Automated Metadata Creation: Possibilities and Pitfalls

WILHELMINA RANDTKE

Presenter

Automated indexing – using a computer to look at individual documents and assign metadata without a person looking at every document– was used to build an interactive online database to store and retrieve pages from a looseleaf resource (i.e., a resource which changes state over time). A database was designed and more than 30,000 pages in the database were indexed. Digitization, optical character recognition, and computer scripting to extract metadata were the methods used to assign most metadata. In places where the computer program could not assign metadata, a person looking at the document assigned metadata. The index was audited for errors. The computer script and the human indexer had comparable error rates but computer indexing spent much less time per value assigned. It is recommended that automated indexing be considered in indexing projects, especially where a large number of similar documents are to be indexed.

KEYWORDS looseleaf, relational database, automated indexing, artificial intelligence, metadata creation, digital libraries

INTRODUCTION: THE HISTORICAL FLORIDA ADMINISTRATIVE CODE

The Florida Administrative Code (FAC) is the official publication containing all agency rules for the State of Florida. Rules are interpretations of law, which are made by an agency as part of a

public notice-and-comment process, and are binding like laws. From 1963 to 1970, the FAC was published as a looseleaf resource with monthly supplements. In 1970, all rules were renumbered, and the entire run of pages was released anew. From 1970 to 1983, the format continued as a looseleaf with monthly supplements which included instructions to remove, replace, and add pages. As pages were removed from the binder, indicating a change in the law stated on that page, the government of Florida did not retain the old pages. The Florida State University (FSU) College of Law and the University of Miami School of Law were the only two publicly accessible archives to retain old pages.

Old rules are binding on actions which took place in the past. If a person does something, and subsequently the law changes, that change in law does not affect the legality of what the person did before the change in law. If an old crime or regulatory violation is newly discovered, the old version of the law matters. For example, where birth defects due to pollution occur today from pollution released in the 1970s, then the allowable limit of pollution approved by a regulatory agency in the 1970s still matters. Often, an old court case or document, which is still binding today, will cite an old version of a rule, but not provide a quote. In order to understand and interpret the citation, one must have a copy of the old rule.

About six interlibrary loan requests per year were made for the 1970 to 1983 FAC to the FSU law library, along with the required search and photocopying fee. Searching in print took significant staff time, because a librarian familiar with the document set had to start at the present version of the rule, look at the history note, and work back to locate the version in effect on the desired date. FSU's old pages were filed by date of removal, so each step back took a search through all pages removed near the revision date. While the historic FAC is in low use,

persons requesting the pages often placed high economic value in accurate copies of old versions of rules.

DIGITIZATION OPPORTUNITY

The opportunity to digitize and make an online searchable version of the FAC came when Florida's First District Court of Appeal (DCA) moved to a new location, and provided FSU with its set of historic FAC pages. The FSU College of Law had bound its pages chronologically by date of removal. The DCA stored its old pages in numerical order by legal citation and page number in ring binders. Material in binders was easy to digitize by feeding through a sheet fed scanner. In contrast, digitization of bound material requires a special mount for book scanning and either labor or expensive apparatus for turning each page. For the first time ever, cost effective on-site digitization of the FAC was possible.

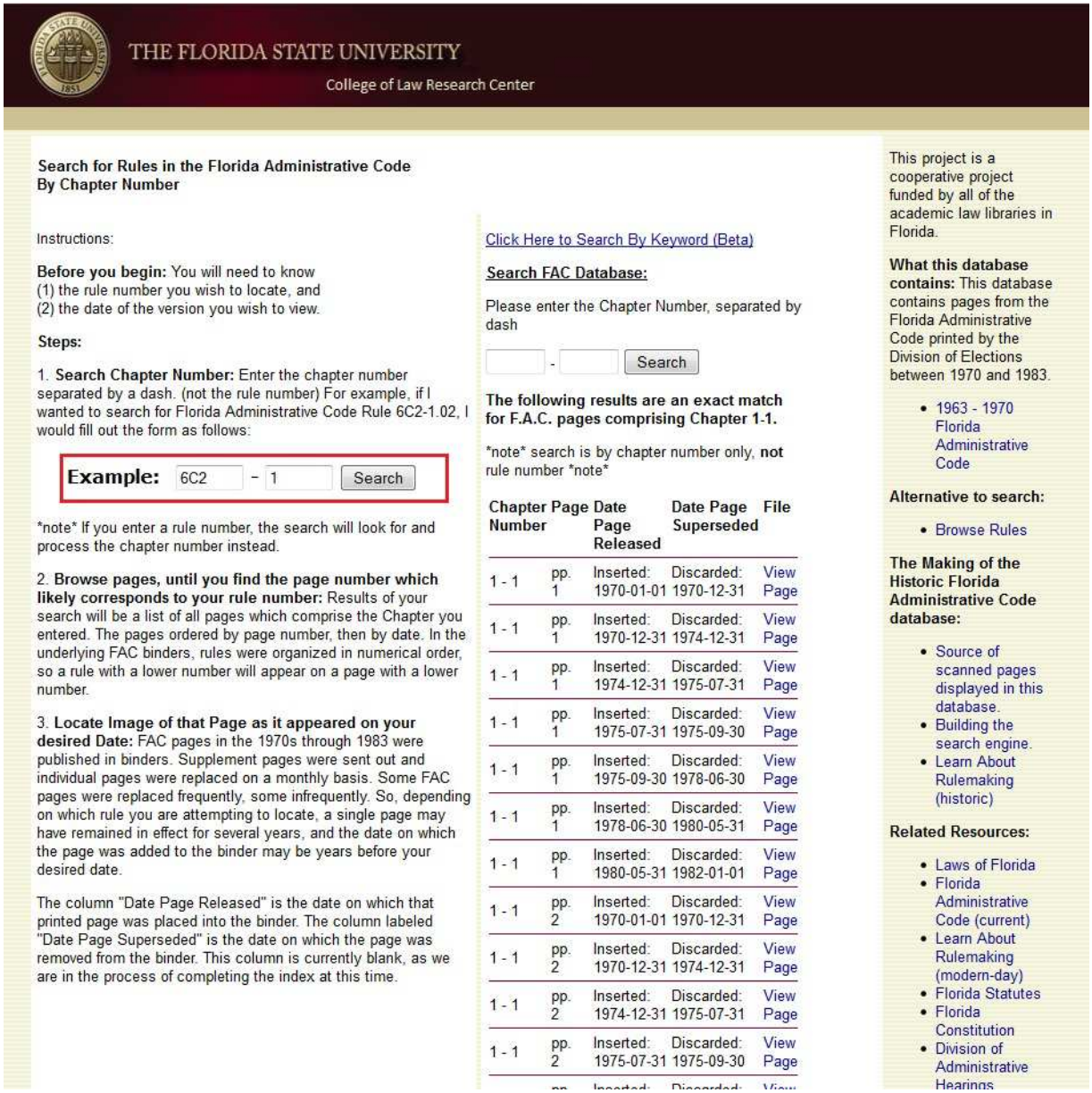
The DCA had also retained all the instruction sheets which went out with FAC supplement packets. This allowed the set of pages to be audited for completeness, and allowed supplement numbers printed on pages to be matched to a date of release.

OVERVIEW OF THE FINAL SEARCHABLE DATABASE

The final searchable database of the FAC from 1970 to 1983 is available at www.fsulawrc.com. The search assumes that a searcher will search using a legal citation to a rule, because this is what interlibrary loan requests for the material typically provided. Figure 1 shows a screenshot of search results in this database. Results are ordered by chapter then page number. Different versions of the same page are displayed next to one another, with a date range listing when each version of the page was in effect.

A compromise is that automatic retrieval by rule number is not possible. Instead, the searcher must retrieve by chapter number, then open some pages to determine what page their

target rule is on, and then retrieve that page on the desired date. Pages were not indexed by rule number, because most pages include several different rules. Rule number was also a field which could not be detected by computer programming, and so would have had to be entered manually. Indexing this field was not possible with the allotted budget and timeframe. A weakness in the project, which became apparent due to our success in lowering barriers to access and opening the resource to new users, is the lack of keyword search capability.



THE FLORIDA STATE UNIVERSITY
College of Law Research Center

Search for Rules in the Florida Administrative Code By Chapter Number

Instructions:

Before you begin: You will need to know
(1) the rule number you wish to locate, and
(2) the date of the version you wish to view.

Steps:

1. **Search Chapter Number:** Enter the chapter number separated by a dash. (not the rule number) For example, if I wanted to search for Florida Administrative Code Rule 6C2-1.02, I would fill out the form as follows:

Example: 6C2 - 1 Search

note If you enter a rule number, the search will look for and process the chapter number instead.

2. **Browse pages, until you find the page number which likely corresponds to your rule number:** Results of your search will be a list of all pages which comprise the Chapter you entered. The pages ordered by page number, then by date. In the underlying FAC binders, rules were organized in numerical order, so a rule with a lower number will appear on a page with a lower number.

3. **Locate Image of that Page as it appeared on your desired Date:** FAC pages in the 1970s through 1983 were published in binders. Supplement pages were sent out and individual pages were replaced on a monthly basis. Some FAC pages were replaced frequently, some infrequently. So, depending on which rule you are attempting to locate, a single page may have remained in effect for several years, and the date on which the page was added to the binder may be years before your desired date.

The column "Date Page Released" is the date on which that printed page was placed into the binder. The column labeled "Date Page Superseded" is the date on which the page was removed from the binder. This column is currently blank, as we are in the process of completing the index at this time.

Click Here to Search By Keyword (Beta)

Search FAC Database:

Please enter the Chapter Number, separated by dash

- Search

The following results are an exact match for F.A.C. pages comprising Chapter 1-1.

note search is by chapter number only, **not** rule number *note*

Chapter Number	Page	Date Page Released	Date Page Superseded	File
1 - 1	pp. 1	Inserted: 1970-01-01	Discarded: 1970-12-31	View Page
1 - 1	pp. 1	Inserted: 1970-12-31	Discarded: 1974-12-31	View Page
1 - 1	pp. 1	Inserted: 1974-12-31	Discarded: 1975-07-31	View Page
1 - 1	pp. 1	Inserted: 1975-07-31	Discarded: 1975-09-30	View Page
1 - 1	pp. 1	Inserted: 1975-09-30	Discarded: 1978-06-30	View Page
1 - 1	pp. 1	Inserted: 1978-06-30	Discarded: 1980-05-31	View Page
1 - 1	pp. 1	Inserted: 1980-05-31	Discarded: 1982-01-01	View Page
1 - 1	pp. 2	Inserted: 1970-01-01	Discarded: 1970-12-31	View Page
1 - 1	pp. 2	Inserted: 1970-12-31	Discarded: 1974-12-31	View Page
1 - 1	pp. 2	Inserted: 1974-12-31	Discarded: 1975-07-31	View Page
1 - 1	pp. 2	Inserted: 1975-07-31	Discarded: 1975-09-30	View Page

This project is a cooperative project funded by all of the academic law libraries in Florida.

What this database contains: This database contains pages from the Florida Administrative Code printed by the Division of Elections between 1970 and 1983.

- 1963 - 1970 Florida Administrative Code

Alternative to search:

- Browse Rules

The Making of the Historic Florida Administrative Code database:

- Source of scanned pages displayed in this database.
- Building the search engine.
- Learn About Rulemaking (historic)

Related Resources:

- Laws of Florida
- Florida Administrative Code (current)
- Learn About Rulemaking (modern-day)
- Florida Statutes
- Florida Constitution
- Division of Administrative Hearings

Figure 1. Screenshot of search results displayed in the final searchable FAC database. Effective dates for each page allow a searcher to retrieve the resource as it appeared on any date over a fourteen year period.

DESIGNING A DATABASE FOR STORING A LOOSELEAF RESOURCE

All digital library platforms currently available are designed for holding individual self-contained digital objects. *CONTENTdm, Digital Commons, dSpace, Omeka, Greenstone,* and other platforms, treat each digital object as a unique item. This is appropriate for storing discrete articles, reports, books, or photographs, because those objects are created once and then are finalized. Even a journal issue is created once, then filed, never to change. A looseleaf resource, in contrast, changes state over time. The FAC in May of 1973 has a different set of pages than the FAC in June of 1973, than the FAC in February of 1980, and so on. A page placed into the binder in 1970 might stay in the binder for only one month, or might stay in the binder for the entire fourteen year run.

Even electronic databases which store the *Code of Federal Regulations* (CFR), a similar resource which changes over time, do not store each version of each page of the CFR. The two dominant legal research databases, *LexisNexis* and *WestLaw*, make available the current CFR as it looks today, then archive a version of the CFR annually. Versioning is incomplete, and additional documentation, a chronological listing of changes to the CFR, must be checked in order to determine what a particular portion of code looked like on a specific date. Other projects, like the CFR archive provided by Cornell's Legal Information Institute, do not support versioning at all.

A database had to be designed which had the ability to track and retrieve different versions of a page and provide the page as it appeared on a specific date. To do this, each page

was examined for unique fields. Figure 2 shows a sample page of the FAC, with unique fields indicated. The FAC was organized by discrete chapter numbers. Pages were numbered with the first page of each chapter assigned the value “1”. Chapter number and page number can be used to browse and retrieve a rule. To track changes over time, two additional values are needed: the date the page went into the binder, and the date the page was removed. Pages were labeled with a supplement number which corresponds to a date the page went into the binder. As clerks at the DCA removed pages, they noted the supplement number replacing that page. This corresponds to the date of removal.

by a lower case "x" following the agency control number and the division letter, if applicable. (Examples: 1x-1.04; 1Ax-1.04).

(8) Emergency rules adopted pursuant to Subsection 120.54(8), F.S., shall be submitted in the same form prescribed by Subsection 1-1.02(3), herein, for other rules, except for the rule number which shall be indicated by the letters "ER" and the last two digits of the calendar year when the rule is first effective. Such indicators shall follow the agency control number, and division letter if applicable. Further, emergency rules shall be numbered consecutively each year beginning January 1, 1975, and such numbers used shall be in lieu of the chapter and section numbers required for rules normally promulgated. (Examples: 1ER75-1; 1AER75-2). Such rules will be summarized in the FAW but will not be printed in the FAC. Emergency rules are accepted by the Department of State without the Department making a determination as to whether or not an immediate danger to the public health, safety or welfare, to which the rule is applicable, does in fact exist.

Specific Authority 120.55(1)(d) FS. Law Implemented 120.54(10)(b), 120.55(1)(d) FS. History—Revised 1-1-75. Amended 8-1-75.

1-1.021 Florida Administrative Weekly.

(1) The Florida Administrative Weekly (FAW) shall be published each Friday. All material to be included in each issue of the FAW must be received by the Department of State no later than 12:00 p.m. Wednesday of the week in which the notice is to appear.

(2) The FAW shall contain the information required by Paragraph 120.55(1)(c), Florida Statutes.

(3) Forms FAC 3 through 6, to be published in the FAW, shall be typed on letter size paper (8½x11), double spaced with a one-inch margin on the left and a half-inch margin on the right. The information given thereon shall be typed on a pica typewriter. The amount of space provided in the form under each heading shall be the maximum space used for providing the information required therein, except in Form FAC 5. The lines used in Forms FAC 3, 4 and 6 are to designate the amount of space that may be used under each heading. Do not underline copy when Forms are submitted for publication.

(4) Form FAC 3 shall be submitted in the following form for the filing of proposed rules, amendments and repeals:

DEPARTMENT OF _____: Rule No.: _____
Rule Title: _____
PURPOSE AND EFFECT: _____
_____
_____
_____
_____
SUMMARY: _____
_____
_____
_____
_____
SPECIFIC LEGAL AUTHORITY UNDER WHICH THE ADOPTION IS AUTHORIZED AND THE LAW BEING IMPLEMENTED, INTERPRETED OR MADE SPECIFIC:
_____
_____
ESTIMATE OF ECONOMIC IMPACT ON ALL AFFECTED PERSONS. $_____. IF NOT POSSIBLE TO DETERMINE, THE REASONS WHY THE COSTS OF THE PROPOSED RULE CANNOT BE ESTIMATED: _____
_____
_____
_____
_____
IF REQUESTED A HEARING WILL BE HELD AT:
TIME: _____
PLACE: _____
DATE: _____
A COPY OF THE PROPOSED RULE MAY BE OBTAINED BY WRITING TO: _____
_____
_____
_____

(5) Form FAC 4 shall be submitted in the following form for filing notices of public meetings, hearings, or workshops:

The (name of agency) announces a (public meeting, hearing, or workshop) to which all persons are invited.
DATE AND TIME: _____
PLACE: _____
PURPOSE: _____
_____
_____

A copy of the agenda may be obtained by writing to (name of agency) at (agency headquarters).

(6) Form FAC 5 shall contain sufficient information to advise substantially affected persons of the nature of the proceeding. Form FAC 5 shall be submitted in the following form for Declaratory statement:

NOTICE IS HEREBY GIVEN that the _____ (agency) received the following petition(s) for Declaratory Statements:
(Name of Petitioner): _____
_____
_____

(7) Form FAC 6 shall be submitted in the following form for filing of Emergency Rules:
DEPARTMENT OF _____ RULE NO. _____
SPECIFIC REASONS FOR FINDING AN IMMEDIATE DANGER TO PUBLIC HEALTH, SAFETY AND WELFARE: _____
_____
_____
_____
_____
_____
_____
REASONS WHY PROCEDURE USED IS FAIR UNDER THE CIRCUMSTANCES: _____
_____
_____
_____
_____
_____

Figure 2. A sample page of the FAC. Recorded fields are double circled in black. The following fields uniquely identify each page: Chapter Number (located in the top right of

odd pages, or top left of even pages), Supplement number (corresponding with the date that the page went into the binders; located in the top right of even pages, or top left of odd pages), and Page number (located at the bottom center of each page). Additionally, some pages were hand labeled "Superseded by" which corresponds to the date the page was removed from the binder. This value was used to assign a date of removal.

The relational database schema to track this resource over time is provided below in Figure 3. For persons wishing to build a digital library for looseleaf resources, this schema may be provided to technology staff in order to assist them in understanding the project.

Page (**uniquePage**, suppNo, chNoBeforeDash, chNoAfterDash, pNo, supercededBy, imageName)
Primary Key: uniquePage
Foreign Key: suppNo references Supplement.suppNo

Supplement (**suppNo**, suppReleased, notesAboutSupp)
Primary Key: suppNo

Explanations:
chNoBeforeDash and chNoAfterDash together make up the chapter number.
pNo is the page number.
suppNo is the supplement number, so corresponds to the date the page went into the binder.
supercededBy is the superceding supplement number, so corresponds to the date the page was removed from the binder.
imageName is the file name or URL to retrieve the full text.

Figure 3. Relational database schema for storing a looseleaf resource (ie. a resource which changes state over time).

AUTOMATED METADATA CREATION

In order to populate the database, the following values had to be filled for each page:

- Chapter number (each chapter number had a dash in the middle; the number before and after the dash were recorded separately)
- Supplement number for the date on which the page went into the binder
- Supplement number for the date on which the page was removed from the binder

- Page number
- File name
- Unique identifier for each document

Digitization produced more than 32,000 unique pages to index. With time and funding constraints, it was impossible to manually index pages. Therefore, computer scripting was used to create a large portion of the metadata for the final database.

BACKGROUND: COMPUTER GENERATED METADATA

In order to understand computer generated metadata, and to understand what projects it might be suitable for, it is important to understand a little about how computers assign metadata. The first point is that, computers are very good at making black and white decisions, but poor at making decisions where there is some gray area. So, the more deterministic something is – the more there is a concrete right answer – the easier it is for a computer to make the decision. Likewise, if there is not a single right answer, but perhaps a continuum, this is difficult for a computer to decide. Second, whatever a computer can do, it can do much faster than a person. Third, we may think of some massive indexing projects as having been done by computer, but in reality human indexing may have been used.

Deterministic decisions

Computers are good at making black and white decisions. It is easy for a computer to know whether a word has a "B" in it, but difficult for the computer to know if two words are synonyms. To solve the synonym problem, a large vocabulary must be broken down into rules, including a rule stating that two words mean the same thing. The computer makes value judgments by applying complex, nested rules.

Fast decisions

While computers are not so good at making value judgments, they are much faster than people. Keyword search is easy for computers to build, and has become dominant because eventually any index to a huge document set becomes more useful than a high quality but partial index. By 2008, Google had indexed one trillion (1,000,000,000,000) pages[1] and did so in about a decade. Human indexing can catch fields the computer cannot, but human indexing will never reach that high volume.

Human indexing in action

Human indexing is used in modern search engines. Shopping sites like Amazon and eBay allow search results for clothing to be faceted by item color, and many other fields. This metadata is generally created by humans. For instance, on eBay a seller assigns metadata even to a low cost item, because this increases the chance the item will be discovered and purchased. In a scholarly example, Google repurposed MARC records and publisher metadata to build the book search. Sometimes, employees manually correct errors. Even Google, an expert at computer generated search, uses human created metadata for some massive search projects.[2]

THE HIGH TECH END OF COMPUTER INDEXING: SUBJECT MATTER INDEXING

At the high end of computer generated metadata is subject matter. Some technologies for performing subject matter indexing purely by computer are Unstructured Information Management Architecture (UIMA), General Architecture for Text Engineering (GATE), and Keyphrase Extraction Algorithm (KEA). Implementing these went beyond the scope of this project, however, all work in the same general way, and a discussion is in order to assist in understanding what is possible on the high tech end of automated indexing. All three listed technologies are artificial intelligence programs which allow rules to be applied to large sets of text. Each can assign subject headings in isolation, but the quality will be poor. In order to get

good quality subject headings, the artificial intelligence program must tie in with a subject specific ontology or thesaurus selected based on the overall thrust of the collection to be indexed.

The artificial intelligence program gives general language processing rules. The ontology or thesaurus will give rules specific to the subject area. An ontology or thesaurus includes rules like "this word is a synonym to this word, they mean the same thing", "this word occurs only a few times on the page, but it's a term of art in the field, so it matters more", and "these words that are next to one another are actually a phrase and go together toward this meaning." The ontology or thesaurus may be proprietary and require a fee to use. Figure 4 shows this process.

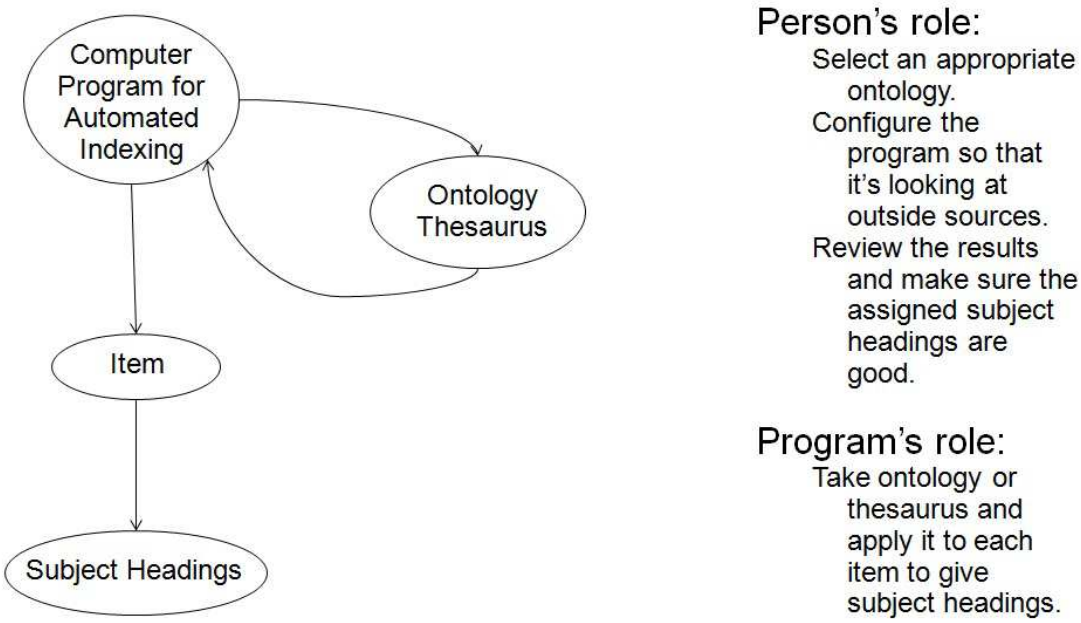


Figure 4. Diagram showing how an artificial intelligence program indexes material. A person looks at the set of documents to index, and selects an ontology or thesaurus which specifies language rules for those documents. The artificial intelligence program connects to the ontology or thesaurus for rules specific to that set of documents, then looks at each item and applies those rules to assign subject headings. Then the person spot checks a

handful of individual items to make sure the indexing program worked in a meaningful way and gave the desired results.

There are several important points about using a computer for subject matter indexing. They are: (1) Traditional indexing is a core skill. Before the artificial intelligence program runs, a person looks at the entire collection and selects which ontology or thesaurus is a good fit for the collection. In traditional cataloging, a cataloger assigns subject headings to one item at a time. Here, a cataloger assigns an ontology or thesaurus to the collection. After the program runs, a cataloger reviews some results to ensure that the program ran as anticipated. If MARC subject headings are assigned, then someone who knows MARC must perform this task. (2) Technology skills come in when the person configures the program to connect with the ontology or thesaurus, and then to run on a set of items. (3) If a librarian and a programmer work together, then each must understand the role he or she plays. The librarians must understand the collection to be indexed and metadata quality. The programmer must understand nuts and bolts of making programs work together. As they trade off, the librarian must do the portions of work which require subject heading expertise: selecting the correct ontology or thesaurus, quality control, and troubleshooting if the assigned subject headings are off. Skill and training in assigning subject headings are central to automated subject matter indexing. Catalogers have a valuable background and their collaboration with programmers allows both to do more than either could do alone.

THE LOW END OF COMPUTER INDEXING: ROTE COPYING FROM THE PAGE

At the low end of computer generated metadata is rote copying of words on a page. Because of the focus on subject matter versus keyword search, we often think of cataloging as oriented around subject matter indexing. Really, a cataloger assigns many fields, like title, author, and

date of publication, which are copied rote off the item. Subject matter indexing takes more training and skill for people, but rote copying fills more total fields in a record.

With the popularity of institutional repositories, many universities are digitizing old theses and dissertations. To make a Dublin Core record for each dissertation, fields are copied: title, author, date of issue, etc. Because dissertations are published with a standard layout, each of those fields appears in a similar location. One is tempted to find a way to push a button and be done with most of the indexing. In fact, some universities have done this. For example, the "thesisbot" on Github.com will extract basic Dublin Core records from theses. See https://github.com/ao5357/thesisbot . Of course, thesisbot will have to be tweaked to fit the layout and format of each university's theses.

The FAC presented a similar problem. Refer back to Figure 2 to see that fields to be indexed (chapter number, page number, supplement number) are all in predictable locations on the page, and all values will be copied from the page. Indexing does not require skill and training for a person or artificial intelligence from the computer. It is much more straightforward.

Both the thesisbot and the script used to index the FAC worked the same way. Each looked at the image, not as an image, but instead as a long string of text. Each looked for patterns in the text, and used the patterns to extract metadata. If you plan a computer program, good patterns to look for should be unambiguous with very clear rules. For example, a computer can "Find the letter 'a' or 'A', then go to the first space to the right of 'a' or 'A', then copy the characters to right of that space until you come to a comma or space." A computer cannot "Find the author's first name."

FINDING ADVISORS ON COMPUTER PROGRAMMING PROJECTS

In planning the FAC project, a professional colleague from law school who now works in network security was consulted. He had a broad technology skill set and also the ability to assess skill sets. He provided assistance in assessing the librarian's skill set, suggesting programming tools with a learning curve which fit the time frame, and gave a sanity check on what quality of results to expect. Without his assistance, this project would not have been possible. His assistance was also unpaid and provided as a professional courtesy. This was done in a single forty-five minute conversation after the librarian had inspected the paper documents, and then two short follow-up conversations later in the project after scripting and automated metadata creation had begun. If you understand your document set, and the other person has a broad and appropriate set of information technology (IT) skills, then forty-five minutes will be enough to get pointers.

Avoid significant misconceptions which routinely hamper technology projects. For example, IT staff at the FSU law school are hired to make video recordings of classes for professors. They do not have skills in networking, database design, or computer programming. These are not failings on their part. It is inappropriate to ask a video equipment operator to assist in computer programming. In a college campus or in an organization, the most visible IT personnel are people who physically install computers, or who perform event set up and recording. They are visible, but may have no need for skills relevant to library projects. Inappropriate technology requests clearly communicate that the librarian does not understand or care to learn what an IT staff member does. When looking for technology advisors, step back, think what skills you do not have but need advice on, and then find someone with those skills. Try to educate yourself about the different types of IT: graphic design, audio editing, networking, programming, etc. Choose an advisor with skills in the area you need.

Another common mistake is taking the wrong people seriously. If you ask someone an IT question and the person does not know the answer, then a best case scenario is that they will say, “I don’t know.” A more likely outcome is that the person will posture, make up an answer, and send you down a blind alley. An IT staff who says "I don't know" should be consulted on future projects, because asking the question is fast, and that person will not waste your time.

Finally, when collaborating with IT staff, the librarian is the expert on the items to be indexed: the nature of the items, how people are likely to use them, and what the desired final product is. The more clearly you understand your goals, the better the technology advisor can assist in planning how to get there, which goals are doable, and which goals are out of reach.

AUTOMATED INDEXING FOR THE FAC STEP-BY-STEP

In order to extract metadata from the FAC pages, each page was stored in PDF, optical character recognition (OCR) was run on the set of pages, the text was pulled to Excel spread sheets and manipulated with Visual Basic to extract metadata.

First, the set of documents was examined. Supplement number was always numeric. All other fields to be extracted were alpha numeric, but all were located in predictable places on the page. No pages had watermarks. Some pages had maps and illustrations. A scan setting of 300 dpi black and white was chosen, based primarily on the perception that the total final size of the project would be manageable at this resolution.

Second, a naming scheme was developed such that no two files would have the same name. The scanner was set to make a separate PDF for the front and back side of each sheet. The DCA had arranged pages by chapter number. Each chapter was scanned in a batch with file names indicating the batch and the scanner set to sequentially number the end of the file names. This allowed all PDFs to be placed together and manipulated, without any having to be later renamed.

Third, after pages were scanned, Adobe Acrobat Professional was used to run OCR on large sets of PDFs at once. Adobe Acrobat Professional is readily available software, which many libraries already own a license for. Versions 9 and later allow OCR to be run on large sets of files all at once. A batch function was created in Acrobat 9 Professional to run OCR on all FAC pages. The function was set up once and run, then left for days until it finished the set. Staff time spent handling files was minimal.

Fourth, after OCR was run on all pages, an appropriate tool was located for extracting text from PDFs to Excel spreadsheets. The final tool chosen was A PDF to Excel Extractor. Many low cost tools which do this are available and offer trial versions to assess output. Text of each PDF was extracted to a separate sheet of an Excel spreadsheet. Figure 5 shows a PDF and next to it an Excel spreadsheet with the extracted text.

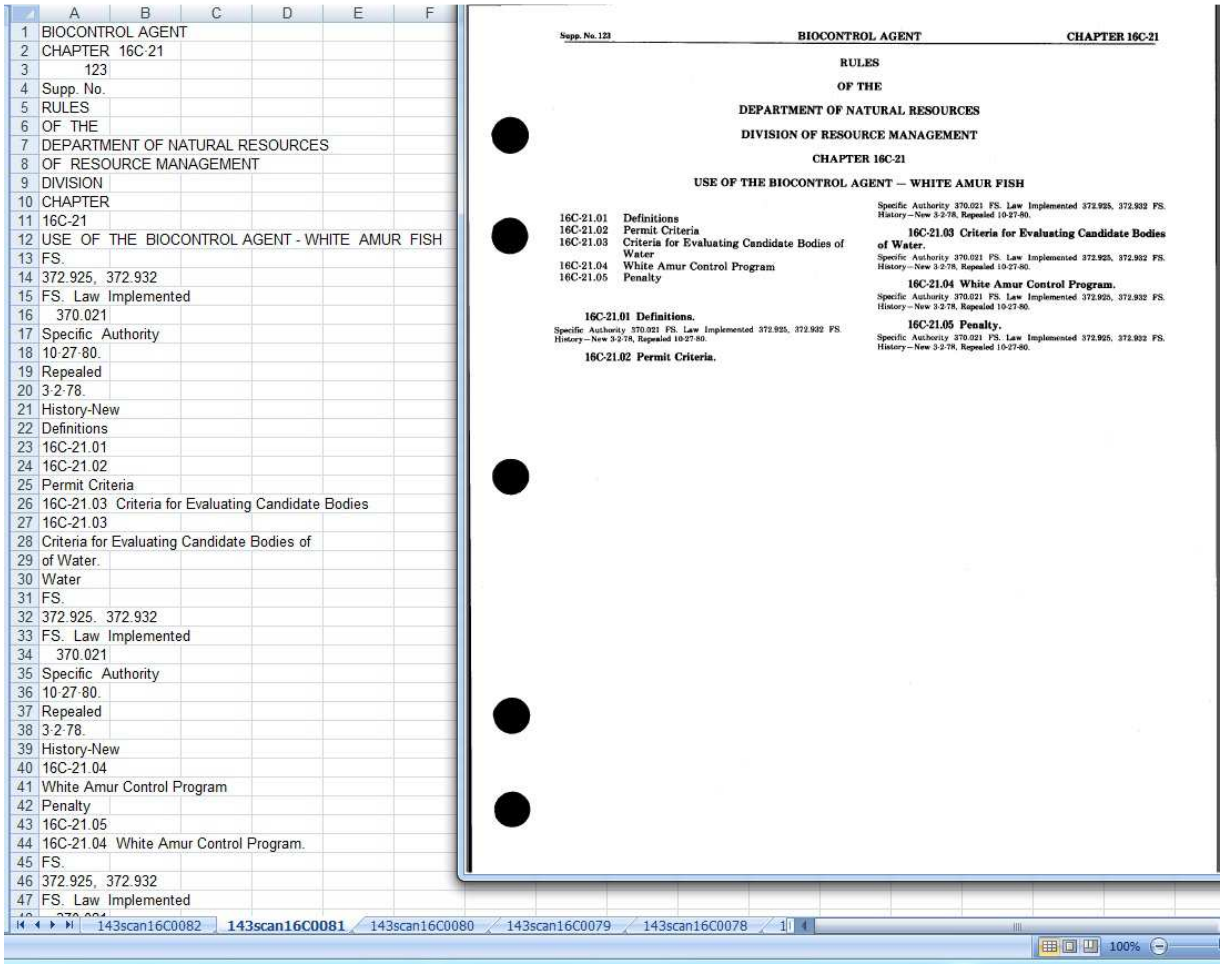


Figure 5. An FAC page and the Excel extraction of text from that FAC page. The full text of each PDF was extracted to a separate sheet in an Excel spreadsheet and the sheet was named with the file name of the PDF. Here you can see both the file naming scheme, and a comparison of the page image with the page represented as a string of text.

Fifth, spreadsheets were examined and compared with images of pages. Step-by-step rules were written for how to locate metadata for each field. Page number was consistently the last row with anything in it on the sheet. Supplement number was usually in the first 5 rows, then in the row containing the word "supp". Chapter number was usually in the first 5 rows, then in the row containing "hap". The dash in the middle of the chapter number was sometimes a dash, sometimes a period, and sometimes a squiggle, so all had to be accounted for. Step-by-step

English language instructions for locating each metadata field were written. Writing the English language instructions is a librarian skill, because it requires familiarity with the documents, and will take a similar amount of time for a librarian as for a programmer. It is possible that with clear step-by-step English language directs, a programmer may translate to code as a favor, because it will not take the programmer much time to do.

Sixth, the English language instructions were translated into Visual Basic. Visual Basic is a scripting language which interacts with Microsoft's line of office products. Because it is designed to work with Excel, the learning curve is much lower than for a full programming language. For example, in Visual Basic, "Worksheet(1)" refers to the first worksheet in a spreadsheet file. Full programming languages are more versatile, but were rejected because there is a steep learning curve to get into the file header before manipulating data. With Visual Basic it was possible to start with the meat of the problem. One of the barriers to writing Visual Basic was the lack of entry level training material. The only truly entry level training on Visual Basic found was a four-step tutorial online.[3] This was enough to get started with the syntax, and understand message board postings addressing specific problems and script to solve them. The other tool which was helpful was to read about string processing. String means a string of letters or text. String processing means manipulating that text. Technical jargon from the field of string processing was used in order to search for instructions on desired tasks.

Seventh, after the script was run, the Excel spreadsheets were cleaned up. Figure 6 shows a sample portion of an Excel spreadsheet before cleanup. Cleanup was performed manually by deleting nonsense values, and reviewing for simple OCR errors such as a number "1" being read as a lowercase "L". The cleaned up values were loaded into an online database. Automated metadata extraction resulted in a value being assigned for: 99.3 percent of chapter numbers

before the dash; 92.2 percent of the chapter numbers after the dash; 93.6 percent of page numbers; and 88.4 percent of supplement numbers. A database with only the automated metadata may be searched by visiting: http://fsulawrc.com/automatedindex.ph . Searching gives a listing of complete matches, and also several listings of partial matches for items with incomplete metadata. Although metadata was filled in for more than 90 percent of most fields, a much higher percentage of PDFs had some missing metadata, because the gaps occurred on different PDFs rather than all fields on the same PDF. Frequently there was no exact match for a desired page, and several listings of partial matches must also be consulted. Search is less time consuming than an in-person visit to the archives, but is still tedious. In the subsequent steps, human indexing was needed to complete the index.

	A	B	C	D	E	F	G	H
1		Identifier	Ch. No. before das	Ch. No. after dash	Page no.	Supp. No.	Filename	
2		143scan16C0081	CHAPTER 16C	21	65		143scan16C0081.pdf	
3		143scan16C0080			64	89	143scan16C0080.pdf	
4		143scan16C0079	CHAPTER 16C20	CHAPTER 16C20	63	89	143scan16C0079.pdf	
5		143scan16C0078					143scan16C0078.pdf	
6		143scan16C0077	CHAPTER 16C	19	62C	87	143scan16C0077.pdf	
7		143scan16C0076	CHAPTER 16C	19 AQUATIC PLA	62B	87	143scan16C0076.pdf	
8		143scan16C0075	CHAPTER 16C	19	62A	87	143scan16C0075.pdf	
9		143scan16C0074	CHAPTER 16C	19	62		143scan16C0074.pdf	
10		143scan16C0073	CHAPTER 16C	19	61	87	143scan16C0073.pdf	
11		143scan16C0072					143scan16C0072.pdf	
12		143scan16C0071			59	7	143scan16C0071.pdf	
13		143scan16C0070					143scan16C0070.pdf	
14		143scan16C0069	CHAPTER 16C	17	57	79	143scan16C0069.pdf	
15		143scan16C0068	16C	16 CHAPTER	HistoryNew41981101		143scan16C0068.pdf	
16		143scan16C0067	CHAPTER 16C	16	53	126	143scan16C0067.pdf	
17		143scan16C0066	16C	16 CHAPTER	52D	125	143scan16C0066.pdf	
18		143scan16C0065	CHAPTER 16C	16	52C	125	143scan16C0065.pdf	
19		143scan16C0064	CHAPTER I6C	16	52B	125	143scan16C0064.pdf	
20		143scan16C0063	CHAPTER 16C	16	52A	125	143scan16C0063.pdf	
21		143scan16C0062	CHAPTER 16C	16 MINE RECLAN	52	125	143scan16C0062.pdf	
22		143scan16C0061	CHAPTER 16C	16	51	125	143scan16C0061.pdf	
23		143scan16C0060	16C	16 CHAPTER	50	125	143scan16C0060.pdf	
24		143scan16C0059	CHAPTER 16C	16	49	125	143scan16C0059.pdf	
25		143scan16C0058					143scan16C0058.pdf	
26		143scan16C0057	APPLICATION AND	15	48G		143scan16C0057.pdf	
27		143scan16C0056	CHAPTER 16C	15	ffl		143scan16C0056.pdf	
28		143scan16C0055	CHAPTER 16C	15	48E	107	143scan16C0055.pdf	
29		143scan16C0054	CHAPTER 16C	15	48D	107	143scan16C0054.pdf	
30		143scan16C0053	CHAPTER 16C	15	48C	107	143scan16C0053.pdf	
31		143scan16C0052	CHAPTER 16C	15	48B	107	143scan16C0052.pdf	

Figure 6. Screenshot of the metadata extracted by computer script before manual cleanup. Some fields, like the chapter number before the dash, can be reviewed then replaced all at once by pasting 16C in. (This spreadsheet came from a batch where only Chapter 16C was scanned; other batches have different chapters.) Items with mostly blank fields are likely back sides of sheets with printing on one side only.

Eighth, missing values were filled in by a part-time student worker using an online form which showed the PDF, then provided a blank for the missing information. A screenshot of this form is included in Figure 7.

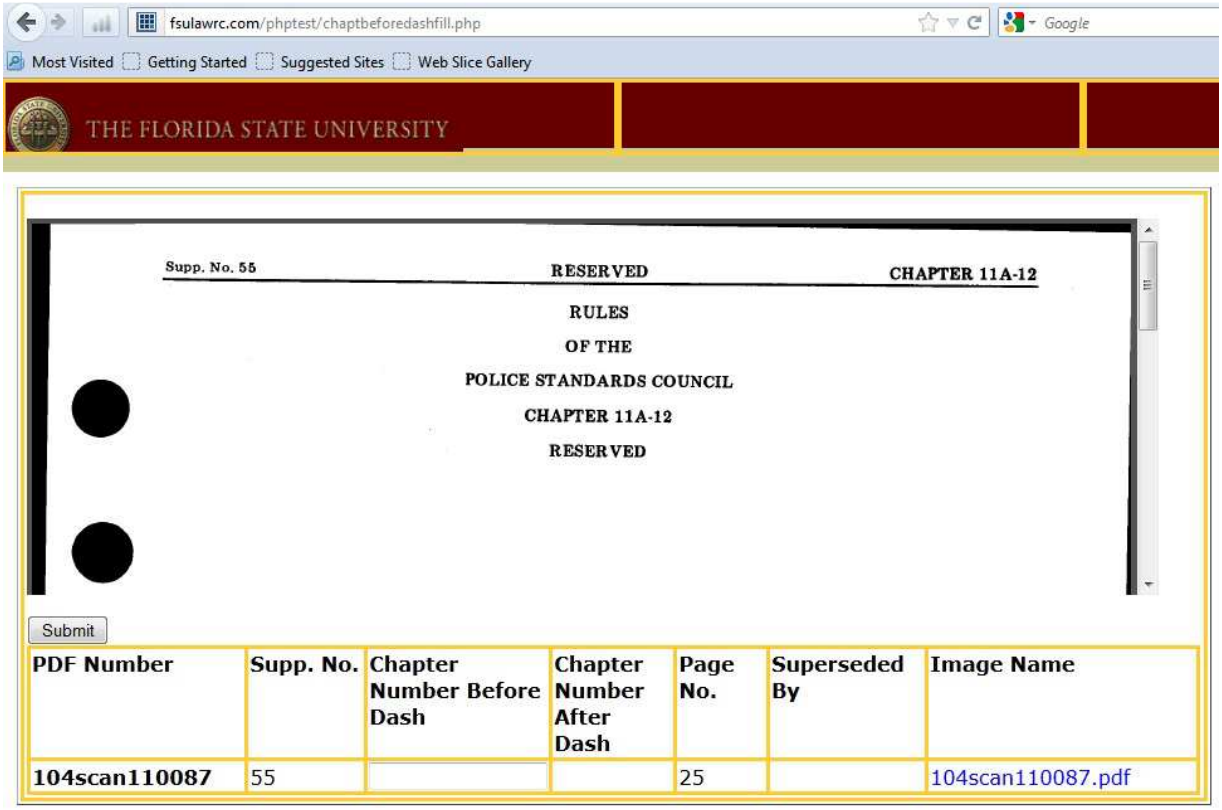


Figure 7. An online form was used to fill in values which computer scripting did not get. This form prompts for a value for the chapter number before the dash, and then the person types it and presses enter to get to submit the value and get to the next PDF.

Ninth, the instruction sheets for each supplement were compared against the database to ensure that where a page was instructed to be taken out of the binders, the database recorded a page coming out, and where a page was instructed to be inserted into the binders, the database recorded a page going in. Blanks were added for missing pages. This audit made the database more authoritative for persons performing legal research, and ensured that no inaccurately dated page will be provided.

QUALITY COMPARISON: ERROR RATES IN COMPUTER AND HUMAN GENERATED METADATA

During the audit for missing pages, if a page was found missing, then a search was made for possible mis-indexing. If an indexing error was found, the error was corrected. The audit caught errors across both the computer generated and human created metadata. This allowed a calculation and relative comparison of error rates. To compare error rates, three spreadsheets were examined: the spreadsheet of metadata after only automated indexing (Spreadsheet Computer), the spreadsheet of metadata after empty fields were filled by a human indexer (Spreadsheet Person), and the final spreadsheet of metadata after the audit (Spreadsheet Audit).

To calculate error rates for the computer, Spreadsheet Computer was compared to Spreadsheet Audit. For each field, only records where the computer had filled a value for that field were examined. Of those, the number of records where the field value filled by computer had changed after the audit (indicating a correction) was divided by the total number of records where the computer had filled that field. A similar comparison was made between the Speadsheet Person and Spreadsheet Audit. Values which had been blank in Spreadsheet Computer and were filled in Spreadsheet Person were checked for errors by comparing with the final value in Spreadsheet Audit.

Error rates for chapter number were not compared, because the physical arrangement of the pages before scanning had been used to game this. Pages were scanned one chapter at a time, and the chapter number was often included in the file name.

For comparison, error rates were as follows:

- Supplement number: Human error rate: 0.8%
- Supplement number: Computer error rate: 2.4%
- Page Number: Human error rate: 3.1%
- Page Number: Computer error rate: 1.0%

We often think of human generated metadata as superior to computer generated metadata. Here, error rates were comparable between the two. Computer errors were due to errors in the OCR as well as errors from the program pulling the wrong part of the text. Reasons for human errors may have come from the repetitive nature of the task. Looking at an image and writing down numbers that appear in that image again and again for several hours is not intellectually stimulating, and the repetitive nature can be very tiring. While it might at first seem that a simple task like this was easy to do because it required little training, it actually may have been harder than a more engaging task.

COST BENEFIT ANALYSIS: HOW MUCH TIME WAS SPENT OR SAVED IN AUTOMATED INDEXING

Staff hours for the project were tracked and sorted into general areas. Some tasks, like digitizing pages and database design, would have been performed regardless of how indexing was done. Both computer indexing and human indexing had some time costs that did not scale with the size of the document set being indexed: writing the script for computer indexing, and writing the online form for the person to fill out. Both types of indexing had some time costs which did scale with the size of the set of documents: cleaning up computer generated metadata, and time spent typing values into the manual data entry form.

A breakdown of time spent in different areas of the project follows:

Database work: 50 hours equal to 8 percent of total time

- Inspecting the printed pages and planning the database: 20 hours (high skill, high training)
- Planning logic behind the script for automated indexing: 20 hours (high skill, high training)

- Loading database and metadata onto a server: 10 hours

Digitization: 35 hours equal to 6 percent of project time

- Digitization with a sheet-fed scanner: 35 hours (low skill, low training)

Automated metadata creation: 70 hours equal to 11 percent of total time

- Writing script to extract metadata: 35 hours (would be much faster for a programmer)
- Running the script, and cleaning up metadata: 35 hours (skilled staff)

Manual metadata creation: 128.25 hours equal to 20 percent of total time

- Coding the online form for data entry: 15 hours (skilled staff)
- Training the student worker on documents and database design: 15 hours (unskilled staff)
- Metadata entry for fields the computer did not get: 98.25 hours (unskilled staff)

Auditing: 342.75 hours equal to 55 percent of total time

- Comparing instruction sheets against the database and verifying that each page which went into or came out of the binders is represented: 342.75 hours

Human indexing and computer indexing took about the same amount of time, but computer indexing got more than 80 percent of the total metadata, providing far more productivity relative to cost. Digitization took only 6 percent of total project time, while metadata creation and the audit of the final database took 86 percent of total project time.

RECOMMENDATIONS AND FUTURE DIRECTIONS

Serials knowledge can break new ground in digital libraries by contributing expertise in unusual resources, like the looseleafs. The digital library and database communities so far have focused on self-contained one-shot resources, like reports, and journal runs. More research should be

done to investigate digital representations of resources with unusual formats, such as the looseleaf.

Practitioners in metadata and cataloging should consider using computer generated metadata. For this project, the computer and the person had comparable error rates, and computer generated metadata took less staff time to create, even with time spent learning new technology skills and scripting. Computer generated metadata may be efficient, even for someone with a limited technology background.

Librarians seeking to collaborate with IT should keep the following points in mind: The librarian is the expert on the collection of items to index, and should not expect the technologist to become an expert in the collection. The librarian needs to be aware of the specific flavor of technology skills, and request advice from someone with the appropriate skill set. Finally, technology assistance is likely to come from outside of the librarian's department or organization, and so the librarian needs to be respectful of the technologist's time. It is easier to find and tap an expert for broad knowledge and pointers, than to find an expert to do the work for you.

ACKNOWLEDGEMENTS

Special thanks to Jason Cronk for assistance with project planning, and to Anna Annino for assistance with metadata creation.

1. Jesse Alpert and Nissan Hajaj. "We knew the web was big..." Google Blog. July 25, 2008. http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html (accessed July 3, 2012).

2. Eric Hellman. "Google Exposes Book Metadata Privates at ALA Forum." *Go to Hellman.* January 18, 2010. http://go-to-hellman.blogspot.com/2010/01/google-exposes-book-metadata-privates.html (accessed July 3, 2012).
3. Pan Pantziarka, "Visual Basic Tutorial," accessed July 3, 2012, http://www.techbookreport.com/tutorials/excel_vba1.html.
4. Github.com, accessed November 19, 2012, https://github.com/ao5357/thesisbot.

CONTRIBUTOR NOTE

Wilhelmina Randtke is Electronic Services Librarian, St. Mary's University School of Law, San Antonio, Texas