Digital Library Curriculum Development

Module 5-b: Application Software

(Last updated: 10/07/2009)

1. Module name

Application software

2. Scope

This module covers commonly used application software, which are specifically designed for the creation and development of digital library (DL) systems and similar types of collections and services, for example, digital repositories or open access archives.

Note: Section 9 "Body of knowledge" lists multiple technologies used in application software. Since the technologies evolve and the applications are being updated, please refer to the documentation on the application software homepages for details of the latest information.

3. Learning objectives

a. Students should know the features and technologies (e.g., OS, servers, indexing/searching system, programming language of the source code, etc.) of the DL application software, which are introduced in this module. Then, students should be able to evaluate the DL application software through critical comparison.

b. Students should be able to search, browse, add and delete items from the digital library systems built by the DL application software.

c. Students should be able to critically compare different application software

Note: The following optional objective, 3.d, might be achieved through a semester-long class project, which is to develop a DL system using application software. For details, please see 'Optional semester-long project 12.d' in the section 12.

d. (Optional) Students are able to both install and configure DL application software. This is to provide practical experiences to students.

4. 5S characteristics of the module

Four S's are present – Streams, Spaces, Scenarios and Structures. However, Societies component (e.g., DL patrons, administrators, etc.) was not considered in this module.

a. Streams: current DL applications are typically designed to deal with various types of data such as multimedia data (e.g., audio, images, videos) as well as text data.

b. Spaces: storage space to store digital contents and the user interface for the DL patrons to communicate with the system are present in the application software.

c. Scenarios: DL application and its patrons interact with each other following a series of steps to achieve tasks.

d. Structures: DL application software has their architecture, metadata formats used, etc., which have the structure.

5. Level of effort required (in-class and out-of-class time required for students)

To achieve learning objectives 3.a, 3.b and 3.c:

a. Out-of-class time:

Preparation for group presentations (Learning activity a-1): 4-6 hours (reading the assigned papers or web pages, creating and submitting concept maps individually and preparing group presentation slides)

Writing a short white paper (Learning activity a-2): 1-3 hours (assuming that the assigned papers are already read)

Review of demos, etc. (Learning activity b): 1-2 hours (visiting the demo sites, trying basic services such as searching, browsing, depositing an item, removing a deposited item or watching a short video tour)

b. In-class time: total 2 hours

1.5 hours for presentations and question/answer session and 0.5 hours to complete the learning activity c (assuming that the assigned papers are already read).

To achieve (optional) learning objective 3.d:

a. Out-of-class time: it depends on the project. It is expected that this learning objective will be achieved through a semester-long project.

6. Relationships with other modules

The module 5-a: Architecture overview/models should be taught in advance so that the students could have the base knowledge about the DL architectures/models to learn about the application software, which were developed based on those knowledge.

After this module 5-b is taught, 9-a: Project management and 9-b: DL case studies module can be taught to provide students the real-world examples of the projects and DL systems created by the application software.

7. Prerequisite knowledge required (completion optional)

If DL application software is to be installed and configured as an optional learning activity and the instructor would like to supervise and help student groups, some knowledge about the pre-requisite software such as database systems (e.g., MySQL), Linux (e.g., Fedora Core, Ubuntu), HTTP server (e.g., Apache) as well as some knowledge about metadata, digital objects, indexing and collection building might be useful.

8. Introductory remedial instruction

None

9. Body of knowledge

Topic: EPrints (version 3)

1. Overview
    a. It was developed in 2000 as a direct outcome of Santa Fe meeting in 1999, where there was the first meeting of the Open Archives Initiative.
    b. It is commonly used as an institutional repository
    c. It has been developed at the University of Southampton School of Electronics and Computer Science
    d. Open source under GPL license
    e. A list of real-life systems using EPrints can be found at: http://www.eprints.org/software/archives/
    f. Or visit the Electronic Theses and Dissertations (ETD) Individuals repository at http://etdindividuals.dlib.vt.edu:9090/ for a specific example
2. Features
    a. Duplicate avoidance
    b. Auto complete for entering metadata
    c. Full-text search
    d. Metadata search
    e. Subscriptions
    f. Multi-language support
    g. Optional multi-lingual metadata

(The benefits of the new features for administrators, developers, researchers, institutions, depositors, etc. are introduced below - excerpt and modified from Eprints homepage at http://www.eprints.org/software/v3/)

a. Repository managers

    i. With metadata auto-completion feature, the collections value and its metadata quality can improve.

b. Depositors

    i. Takes less time to deposit with metadata auto-completion
    ii. Import data from other repositories and services

c. Researchers

    i. Works with desktop applications and new Web 2.0 services
    ii. RSS feeds and email alerts keep you up-to-date

d. Developers

    i. Tightly-managed, quality-controlled code framework
    ii. Flexible plug-in architecture for developing extensions

e. Institutions

    i. Can create high quality institutional open access collections
    ii. Conforms with research funding agency's open access mandates

3. Content types
    a. Text
    b. Multimedia (image, audio, video)
4. Technologies used
    a. Unix-like OS (e.g., Linux)
    b. Written in Perl (allows rapid development and modification)
    c. XML (for import/export of data, partial configuration)
    d. Apache server with mod_perl installation
    e. MySQL database
    f. Unicode (UTF-8 encoding)
    g. OAI-PMH support

Topic: DSpace

1. Overview:
    a. It was developed as a collaboration between MIT libraries and Hewlett Packard Research Lab

   b. Research institutions use it to build various digital archives for institutional repositories, learning object repositories, digital preservation, publishing, etc.
   c. Open source under BSD license
   d. A list of repositories using DSpace can be found at: http://www.dspace.org/index.php?option=com_content&task=view&id=596
   e. Or visit the Electronic Theses and Dissertations (ETD) repository in the University of North Carolina at Chapel Hill at http://etd.ils.unc.edu/dspace/ for a specific example
2. Features
   a. Long-term preservation supported
      i. There are three types of data formats (supported, known and unsupported types)
      ii. For all three types, DSpace does bit preservation: the preserved file remains exactly the same over time – not a single bit is changed
      iii. For supported types, DSpace does functional preservation: the file changes over time so that the material can be immediately usable in the same way it was originally, while the physical media and digital formats change
   b. Interoperability
      i. It can export digital content with its metadata in an XML-encoded file or METS
      ii. DSpace Java API can be customized to allow interoperation with other systems
      iii. Handle System from CNRI is assigned to each digital item as a persistent identifier
   c. Support for Open Archives Initiative's Protocol for Metadata Harvesting (OAI-PMH)
      i. DSpace supports OAI-PMH v.2.0 as a data provider
      ii. OAI support was implemented using OCLC's OAICat
      iii. Institutions running DSpace can turn on and off OAI and choose to register as a data provider or not
3. Content types
   a. Text
   b. Multimedia (image, audio, video)
4. Standards
   a. Well-defined APIs for interoperability with other systems
   b. CNRI handles for persistent identifiers
   c. X.509 certificate-based access control
   d. Dublin Core metadata for digital objects
   e. OAI-PMH for metadata harvesting/providing
   f. METS profile can be used to export digital items
5. Technologies used
   a. Operating system: Linux, Solaris, HP/UX, etc.
   b. Server: Apache, Tomcat, OpenSSL

c. Indexing/searching: Lucene
d. Database system: PostgreSQL, JDBC
e. CNRI Handle System
f. Jena (RDF history system)
g. Java, JSP, Servlets
h. JUnit (testing) and Log4j(logging)

Topic: Greenstone

1. Overview
    a. It was developed and distributed as an international cooperative effort established in 2000 by the University of Waikato with UNESCO and Human Info NGO, "New Zealand Digital Library Project."
    b. It helps the universities, libraries and public service institutions build their own digital libraries.
    c. It is a suite of software that has ability to build new digital library collections and provide services for them.
    d. Open source under General Public License (GPL)
    e. A list of systems using Greenstone is at: http://www.greenstone.org/examples
    f. Or visit Oxford Digital Library at http://www2.odl.ox.ac.uk/gsdl/cgi-bin/library/ for a specific example
2. Features
    a. Installation of Greenstone digital library (GSDL)
        i. It runs on Windows, Unix/Linux, and Mac OS/X. It can be installed easily by using the ready-to-use binaries which are included in the distribution (but some functionality is limited).
        ii. It might be installed on a laptop for personal use (built-in web server), or run on the main web server (Apache or Windows IIS).
    b. Collection building
        i. It can harvest documents over OAI-PMH to include them in a collection
        ii. Full text tagging is supported for hierarchical document browsing
        iii. Automatic text extraction and indexing are provided
        iv. Data compression is supported
        v. Metadata
            1. Automatic extraction of simple metadata
            2. Explicit metadata via classifiers
            3. Used for browsing and searching
        vi. Multiple languages supported via Unicode
    c. Browse and search provided
        i. Full text search
        ii. Metadata field search
        iii. Either Boolean or ranked (when indexed with MG indexer)
        iv. Search history, search term highlighting, etc.
    d. Presentation

        i. Search results formatting available
        ii. Homepage customization available
    e. Collection administration
        i. Adding new documents (batch operation)
        ii. Usage monitoring
        iii. Security
    f. Interoperability
        i. Any Greenstone collection can be exported to DSpace
        ii. Any DSpace collection can be imported into Greenstone
        iii. Any collection can be exported to METS (in the Greenstone METS Profile) and Greenstone can ingest documents in METS form
    g. Customizable, extensible
        i. New document and metadata formats can be accommodated by writing 'plug-ins' in Perl
        ii. New metadata browsing structures can be implemented by writing 'classifiers.'
        iii. User interface can be customized using 'macros' written in a simple macro language
        iv. CORBA protocol allows agents (e.g., written in Java) to use all the facilities associated with document collections
3. Architecture
    a. Receptionist
        i. Provide user interface
        ii. User input accepted
        iii. Page generation
        iv. Send to appropriate collection server
    b. Collection server
        i. Collection content management
        ii. Search/filter information
        iii. Return results
        iv. Handle multiple collections
    c. Metadata supplied by communities
4. Content types
    a. Text
    b. Multimedia (image, audio, video)
5. Standards
    a. Dublin Core metadata for digital items
    b. Z39.50 client-server protocol for searching and retrieving information from remote computer databases.
    c. Support for OAI-PMH both as a client and a server
    d. Unicode for multiple language support
6. Technologies used
    a. Greenstone runs on all versions of Windows and Unix/Linux and Mac OS-X.
    b. Apache HTTP server

c. Source code in C++ (experimental Greenstone v.3 is written in Java) and Perl available
d. Greenstone provides a choice of three indexing tools
    i. MG is the default indexer. It does section level indexing and the searches can be either Boolean or ranked. For phrase searching, Greenstone does 'AND' search on all the terms.
    ii. MGPP (MG plus plus, new version of MG). It does word level indexing, which provides fielded, phrase and proximity searching. Boolean searches can be ranked. Document/section levels and text/metadata fields are all handled by the one index. It's a bit slower compared to MG when large data is to be indexed considering MGPP does word level indexing.
    iii. Lucene was added for incremental collection building, which cannot be provided by MG and MGPP. It handles field and proximity searching but only at a single level for example, complete documents or individual sections but not both. It also provides single-character wildcards and range searching.
e. Multiple GNU software are integrated
    i. Apache web server
    ii. Perl
    iii. wget to download pages from the web
    iv. XML::Parser used to read and write internal XML documents
    v. Stemmer for English documents
    vi. CVS for version control
    vii. GDBM for database
    viii. and many more

Topic: CONTENTdm

1. Overview

    a. It was conceived by the Center for Information Systems Optimization (CISO) at the University of Washington. It was then taken over and extended by the Online Computer Library Center (OCLC).
    b. It is commercial software.
    c. Its users are universities, public libraries, government entities, museums, non-profit organizations, etc.
    d. It is 100 percent web compatible so the servers and collections can be administered remotely. There could be a maximum of 50 'acquisition stations', which are remote locations for items and their metadata entry. Those data entered through the acquisition stations are stored and provided by the central CONTENTdm server.
    e. Collection sharing is supported.

i. Collections can be added to OCLC WorldCat catalog system so that the user collections can be part of WorldCat's 80 million record global catalog.
ii. CONTENTdm functions as OAI data repositories for the users who want their metadata available for harvesting.
iii. Its Multi-Site Server allows users to query multiple CONTENTdm servers from a single user interface.

f. Example collections can be browsed at http://www.oclc.org/contentdm/collections/
g. Or visit the Virginia Commonwealth Univ.'s PS Magazine, the Preventive Maintenance Monthly collection at http://dig.library.vcu.edu/cdm4/index_psm.php?CISOROOT=/psm

2. Features (based on http://www.oclc.org/contentdm/about/default.htm)

a. It supports both text documents and multimedia. For example, it builds documents, books and other multiview and multipage materials. It can also present video and audio files with related transcripts.
b. By using the batch import tools, it can import images and metadata quickly and easily as well as text files for full-text searching.
c. By utilizing the compound object import wizard, CONTENTdm can import multiple compound objects, such as newspapers, in batches. I also can queue multiple compound objects and process them during off-hours to not slowdown the system use.
d. It supports JPEG2000, which is a format for high-quality and large format images without a browser plug-in.
e. To prevent unwanted copying of images it manages, CONTENTdm has three different options for image rights: band, brand or watermark. Band uses a band of color and words (in here, a 'band' means a layer in a digital image. The term originally came from electrical engineering field to represent a range of wavelengths or colors). Brand uses icons and words. Watermark uses grayscale images.
f. For digitized text documents, CONTENTdm provides an integrated Optical Character Recognition (OCR) capability for full-text searching. Users will be able to search words in the digitized text in addition to searchable metadata fields within your collections. When viewed, items prepared with this feature will display highlighted search terms within the digitized document image.
g. To index subjects of various still images (so that they can have consistent and uniform metadata), CONTENTdm uses the *Library of Congress Thesaurus for Graphical Materials I* (TGM I), which provides a controlled vocabulary to describe activities, objects, types of people, events or places. Proper noun names of those are excluded. As an option, you can develop your own controlled vocabulary to index images.

h. It provides customizable user interfaces—Create predefined queries and customized interfaces to collections.
i. Its flexible search features include Dublin Core and Latin-1 character set support, Boolean search and advanced search option. Advanced search option provides search-by-fields, across all fields, by proximity, and across one or many collections. CONTENTdm also auto-generates the search terms based on the existing metadata.

3. Content types

    a. Text
    b. Multimedia (e.g., image, video, audio)
    c. Compound objects (items which consist of multiple views. For example, two-sided objects such as postcards, brochures, ticket stubs, or six-sided objects such as images of a chair seen from six different directions)
        a. CONTENTdm allows the users to define compound objects so that all the views of a compound object can be retrieved.
    d. Null data type support for the items not yet in the system
    e. URL data type support allows lengthy video and audio files stored in the streaming media server to be accessed through CONTENTdm.

4. Standards and technologies

    a. CONTENTdm is fully compliant with OAI-PMH v.2.
    b. Its default metadata templates are Dublin Core and Visual Resource Association (VRA) Core. Collection admins can still add their own descriptions.
    c. It is Z39.50 (client-server protocol to access and retrieve information in remote computers) compatible through ZCONTENT, open source software developed by the Univ. of Utah Marriott Library. ZCONTENT allows users to access the collections of CONTENTdm and download items.
    d. XML is used for all the internal structure description. For example, it is used to export the metadata descriptions in order to work with other systems that have different metadata standard.

(Optional) Topic: Critical Comparison of the DL application software

Based on the resources in the next section 10. Resources (especially 'Comparing the DL application software'), a comparison table can be built to show the similarities and differences of the DL application software. This will provide students with ability to think critically when they need to select DL application software to set up a DL.

10. Resources

Note: Feel free to read about features, technologies and (optionally) installation and configuration manuals as well as the assigned portion in the software homepages.

- Eprints 3
  - Reading for students
    - Read 'Introducing EPrints 3' and watch short QuickTime video clips at http://www.eprints.org/software/v3/
- DSpace
  - Reading for students
    - Visit DSpace homepage at http://www.dspace.org/ and read 'About DSpace' under 'New to DSpace?' on the top left pane.
  - Advanced reading for students (optional) and instructors
    - DSpace architecture review group, "Toward the next generation: Recommendations for the next DSpace Architecture", January 24, 2007. http://wiki.dspace.org/static_files/0/0e/DSpace-recs.pdf
- Greenstone
  - Readings for students
    - Ian H. Witten and David Bainbridge, A brief history of the Greenstone Digital Library Software, at http://wiki.greenstone.org/wiki/gsdoc/others/Greenstone_history.htm
    - Katherine J. Don, David Bainbridge, and Ian H. Witten, The design of Greenstone 3: An agent based dynamic digital library, at http://www.greenstone.org/docs/greenstone3/gs3design.pdf
  - Advanced readings for students (optional) and instructors
    - Ian H. Witten and David Bainbridge. (2003). How to build a digital library. Morgan Kaufmann.
- CONTENTdm
  - Readings for students
    - Visit http://www.oclc.org/contentdm/about/default.htm and read the topics under 'About' on the left pane.
- Comparing the DL application software
  - Readings for students
    - Witten, I. H., Bainbridge, D., Tansley, R., Huang, C. & Don, K. J. (2005). StoneD: A Bridge between Greenstone and DSpace. *D-Lib Magazine*, 11(9). http://www.dlib.org/dlib/september05/witten/09witten.html
    - Wang, J. Y., Assion, M. & Matthaei, B. (2003). Open Archives Forum: Inventories-Open Archives Software Tools. http://www.oaforum.org/otherfiles/tv-tools.pdf
    - William Nixon. DAEDALUS: Initial experiences with EPrints and DSpace at the University of Glasgow. Article is in http://www.ariadne.ac.uk/issue37/nixon/
    - Goh, D. H.-L., Chua, A., Khoo, D. A., Khoo, E. B.-H., Mak, E. B.-T., & Ng, M. W.-M. (2006). A checklist for evaluating open source

digital library software. Online Information Review, 30(4), 360-379. http://dx.doi.org/10.1108/14684520610686283

11. Concept maps (created by students)

After studying the material in this module, students will create a concept map, which represents the concepts in the module and their relationships with one another. By transforming the knowledge in their mind into a graphical representation, students will have a 'clearer picture' of the content.

Students might create concept maps not only for the content in the body of knowledge section, but also for the learning activities section. For example, students may show the steps to search, browse, add, delete, import or export an item in a concept map. Or, they can list features of different DL application software and compare them to promote critical thinking. Even a concept map (or multiple concept maps) can be created for the semester-long DL development project, showing several phases of the project such as preparation step, actual installation and configuration of the software, content selection, collection development, etc.

Note: IHMC Cmap Tools is an open source client tool to create concept maps. CmapServer enables the users to collaborate and share concept maps anywhere on the internet. Both software can be downloaded freely for educational purposes from http://cmap.ihmc.us/download/index.php

12. Exercises / Learning activities

a-1. Group presentations on specific DL application software

Note: These group presentations will substitute for a formal lecture by the instructor. The instructor should be prepared to fill in gaps or make corrections if any of the presentations are incomplete or misleading.

- During the previous class, students form into groups. Each group chooses a particular DL application software for their group presentation. Readings are assigned from the Resources list in section 10. Instructors may provide guidelines such as what to address and time limits for each group presentation.
- The students in a group should work together to create their presentation slides explaining the features and other information of the software such as services, technologies and standards used.
- In the class, each group gives a presentation about their application software followed by a question and answer session. Each presentation should be done within a time limit given by the instructors.

a-2. Writing a short white paper on specific DL application software (suggested by Bob Allen, iSchool at Drexel)

Note: This individual activity may be used instead of a-1: Group presentation activity. This exercise can provide details of the DL application software to students. Instructors should select appropriate class activity based on the students' learning styles.

- The instructor assigns (or each student selects) a topic (e.g., one of DL application software) and write a short white paper about it.
- The instructor reviews students' white papers for accuracy and provide necessary comments to students.
- Students revise their white paper based on the instructor's comments and re-submits it.
- The instructor makes the white papers available to all students by posting them to the class website so that the papers could be used as resources.

b. Individual learning activity: Interacting with software demos

Prior to the class session, each student should complete the following activities. Students may work individually or together.

- (EPrints demo) Try searching and browsing. You need to create an account if you want to try depositing an item. Examine the metadata fields when you enter the metadata while depositing an item.
  - Demo site at http://demoprints3.eprints.org
- (DSpace demo) Interactive demo for students
  - Learn how to submit an item at http://libraries.mit.edu/dspace-mit/build/dspace-demo.html
  - Try searching and browsing at http://dspace.mit.edu/
- (Greenstone demo)
  - Demo page for searching for an item at http://diglib.auburn.edu/gsdlr/cgi-bin/library?site=localhost&a=p&p=about&c=demo&ct=0&l=en&w=utf-8
- (CONTENTdm) Watch the four minute tour video clip at
  - http://www.oclc.org/contentdm/tour/tour.htm

c. Scenario-based application software selection (prior to this exercise, please read the papers under 10.Resources → Comparing the DL application software)

In this exercise, three scenarios are given. For each scenario, your job is to select the most appropriate digital library (DL) application software, which might satisfy the requirements described in the scenario. In addition, you need to provide your reasons to do that.

In case students have diverse opinions about the software selection, there will be a class debate to resolve the issues.

- Scenario 1:
  As a member of an NGO, Susan has been working with indigenous people in country X, which is one of the third world countries. This tribe has a unique wood carving techniques using hand tools and a variety of traditional songs that are sung when they work together to make a huge sculpture. These techniques and songs have been passed down from generation to generation until recently.

  However, things have been changed significantly. Younger generation is not interested in learning the wood carving techniques and songs anymore. To them, movies and rock music are new and much more fun culture. But the chief of the tribe and older generation realize the importance of preserving their unique culture, so they consulted Susan to develop a local digital library. She has a background in Library and Information Science and could educate people about the basics of librarianship.

  With the help of the tribe, Susan could take pictures of the wood artifacts. She also collected traditional songs using a digital recorder. Since the data is prepared, the next step is to set up a digital library system on a computer. But, the problem is that there is no internet connectivity in that region. There are some computers for educational purposes, but they are old, 386-processor computers. Considering the situation in this village, what kind of digital library application software should Susan choose? And why?

- Scenario 2:
  The professors in the school of engineering at Y Tech are very active in their research projects. Every year, they and their graduate students publish various journal articles, conference proceeding papers, posters, demos, as well as experimental program codes. The dean made a decision to set up a central repository to collect all those published materials for internal use (and possibly open to public in the future).

  She puts much emphasis on the 'preservation' of the resources in this repository because there are foundational theories developed and published in this school, whose access should be guaranteed even after several decades. In addition, scholars should be able to access this repository on the internet. Since this system is used by faculty members and graduate students themselves, the process of submission should be simple and does not require knowledge of a librarian. To set up this central repository, the school recently purchased a computer, which is equipped with quad-core processors, internet connectivity and a Linux operating system. Which digital library application software should be selected to set up a repository in this school? Please provide your reasons, too.

- Scenario 3:
  A research institution, S, produces tons of text-based content such as peer-reviewed journal articles, theses and dissertations as well as some multimedia content such as data sets, images from experiments, lecture video clips and audio

files. In addition, the institution plans to import educational resources in a couple of months later from other institutions. Those materials need to be preserved for a long period of time considering their values.

Therefore, a project was set up to establish an institutional repository, which would hold the content mentioned above (mostly text-based material). The principal investigators of the project made a list of requirements that should be satisfied by the repository. They are as follows:

- Students and faculty members should be able to easily deposit and manage their submitted content such as theses, dissertations, lecture slides, lecture video notes, or audio data files
- The administrator's user interface in the repository software should support easy maintenance of the repository software settings
- The repository software should be OAI-compliant
- The repository software should support a type of educational material element set (e.g., SCORM).

Which application software will be appropriate in this institution? EPrints? DSpace? CONTENTdm? Or would it be better to create two repositories using the different application software then connect the two repositories seamlessly? If you were one of the principal investigators, what would you say in the PI meeting? Please provide your reasons to choose the software.

d. Optional semester-long project (group activity)

- Step 1: Students form a group and meet with clients who want to have a customized DL system developed.
- Step 2: The clients give the student groups specifications of the DL systems they want.
- Step 3: Each student group explores different DL application software to find the most appropriate application to meet their client's needs.
- Step 4: Student group installs the application software chosen in step 3, including installation of any pre-requisite software. For example, to install EPrints 3 in a linux machine, the Perl programming language along with its multiple modules, MySQL database and Apache server should also be installed in advance.
- Step 5: The installed application software is customized. For example, the students might configure the subject classification system as the Library of Congress (LOC) system or ACM classification, change the appearance of the user interface, modify the metadata fields used, etc.
- Step 6: The client verifies whether the installed DL application software is appropriately configured to meet his/her needs.

- Step 7: Student groups begin to create collections by adding the items provided by their clients to develop a DL (e.g., adding a group of pictures to create the Digital Library of Native American History or the Virginia Digital Museum of Cars, etc.).
- Step 8: Student group members make sure all the services of the developed DL system work well.
- Step 9: Clients evaluate the developed DL system and the student group refines it based on the feedback.

13. Evaluation of learning objective achievement

Note: Since the learning objectives and the learning activities are in one-to-one mapping relationships, the performance and the quality of the learning activities achievements are evaluated as the means to evaluate the learning objectives of this module.

a. Group presentations on specific application software

The group presentations described in section 12 could be graded, to evaluate students' learning.

- Group presentations might be evaluated in terms of their comprehensiveness (did they include the important features and characteristics of the software?), their clarity (did they explain the software in a way that it could be distinguished from the alternative software packages?), and the quality of the presentation (e.g., slide quality, presentation style, use of time, and Q/A session).

b. Individual concept maps on specific application software

- After the class, each individual student creates one or more concept maps for the different application software packages and submits them to the instructor. The concepts maps are expected to demonstrate the student's overall understanding of all four software packages introduced in this module.
- The concept maps should be evaluated in terms of their comprehensiveness (did they include all the major concepts covered in the module?), their richness (were the concepts well-connected?), and their organization (was there a clear depiction of the concepts and their relationships?).

c. After reading each of three scenarios, students will answer the questions at the end of each scenario and participate in the class discussion. Instructors will assign points based on the participation and the clarity of the reasons.

d. (Optional) semester-long project

- Each instructor may develop a different method for evaluating the learning achieved through this project. We suggest that points might be assigned as follows:

- The DL application software is incorrectly installed and not working (0 points)
- DL application software as well as all the pre-requisite software is correctly installed (3 points)
- DL application software is installed and fully configured (6 points)
- The DL system is fully configured and a collection(s) is created with the data provided by the clients (10 points)
- All the features of the new DL system are fully functional (15 points)

14. Glossary

Application software is a complete, self-contained program that performs a specific function directly for the user. This is in contrast to system software such as the operating system kernel, server processes, libraries which exists to support application programs and utility programs. – Dictionary of Computing –

API (Application Programming Interface) The interface (calling conventions) by which an application program accesses operating system and other services. An API is defined at source code level and provides a level of abstraction between the application and the kernel (or other privileged utilities) to ensure the portability of the code. An API can also provide an interface between a high level language and lower level utilities and services which were written without consideration for the calling conventions supported by compiled languages. In this case, the API's main task may be the translation of parameter lists from one format to another and the interpretation of call-by-value and call-by-reference arguments in one or both directions. – Free On-Line Dictionary Of Computing –

CORBA is the acronym for Common Object Request Broker Architecture, OMG's open, vendor-independent architecture and infrastructure that computer applications use to work together over networks. Using the standard protocol IIOP, two application programs that are based on CORBA but developed by different vendors, on different operating systems, programming languages can interoperate with each other. – Object Management Group (OMG) –

OpenURL is a type of URL that contains resource metadata for use primarily in libraries. The National Information Standards Organization (NISO), has developed OpenURL and its data container (the ContextObject) as international ANSI standard Z39.88. On 22 June 2006, OCLC was named the maintenance agency for the standard.

Dublin Core metadata element set is a standard for cross-domain information resource description. It provides a simple and standardized set of conventions for describing things online in ways that make them easier to find. Dublin Core is widely used to describe digital materials such as video, sound, image, text, and composite media like web pages. Implementations of Dublin Core typically make

use of XML and are Resource Description Framework based. Dublin Core is defined by NISO Standard Z39.85-2007

Z39.50 is a client-server protocol for searching and retrieving information from remote computer databases. It is covered by ANSI/NISO standard Z39.50, and ISO standard 23950. The standard's maintenance agency is the Library of Congress. Z39.50 is widely used in library environments and is often incorporated into integrated library systems and personal Bibliographic Reference software. Interlibrary catalogue searches for interlibrary loan are often implemented with Z39.50 queries.

OAI-PMH (*Open Archives Initiative Protocol for Metadata Harvesting*) is a protocol developed by the Open Archives Initiative. It is used to harvest (or collect) the metadata descriptions of the records in an archive so that services can be built using metadata from many archives.

XML (**Extensible Markup Language**) is a general-purpose markup language. It is classified as an extensible language because it allows its users to define their own tags. Its primary purpose is to facilitate the sharing of structured data across different information systems, particularly via the Internet.

SOAP (Service Oriented Architecture Protocol) is a protocol for exchanging XML-based messages over computer networks, normally using HTTP/HTTPS. SOAP forms the foundation layer of the Web services stack, providing a basic messaging framework that more abstract layer can build on.

15. Additional useful links

16. Contributors

Seungwon Yang (Virginia Tech, development), Edward A. Fox (Virginia Tech, evaluation), Eunyee Koh (Texas A&M, evaluation), Uma Murthy (Virginia Tech, evaluation), Jeff Pomerantz (UNC, evaluation), Sanghee Oh (UNC, evaluation), Barbara Wildemuth (UNC, evaluation), Robert B. Allen (iSchool at Drexel, evaluation), Sally Jo Cunningham (Univ. of Waikato, evaluation)