, at this point none of these 150 registered users have rated models or written reviews.

Future Work

Our work in the 3D Repository has led to several interesting new avenues for research and development of 3D content interoperability. Much of our future work will involve expanding the base of available inputs and outputs for the 3DR content. Specifically, we would like to look into supporting more file formats and improving support for existing formats. The open source parsers we rely on lack some important features and we believe our service would greatly benefit from including these features. Notably, the current support for animations in FBX and COLLADA files is incomplete. We will also research the possibility of supporting several run-time formats. This would create a huge benefit for the community by freeing content from vertical silos. We would also like to investigate the possibility of supporting some authoring tool formats directly. This would allow us to take control over the mapping between authoring formats and interchange formats and to specify the most successful techniques.

We will also build tools to allow for more control of the distribution of content. Currently, anyone can create a user account and access models. We will research techniques for user authentication and build tools to allow content authors to specify exactly who may and may not access their content. This will allow many organizations to host content that currently don't due to concerns about piracy and copyright.

Finally, we would like to demonstrate expanded support for integrations between the 3DR and other content tools like game engines or artistic modeling software and apply the lessons learned from this experience to other forms of digital learning content such as video, audio, simulations and games.

CONCLUSION

There are many benefits to reusing content for education and training. There are also many challenges. Many efforts have attempted to realize these benefits and overcome these challenges. Yet many challenges still remain. We believe that one of the most pressing challenges that the 3D Repository helps to address is the difficulty of accessing content. While discovery is important, without subsequent access, discovery is of little value. One of the primary obstacles to implementing reuse strategies is having an easy-to-use repository. The assumption that individuals will establish their own repositories that span organizational boundaries does not seem valid.

While we propose that the 3D Repository prototype addresses technical challenges in the metadata and repositories field, we also understand that, while addressing these technical issues is necessary, it is not sufficient. The complex legal and security challenges remain. While the 3D Repository might provide a platform for sharing legally reviewed and publicly released content, there is a large volume of content that does not have clear legal or release attributions. Once we overcome the technical, legal, and security issues, we will still have to address more practical issues. Many individuals will still hesitate to invest the time required to share content. We will need to continue to work on making the sharing of content as easy as possible for those who choose to do so. We will also have to encourage those that would prefer not to share to do so by providing examples of effective sharing that benefits the greater community.

REFERENCES

Shanley, M.G., Lewis, M.W., Straus, S.G., Rothenberg, J., & Daugherty, L. (2009). *The Prospects for Increasing the Reuse of Digital Training Content*. Santa Monica, CA: RAND Corporation.

McSweeney, P., Borthwick, K., Hargood, C., Millard, D., & Howard, Y. (2011). *The Mechanisms and Impact of Encouraging Community Engagement in Teaching Repositories*. Retrieved from http://eprints.ecs.soton.ac.uk/22063/