The Texas Course Redesign Learning Object Repository:
Research and Development for a Production System

THECB Learning Object Repository Submission Tool

Prepared for

The Texas Higher Education Coordinating Board

by

The Texas Course Redesign Repository Team

under the supervision of
Dr. William E. Moen
<william.moen@unt.edu>

May 2010

Texas Center for Digital Knowledge
College of Information, Library Science & Technologies
1155 Union Circle 311068
Denton, Texas 76203-5017
Phone: 940-565-2473
Fax: 940-369-7872
Web: http://www.txcdk.unt.edu

Table of Contents

THECB Learning Object Repository
Submission Tool

1. Introduction

This document describes an alternative to the DSpace submission system: a custom metadata creation and submission tool that provides useful extensions such as metadata templates, multi-file uploader, and connection to the automatic keyword extraction service.

The following sections describe Submission Tool in details.

2. Terminology and acronyms

The following terminology conventions are used throughout this document:

- *Current template:* An item from which the metadata is loaded when describing a working item
- *Current working item:* An item that currently being uploaded to the submission tool server, edited, or deposited to DSpace repository. All submission tool actions are performed on current working item.
- *DSpace:* An open source digital assets management system; digital repository system
- *Item:* An autonomous digital object (like learning object) that comprises of one or more content files that normally hyperlinked. An item may reside on a client's machine, submission tool server, or a DSpace repository. Additionally, an item also includes metadata and other system files when prepared for depositing or deposited into a DSpace repository.
- *LOR*: Learning Object Repository
- *SAF:*: Simple Archive Format
- *TCRR:* Texas Course Redesign Learning Object Repository for learning objects that uses DSpace repository system

3. Implementation

3.1. Problem statement

The proof-of-concept LOR implementation used the basic manual submission workflow provided in DSpace. The project team configured the DSpace metadata registry to accommodate the elements needed, and customized the submission pages to assist in metadata creation. However, the project team encountered several deficiencies of the standard DSpace submission system and proposed solutions and extensions.

- System does not allow uploading multiple files at a time. Having items that consist of hundreds of files slows down submission process and makes it prone to errors.
- Submitting items that share many metadata values without reentering these values requires setting per-collection item templates. This procedure requires administrative privileges and does not allow needed flexibility.
- System does not allow easy integration of external keyword extraction services that need the access to the content files of the items being deposited before an item is deposited into the repository.

3.2. Proposed Submission Tool

The project team has developed web-based submission tool that is separate from DSpace. This tool supports both distributed and centralized submission models. Submitters use a web browser to upload files (i.e., bitstreams) that comprise a learning object to a file system on a server, describe items (i.e.

create metadata), and generate items in DSpace's Simple Archive Format. Figure 1 illustrates the components and flow in the submission tool.

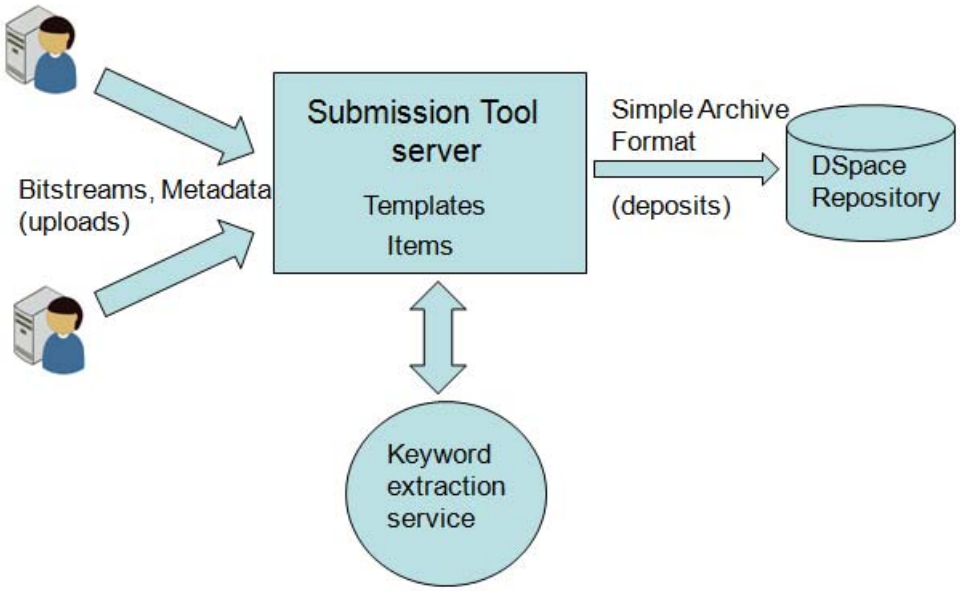

Figure 1: Architecture and submission workflow

Items in Simple Archive Format are used to import items into DSpace repository using DSpace's importer plugin. This procedure can be done by the authorized submitters or administrator(s) of the repository.

The submission tool includes templates for metadata creation that speed up submission of items that share common values for metadata elements or inherit those values from parent items. For example, a Lesson-level learning object (LO) may encapsulate a smaller Topic-level LO and because of the relation of these two LO metadata for each will have the same data values (e.g., creator, subject terms, etc.). Topic LO within one Lesson LO may share many values. Temples may be specifically created or based on previously described items.

The submission tool allows generating values for subject related elements using an external keyword extraction service developed by the project team. Content files of an item that is being described are sent to the service and the service generates proposed keywords and provides them to the submitter in a separate window. Submitter may consider these keywords when assigning values to the subject-related metadata elements.

4. User's reference guide

The web-based submission tool is located at the following address:
http://txcdk1.unt.edu/mt/

The functionality and workflow of the submission tool is described by going over the main menu items (see Figure 2).

4.1. Menu About

This menu contains brief information about the functionality of the submission tool. Items are first uploaded onto the submission tool server, described, and then deposited into the target DSpace repository.

4.2. Menu Login

Anonymous users can only view items on the submission tool server. Only authorized users may upload, describe, edit, and deposit items. Submission tool is set to work with one specific target DSpace repository. The users of the submission tool are limited to the valid users of the specific target repository.

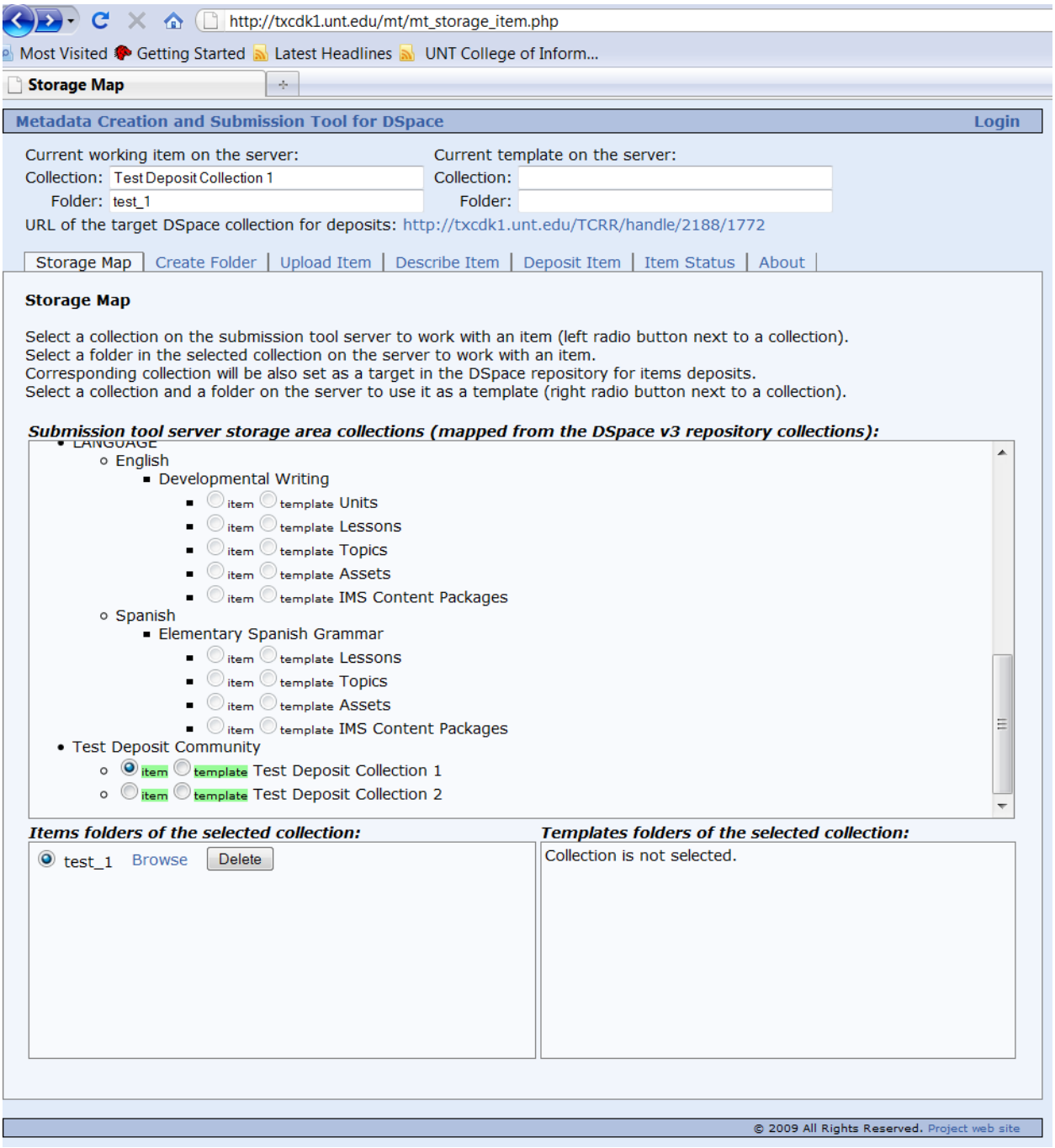

Figure 2: Submission Tool, menu Storage Map

4.3. Menu Storage Map

Objects/items in the DSpace target repository are organized into the collections and communities. Submission tool server storage area (see Figure 2) provides an intermediate location for the items between the user's client computers and DSpace target repository. Items are being described by submitters while they are located on the submission tool server storage area. Submission tool server storage area is organized in the collections that mirror collections and communities structure of the DSpace target repository.

Each item on the submission tool server is stored in a folder located under the corresponding collection. The name of a folder should be similar to the title of an item or somehow uniquely identify an item within a collection (see Create Folder menu).

This approach provides organized intermediate storage of the items when submitters are working with them on the server of the submission tool. As it was mentioned above, the submission tool is set to work with one specific repository and map of submission tool storage area replicates the structure of the communities and collections of that DSpace repository.

Only collections open for submission are selectable (marked green).

Each collection on the submission tool server may have two roles: current working item collection and current template collection. Each item on the submission tool server may have two roles: current working item and current template. Working item is an item currently being uploaded to the submission tool server, edited, or deposited to DSpace repository. Template is an item from which the metadata is loaded when describing a working item (see Describe Item menu).

After selecting current collection for the item (Test Collection 1 on Figure 2) the name of the selected collection is shown on the top part of the window. URL of the selected collection in the DSpace repository is shown on the top part of the window. This URL can be used to make sure that target collection at the DSpace repository is correct and to browse the content of the collection.

All previously uploaded items in selected collection are listed by the name of a folder that should be similar to the title of an item. Items may be deleted by authorized users. Files of the items may be browsed and viewed. Items may include content files (bitstreams) and, if previously described by submitters, metadata and other system files. Items that in addition to the content files include metadata and system files constitute items in DSpace's Simple Archive Format (SAF).

Metadata and system files are:

```
dublin_core.xml
metadata_lom.xml
metadata_gem.xml
license.txt
contents
```

After selection of the current item folder the folder name is shown on the top part of the window (test_1 on Figure 2). An item becomes a working item. All action performed in various sections of the submission tool will be applicable only to this item. To change working item user should return to the Storage Map menu and select different item.

When a working item is selected and user selects different collection, the working item will be reset to none.

Template collection and template item have to be selected only if it is necessary to load the metadata of previously edited item serving as a template (see Describe Item). Similarly to working items, a collection and a folder with an item that serves as a template can be selected. Name of the selected current

template collection and folder is shown on the top part of the window. Templates can be browsed but not deleted in this list.

4.4. Menu Create Folder

Each item on the submission tool server is stored in a folder located under the corresponding collection. Folders are created within the current working item collection. The name of a folder should be similar to the title of an item or somehow uniquely identify an item within a collection.

4.5. Menu Upload Item

Items should be located on the local submitter's machine in order to be uploaded to the current folder for working item on the submission tool server. Files but not folders can be selected. File(s) from each subfolder should be uploaded separately. Multiple files can be selected (use "Shift"/"Ctrl").

4.6. Menu Describe Item

Submission form uses the THECB LOR Metadata Application Profile. Mandatory requirements are relaxed for all elements except Title. Repeatable elements can be added by using button Add more. They can be removed using Remove button. Input rules are linked under each field.

Subject field is using keyword extraction service and also may use various vocabularies (see Appendix B for details about the service).

Fields with multiple choice options are presented as sets of checkboxes versus dropdown lists with multiple choice options according to W3C usability recommendations.

Save button writes metadata into metadata and system files for SAF. The resulting metadata and system files may be viewed from the Storage Map menu through Browse.

Load button loads values of metadata of the current template to the form. These values can be modified as necessary and saved as metadata of current working item.

4.7. Menu Item Status

All events related to each item uploading, editing, and depositing are recorded in the Item Status log. Log includes the following events:

- Folder creation, user, date and time
- File uploading, user, date, time
- Metadata saving, user, date, time
- Depositing output and success or error code

4.8. Menu Deposit Item

This menu is used to deposit current working item to the corresponding collection in the DSpace repository. All parameters of the deposit are described on the top part of the window.

The following output end means successful deposit:

```
"Processing handle file: handle
It appears there is no handle file -- generating one
0 SAF"
```

This output is added to the Item Status Log. After successful deposit URL (handle) for the deposited item appears on the top part of the window.

Deposited item may be accessed in the target DSpace repository using given URL. Primary bitstream for deposited item should be set using DSpace repository interface.

4.9. Validation rules

Submission toll uses various validation and cross checking rules to prevent unauthorized users from altering any item or from uploading and depositing. Some of these rules also apply to authorized users to enforce validation and conflicting processes.

5. Conclusion

The submission tool allows:

- Uploading multiple files at a time
- Pausing, resuming, and committing process when preparing LOs for submission
- Using metadata templates as well as using previously submitted items as metadata templates
- Connecting to the keyword extraction service to assist metadata generation
- Creating items in Simple Archive Format for batch import into DSpace

References

The DSpace Foundation (2009). DSpace 1.5.2 Manual. Available at: <http://www.dspace.org/index.php/Architecture/technology/system-docs/index.html >

Appendix A. Technical specification

Metadata Creation Submission Toll is a standalone multi user Web based application that complies with W3C XHTML 1.0 Transitional and W3C CSS level 2.1 specifications.

The tool is built on an open source software stack described in the following table.

Class of software	Name
Operating System	Linux Ubuntu Server 8.04 Long Term Support
Web server	Apache 2.2
Web server specialized modules	suPHP module for managing DSpace connectivity permissions
Hypertext preprocessor	PHP 5.2
Database Server for controlled vocabularies	MySQL 5.0
Programming, scripting, markup, and style languages for web application	PHP, JavaScript, XHTML, XML, SQL, CSS
Remote file browser and search service	AutoIndex PHP Script
JavaScript/Flash library for multi-file uploader	SWFUpload
MySQL table manager for controlled vocabularies	phpMyEdit
Keyword extraction service	See Appendix B. Keyword extraction service

Appendix B. Keyword extraction service

Introduction

We designed and implemented two techniques for automatic keyword extraction: (1) a method relying on graph-based centrality algorithms, which we refer to as TextRank; and (2) a method based on information drawn from Wikipedia, which we refer to as Wikifier. Finally, we implemented several hybrid methods that combine the two basic algorithms and have chosen a method based on the longest common substring (LCS) as the most effective four our task (Coursey, Mihalcea, & Moen, 2008).

TextRank

The first keyword extraction method that we use relies on an unsupervised graph-based ranking algorithm, which we refer to as TextRank (Mihalcea & Tarau, 2004).

Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based ranking model is that of “voting” or “recommendation.” When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

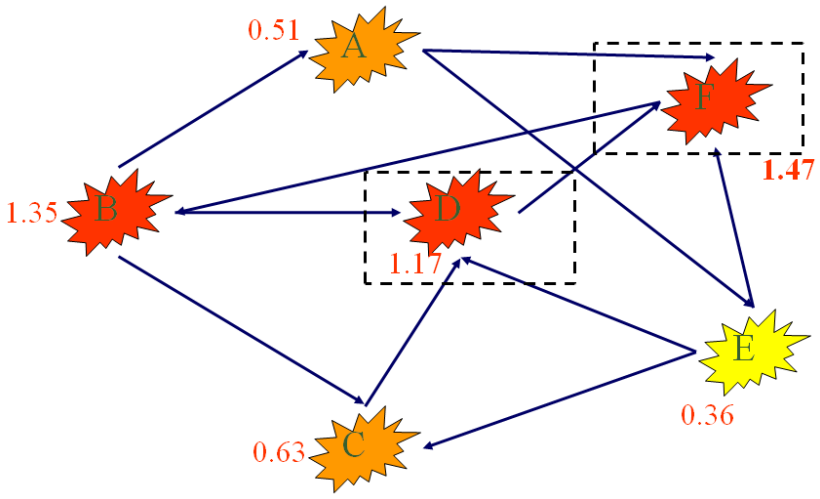

Figure 1. (Mihalcea & Tarau, 2004).

For the task of keyword extraction, the units to be ranked by the graph algorithm are sequences of one or more lexical units extracted from text, and these represent the vertices that are added to the graph.

Any relation that can be defined between two lexical units is a potentially useful connection (edge) that can be added between two such vertices. We are using a co-occurrence relation, controlled by the distance between word occurrences: two vertices are connected if their corresponding lexical units co-occur within a window of maximum N words, where N can be set anywhere from 2 to 10 words.

The vertices added to the graph can be restricted with syntactic filters, which select only lexical units of a certain part of speech. One can, for instance, consider only nouns and verbs for addition to the graph, and consequently draw potential edges based only on relations that can be established between nouns

and verbs. The best results were obtained when using nouns and adjectives. To avoid excessive growth of the graph size by adding all possible combinations of sequences consisting of more than one lexical unit (n-grams), we consider only single words as candidates for addition to the graph, with multi-word keywords being eventually reconstructed in the post-processing phase.

Next, all lexical units that pass the syntactic filter are added to the graph, and an edge is added between those lexical units that co-occur within a window of N words. After the graph is constructed (undirected unweighted graph), the score associated with each vertex is set to an initial value of 1, and the ranking algorithm described in the previous section is run on the graph for several iterations until it converges – usually for 20-30 iterations, at a threshold of 0.0001.

Once a final score is obtained for each vertex in the graph, vertices are sorted in reversed order of their score, and the top T vertices in the ranking are retained for post-processing. During post-processing, all lexical units selected as potential keywords by the TextRank algorithm are marked in the text, and sequences of adjacent keywords are collapsed into a multi-word keyword.

Wikify!

In order to automatically identify the important encyclopedic concepts in an input text, we use the system Wikify! (Mihalcea and Csomai, 2007), which identifies the concepts in the text that are likely to be highly relevant (i.e., “keywords”) for the input document, and links them to Wikipedia concepts.

Wikify! works in three steps, namely: (1) candidate extraction, (2) keyword ranking, and (3) word sense disambiguation. The candidate extraction step parses the input document and extracts all the possible n-grams that are **also** present in the vocabulary used in the encyclopedic graph (i.e., anchor texts for links inside Wikipedia or article or category titles).

Next, the ranking step assigns a numeric value to each candidate, reflecting the likelihood that a given candidate is a valuable keyword. Wikify! uses a “keyphraseness” measure to estimate the probability of a term W to be selected as a keyword in a document, by counting the number of documents where the term was already selected as a keyword $count(D_{key})$ divided by the total number of documents where the term appeared $count(D_W)$. These counts are collected from all the Wikipedia articles.

$$P(keyword|W) \sim= count(D_{key})/count(D_W)$$

This probability can be interpreted as “the more often a term was selected as a keyword among its total number of occurrences, the more likely it is that it will be selected again.”

Finally, a simple word sense disambiguation method is applied, which identifies the most likely article in Wikipedia to which a concept should be linked to. This step is trivial for words or phrases that have only one corresponding article in Wikipedia, but it requires an explicit disambiguation step for those words or phrases that have multiple meanings (e.g., “plant”) and thus multiple candidate pages to link to. The algorithm is based on statistical methods that identify the frequency of meanings in text, combined with symbolic methods that attempt to maximize the overlap between the current document and the candidate Wikipedia articles.

Hybrid Methods

The two systems have different strengths that have been combined. TextRank provides an estimate of the attention a reader would give terms in any given text. Wikifier recognizes phrases that large numbers of people consistently use to point to Wikipedia articles. Wikifier uses the broad coverage of Wikipedia to recognize those entities that human annotators have felt were important to reference, while TextRank provides the ability to handle novel yet important topics for which no Wikipedia article may yet exist.

To find commonality between the outputs of the two systems three methods were tested: set union, set intersection union and longest common substrings (LCS). Overall, using LCS on the output of both TextRank and Wikifier provides a higher precision output than either alone.

References for Appendix B

Coursey, K. H., Mihalcea, R., & Moen W. E. (2008). Automatic Keyword Extraction for Learning Object Repositories. In Grove, A. (Ed.), *71st ASIS&T Annual Meeting: Vol. 45. People Transforming Information - Information Transforming People.* Richard B. Hill.

Mihalcea, R. & Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, Lisbon, Portugal.

Mihalcea, R. & Tarau, P. (2004). TextRank: Bringing Order into Texts, In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP 2004)