Introduction

The Digital Library on Societal Impacts (DLSI) is a Web-based resource for collecting and disseminating research findings related to the use and value of weather forecasts. The goal of the library is to make research results, including case studies, Web sites, decision support tools, and other resources accessible to appropriate communities. In particular, results should be synthesized so that they are useful to policy makers and researchers interested in the use of weather forecasts.

The purpose of this document is to describe the system architecture and steps needed to develop the digital library. DLSI will be a Web application. The library will exist on a Web server, and you will use a Web browser to connect to the library. The three critical elements needed to implement the library are a user interface (the presentation of information to users), infrastructure (the server-side storage and manipulation of data), and content (the resources that populate the library). This document will describe the necessary steps toward creating these elements.

This document will first describe an architecture for the digital library and the current work that has led to this design. It will then present a set of steps and milestones in the development of the library. Finally, potential problems and possible solutions will be presented in the Exposures section.

System Description

DLSI follows the design of most Web applications with three layers of architecture. Figure 1 provides an overview of the proposed architecture. Users interact with the library through a Web browser, and there are interfaces for different tasks. For example, there will be an interface for browsing the library contents, another for searching, and a third for adding or changing information contained in the library. There isn't necessarily a one-to-one correspondence between Web pages and tasks. The user interface responds to user actions and presents the results of the user actions in the form of Web pages. The results of user actions are passed to the middle layer, which is the core of the digital library. In general, this layer takes user queries and searches the library, returning results to the user. In DLSI, the DLESE tools handle much of the complexities of doing the work of managing stored records. We will build a translation layer to
An example of library use may help illustrate the role of each layer. A user may use a search interface to retrieve case studies of blizzards in New York between 1980 and 1990. The user would fill out a form in a Web browser. This form is generated by the user interface layer. When the user fills out (submits) the form, the user interface layer initiates a search action. The infrastructure layer provides the functionally to perform the search. It transforms the user request (blizzard, New York, 1980–1990) into categories used in digital library entries (storm type, location, time). It then searches the database for items that match the search parameters. The database in the content layer returns records, each of which is an entry to a library resource. The infrastructure layer passes these records to the user interface, which turns the library record into text that populates a Web page.

In the section that follows, each of the three layers and the implementation requirements of each layer will be discussed.
User Interface

The user interface provides the mechanism by which people interact with the computer's functionality. Typically a user's understanding of a computer application is governed by the behavior of the user interface. In a Web application, the interface is hosted within a user's Web browser. The user clicks on information and fills out forms. Because of the architecture of the Web, search is exclusively user-driven. A user requests a piece of information (by clicking on a link) and the server responds. The important ramification of this is that new pages are always requested by the user—the server does not spontaneously push new information to the client.

What a user interface must provide depends intimately on the tasks that the application must support. This is based on the needs of the users and representative tasks. In DLSI, the users include:

- Policy makers
- Research funding organizations
- Researchers
- Private sector representatives

Representative tasks include:

- Retrieving contents from the library
  - Finding interesting tools
  - Finding research results
  - Finding summaries of research results
  - Finding colleagues
  - Tracking ongoing projects

- Adding data to the library

Designing an effective application requires designing a user interface that is appropriate for the users and which helps them complete their tasks quickly and efficiently. Understanding user needs and testing possible interface designs with users is essential.

We will employ a task-based iterative prototyping methodology for DLSI. We will first create lists of tasks and create designs that can help users accomplish those tasks. We will then create prototypes (such as paper sketches, storyboards, and screen-shot mockups) so that we can evaluate the design with users. Based on increased understanding of user tasks and user feedback, we will refine the designs.

A Web application design is implemented by writing HTML which is presented in a user's Web browser. Because of the technologies used in the infrastructure layer, the design will use Java Server Pages (JSP) technology. Several Web design tools ease the transition between informal mockups and a working dynamic Web site.

Infrastructure

The infrastructure code is the heart of the digital library. Using the metaphor of a physical library, the infrastructure helps query the contents of the library collections. The infrastructure determines the kinds of searches that can be performed on the content, the ways to filter relevant content, and mechanisms to categorize new materials as they become available.

Several online digital libraries create software that aids in the creation of new digital libraries. These systems simplify the library design task by providing search functionality, managing the library content, checking that the library content is consistent (i.e., no broken links), and provide ways to automatically query other Web-based libraries. The Digital Library for Earth System education (DLESE) is one such system. DLESE is particularly well suited for DLSI for sev-
eral reasons. DLESE is well established in the geoscience education community, and as such their software is mature and has a proven track record. Software to manage the library is open source and such is freely available. DLESE's success with its educational audience has led to a successful user interface for that community that can be used as a starting point for our library.

DLESE provides general purpose library mechanisms such as textual search for content, categories to organize related links, and metadata (data that formally describes what the content contains) which can be used for custom search and presentation. Because DLESE's content and metadata was created for educational purposes, it will require some modification for use in a digital library not focused exclusively on cataloging educational resources. For example, whereas “intended audiences” in DLESE would include K-12, undergraduate, or post-graduate classes, intended audiences in DLSI would include policy makers and representatives from various sectors (energy, transportation, and so on.) It will therefore be necessary to customize the data stored in DLESE and the ways to query the data in a way that is appropriate for DLSI users.

**Content**

Useful content is the main reason why users will visit the digital library. It is desirable to index as much content as possible so that the information that users need is available to them.\(^1\) The content of DLSI will include articles, links to tools, Web sites, and projects.

In a physical library, the library owns the collection of materials. Library books appear on library shelves. In a digital library, the situation is more complicated. A digital library's servers do not necessary contain the complete contents of each resource. DLESE does not host any of the Web content to which they point, they only store the content to (other people's) content. There are trade-offs to whether holdings should be stored locally or distributed. In practice, there are frequently legal and practical barriers to owning all library content. In DLSI, it will be necessary to host some content (in cases where resources do not have their own home) and link to other content.

Just as the catalog information used to find information is essential in a physical library, the information that describes the data (or metadata) is essential in a digital library. Good metadata is the key to making user tasks effective. For example, if users are only interested in items related to high winds, the search is dramatically simplified if the metadata contains information about high winds. Typical generic metadata include titles, authors, and abstracts. DLESE provides an elaborate metadata framework that are geoscience-specific such as geographic location and time scales of studies. For DLSI, it will be necessary to extend existing metadata based on the needs of the particular audience using the library.

The content problem can be broken down to the problems of identifying useful resources, cataloging the metadata about each resource, and incorporating the content into the digital library. The storage of data is a relatively straightforward problem, but identifying and cataloging can be extremely labor intensive. Once a library has an active user community, it is often possible to utilize members of the community to assist in the cataloging process. When that happens, it will be necessary to provide an interface to allow people to add their own content to the library. As a first step, an elaborate creation interface is unnecessary, as researchers finding content can create the library records directly.

**Work Already Accomplished**

The first task was to explore some of the initial content that would be relevant to the digital library. Richard Katz from NCAR and Roger Pielke, Jr. from the University of Colorado both maintain bibliographies\(^2\) on the value of weather Forecasts. Unfortunately, both of these resources emphasize published papers and scholarly publications. This means that the references tend to be at least several years old, and most are not available electronically. Electronic resources

\(^1\)It is generally not desirable to present overwhelming amounts of data to users, so it is good to have a lot of content for coverage, but only return the subset that is useful.

(presentations, decision support tools, home pages) are not well indexed in current systems. However, the categorization scheme used in these sites will be valuable in creating initial categories for DLSI.

There are several digital libraries in the geosciences that were investigated, more than can be mentioned here. What this suggests is that while the geoscience research community is well served with bibliographies and search tools, the policy makers and users of weather information have significantly fewer resources. There is therefore potential to serve a user community that currently is underrepresented.

DLESE, which is mentioned throughout this document, is not the only digital library infrastructure that I considered. There are several digital library software packages. One of the most similar systems is Morpho and Metacat [http://knb.ecoinformatics.org/informatics/], used in the Knowledge Network for Biocomplexity. KNB is particularly interesting because they reference not only journal articles, but live data sets. Thus, researchers using their digital library get digital resources (software, data sets), not just static journal articles. DLESE seems to be the most suitable infrastructure for this digital library for several reasons, including the considerable in-house expertise, the free availability of the software, and the proven robustness of the digital library.

**Development Milestones**

Fortunately, all three aspects of the architecture can be developed in parallel. Since completing one aspect is generally not a precondition for beginning work on another aspect, each can be incrementally developed. This allows for incremental development of all aspects of the architecture. This development plan calls for small progress on each aspect to be made concurrently, in order to minimize the likelihood that an unforeseen issue in one aspect of the project will cause disruption of the other aspects.

Because user involvement is so important in library development, it is important not to over-commit in an implementation without understanding user requirements. It is often possible to invest considerable time in implementing features that users don't use, and omitting features that users deem critical. Testing with the user community is of course important, but equally important is designing a flexible infrastructure. Since a healthy digital library is one that is constantly changing (contribution of new resources, changing user community, and so on), evolution and incremental development should be considered as an important design goal.

**User Interface**

The design of the user interface includes not only the determination of how the digital library should look. Instead, it subsumes the more important task of determining who makes up the community of potential users and what they need to do. Designing a user interface requires as much interaction with users as is reasonable. This does not mean that all user interaction is time consuming, or that user interaction is needed for all phases. The cognitive walkthrough is a well-established technique for designing user interfaces where designers can quickly test and refine designs without involving users by evaluating the quality of designs against the lists of required tasks.

The important tasks (in order of completion) include:

1. Interviews / discussions with potential users
2. Survey to collect user needs
3. Inventory of core tasks
4. Mockups of User Interface
5. Evaluate mockups using cognitive walkthrough (quick evaluation without user testing)
6. User testing
7. Iteration of the previous steps

---

3. In fact, the entire design of the user interface is based on the task-based analysis user interface design methodology, of which the cognitive walkthrough is just one part.
The primary deliverables include:

1. Paper mockups of the user interface
2. A list of core tasks the library must support
3. Templates for Web pages that will be implemented in the infrastructure

**Infrastructure**

The development of the infrastructure involves the modification of DLESE functionality based on the unique needs of DLSI. The modification requires a knowledge of what changes will be necessary, which will be identified through the user interface testing. Nevertheless, the infrastructure can be developed without intimate knowledge of user needs. Regardless of the precise categories and features that users need, the library contents must be stored in some scheme and stored in some database. Because of the nature of the data storage, new fields (such as new categories) can be added at any time. Furthermore, while the sooner that actual content can be added to the infrastructure the better, the infrastructure can be tested with “mock content” until authentic content becomes available.

The important tasks (in order of completion) include:

1. Setting up the development server (Eric's desktop PC) with necessary software
2. Getting DLESE running on development server
3. Detailed design document
4. Create database schema
5. Modify DLESE search system
6. Basic browsing feature (list of all records)
7. Basic search feature (textual search)
8. Migrate from development server to public test server
9. Advanced browsing (browse by categories)
10. Advanced searching (search by custom metadata)

The primary deliverables include:

1. Design document
2. Basic features using unmodified DLESE interface
3. Basic features using modified DLESE interface
4. Library running on a publicly accessible machine

**Content**

The identification and categorization of content is probably the most labor intensive aspect of the digital library, which probably makes it the libraries biggest exposure. For example, it is probably desirable to create at least one or two sentence summaries for each resource, and only academic papers have formal abstracts from which this can be drawn. For other types of materials, we will have to create our own abstracts. To aid in the categorization process, we will contact the cataloging experts at DLESE and determine if they can be contracted to categorize resources which we provide to them.

The important tasks (in order of completion) include:

1. Identify initial content
   - Articles (from existing bibliographies by R. Katz and R. Pielke)
   - Web sites
   - Talk slides (PowerPoint)
   - UCAR communications
2. First category identification  
3. Give records to DLESE for cataloging  
4. Extract relevant information from content

The primary deliverables include:

1. Around 30 records for Version 0 library testing, cataloged by hand  
2. List of categories and rubric (categorization guidelines) to give to catalogers  
3. Records in DLESE internal format for use in infrastructure

Potential Exposures

This section is an attempt to describe the greatest risks in the design of this system and describe how they can be avoided. It is not an exhaustive list, and of course the greatest dangers are those which are unexpected.

• The user community for this library does not yet exist, nor do the individuals who contribute and use content consider themselves a “community.” Because of this, getting precise requirements for user needs may be difficult. Users do not know that they need a digital library because they do not have a mental model on how they might use one. What this implies is that it is important to give as many examples and scenarios as possible to users to discover what scenarios the users find the most exciting.

• The user community may be extremely small and may not have the time or motivation to contribute resources themselves. We cannot assume that there will be active participation from the user community. “Success” should probably not be measured in number of cataloged entries or number of hits, but instead by an evaluation of how effectively the library synthesizes the resources in a way that has not been done before. Nevertheless, the library should not be over-designed and rely on heavy user involvement.

• There may not be a large amount of content that is targeted directly at the societal impacts and economic value of weather forecasts, especially in the first content area of hurricanes. This presents the danger that, in the pressure to add any content to the library, that DLSI will include resources which are only peripherally and tenuously related to the theme, and hence are not valuable to the users. Even if the library holdings are small, they should be high quality, and a small but detailed collection is likely more useful than a large and minimally useful set.

• Cataloging content is extremely time-consuming because it requires someone to read and comprehend each resource in the library. The DLESE cataloging team may not be available for the task we need in the scale or time frame that meets our requirements. If cataloging turns out to be a problem, it is probably worthwhile to invest in the design of well-defined cataloging rules. Having a few very well cataloged records as well as instructions on how to catalog new records can be useful to delegate the task to future staff who need only follow the cataloging instructions.

• It is possible that DLESE does not provide appropriate infrastructure for this library, perhaps because some of the important user tasks cannot be captured in the DLESE software model. If this is the case, there are several other infrastructures upon which the library can be built, but it will delay some of the deliverables.

Deliverables

Consistent with the iterated design and rapid prototyping described in previous sections, the short-term implementation plan calls for three major library deliverables:

• Version 0 (August 2004)
  • “Canned data”
  • Prototype pages for searching and browsing.
  • Around 30 resources
• Not hooked to DL infrastructure (static Web pages)
• Used to test categories and features

• Version 1 (December 2004)
  • Fully hooked to infrastructure
  • As many records as possible
  • Basic search feature
  • Browsing capability

• Version 2 (Mid 2005)
  • Advanced Browse
  • Advanced Search
  • Interface to contribute / edit resources
  • Incorporation of our records into DLESE Catalog