DOCUMENT RESUME

ED 459 845 IR 058 381

AUTHOR	Marisa, Richard J.
TITLE	HeinOnline: An Online Archive of Law Journals.
SPONS AGENCY	National Science Foundation, Arlington, VA.
PUB DATE	2001-06-00
NOTE	8p.; In: Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (1st, Roanoke, Virginia, June 24-28, 2001). For entire proceedings, see IR 058 348.
CONTRACT	IIS-98174416
AVAILABLE FROM	Association for Computing Machinery, 1515 Broadway, New York NY 10036. Tel: 800-342-6626 (Toll Free); Tel: 212-626-0500; e-mail: acmhelp@acm.org. For full text: http://www1.acm.org/pubs/contents/proceedings/dl/379437/.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Access to Information; Archives; *Computer System Design; Electronic Journals; Information Services; Information Sources; Law Libraries; *Online Systems; Scholarly Journals
IDENTIFIERS	*Legal Reference Materials; *Legal Research

ABSTRACT

Law is grounded in the past, in the decisions and reasoning of generations of lawyers, judges, juries, and professors. Ready access to this history is vital to solid legal research, and yet, until 2000, much of it was buried in vast collections of aging paper journals. HeinOnline is a new online archive of law journals. Development of HeinOnline began in late 1997 through the cooperation of Cornell Information Technologies, William S. Hein & Co., Inc. of Buffalo, New York, and the Cornell Law Library. Built upon the familiar Dienst and new Open Archive Initiative protocols, HeinOnline extends the reliable and well-established management practices of open access archives like NCSTRL and CoRR to a subscription-based collection. The decisions made in creating HeinOnline, Dienst architectural extensions, and issues that have arisen during operation of HeinOnline are described in this paper. The paper discusses Dienst, a framework for implementing digital library systems; the HeinOnline design; creating a working library; production experience; and future plans. (Author/AEF)

HeinOnline: An Online Archive of Law Journals

Richard J. Marisa
Cornell Information Technologies
Cornell University
Ithaca, NY 14853
+1-607-255-7636
rjm2@cornell.edu

ABSTRACT

HeinOnline is a new online archive of law journals. Development of HeinOnline began in late 1997 through the cooperation of Cornell Information Technologies, William S. Hein & Co., Inc. of Buffalo, NY, and the Cornell Law Library.

Built upon the familar Dienst and new Open Archive Initiative protocols, HeinOnline extends the reliable and well-established management practices of open access archives like NCSTRL and CoRR to a subscription-based collection. The decisions made in creating HeinOnline, Dienst architectural extensions, and issues which have arisen during operation of HeinOnline are described.

Categories and Subject Descriptors

H.3.7 **[Information Storage and Retrieval]**: Digital Libraries – *Collection, System issues.*

General Terms

Management, Design, Experimentation.

Keywords

Dienst, digital library, document structure, law journals, metadata, system design.

1. INTRODUCTION

Law is grounded in the past, in the decisions and reasoning of generations of lawyers, judges, juries, professors. Ready access to this history is vital to solid legal research, and yet, until 2000, much of it was buried in vast collections of aging paper journals.

Enter HeinOnline, an ambitious project to make complete runs of US law school journals available on the Web, and then to expand this to include an array of other classic legal materials.

HeinOnline is a collaboration among William S. Hein and Co. Inc., the world's largest distributor of legal periodicals, Cornell Law Library and Cornell Information Technologies. The collection currently contains more than 55 journals, comprising over 1.4 million pages (as of 4/1/01) and is growing at over 150,000 pages a month.

Before HeinOnline, searching these materials was often difficult. Researchers depended on the journals being on the shelf when they needed them, and the only tool for finding articles was paper-based indexes — often constructed with outdated legal terminology and lacking topics of contemporary interest.

Browsing the collections and searching for specific information are both supported, which means researchers don't have to trade the convenience of online access for the ability to "flip" through the pages of journals. Another feature is being able to enter a standard citation and instantly pull up the article.

Cornell's involvement began in 1997 when Hein was investigating ways to put its collections on the Web and Cornell University Library's *Making of America*[1] project caught its attention. *Making of America* is a collection of 19th-century periodical literature which serve as primary sources for American social history. Like HeinOnline, it relied on a digital library protocol called Dienst [1]. Cornell's Office of Information Technology had provided technical support for the *Making of America* project, and became technical lead for HeinOnline.

2. DIENST

Dienst is a framework for implementing digital library systems. The Dienst architecture specifies a set of distributed services which allows access to documents, components of documents and aggregations of documents. Dienst was attractive because it offered a clearly defined set of document *services* which together comprise a capable repository, an open, proven *protocol* [5] to communicate with those services, upon which to build a user interface and application layer, and it encompassed a *document model* which was applicable to law journals.

Dienst was developed as part of the Arpa-funded CS-TR project, which was undertaken to make computer science research available over the Internet and to undertake basic research in digital libraries [2]. It formed the basis of the Networked Computer Science Technical Report Library [4], and is used by

- CoRR, the Computing Research Repository[2],
- the Open Archives Initiative[3],
- ETRDL, the ERCIM Technical Reference Digital Library[4], and

JCDL '01, June 24-28, 2001, Roanoke, Virginia, USA.
Copyright 2001ACM 1-58113-345-6/01/0006...$5.00.

[1] http://cdl.library.cornell.edu/MOA

[2] http://xxx.lanl.gov/archive/cs/intro.html

[3] http://www.openarchives.org

[4] http://www-ncstrl.inria.fr/Dienst/htdocs/

- the Cornell University Library Historical Math Book Collection[5].

2.1 Services

Services defined within Dienst include:

- a *Repository Service* which stores digital documents according to the defined document model, each of which has a unique name and may exist in multiple versions, each with different components and formats,
- an *Index Service* which accepts queries and returns lists of documents identifiers matching those queries,
- a *Collection Service* which provides information on how a set of services interact to form a logical collection., and
- a *User Interface Service*, through which human interaction with the other services and their protocols is mediated.

2.2 Protocol

Dienst protocol requests are expressed as URLs embedded in HTTP requests A typical implementation uses a standard Web server, such as Apache, that is configured to dispatch Dienst URLs to the appropriate Dienst service.

Responses to protocol are formatted as HTTP responses. The content type of the response will vary according to the type of reply. For example:

- `text/xml` is used for responses that contain structured information (such as the protocol request for the internal structure of a digital object), and
- content specific types such as `image/gif` and `application/postscript` are employed for disseminations from digital objects.

2.3 Document Names

Documents in Dienst Repositories are named by assigning them a *handle* [3]. The Handle System is a comprehensive system for assigning, managing, and resolving persistent identifiers for digital objects and other resources on the Internet. Every handle consists of two parts: its naming authority (the "prefix"), and a unique local name under the naming authority (the "suffix"). The naming authority and local name are separated by the ASCII character "/". Naming authorities are defined in a hierarchical fashion resembling a tree structure.

2.4 Document Model

The Dienst document model allows for the storage and dissemination of documents in multiple forms. While documents in some other collections are "born digital" and have primary representations in Postscript or other electronic formats, HeinOnline journal volume documents are stored as sets of high resolution scanned images and as text files derived from those images. Derivative representations of the page images are disseminated at different resolutions and in different image formats for viewing and for printing and as text for search procedures.

The Dienst service responsible for document storage and dissemination is the *Repository*. The Repository service processes commands ("verbs" in Dienst parlance) to deposit documents, discover their structure, and to provide disseminations of documents.

The *Structure* verb allows the user to discover the logical structure of a document, the *Formats* verb returns the list of formats ("content-types") which may be requested from a document, and the *Disseminate* verb allows a client to request a dissemination of the document by specifying a structural component and content-type.

The Dienst Repository service supports other concepts, such as "versions" of a document, which are not employed in the HeinOnline system.

2.5 Document Structure

Dienst maintains structural and descriptive metadata – information about the tables of contents, indexes, chapters, and so on – along with descriptive (cataloging) information about each structural component. The cataloging information is used to facilitate discovery and location of the subcomponents, for example, finding articles by author and/or title.

The only requirement of the descriptive metadata used by Dienst is that it is transported (formatted) in XML. The Dienst Protocol Specification offers examples of descriptive metadata using rfc1807, Dublin Core [7], and OAMS (Open Archive Metadata Set) elements. HeinOnline formats descriptive metadata using Dublin Core elements.

As represented in the Dienst Structure file, a document may contain zero or more views. Each view is an alternative expression or structural representation of the content encapsulated in the digital object. Two examples from the Dienst Protocol Specification illustrate the notion of alternate views:

A digital object representing a musical work may contain three views:

- the audio view of the music
- a textual view of the lyrics
- a video view of a performance of the musical piece.

A digital object representing a scholarly paper may contain two views:

- the complete content (body view) of the paper, including text and tables
- a table view that only provides access to the tables in the paper.

A Dienst view may then be hierarchically structured using nested divisions ("*divs*"). Two examples illustrate the purpose of a div hierarchy:

- A book view may contain a hierarchy indicating sections and chapters
- A scholarly journal view may contain a hierarchy containing issues and articles.

Each div may contain descriptive metadata and may then contain one or more *terminal elements* that are individually disseminable components of the document instance. At present Dienst only supports a single terminal element, *pageimage*, which represents

[5] http://cdl.library.cornell.edu/math.html

an individual page of text. Future versions of the protocol may support other terminal elements such as frames in a movie or samples in a digital audio format.

3. HEINONLINE DESIGN

3.1 Dienst Repository Server

A Dienst Repository server was written in Perl for HeinOnline according to the Dienst Protocol specification. This new implementation, distinct from the Cornell Computer Science/ NCSTRL implementation of Dienst, is a CGI program which works with the Apache web server. The Dienst server is used under MS Windows operating systems (NT, 2K, 98) for internal users; the production server for external users runs under Solaris 2.6 on an UltraSparc 10.

3.2 Document Names

The documents in the HeinOnline law journals collection are assigned handles using the naming authority `hein.journals`. The local name part of the handle specifies a specific document in the collection, i.e., a journal volume. Thus, the handle for Texas Law Review, volume 50, may be `hein.journals/tlr50`.

3.3 Document Structure

The law journal volume view used in HeinOnline consists of a sequence of page images organized into a hierarchy of issues, articles, indexes, cases, and so on. The volume and each level of the hierarchy (the divs) may have an associated metadata record, which is marked up using Dublin Core elements. Listing 1 shows the beginning of the structure file for Cornell Law Review volume 85 (1999), including the Dublin Core record for the volume as a whole. Listing 2 shows a fragment from the middle of the file, including an article and its Dublin Core record and some of its pageimage elements.

Listing 1. Beginning of Structure File

```
<?xml version="1.0" encoding="UTF-8"?>
 <Structure>
  <meta-formats>
   <dc xmlns:dc=
     "http://purl.org/dc/elements/1.0/">
     <dc:Title>
        Cornell Law Review
     </dc:Title>
     <dc:Series>85</dc:Series>
     <dc:Date>1999-2000</dc:Date>
     <dc:Identifier>
        citstring:Cornell L. Rev.
     </dc:Identifier>
   </dc>
  </meta-formats>
  <view type="volume">
    <div id="misc1" desc="titlepage">
      <meta-formats>
         <dc xmlns:dc=
     "http://purl.org/dc/elements/1.0/">
           <dc:Description>
             Title Page
           </dc:Description>
           <dc:Identifier>
         citation:85 Cornell L. Rev. []
           </dc:Identifier>
         </dc>
      </meta-formats>
    <pageimage id="1" native="[]"/>
    <pageimage id="2" native="[]"/>
    </div>
...
</Structure>
```

Listing 2. Fragment of Structure File

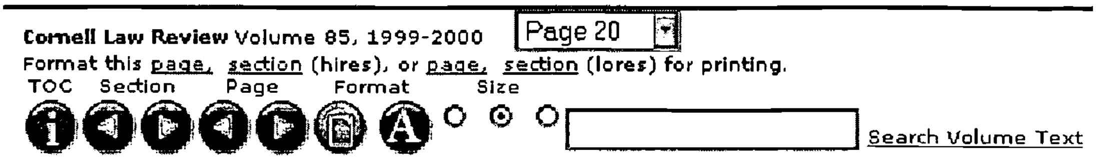
```
...
<div id="misc7" desc="article">
  <meta-formats>
    <dc xmlns:dc=
     "http://purl.org/dc/elements/1.0/">
      <dc:Title>
   Efficiency of Managed Care Patient
   Protection Laws: Incomplete
   Contracts, Bounded Rationality,
   and Market Failure
      </dc:Title>
      <dc:Creator>
         Korobkin, Russell
      </dc:Creator>
      <dc:Identifier>
         citation:85 Cornell L. Rev. 1
                  (1999-2000)
      </dc:Identifier>
    </dc>
  </meta-formats>
  <pageimage id="9"  native="1"/>
  <pageimage id="10" native="2"/>
  <pageimage id="11" native="3"/>
  <pageimage id="12" native="4"/>
  <pageimage id="13" native="5"/>
 ...
</div>
...
```

3.4 User Interface

The HeinOnline user interface was developed at Cornell Information Technologies, with input from Daniel Rosati, Senior Vice President of William S. Hein Co., Claire Germain, professor of law and Edward Cornell Law Librarian, and her colleagues at Cornell Law Library, and law school librarians at several other institutions.

JSTOR, Lexis-Nexis and Westlaw — other popular online research tools — were used as the comparison standards, and second- and third-year Cornell law students in advanced legal research courses tested system prototypes.

HeinOnline displays the exact image of a page. This protects the integrity of the original document and ensures its authenticity. Additionally, since users have access to text derived from the images, they can copy text and paste it into their notes and papers. Both metadata and full text are searchable.



Cornell Law Review Volume 85, 1999-2000

Format this page, section (hires), or page, section (lores) for printing.

Figure 1. HeinOnline Document Browsing Control

3.4.1 *Hunter Routines*

The HeinOnline system employs a suite of Perl and JavaScript routines, collectively known as "Hunter", to navigate the journal volumes as structured Dienst documents. When a volume is opened, a server CGI routine issues a Dienst request for the associated structure file. After analyzing that file, JavaScript objects representing the page data and hierarchic structure of the volume are generated. These are sent with the Hunter JavaScript routines to the client web browser. The client can now access any of the named (numbered) pages of the volume, and browse from section to section (e.g., article to article) via direct Dienst requests to the HeinOnline repository. This design relieves the server of maintaining state information on the document the user is accessing and of re-parsing a representation of the document structure for every client interaction.

Hunter client functions include:

- next page
- previous page
- next section
- previous section
- go to (named) page
- format page for printing
- format section for printing
- select image size
- display page image / display OCR text of page (toggle)

The HeinOnline end-user interface to access these function is shown in Figure 1.

The "go to (named) page" drop-down menu is used to allow the user to browse the page sequence as it is bound in the printed volume without knowing the page number sequence. Because law school journals are often published by students, page numbers are occasionally non-sequential and even repeat within a volume. Navigating a list of actual page labels allows the reader to choose any page in spite of such anomalies.

3.4.2 *Citation Access Widget*

One of the most time-consuming aspects of writing and reviewing legal papers is the checking of numerous citations. A citation-based navigation widget was one of the most requested features in early prototypes of the system. The citation widget allows a user to enter a citation in a standard Bluebook[6] format, and the system opens the journal volume to the requested page.

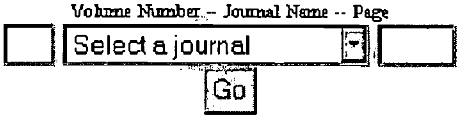


Figure 2. Citation Navigation Widget

Law librarians often receive requests from patrons for copies of journal articles. Satisfying these requests has been labor intensive, involving retrieving the paper volume (perhaps recalling it), copying the article, then generally faxing it to the patron. Libraries using HeinOnline report that they use the citation widget to access the article, format it, save it as a web page, and e-mail the result to the patron, with great labor and copying savings.

The printing formatter complements the citation widget by placing the standard page citation on each printed page. This simplifies record keeping for researchers while they are collecting references.

3.5 Preservation

Digital libraries are increasingly being considered by libraries as a way to ease shelf-space pressures. Now that HeinOnline has put historical law journals online, law libraries have the option of keeping only single copies of those journals on their shelves, putting them in storage or discarding them, depending on their preservation policies.

Hein is a republisher of historical legal materials both in print and microfiche and, as such, has long been interested in preservation issues. Since 1920, William S. Hein & Co., Inc. has specialized in locating rare and out-of-print collections, reprinting government documents and periodicals, converting archive collections to microforms, and preserving legal classics. Hein's holdings include over 75 million pages.

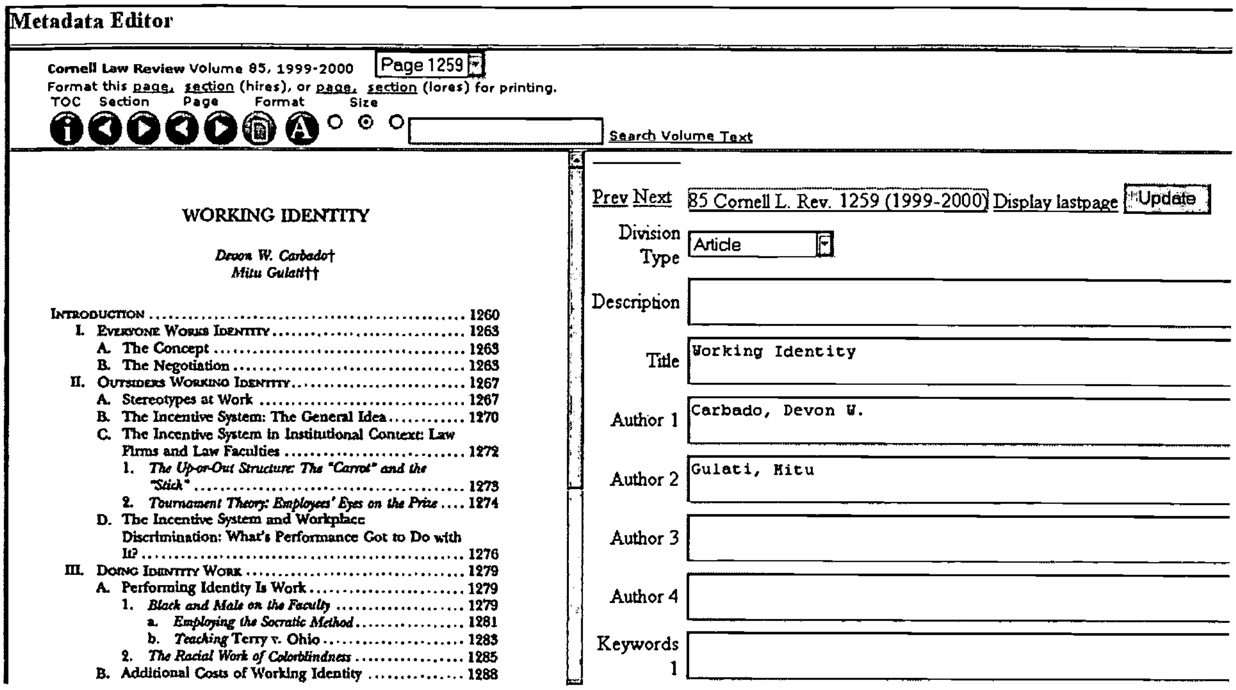
Print and microfiche archives of source materials are maintained in the firm's Buffalo, NY and Littleton, CO facilities. Electronic images of pages, generally bi-tonal 300-dpi, group-4 compressed TIFF format images on optical media are maintained in two separate Buffalo facilities. These images, originally created to enable print reproduction, are the masters for the HeinOnline system.

4. CREATING A WORKING LIBRARY

4.1 Work Flow

Hein scans pages of unbound journal volumes using a 2-sided scanner with a straight-line paper path – a must for old, and often brittle, pages. Page images are collected and organized using the Xerox Digipath[6] scanning system.

[6] http://www.xerox.com



Metadata Editor

Cornell Law Review Volume 85, 1999-2000 Page 1259

Format this page, section (hires), or page, section (lores) for printing.

WORKING IDENTITY

Devon W. Carbado†
Mitu Gulati††

85 Cornell L. Rev. 1259 (1999-2000)

Division Type: Article

Description:

Title: Working Identity

Author 1: Carbado, Devon W.

Author 2: Gulati, Mitu

Author 3:

Author 4:

Keywords 1:

Figure 3. HeinOnline Metadata Editor Screenshot

A Digipath operator records the "native" page numbers (those which appear on the page image) and identifies the beginning and end of each journal section (article, index, editorial, etc.). This information is exported from the Digipath system as a PDF file, which contains the page number and structure information as "bookmark" data.

The PDF file is then processed to extract the bookmark and page image data to non-proprietary formats using ISIToolBox[7] and Ghostscript[8]. A locally developed Perl script converts the bookmark data to an XML-encoded Dienst structure file. The text of the TIFF page image is extracted using an OCR ("optical character recognition") program, ScanSoft TextBridge[9]. A word occurrence index for each journal volume is generated from the OCR text files.

4.1.1 Entering Metadata

Fashioned from the Hunter suite and an additional server-side script, a web-based metadata editor (Figure 3) allows a production operator to enter or update the descriptive Dublin Core metadata for any division of the structure file hierarchy. Additional controls allow the operator to navigate the structural hierarchy and display the associated pages, for example accessing the last page of a section (where in some journal styles, the author's name is printed).

4.1.2 Quality Control

When scanned pages are collected and organized, every page image is viewed to be sure that it is present, readable and not excessively skewed. Similarly, the output of each operation (OCR, image extraction) is inspected to verify that the operation completed successfully.

The digital library is dependent on the quality of its metadata. As descriptive metadata is entered into the metadata editor, an operator checks the structural metadata for consistency by reference to a bound copy of the journal. After metadata is entered, a second operator copyedits the descriptive metadata record for each division in the structure file, using the metadata editor to display images of the original pages.

A style guide was developed for entry of article titles, author names, and other descriptive entries. Division types are assigned by selecting from a controlled vocabulary of types on a pull-down menu.

4.1.3 Placing Content into Production

The TIFF images, OCR text, structure file and index comprise the primary data served by Dienst. These are archived at Hein locations on optical media and copies are shipped on CDs to Cornell University, which houses the external server. The CDs are copied onto spinning magnetic media and integrated into the collection using automated scripts to build searchable metadata and full text databases, as described below.

4.2 Derivative Data

Page images, such as multi-tonal PNG images formatted for display by web browsers, are generated on the fly from the bi-tonal TIFFs by the Dienst Repository server. Derivative image

[7] http://www.imagesolutions.com/isi_software.htm

[8] http://www.cs.wisc.edu/~ghost/

[9] http://www.scansoft.com/products/tbpmill/

generation is done at user-selectable resolutions to accommodate various client configurations and to aid the visually impaired.

While OCR text data could, in principle, be generated on the fly, the latency time for generation and the need to access many pages in single search operations led us to store a text file of each journal volume page.

4.3 Subscription Management

The initial set of subscribers to HeinOnline are primarily law libraries in universities or state and federal government offices. HeinOnline is made available to their patrons within local facilities or across campuses or other facilities.

HeinOnline downloads a JavaScript application to client web-browsers to enable browsing of journal pages. The application makes direct requests to the Dienst Repository for content, therefore subscription enforcement must be integrated directly into the Dienst Repository server.

The initial implementation of subscription enforcement in HeinOnline relies on IP address restrictions. The Dienst Repository service checks a database of allowed IP address ranges which correspond to subscriber campuses or facilities. The database is administered by Hein, which does all subscription servicing. This strategy works well for these customers which are often assigned blocks of static IP addresses. Customers with dynamically assigned addresses, but which use an IP proxy server, can also be serviced with an IP address enforcement scheme because the proxy has a fixed IP address.

However, enforcement of subscriptions by IP address does not work for individuals who are not affiliated with subscribing institutions or who do not have fixed addresses (e.g., dial in or cable modem customers), so an alternate method for authentication and authorization is necessary. For this reason, a cookie-based session management system was built into the subscription enforcement mechanism. This system is currently only used to facilitate guest access to the system, and will be expanded to general subscriber use as system usage grows.

4.4 Full Text Searching

4.4.1 Uncorrected OCR

The text generated from the page images usually contains recognition errors, which result in misspelled or unrecognizable words. These errors are more frequent in pages with ornamented or antique fonts, or in volumes produced with less precise printing technologies. Broken letters characteristic of early printing processes introduce some systematic errors.

For financial reasons, the generated text used in HeinOnline is not corrected or edited. Experience has shown that OCR text errors have minimal effect on search function ability to find relevant articles, thanks to the redundancy of word usage in English prose.

4.4.2 Index Files

Two levels of index files are created to facilitate full-text searching. For each volume, the set of derivative text pages is read to determine a list of unique words and the list of page images on which each word occurs. This list is alphabetically sorted and written as a text file (a volume index), one word (and list of page images) per line.

A second index is created, this time by using the indexes previously generated to create a list of all words which occur in the entire collection. The list of volumes in which each word occurs is recorded, and the alphabetized list of words and volumes is written as another text file (a collection index).

When a user enters a set of search terms, the collection index is consulted to determine which volumes contain the referenced terms. Then the volume indexes for those volumes are consulted to determine if the user's terms occur together (or in a user specified combination) on the same page. The list of selected volumes is then displayed to the user, with an indication of how many pages in each volume may be of interest.

When the user selects one of the relevant volumes, the derivative text files for the specific pages containing the search terms are accessed, and the lines of text containing the terms in the user's search request are displayed in the user's browser, along with links which will take the user to the specified page.

This simple structure has several advantages. It performs well with a large number of pages. It is easy and quick to update the indexes as the collection grows, although care must be taken to use efficient algorithms when processing a million-plus pages of text. It is easy to limit searching to specific titles or volumes.

4.4.3 Metadata Harvesting

To enable searching by article title and author, the descriptive metadata in the Dienst structure files is harvested into an SQL database table. This is done automatically by issuing a Dienst Repository List-Contents command to determine the set of documents currently in the repository, and then sequentially requesting the structure file for each volume. Each structure file is parsed and analyzed and the SQL table is updated appropriately.

4.4.4 Open Archives Initiative Server

This same metadata harvesting technique may be used to update a database which supports an OAI (Open Archives Initiative) server.

The OAI protocol[10] is an application-independent interoperability framework for *metadata harvesting*. *Data Providers* like HeinOnline use the OAI protocol as a means of exposing metadata which describes their content. *Service Providers* may issue OAI protocol requests to data providers for the metadata (the "harvesting") and use the metadata as a basis for building value-added services.

HeinOnline participated in the alpha-test of the OAI protocol in 2000-2001.[11] Volume level and/or article level metadata from HeinOnline may be served via OAI, per the subscription policy.

5. PRODUCTION EXPERIENCE

Working two shifts, six days per week, HeinOnline has been in full production since mid-2000. Since coming online in July, 2000, the number of subscribing institutions has grown to over 125.

[10] http://www.openarchives.org/OAI/openarchivesprotocol.htm

[11] http://www.openarchives.org/OAISC/alpha-testing-press-release.htm

The HeinOnline collection has grown to over 1,600 volumes, encompassing over 40,000 articles. Titles include *Cornell Law Review, Texas Law Review, Harvard Journal of Law and Technology, University of Pennsylvania Law Review, Tulane Law Review*, among many others. Some titles include full runs from inception to the current volume; for other titles only the earliest volumes have been processed.

6. FUTURE PLANS

6.1 Collections

Following the initial focus on law journals, plans are underway to add additional collections to HeinOnline including:

- *International Documents* which will include items such as the Nuremburg Trials and Classics of International Law,
- *Case Law* which will include exact reproductions of the first one hundred volumes of U.S. Reports, and
- *Legal Classics* which will initially include collections such as Blackstone's Commentaries and Elliott's Debates.

6.2 Repositories

The Dienst architecture is designed to work seamlessly with collections of materials which are distributed across the Internet.

To accommodate the growing body of material, the HeinOnline system will be split across multiple Dienst servers in 2001. In addition, we have plans to duplicate the collection in multiple locations to insure reliability and performance. To implement and manage this configuration, we expect to use the Dienst Collection service.

The Collection service maintains a registry of Repository servers (as well as other data), and allows clients to discover which servers are operating and which hold documents under the various naming authorities.

6.3 Full Text Searching

The full text search capability in HeinOnline is basic, but performs its main objective well, that is, to locate the set of articles relevant to the users' search criteria.

We have experimented with fuzzy text matching which overcomes some OCR recognition errors at the expense of additional false positives. The fuzzy match technology also assists matching over singular and plural terms and over words with the same word-stem.

We anticipate adding more functionality to the full text search modules as the archive grows.

7. ACKNOWLEDGEMENTS

Thanks to Steve Worona for launching and shepherding the formation of the HeinOnline project at Cornell. Thanks to Carl Lagoze for many discussions on the Dienst protocol. Thanks to Beth Goelzer Lyons for contributions to the Introduction.

The HeinOnline project was supported by the William S. Hein & Co., Inc. and Cornell Information Technologies. The Cornell Law Library staff contributed invaluable suggestions and insight. Support for the Dienst protocol work came from the National Science Foundation through grant IIS-98174416.

8. REFERENCES

[1] Cornell Digital Library Research Group. Dienst Overview and Introduction. http://www.cs.cornell.edu/cdlrg/ - dienst/DienstOverview.htm

[2] Corporation for National Research Initiatives. CSTR Computer Science Technical Reports. http://www.cnri.reston.va.us/home/cstr.html

[3] Corporation for National Research Initiatives. Handle System. http://www.handle.net

[4] Davis, James R., Lagoze, Carl. The Networked Computer Science Technical Report Library. http://cs-tr.cs.cornell.edu/Dienst/UI/1.0/Display/ - ncstrl.cornell/TR96-1595

[5] Davis, J., et. al. Dienst Protocol Specification. http://www.cs.cornell.edu/cdlrg/dienst/protocols/ - DienstProtocol.htm

[6] Harvard Law Review Association. The Bluebook: A Uniform System of Citation.

[7] OCLC. Dublin Core Metadata Initiative. http://purl.org/DC