Technology and Trends
Wrangle Your Data Like a Pro With the Data Processing Power of Python

Geoffrey P. Timms, Librarian for Marine Resources, College of Charleston/SCDNR/NOAA

Jeremy M. Brown, Assistant Dean for Technical Services and Systems, Mercer University

Abstract

Management, delivery, and marketing of library resources and collections necessitate interaction with a plethora of data from many sources and in many forms. Accessing and transforming data into meaningful information or different formats used in library automation can be time consuming, but a working knowledge of a programming language can improve efficiency in many facets of librarianship. From processing lists to creating extensible markup language (XML), from editing machine-readable cataloging (MARC) records before upload to automating statistical reports, the Python programming language and third-party application programming interfaces (APIs) can be used to accomplish both behind-the-scenes tasks and end-user facing projects. Creating programmatic solutions to problems requires an understanding of potential. Here we summarize the data sources, flows, and transformations used to accomplish existing projects at Mercer University and the College of Charleston. Foundational programming techniques are explained and resources for learning Python are shared.

Introduction

Libraries generate and have access to more data than ever before, which presents exciting opportunities to take library services to new heights and enhance or personalize patrons’ experiences. All it takes is data transformation. There are many reasons why a librarian might seek to transform data: Perhaps to add value by presenting them in a different form, or to prepare them structurally for ingestion into a database or library automation system. Unfortunately, working with data can be a time-consuming process. Some tasks, such as documenting library use, may be demanded monthly. Others, such as processing e-resource usage data, may only be tackled annually but consist of great quantities of data. Whatever the reason, data transformation is undertaken to restructure, synthesize, or substitute data to achieve a higher level of utility.

The transformation of data is part of a process that can be automated if data is available in a machine-readable form and is of a predictable structure. Ideally, data should be accessible with an automated process, and the transformed data should be deliverable in an appropriate format and location automatically. However, this is not always technically possible, and human intervention may be required during the transformation process. Regardless, automated transformation can accomplish in fractions of seconds what may consume hours of a person’s time.

Once one or more datasets are obtained, they must be ingested by a script and then organized in a manner suitable for the transformation process. The transformation itself may be as simple as rearranging the data or tidying them by removing superfluous data. Transformation may involve generating visual representations of data or using parts of the data to reference other information stored online, such as book cover art. Once transformed, the output is then delivered in the appropriate form, be it XML, HTML, or images for on-screen display; stored as data in an SQL database; or inserted into a file. The following examples of data transformation projects demonstrate the data sources and the output of new information. See Appendix for figures, data, and tables.

Virtual New Bookshelf

- Transformation summary: Bibliographic data transformed to XML for Web display.
- Data input: Text file manually downloaded from library catalog (Table 1) consisting of record number, ISBN string, title, call number, and publication information for recently purchased books.
• Output: Write XML to an XML file (Figure 1), which is then styled using XSL for HTML display (Figure 2).

Map of the Library Shelving Highlighting the Range for a Chosen Item

• Transformation summary: Call number and location code used to create a visual map of the chosen item’s location.
• Data input:
  o Catalog record for chosen item information (Figure 3).
  o SQL database tables including:
    ▪ Range number, location code, call number of first book on range (Table 2).
    ▪ Map coordinates of each range in each location code (Table 3).
    ▪ Waypoints for each range in each location code (Table 4).
  o Base map image file for each floor showing all ranges.
• Output: Return custom map image to the screen (Figure 4).

Conversion of Monthly Transaction Logs to a Multisheet Excel Summary Report

• Transformation summary: Transaction logs from ILL borrowing and lending, and reference analytics transformed to multisheet Excel workbook documenting multiple agencies’ use of the library.
• Data input: CSV and Excel files manually downloaded monthly from Interlibrary Loan (ILLiad) (Table 5) and Reference Analytics.
• Output: Excel workbook with three summary report sheets (Table 6).

Foundational Programming Techniques

Some programming techniques are fundamental to most of the projects that the authors have undertaken. Several are presented below.

Variables, Dictionaries, and Lists

In Python, as in most programming languages, we deal with data of various types: Among other types, there are integers (whole numbers), floating point numbers (such as 3.14), and strings (alphanumeric data of any length). Python detects the type of variable based upon the data that we put into it. For example:

```python
integer_value = 4
string_value = 'dog'
float_value = 3.14
```

Frequently, we deal with data that makes sense to group together. For example, we need to make a list of similar values. This list could represent a row or a column of a spreadsheet, or it could simply be a list of values. In Python, we create a list with square brackets, separating the values by commas:

```python
a_list = [1, 3, 4, 5, ]
```

Proper form dictates that we leave a trailing comma, in case we want to add another value. We could create an empty list in a couple of different ways:

```python
a_list = []
another_list = list()
```

Lists are mutable, meaning they may be changed after creation, so we could take our list, and add an element to it:

```python
a_list.append(6)
```

Or we could sort it:

```python
a_list.sort() # sorts in ascending order
a_list.sort(reverse=True) # sorts in descending order
```

To access data in our list, we simply call it by its position within the list:

```python
a_list[0]
```

In our first list, which contained the values 1 through 5, we would receive the first one, which is the number 1. Note that list indexes always begin with zero.

Other times, we have data that belongs together but needs some amount of compartmentalization. In
Python, a useful data structure for that kind of data is a dictionary. In Python, a dictionary is a pair of data that is made up of a key and a value. Typically, a simple data type (such as a string or an integer) is used as the key, but the value could be anything. Curly braces denote the beginning and end of a dictionary. For example:

```
a_dictionary = {
    'name': 'Hans Mustermann',
    'surname': 'Mustermann',
    'given_name': 'Hans',
    'age': 35,
    'children': ['Frank','Annie','Jim Bob'],
}
```

In the above example, note that we have a mix of data types. We have three strings: name, surname, and given_name, followed by an integer (age) and then a list (children). Values that we store in dictionaries can be retrieved with the same syntax as the list uses:

```
a_dictionary['children']
```

This would give us the list we stored:

```
['Frank','Annie','Jim Bob']
```

The thing to remember about dictionaries is that there is no guarantee that anything will be stored in any particular order.

**Iterating Over/Comparing Variables and Lists**

Iterating over a list allows us to examine each element in the list and potentially do something with it. Consider a case where we declare a list full of numbers and want to print each number on its own line:

```
a_list = [1,2,3,4,5]
```

```
for item in a_list:
    print(item)
```

This “for” statement proceeds through a list and assigns a value to “item.” This is convenient when we just need to use the item and move on. If, however, we want to compare two lists of values to see how many times items in one list appear in another list, we must proceed differently. We declare two lists: a data list and a list with values to look for and count. Since the second list really has two items for each element (a value and a count), we could use a dictionary to keep these values separate, so we put together a list of dictionaries.

```
data_list = [6, 5, 9, 1, 8, 2, 9, 7,
            1, 4, 2, 1, 9, 3, 1,
            3, 5, 4, 7, 2, 5, 1,
            6, 7, 3, 2, 4, 9, 8, 3,
            2, 5, 6, 9, 7, 1, 8,
            6, 1, 2, 8, 7]
```

```
a_list = [
    {'value':1, 'count':0},
    {'value':2, 'count':0},
    {'value':3, 'count':0},
    {'value':4, 'count':0},
    {'value':5, 'count':0}
]
```

Now that we have our data, we can prepare to look through our two lists. Because we have two lists, we will need to have two loops. This is called a nested loop because one loop is within the other loop:

```
for item in a_list:
    for r_item in data_list:
        if item['value'] == r_item:
            item['count'] += 1
        else:
            pass  # do nothing
    print(item)
```

The first loop assigns each value of a_list to the variable “item.” Then, we immediately proceed into the data_list. We assign a value from data_list to the variable “r_item,” and then we compare the value of the item to r_item. We use the “equals” operator, which is a double equal sign (==). If the two integer values are equal, this returns true, and we then add one to item’s count argument. If the two items are not equal to each other, we don’t do anything. Instead, we just move on to the next element in data_list. When we have moved through all 45 values in data_list, we print the values that we just stored in item, and then move on to the next item in a_list and repeat the comparison process. This renders the output:
Reading and Writing Files (Input/Output)

In practice, we encounter data files that contain lists. These could originate from Excel, or they could be reports from another system. Frequently, the data fields are separated by special characters, such as commas, tabs, or some other character. In this example, we are reading two lists of ISSN data. The task consists of the following steps:

1. Read in all the lines of two files.
2. Create results.
3. Output this to a new file

```python
with open('input1.csv','rU') as L1:
    reader = csv.reader(L1, delimiter="\t")
    list1 = []
    for sublist in list(reader):
        list1.append(sublist.pop())
    list1.pop(0)  #Remove the first row from list1, which is the column names
```

The code above gets us a file handle (L1) pointing at the data file. We point a CSV reader at that, configured to handle tab-delimited data. We create an empty list (list1) to store our data. The for loop is dense code: We are converting all lines in the file to a list. Each list here represents the rows in a spreadsheet, and each contains a sublist of column values. Because we only have one column of data, we can pop the only item off the sublist and append it to our results list, list1. This takes care of our input.

Once we have processed our input, we will need to output the data. For this example, we have our output in a dictionary called finaldata, which we wish to store in a Microsoft Excel file. Using xlwt makes this almost as simple as the prior example:

```python
wb = xlwt.Workbook()
Sheet1 = wb.add_sheet('Journals found in both lists')
Sheet1.write(0,0, 'Linking ISSN-L')
Sheet1.write(0,1, 'Title')
n=1
for issn,title in finaldata.items():
    Sheet1.write(n,0,issn)
    Sheet1.write(n,1,title)
    n+=1
```

Regular Expressions (Pattern Matching)

In our case study, we face data of varying formats. Really, the values are all the same, but punctuation and notation differ. As long as this is somewhat regular, we can anticipate this variance and obtain the data we need. Pattern matching is a very broad topic in itself, but it is simple to learn the basics and begin extracting data.

Before we begin, it is important to realize that most every character we can type will match itself and become part of the pattern. A “2” generally matches a typed number two, for example. However, we need to be aware of some special characters that are used in the syntax of this powerful programming language: .^$*+?-{[]}(). These each have special meanings.

Before we can start using regular expressions in Python, we need to import the library using an import statement:

```python
import re
```

We can then proceed to use it in our subsequent code. We will focus on the convenience functions, re.sub() and re.match(). Consider the case of an ISSN. An ISSN is typically eight meaningful digits, except when the final character also happens to be an X. It could be written in several ways. Let us consider three cases:

1. Written with a hyphen between two sets of four characters: 0022-510X.
2. Written without the hyphen: 0022510X.
3. There could be extra characters before or after our ISSN: ISSN: 0022-510X Journal of the Neurological Sciences.
An easy way to attack this problem is to detect which case we have and deal with that with an if/elif statement. However, we have to use the match function first. Match returns a match object if it does indeed match something, or if it matches nothing, it returns a value of None. If we receive a value of None, we know we did not get a match.

Examining the first pattern, we see four digits in a row followed by a hyphen, followed by three digits and an X. That X could also be a digit, but this informs us as to how to write our pattern:

```python
re.match(r'\d{4}-\d{3}X', '0022-510X')
```

The string itself specifies that we want two groups of characters: Four digits, a hyphen, and three digits followed by anything zero through nine, or an upper or lowercase X. The second value is our ISSN string.

In our second case, perhaps we have data where we do not know if we have a hyphen or not. Both could be correct, but if we want to match that, we could use much the same search string, but then address the number of hyphens. We could have zero hyphens, or we could have a single hyphen. This statement detects either of those cases:

```python
re.match(r'.*\d{4}-(0|1)\d{3}[0-9Xx]', '0022510X')
```

Our third and final case is an ISSN with extra text before or after the ISSN. We can craft a search pattern that looks for any character (a period: .) any number of times (an asterisk: *), and include that before and after our search pattern:

```python
r'.*\d{4}(0|1)\d{3}[0-9Xx])\.*
```

The next step is to clean up that regular expression. We can use the re.sub() function to do just that. Sub stands for substitution, and it allows us to use regular expressions to replace characters within a string. There are three arguments that we need to provide to this function:

1. A pattern, which we have already seen in our handling of the match function.

2. A substitution pattern, and finally

3. A string to examine.

The substitution pattern is really the only new thing here. In our regular expression that perfectly matches our three cases, we already have two sets of parentheses. Each set captures a portion of the ISSN, and each portion is exactly four characters. We could add that to a substitution call and cause it to reformat any of the three cases into our preferred format: 0022-510X, which looks like the following:

```python
re.sub(r'.*\d{4}-(0|1)\d{3}[0-9Xx])\.*', 'ISSN: 0022-510X Journal of the Neurological Sciences')
```

**Using Data from APIs**

The acronym API is used very frequently in software circles. These days it seems that it has penetrated our conversations a little further, since many publishers and most systems vendors tell us they have their own APIs that we only need to reach out and use. API stands for application programming interface, and defined, an API provides a means for a programmer to write a program that uses their software.

Since an API is essentially a way for us to communicate with another application, we need to know what data to send it and what to expect from the API when it gives us data back. In our next case study, we use the OCLC xISSN Web service (http://xissn.worldcat.org/xissnadmin/doc/api.htm). The documentation tells us that we can obtain journal metadata from the API (namely, we want the title that corresponds to the ISSN) if we query it using a URL of the form http://xissn.worldcat.org/webservices/ixsissn0022-510X?method=getEditions&format=json

There are many ways to access URLs in Python. A popular one is to use the Requests library (http://docs.python-requests.org).
import requests, json


r = requests.get(url)
formatted_json = json.dumps(r.json(), sort_keys=True, indent=4, separators=(',', ': '))
print(formatted_json)

In this example, we first import the two libraries we need: requests and JavaScript Object Notation (JSON). Then we define a URL. Next, we create a requests object for our URL. This requests object fetches the data from the URL when we create it, so this is where we have done all the heavy lifting. At this point, we have all the data we need, but long blocks of JSON are somewhat tough to read. The next thing is to reformat the JSON so we can easily read it. We use Python's built-in JSON library to sort the keys, indent each level four characters, and use a comma as a list separator and the colon as the assignment indicator.

When we examine this output (Appendix, Figure 5) we see two data structures: stat and group. Inside "group," we find our data. The trick to accessing this data is to pry the dictionary apart. If we want the title, we would have to access it from within the dictionary contained in list. This looks like so:

r.json()['group'][0]['list'][0]['title']

This is because the title is contained by a dictionary and contained by a list, which is in a dictionary called "list," which is a key in a dictionary contained within a list called "group," which is a key in a dictionary.

Case Study

The following case study utilizes the above programming techniques. Initial data are provided in two differently formatted lists of ISSNs stored in two CSV files (Appendix, Table 7). One list (~11,000 ISSNs) represents a broad array of journals. The other list (49 ISSNs) represents locally held titles. The lists are transformed into an Excel spreadsheet that presents the corresponding linking ISSNs (ISSN-Ls) and journal titles only for the ISSNs that occurred in both lists, using an OCLC API. The Python code is broken into discreet chunks called functions, representing each phase of the transformation. The full code can be found at http://libraries.mercer.edu/ursa/handle/10898/3687.

Phase One

The make_lists function opens each CSV file in turn, reads it as a tab-delimited structure, and puts its content into a list.

Phase Two

The clean_list function is called once for each list of ISSNs. Each list item is taken in turn and tested against a regular expression to see if a properly formed ISSN can be extracted. If so, it is reformatted and added to a new list. Otherwise, an error is reported, and the script moves to the next item.

Phase Three

The get_dupelist function compares both cleaned lists to find duplicate ISSNs using an internal method called “intersection.” To use this method, each list must be converted into sets. Sets differ from lists because they are unordered and can contain no duplicates. The intersection method results in a set of items found in both sets. This set is converted to a list for use in Phase Four.

Phase Four

The get_oclc function is used to create a dictionary of key:value pairs representing ISSN-L (the key) and journal title (the value) obtained from the OCLC API described earlier. For each ISSN in the list generated in Phase Three, the API URL is formed using the ISSN, and the request is sent. The ISSN-L and title are extracted from the response.

Phase Five

The write_sheet function creates an Excel workbook and sheet with column headings. It then iterates over the data dictionary to write the key:value pairs of ISSN-L and title to the spreadsheet. Once every dictionary item has been written, the spreadsheet is saved, and the operation is complete (Appendix, Table 8).
Communicating With Systems People

Realistically, most complex programming will be undertaken by those who already possess the experience to deliver programmatic solutions to particular problems. As such, a data transformation project will likely involve communication between detail-oriented programmers and other staff who may have a different perspective. In order to help a programmer deliver a solution to a problem, it is important to be ready to:

- Explain what you want to accomplish.
- Describe the desired end result in detail.
- Detail data sources, formats, and locations.
- Outline data transformations.
- Discuss potential data anomalies.
- Provide data samples.

Participating in such a project provides the perfect opportunity to contribute to and learn about the solution put into place by the programmer. Such inquiry may also contribute to one’s own development as a novice programmer. Learning by doing is perhaps the best way to get started.

Getting Started With Python

Python may be installed and operated on a workstation or server. Python libraries may also be added in both these contexts. For Python to interact with SQL-based databases, software such as PostgreSQL, MySQL, or SQLite must also be installed. User interaction with Python may be undertaken in a variety of ways. An integrated development environment (IDE) is separate software used to make the creation, testing, and preservation of code as convenient as possible. Alternatively, Python may be developed and run from a command line interface, which is a less convenient method.

Software

- Python: https://www.python.org/
- Python’s built-in libraries: https://docs.python.org/3/library/
- Eclipse IDE: https://eclipse.org/ide/
- SQL databases
  - MySQL: https://www.mysql.com/
  - SQLite: https://sqlite.org/
  - PostgreSQL: https://www.postgresql.org/
- Resources for Learning Python
  - A Byte of Python: https://python.swaroopch.com/
  - Codecademy: https://www.codecademy.com/learn/python
  - Stack Overflow discussion list: http://stackoverflow.com/questions/tagged/python

Conclusion

Data transformation with Python may be achieved at various levels of complexity. At its simplest, we have cleaned and compared two lists of data and referenced an API. Our examples of more complex projects undertaken at our libraries recently demonstrate the potential for transformation of data into visual or graphical information. Successfully learning a programming language is within reach of self-motivated individuals who possess a curious and systematic mindset. Libraries that are fortunate enough to have programmers on staff will be better able to utilize these unique skills when nonprogrammer staff have a basic appreciation for the potential offered by programmatic solutions to library tasks and problems involving data.
## Appendix

Table 1. Bibliographic data exported from Library Management System.

<table>
<thead>
<tr>
<th>RECORD # (BIBLIO)</th>
<th>STANDARD #</th>
<th>TITLE</th>
<th>CALL INFO</th>
<th>PUB INFO</th>
</tr>
</thead>
</table>

Figure 1. XML output following transformation.
Figure 2. Virtual bookshelf web page resulting from XML/XSL data.

Figure 3. Catalog record excerpt for a book, showing location and call number.
Table 2. PostgreSQL data excerpt for shelf ranges showing collection and normalized call number for the first item on each shelf range.

<table>
<thead>
<tr>
<th>id</th>
<th>shelf</th>
<th>location</th>
<th>call_start</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>24</td>
<td>T3mstk</td>
<td>BX0877400596000</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>T3mstk</td>
<td>D00000200A70000</td>
</tr>
<tr>
<td>26</td>
<td>26</td>
<td>T3mstk</td>
<td>D00052100H35000</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>T3mstk</td>
<td>D00075700A64000</td>
</tr>
<tr>
<td>28</td>
<td>28</td>
<td>T3mstk</td>
<td>D00080430L35700</td>
</tr>
<tr>
<td>29</td>
<td>29</td>
<td>T3mstk</td>
<td>DA00228001L30000</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>T3mstk</td>
<td>DA0039600A20000Y60000</td>
</tr>
<tr>
<td>31</td>
<td>31</td>
<td>T3mstk</td>
<td>DA0053000N53000</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>T3mstk</td>
<td>DC0010300S20000</td>
</tr>
</tbody>
</table>

Table 3. PostgreSQL data excerpt for shelf ranges showing collection location, map coordinates, arrowhead aspect (aisle), waypoints, and fill style.

<table>
<thead>
<tr>
<th>id</th>
<th>shelf</th>
<th>location</th>
<th>coordinates</th>
<th>aisle</th>
<th>waypoints</th>
<th>style</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>23</td>
<td>T3mstk</td>
<td>[[133,287,135,287,135,384,133,384]]</td>
<td>w</td>
<td>[300,301]</td>
<td>f</td>
</tr>
<tr>
<td>25</td>
<td>24</td>
<td>T3mstk</td>
<td>[[133,287,135,287,135,384,133,384]]</td>
<td>e</td>
<td>[300,301]</td>
<td>f</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>T3mstk</td>
<td>[[147,73,149,73,149,132,147,132]]</td>
<td>w</td>
<td>[300,301,302,303]</td>
<td>f</td>
</tr>
<tr>
<td>27</td>
<td>32</td>
<td>T3mstk</td>
<td>[[147,73,149,73,149,132,147,132]]</td>
<td>e</td>
<td>[300,301,302,303]</td>
<td>f</td>
</tr>
<tr>
<td>28</td>
<td>32</td>
<td>T3mstk</td>
<td>[[147,160,149,160,149,257,147,257]]</td>
<td>w</td>
<td>[300,301,302]</td>
<td>f</td>
</tr>
<tr>
<td>29</td>
<td>31</td>
<td>T3mstk</td>
<td>[[147,160,149,160,149,257,147,257]]</td>
<td>e</td>
<td>[300,301,302]</td>
<td>f</td>
</tr>
<tr>
<td>30</td>
<td>29</td>
<td>T3mstk</td>
<td>[[147,331,149,331,149,384,147,384]]</td>
<td>w</td>
<td>[300,301]</td>
<td>f</td>
</tr>
<tr>
<td>31</td>
<td>30</td>
<td>T3mstk</td>
<td>[[147,331,149,331,149,384,147,384]]</td>
<td>e</td>
<td>[300,301]</td>
<td>f</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>T3mstk</td>
<td>[[161,73,163,73,163,132,161,132]]</td>
<td>w</td>
<td>[300,301,302,303]</td>
<td>f</td>
</tr>
</tbody>
</table>

Table 4. PostgreSQL data excerpt for waypoints showing waypoint coordinates.

<table>
<thead>
<tr>
<th>id</th>
<th>wpid</th>
<th>waypointcoords</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>104</td>
<td>[302,226]</td>
</tr>
<tr>
<td>6</td>
<td>105</td>
<td>[302,210]</td>
</tr>
<tr>
<td>7</td>
<td>106</td>
<td>[302,83]</td>
</tr>
<tr>
<td>8</td>
<td>107</td>
<td>[415,83]</td>
</tr>
<tr>
<td>9</td>
<td>200</td>
<td>[315,409]</td>
</tr>
<tr>
<td>10</td>
<td>201</td>
<td>[315,400]</td>
</tr>
<tr>
<td>11</td>
<td>202</td>
<td>[315,340]</td>
</tr>
<tr>
<td>12</td>
<td>203</td>
<td>[315,315]</td>
</tr>
<tr>
<td>13</td>
<td>204</td>
<td>[315,245]</td>
</tr>
</tbody>
</table>
Figure 4. Map of library third floor showing shelf range location following data synthesis and transformation.
Table 5. Edited excerpt of Interlibrary Loan borrowing transaction log.

<table>
<thead>
<tr>
<th>Request Type</th>
<th>Transaction Status</th>
<th>Transaction Date</th>
<th>Status</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Article</td>
<td>Request Finished</td>
<td>10/14/16</td>
<td>Staff</td>
<td>NOAA Hollings Marine Laboratory</td>
</tr>
<tr>
<td>Article</td>
<td>Request Finished</td>
<td>10/13/16</td>
<td>Faculty</td>
<td>Library</td>
</tr>
<tr>
<td>Article</td>
<td>Delivered to Web</td>
<td>10/25/16</td>
<td>Faculty</td>
<td>GPMB/Grice Marine Laboratory</td>
</tr>
<tr>
<td>Article</td>
<td>Request Finished</td>
<td>10/29/16</td>
<td>Faculty</td>
<td>NOAA CCEHBR Charleston</td>
</tr>
<tr>
<td>Article</td>
<td>Request Finished</td>
<td>10/8/16</td>
<td>Faculty</td>
<td>GPMB/Grice Marine Laboratory</td>
</tr>
<tr>
<td>Article</td>
<td>Request Finished</td>
<td>10/29/16</td>
<td>Faculty</td>
<td>Library</td>
</tr>
<tr>
<td>Article</td>
<td>Request Finished</td>
<td>10/24/16</td>
<td>Faculty</td>
<td>NOAA CCEHBR Charleston</td>
</tr>
<tr>
<td>Loan</td>
<td>Cancelled by ILL Staff</td>
<td>10/12/16</td>
<td>Faculty</td>
<td>Library</td>
</tr>
<tr>
<td>Article</td>
<td>Request Finished</td>
<td>10/24/16</td>
<td>Faculty</td>
<td>NOAA CCEHBR Charleston</td>
</tr>
<tr>
<td>Article</td>
<td>Request Finished</td>
<td>10/26/16</td>
<td>Staff</td>
<td>NOAA CCEHBR Oxford</td>
</tr>
<tr>
<td>Article</td>
<td>Delivered to Web</td>
<td>10/27/16</td>
<td>Faculty</td>
<td>GPMB/Grice Marine Laboratory</td>
</tr>
<tr>
<td>Article</td>
<td>Request Finished</td>
<td>10/24/16</td>
<td>Faculty</td>
<td>NOAA CCEHBR Charleston</td>
</tr>
</tbody>
</table>

Table 6. Interlibrary Loan borrowing summary report following data transformation.

<table>
<thead>
<tr>
<th>Agency</th>
<th>Oct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requested Book</td>
<td></td>
</tr>
<tr>
<td>CoFC 1</td>
<td></td>
</tr>
<tr>
<td>SCDNR 0</td>
<td></td>
</tr>
<tr>
<td>NOAA 0</td>
<td></td>
</tr>
<tr>
<td>NOAA OCM 0</td>
<td></td>
</tr>
<tr>
<td>NIST 0</td>
<td></td>
</tr>
<tr>
<td>Article</td>
<td></td>
</tr>
<tr>
<td>CoFC 10</td>
<td></td>
</tr>
<tr>
<td>SCDNR 0</td>
<td></td>
</tr>
<tr>
<td>NOAA 12</td>
<td></td>
</tr>
<tr>
<td>NOAA OCM 0</td>
<td></td>
</tr>
<tr>
<td>NIST 0</td>
<td></td>
</tr>
<tr>
<td>Received Book</td>
<td></td>
</tr>
<tr>
<td>CoFC 0</td>
<td></td>
</tr>
<tr>
<td>SCDNR 0</td>
<td></td>
</tr>
<tr>
<td>NOAA 0</td>
<td></td>
</tr>
<tr>
<td>NOAA OCM 0</td>
<td></td>
</tr>
<tr>
<td>NIST 0</td>
<td></td>
</tr>
<tr>
<td>Article</td>
<td></td>
</tr>
<tr>
<td>CoFC 7</td>
<td></td>
</tr>
<tr>
<td>SCDNR 0</td>
<td></td>
</tr>
<tr>
<td>NOAA 10</td>
<td></td>
</tr>
<tr>
<td>NOAA OCM 0</td>
<td></td>
</tr>
<tr>
<td>NIST 0</td>
<td></td>
</tr>
</tbody>
</table>
Figure 5. Output from OCLC xISSN API.

Table 7. Sample of raw ISSN data from each of two CSV files.

<table>
<thead>
<tr>
<th>ISSN</th>
<th>Local Holding ISSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>02652048</td>
<td>ISSN: 1387-974X</td>
</tr>
<tr>
<td>02506807</td>
<td>ISSN: 1110-662X</td>
</tr>
<tr>
<td>16101928</td>
<td>ISSN: 0022-3131</td>
</tr>
<tr>
<td>13684302</td>
<td>ISSN: 1103-8128</td>
</tr>
<tr>
<td>07415214</td>
<td>ISSN: 0257-9731</td>
</tr>
<tr>
<td>1721727X</td>
<td>ISSN: 0368-492X</td>
</tr>
<tr>
<td>05874254</td>
<td>ISSN: 1383-469X</td>
</tr>
<tr>
<td>00014575</td>
<td>ISSN: 00297828</td>
</tr>
<tr>
<td>17588251</td>
<td>ISSN: 0015-7120</td>
</tr>
<tr>
<td>00218456</td>
<td>ISSN: 0742-4787</td>
</tr>
</tbody>
</table>
Table 8. Sample of the final ISSN-L and title data in Excel.

<table>
<thead>
<tr>
<th>Linking ISSN-L</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1523-3790</td>
<td>Current oncology reports</td>
</tr>
<tr>
<td>1096-6218</td>
<td>Journal of palliative medicine</td>
</tr>
<tr>
<td>0014-4797</td>
<td>Experimental agriculture</td>
</tr>
<tr>
<td>0102-8650</td>
<td>Acta cirúrgica brasileira / Sociedade Brasileira para Desenvolvimento Pesquisa em Cirurgia</td>
</tr>
<tr>
<td>1103-8128</td>
<td>Scandinavian journal of occupational therapy</td>
</tr>
<tr>
<td>1387-974X</td>
<td>Photonic network communications</td>
</tr>
<tr>
<td>1610-8167</td>
<td>Urban forestry &amp; urban greening</td>
</tr>
<tr>
<td>0737-3937</td>
<td>Drying technology</td>
</tr>
<tr>
<td>1368-4221</td>
<td>The econometrics journal</td>
</tr>
<tr>
<td>1742-4755</td>
<td>Reproductive health RH</td>
</tr>
</tbody>
</table>
Head in the Clouds: Will a Next-Generation Library Management System Bring Clear Vision?

Denise M. Branch, Head, Continuing Resources, Virginia Commonwealth University

Who are we, and what are we seeking?

Virginia Commonwealth University is a large urban research institution located in Richmond, VA, whose library collections include almost 3 million volumes, 61,000 serials, over 1 million e-books, and nationally prominent collections in the health and biological sciences, social work, the arts, the history of medicine, and comic and book arts.

VCU Libraries was looking for a management system that would allow integration, flexibility, and interoperability, especially in the handling of e-resources. It wanted to break away from the stringent functionality of the traditional integrated library system (ILS) and be transported into a dynamic new system that would support the university’s mission of advancing knowledge and ensuring student success. It desired increased efficiencies, streamlined workflows, automated processes, and engaged users.

The libraries’ legacy ILS, Aleph, was 10 years old and needed to be replaced. Aleph was nearing the end of its life as a system under continuing enhancement, and expensive investments were anticipated if it were to remain viable. After researching the ILS marketplace and finding a next-generation management system that was a good fit, the libraries moved forward with implementation. A number of Ex Libris products had already been purchased, including the discovery tool Primo, and the library felt Alma would complement its existing systems.

VCU Libraries became an Alma “early adopter” and completed a fast-track implementation in six months from April 2012 up until “go live” in October 2012. On October 24, 2016, the libraries celebrated four years with Alma.

Cloud Climate

The implementation journey involved some hiccups but was mostly successful. Alma was a brand new system that was actively being developed when VCU Libraries decided to implement. Since Alma was a new system, it wasn’t exactly clear as to what information would migrate from Aleph and how it would be presented in Alma.

Overall, the migration from Aleph to Alma went as planned. Some migration issues were resolved quickly, and others have continued to haunt the libraries for years. For example, e-serials order records migrated to Alma as print order records. Ex Libris worked collaboratively with the library to identify those records and was able to switch them to electronic orders. Unfortunately, some records were not identified at the time and had to be manually corrected over the years as discovered.

Course reserves migrated with lots of duplicates. Cataloging had to suppress everything that was a course reserve and then identify the active course reserves so they could be unsuppressed. The library was plagued with course reserves cleanup for several years after migration.

When first going live, a lot of the staff felt fearful and somewhat disoriented but also excited. Here was a powerful new cloud-based system that could consolidate disparate systems, simplify user interactions, optimize workflows, and push out analytics. Staff were still in the throes of learning new workflows, becoming acquainted with Alma language and completing the processing of materials that had been put on hold before migration. Training sessions were held, functionality was tested, and documentation was created. The new system facilitated collaboration as staff worked together to develop workflow solutions.

Circulation

There is always the possibility of losing data with any migration, and Circulation was wary of losing item histories. Since there was uncertainty about some of the data migration, a decision was made to save item histories in a spreadsheet before migration. It was good that this data was captured because it did not migrate to Alma. Ex Libris has now resolved item
Course reserves have been a problem ever since migration. Circulation was hoping Alma would have this functionality. The functionality was there, but it wasn’t performing as expected. This was a big disappointment. To address the functionality issue, the libraries created a homegrown online form that interacts with the discovery tool, Primo. Course reserves seemed to be an afterthought on Ex Libris’ part, but they did eventually develop a cloud-based service for managing course reading lists in 2015 called Leganto. However, the libraries decided to keep the methodology that it had developed since it seems to work.

Alma works seamlessly to simplify user interactions, enabling users to locate resources and place loan requests anytime and anywhere. They can check their accounts and view requests through the integrated discovery system Primo with their mobile devices. User interaction is simplified, and Alma is a single source for users to find resources.

Circulation staff are comfortable with fulfillment functionalities and have witnessed many efficiencies. Alma has retained many traditional circulation functions that staff are comfortable using. There was a learning curve, but once accustomed to the new environment, it became easy to maneuver. Staff can take a mobile device into the stacks and work. Downtime is rare. They are happy with Alma and report that fewer things go wrong.

**Resource Sharing and Delivery**

Implementation of the new system brought high expectations after hearing that Alma could automate routine interlibrary functions and integrate with the interlibrary loan (ILL) system, ILLiad. Staff were excited in discovering that some manual functions could be eliminated, and there are points of integration in ILL borrowing and lending, but not as much as expected. Our Resource Sharing and Delivery Department (RSD) has been mostly disappointed in Alma as a resource sharing tool.

The resource sharing functionality of Alma was available when going live but was not fully developed. The libraries decided not to implement at the time. Two years after implementation and developments, the libraries were able to take advantage of the integration between Alma and ILLiad. However, it discovered the two systems did not work well together. Alma was not always successful in filling requests. For example, if a distance user lived in Kalamazoo and submitted a request through Alma, there was no way for staff to know that the requested item needed to be shipped to Kalamazoo. This lack in functionality caused frustration in both staff and users. Because of this, staff have decided to continue to process requests through ILLiad.

Alma manages resource sharing in a variety of ways, including e-mail, ISO ILL, British Library ARTEMail, and NISO Circulation Interchange Protocol (NCIP). VCU Libraries uses the NCIP add-on, which covers borrowing and lending. This add-on supports the integration between Alma and ILLiad. When an item is borrowed, the add-on creates an NCIP request with the user’s ID and a brief bib record. When an item is lent, the add-on verifies the barcode match and moves the item from the home location to the resource sharing location in Alma. The add-on also updates Primo to show the item is unavailable. When the item is returned, the add-on will send a message to Alma where the status becomes “in transit.” It will have to be scanned in Alma to return to its home location.

Users are comfortable with the information that is presented to them in the integration. When an item is checked out, it displays as a loan in the user’s account with a due date. The user can quickly check his or her account to see these checkouts.

RSD is satisfied with borrowing and lending functions for now with the support of the add-on, but workflows are convoluted. It is hoping to see more efficiencies as Alma continues to be developed.

**Cataloging**

Cataloging staff were anticipating two major areas of efficiencies in Alma. The first one was in e-book management. Titles in e-book collections are constantly being added and dropped. Before Alma, staff had to constantly check for new records, delete titles that had left the collection, and load new records. This process was time consuming and tedious. Alma allows staff to activate collections in the community zone (CZ) and work on other tasks,
while Ex Libris manages titles that are added or deleted from subscribed collections. Providers don’t always provide the timeliest and best quality data. There are still things that fall through the cracks, and staff still feel obligated to spot check collections. However, Ex Libris has been very responsive in working with providers to keep the knowledge base up-to-date.

The other area of anticipated efficiencies was in the centralized authority records and processes. Alma is more efficient in this area than Aleph. The process was all manual and very time consuming in Aleph. A lot of searching, loading, and correcting was necessary. With Alma, the staff no longer have to maintain local authority records. CZ authority records are maintained centrally by Ex Libris and are updated frequently. Alma connects authority records to bibliographic headings automatically and corrects bibliographic headings to match the authority preferred headings. There were some minor glitches when we first implemented this heading correction functionality, but after Ex Libris refined the preferred term correction processes, these corrections have become much more accurate and reliable.

Inefficiencies are encountered in the workflow of new print books. After new books are received by the Acquisitions staff and sent to Cataloging, they don’t get processed in one sequence. There is a sorting process in Cataloging. Students sort the books in two ways: Books requiring easy processing and those requiring complex cataloging. The students made these determinations by viewing the machine-readable cataloging (MARC) record for each book and looking for several distinguishing characteristics. With the availability of locally customized brief levels in Alma, the student sorting process has become much simpler and quicker. The student simply looks for the value of the brief level and acts accordingly, but the fact remains that someone still has to touch every book after receipt and look at every bib record to determine which ones can get processed the fastest. It would be even more efficient if, at the point of receipt, staff could easily see the brief level of each book and determine which stream the book should go in.

Alma has functionality for making global changes to records, but some areas are more labor intensive than Aleph. Staff can identify the records that need a change, save them as a set, create a norm rule, and run a job to apply the change to all the records in the set. However, if it’s a one-time correction, this approach may be more trouble than it’s worth. In this case, we have found it just as effective to identify the records, save them as a set, export the set out of Alma, open the file, and correct the records using a third-party application such as MARCEdit, and then re-import them back into Alma.

**Serials**

Workflow analyses performed over the years had identified e-serial workflow inefficiencies. E-serial management involved managing data in multiple systems before Alma producing information silos and ineffective communication. It required two systems, Aleph and SFX, to maintain e-serials. Someone had to remember to update data in both systems. Sometimes this happened, and sometimes it didn’t. When it didn’t happen, users were exposed to outdated information. Staff were excited to learn about the Alma task-oriented workflows that would deliver a unified management environment.

Manual processing has been reduced by Alma’s capability for “pushing” tasks. When an e-resource is ordered, it is automatically pushed to an unassigned task list where it awaits activation. The library can define its rules for workflows. Once set, tasks are automated, and staff only have to handle workflows that are exceptions. For example, when an e-resource is ordered, it will automatically be pushed to an unassigned task list where it waits activation. There is no need to worry about determining which titles have been ordered or if they need to be activated. It’s all clearly displayed in the task list. Upon activation, it can be deleted from the task list.

Tasks can be easily assigned electronically for completion of processing instead of manually filling out paper routing forms, which had formerly been the routine. When a task is assigned, it appears in the individual’s personal task list, allowing quick access to the assigned task and determination of the work that needs to be performed. Every screen in Alma has a persistent link to the task list, ensuring that workflows are focused on the task at hand. Once the task has been completed, it can be deleted from the task list or assigned to another staff member for further processing.

Serials patterns were not available in Alma when VCU first went live but became available about a
year ago. However, the Serials Unit made the decision not to create patterns. The staff figured out how to manipulate Alma to correctly expect issues by using dates and felt that it would be a waste of time to create patterns for print serials in response to the libraries’ policy to switch to electronic subscriptions whenever available. Staff can focus on other priorities that warrant their attention.

Alma eliminates silos so that updates made in Alma are automatically updated in Primo. The two systems work well together most of the time. E-resources are activated in Alma and can display in Primo on the same day. However, there are occasions when resources activated in Alma do not show up in Primo. This is when staff have to call on Ex Libris support to get resources to display in Primo.

Acquisitions

Fiscal year rollovers have been a dreadful task for Acquisitions for many years. Lots of time-consuming steps were wrapped up in the rollover process in Aleph. Processes could be put on hold for a week while a rollover took place. Funds had to be checked and updated, and a multitude of cleanup was involved. Now annual rollovers take approximately 15 minutes in Alma. A test rollover is performed to catch errors. Once the errors are corrected, the live rollover is performed. Processing holds have been eliminated, and staff are elated that it’s so easy.

Alma is changing the way staff interact with data. Analytics could not be easily extracted and compared in Aleph. The analytics tool in Alma is very useful and plays an important role in the libraries’ decision making as it attempts to bring together appropriate data. Out-of-the-box reports are provided, and staff can create their own reports. A dashboard is available for displaying customized reports, and this robust tool allows reports to be run on library activities and graphs to be added to the analysis. Analytics can be compiled that are as simple as showing a comparison or as complex as revealing a trend.

The order staff enjoy the simplicity of initiating orders for print and e-resources within or outside of Alma. Orders within Alma can be created in the institution zone or the community zone. Staff can go to the metadata editor to import records from an external vendor and quickly place an order, and once an order is placed, the order workflow will commence in Alma.

Staff don’t have to concentrate heavily on data entry but can focus on decision-making issues that require more thoughtful consideration. The library can define how it wants the order workflow to progress and the degree of automation. For example, staff can choose to have invoices that come through the electronic data interchange (EDI) process to be automatically approved without staff intervention or have staff review and approve at specific points along the way.

Digital Technologies/Systems

The Digital Technologies Department (DigTech) was anticipating two major efficiencies in the cloud. One was the release from server upkeep. Before implementing Alma, the applications administrators had to manage three different points of upgrades: (1) The OS. Running SunOS updates is complicated, and at the time, the libraries were contemplating a migration to Linux (a different operating system) before deciding to move to Alma; (2) The database. There was a database administrator in addition to the OS administrator because of the complexities of Oracle; (3) The application Aleph. The last Aleph upgrade involved changes to all three components. Managing the upgrade was a full-time job for the better part of three months. After Alma implementation, fewer administrators were needed to monitor servers, and Ex Libris takes care of the upgrades in the cloud. The applications administrators can focus on developing other applications to enhance user experience and not be bogged down by server maintenance.

Clients had to be constantly managed in Aleph with every upgrade. Back office customizations in Aleph required five different client versions to be installed and maintained with every upgrade. DigTech had to determine which version each staff member should have installed on his or her computer. The burden of installing clients has now been removed in the cloud. Alma doesn’t have all these different versions, and all libraries are on the same version and are upgraded at the same time.

DigTech encountered difficulties when it had to assist the Circulation Department in setting up policies in Alma. Aleph was relatively flexible in
allowing a broader base of policies. The policies had been configured for material type, such as book, DVD and journal. Alma’s policies are not as flexible and are based on location. It was hard to conceptualize a policy shift from material type to location during implementation. If Alma’s location policies were used, the library would have to determine what location to use. However, there was an option to take a carbon copy of the Aleph policies and make them fit into Alma. After discussion, the library decided to create a hybrid between the two different sets of policies. This seemed to work well since we did not copy the items where the locations had several material types and circulated differently.

DigTech found it difficult to let go of control over some things. Staff liked the idea of knowing that data was within their control. However, they realized they must relinquish this control to Ex Libris and trust that our success is built in with Ex Libris’ success. Neither party wants to fail but wants to have the best cloud system possible.

Moving Forward

The VCU Libraries is committed to broadening research capabilities in the cloud. It is looking for ways to facilitate collaboration among faculty, staff, and departments and partner with other institutions to enhance research and discovery.

Here are some ways in which the VCU Libraries plans to move forward in the cloud:

• Seek further points of integration, such as ILL, bursar, and altmetrics.
• Explore collaborative collection management with other libraries.
• Push Ex Libris to develop better integration with ILLiad and DOCLINE.
• Review internal policies to determine barriers of integration.
• Determine methods to expose rich level of data that brings different paths for information gathering.
• Get a better understanding of how data can be used to support university’s scholarship and learning.
• Investigate opportunities for new formats such as linked data and BIBFRAME.
• Investigate collaborative opportunities with other departments within the library.

Conclusion

A number of the VCU Libraries’ objectives have been fulfilled in the cloud. Alma was the right decision, and staff have adapted well to the new workflows. The chains of maintaining servers and clients have been broken. It’s not a perfect system, but there is potential for creating efficiencies. It has saved money and time on hardware and software, and we have witnessed a net decrease in operating costs. There wasn’t as much change in the way of organizational structure as expected, but Alma is still being actively developed, and there will be more opportunities for transformation.
A Tale of Two Campuses: Open Educational Resources in Florida and California Academic Institutions

Alejandra Nann, Electronic Resources and Serials Librarian, Copley Library, University of San Diego
Julia Hess, Head of Metadata and Copy Cataloging, Ball State Libraries, Ball State University
Sarah Norris, Scholarly Communication Librarian, John C. Hitt Library, University of Central Florida
John Raible, Instructional Designer, Center for Distributed Learning, University of Central Florida

Abstract

Open educational resources (OER) provide a high-quality and low-cost alternative to traditional textbooks. The University of Central Florida (UCF) and the University of San Diego (USD) have been engaged in a multitude of efforts related to OER and textbook affordability. This article will discuss the textbook affordability climate at the state (Florida and California) and institutional (UCF and USD) level. Macro and microventures and lessons learned will be shared by both institutions ranging from perceptions of open education resources by the universities to collaborating with constituents across campus, in addition to specific case studies with UCF faculty teaching online and face-to-face courses as well as USD’s stipend program. Lastly, the article will discuss future developments and continuous improvements by educating UCF and USD campus communities through several initiatives and new partnerships with stakeholders.

OER at the State Level

Florida

The State of Florida began its textbook affordability efforts in 2008 with the passage of Florida Statute 1004.085. The direct student benefit of this law requires each public higher education institution to post a public list of required textbooks at least 30 days before each semester begins (Textbook Affordability Act, 2008). The spirit of the law is to allow the purchasing required materials through other, possibly cheaper, retailers.

Raible and deNoyelles (2015, p.6) conducted an analysis of bookstore contracts in the State University System of Florida. They found two universities whose bookstore contract directly addressed textbook affordability beyond providing rental programs. One university adopted specific textbook affordability language allowing the university to pursue alternative content delivery methods without the bookstore’s permission or involvement. The university cited Florida Statute 1004.085 as the rationale for this contract language.

The national conversation about textbook affordability continued after the passage of the 2008 textbook affordability law. In the 2015–2016 Florida legislative session, various updates were passed. Major changes included the extension to 45 days from 30 days for institutions to publish required course materials. The term instructional materials was introduced to broaden the scope of the law. Open educational resources (OER) are encouraged to be adopted in general education courses. Perhaps the biggest change is requiring each institution to document annually report textbook and instructional material affordability efforts (Textbook Affordability Act, 2016).

California

California tackled the textbook affordability issue by focusing specifically on open education resource (OER) initiatives. In 2008, the state enacted its first OER law, authorizing the Board of Governors of the California Community Colleges (CCC) “to establish a pilot program to provide faculty and staff from community college districts around the state with the information, methods, and instructional materials to establish open education resources centers” (Wiley, 2008).

In 2012, the state legislature directed that the state’s public higher education systems develop an OER
digital library, and in 2013, they created and funded the California OER Council. The California OER Council, run by members of the California Community Colleges (CCC), California State University (CSU), and the University of California (UC), was charged with implementing a variety of state OER programs (California Open Educational Resources Council, 2011). CSU, already experienced in managing such efforts, was directed to lead the establishment of the California Digital Open Source Library, seek private funds, and administer the matching funds by the state (About, 2015).

California continues to implement new initiatives at the state level. The 2016–2017 California state budget, for example, includes a one-time $5 million fund to provide a competitive grant for community colleges. Grants will provide institutions with up to $200,000 for each zero-textbook-cost associate’s degree or certificate, allowing students to enroll in a program guaranteed that they will not have to buy expensive textbooks (Lesko, 2016).

OER at the Institutional Level

Florida

At the University of Central Florida, no institutional initiatives exist to promote textbook affordability. A working textbook affordability group was established by the John C. Hitt Library and the Center for Distributed Learning. The goals of this working group is to create a campus environment favorable to textbook affordability and systematically promote high-quality, ADA-compliant, open educational resources, and library-sourced content to reduce student costs.

The working group’s goals are limited to OER and library-sourced materials due to restrictions of the institution’s bookstore contract. The current contract’s language includes an exclusivity clause stating the bookstore is the “exclusive seller of required, recommended, suggested, course packs, no exceptions” (University of Central Florida, 2003). The bookstore contract expires September 30, 2017, and the working group has provided value input to the institution for including textbook affordability language and practices for the next contract term.

Case Studies. OER efforts on UCF’s campus can best be illustrated be several case studies, which highlight the various types of activities related to open educational resources in the classroom. These activities are categorized in three distinct ways. These include:

- Determining if present materials used in the classroom are openly accessible through other avenues;
- Offering open and/or library-sourced materials as an optional replacement for an existing textbook; and
- Adopting an open and/or library-sourced book as a required textbook.

The first case study explores determining if present materials used in the classroom are openly accessible through other avenues. In this particular instance, a lecturer teaching an English literature course utilized public domain and/or creative commons licensed materials. The lecturer’s required texts focused on works from the medieval period through the late 18th century, much of which existed in the public domain. The lecturer initially became interested in the project after a UCF Libraries subject librarian reached out to discuss open alternatives to their required text. Assessing the required reading, conducting an analysis, and ultimately creating an OER anthology for the course was a collaborative effort that included the library’s Office of Scholarly Communication, subject librarian, and instructional designers. Each played an important role in vetting the content for appropriateness and copyright compliance. With an ePub version of required readings available free of charge, the only potential out-of-pocket expense to the student was a text purchase for one work still protected by copyright and available at a nominal fee. Despite having to purchase a text, the course has still seen significant savings to the student.

The second case study explores offering open and/or library-sourced materials as an optional replacement for an existing textbook. In this case, a microeconomics professor had been utilizing an OpenStax open textbook as an alternative text for their course. The faculty member utilized this as a free alternative for his students and did not seek library and/or CDL intervention for implementation; in fact, an instructional designer discovered that the faculty member was using the OpenStax book while researching high-quality open textbooks. The working group members conducted a survey of the faculty member’s students in the spring of 2016 and
the summer of 2017 to survey which book(s) they
used and their perceptions of OER content, all of
which were received positively.

The final case study explores adopting an open
and/or library-sourced book as a required textbook.
In this particular case, an American history lecturer
attended a campus presentation on OER given by
the working group. This prompted interest in
OpenStax as an alternative to the currently offered
textbook for their course. Working closely with the
instructional designers, the faculty member
ultimately decided to utilize the OpenStax book as
the primary required textbook for their course.
Like the previous case studies, a survey was
distributed to students, with positive feedback from
students.

Overall, each of these courses saw significant
student savings, ranging from $1,800 a semester to
over $200,000 per semester—with additional
significant savings over time should these OER
materials continue to be used by the faculty
members and students. Additionally, both students
and faculty have indicated positive feedback on
surveys administered to each course.

California

During the 2014–2015 academic year, some librarians
formed a small working group with the goal of gauging
interest in OER on campus through surveys, focus
groups, and workshops. These efforts were less
successful than hoped, so the working group explored
an alternative: Financial incentives. Following an
application process, four faculty were selected to
receive $1,000 stipends in exchange for attempting to
replace a textbook in one course with OER during the
2015–2016 academic year and write a two- to three-
page report by the end of the following summer.
Members of the group met individually with the
participants over the summer to introduce them to the
project, and they were aware that they could contact
the librarians at any time with questions, but
otherwise the librarians did not regularly communicate
with the faculty throughout the academic year. An
analysis at the end of the year showed that the pilot
program had saved 118 students approximately
$12,000 in textbook costs.

At the end of the first year, the working group took
time during the spring of 2016 to evaluate lessons
from faculty feedback and their own observation.

There were three key notes made during the
evaluation process. First, the librarians had assumed
that if professors had any questions, they would reach
out, but this turned out not to be the case; in some
cases, they just gave up. Second, because the
university’s fiscal year runs July to June, reports need
to be turned in by early June. That allows faculty to
receive their stipends in the correct fiscal year. Lastly,
the librarians should begin contacting other
stakeholders on campus (e.g., the bookstore and
student groups) to come up with potential avenues for
collaboration, offer a smoother transition to faculty
implementing OER, and increase promotional efforts.

Promotion and marketing were not key strengths
during the first year pilot. E-mail blasts to faculty
work on occasion, but the librarians wanted to get
more creative with promotional activities. One
librarian from the OER working group created an
OER poster for Copley Library’s Salon in 2016. The
Salon is a poster session where faculty are invited to
learn about resources and services the library offers.
She spoke to a number of faculty about OER and
Copley Library’s OER initiative. Another event the
librarians participated in was the New Faculty
Reception that is hosted annually by Copley Library.
The librarians requested a time slot to introduce
new faculty to the Copley Library OER initiative and
ways faculty can get involved.

Since more funding was provided for the 2016–2017
academic school year, the librarians sent out a call
for proposals for faculty to participate in the OER
initiative. The librarians created a rubric to evaluate
applications fairly. The rubric consisted of three
components:

- Cost savings: How much money will the
  faculty member save the students by
  replacing the textbook with OER?
- Feasibility: Can their project be reasonably
  accomplished during the academic year?
- Impact: How will it benefit their students
  and is it sustainable?

During the course of the initiative’s first year, the
Dean spoke with several stakeholders including the
Provost, Dean’s Counsel, and board meetings
regarding the OER initiative. Additionally, the
librarians met with the Textbook Manager from the
university bookstore to discuss a potential
partnership. Through that meeting, they were able to come up with an alternative to students and faculty who prefer to study with print material. If a stipend recipient implements OER that fall under the public domain or creative commons license that allows for distribution, the bookstore will work with the university print shop to print and bind the course material, similar to a course pack, and sell it for a nominal fee.

In the 2016–2017 academic year, the librarians were able to accept nine faculty members to join the Copley OER initiative and replace a textbook with OER. The librarians hosted two group meetings during the summer as an opportunity to introduce the faculty to OER and gave them an idea of what to expect from the program. Additionally, the librarians are offering two nonrequired meetings per semester as an opportunity for the nine faculty to meet and ask questions, discuss successes and challenges, and provide information about their experiences with OER. Lastly, the librarians email the faculty members monthly to check-in and offer one-on-one meetings in case a stipend recipient needs additional help locating the right OER for their class.

Although the current OER initiative stipend program is doing well, the librarians continue to explore other options to raise awareness and educate faculty on OER. After receiving more funding, the librarians launched Copley Library Open Textbook Review. The program offers a $250 stipend to faculty who find a suitable textbook in their field and write a short review. A call for proposals was sent out at the beginning of the academic year, and the librarians were able to accept 23 faculty applicants into the program. Finally, Copley Library is creating a committee beginning in the spring of 2017. The charge of the committee will be to discuss current news and events related to OER, educate the campus community on OER, and continue creating innovative ways to improve the OER initiative.

Conclusion

As institutions of various sizes continue to explore the impact of OER and textbook affordability on their campus, it is important for constituents to identify and focus on ways to educate members of the community on the importance of OER and how it can largely benefit students. Integrating the use of OER can range from presenting workshops to members of the university community to creating initiatives that offer stipends to faculty who implement OER or create OER for their class.

References


University of Central Florida. (2003). Agreement for Bookstore Management and Related Services between The University of Central Florida Board of Trustees and Barnes & Noble College Bookstores, Inc. [Contract]. Orlando, FL.

Finding the Right Fit for Article Delivery: Using Resource Sharing Technology to Provide Enhanced Access

Shannon Pritting, SUNY Polytechnic Institute

Abstract

Electronic subscriptions occupy an ever increasing majority of budgets and prevent libraries from expanding services in other areas. There are few attractive options for libraries who want to provide access to research materials to users in a way that is cost effective and simple for users. Libraries are caught between subscriptions to single journals, large packages, or article-level purchasing that is either not instant or must allow access to everyone, which can quickly become costly. The IDS Article Gateway platform, developed by the IDS Project and SUNY Polytechnic Institute Library, uses resource-sharing technology and workflows to deliver fast or instant access to research material to users in a way that involves little or no staff time and removes as many barriers to user access as possible. Where resource sharing has typically sought to deliver articles in one to two days, libraries using Article Gateway deliver articles either within a few minutes or a few hours.

Introduction

In the past, resource sharing and interlibrary loan (ILL) have helped users gain access to research material far beyond what their libraries could afford as a single institution. Going forward, resource sharing can offer many options for libraries to provide users instant access to research material in a way that is convenient yet cost effective, allowing users to have expanded access without the need to wait a day or more for access. The Article Gateway application integrates with various Web services, providing IDS Project libraries with more options for access to research materials while also removing the need for staff to review requests.

Using IDS-developed resource-sharing technology as the foundation, the IDS Article Gateway can automate verification of copyright and licensing for the requested article, determine the best price for purchase from article vendors, and can apply user profiles and system configuration that will allow more refined instant purchasing of research articles. The Article Gateway platform opens up another method for libraries to meet the information needs of its researchers and students and helps to ensure that content is delivered in the most seamless and cost-effective manner. IDS Article Gateway fully automates resource-sharing article requests and allows for automatic borrowing of articles from other libraries or purchasing articles from document providers. Article Gateway also standardizes and completes citations to ensure that the data used for the automated decision is accurate and that staff time is not needed to correct request information.

Literature Review

The development of modern library systems allows connections through applications or middleware that can provide increased options for patron access and enhance library services. Article Gateway utilizes IDS Logic, which is a recently developed middleware program that connects various library software systems and can serve to apply complex rules based configuration. In the past decade, there are many examples of individual libraries using application programming interfaces (APIs) or Web services to address issues, streamline work, or enhance library functions. Additionally, library vendors are beginning to include access to data within systems through APIs and allow for connection to systems via Web services and APIs. However, beyond developer networks such as Online Computer Library Center’s (OCLC) developer’s network (https://www.oclc.org/developer/home.en.html) or the Ex Libris Developer network (https://developers.exlibrisgroup.com/), which are meant for sharing ideas and code, there is not a large community based on technology integration and development. The IDS Project brings together a community with ideas and strategies about how to improve libraries and connects them with a platform and development expertise to integrate systems and foster innovation.

Recent examples of resource-sharing system integration through application and software development reveal how much effect software solutions can have on library functions. For example, in 2011, Wayne State University created an application that connected data from its two
systems, ILLiad and ArticleReach, and submitted orders to the Copyright Clearance Center (CCC) API, saving over 500 staff hours per year spent paying royalties (Sharpe & Gallagher, 2011, p. 137). Services such as CCC and the global library cooperative OCLC are ripe for integration, and the positive effect in saved time is evident, even with applications that are limited in scope.

As resource sharing and library cooperation expands, especially outside of North American libraries, there will be an increased need to connect disparate resource sharing and library management systems, as there are different systems used by international libraries. As OCLC has expanded resource sharing in Spain and other countries, Rodriguez-Gairin and Somoza-Fernandez (2014, p. 487) identified a need to connect OCLC’s WorldShare Management platform with the GTBib-SOD interlibrary loan system already in use in Spain. Further, the solution identified by Rodriguez-Gairin and Somoza-Fernandez (2014, p. 487) suggests using web services and APIs to connect the two resource-sharing systems and remove the need for duplicate work in systems.

The software platform that allowed the most customization and optimization has been ILLiad, developed by Atlas Systems, and supported by OCLC. In many cases, ILLiad has been extended beyond its core purpose of being a complex hub for resource sharing into a core system that libraries have integrated into many areas and departments. With the experience developed by many libraries and the IDS Project in integrating library services platform components, OCLC services, ILLiad, and other software systems, the potential is great for further development.

As the IDS Project developed a community of talented librarians and staff, and systems matured to become more open to integration, there is now the ability to connect “mission critical” systems that will “support better, more informed decisions and free employees to undertake higher-value tasks,” which will ultimately “offer the capability to unlock talent and time” (Oberlander, 2012, p.15). The promise of freeing time and talent through improved systems was at the core of many of the software projects that have come from the IDS Project, which has resulted in staff who have more time for professional development, are more engaged with an innovative community, and can contribute more to their individual libraries and the IDS Project community.

The development of resource-sharing systems has automated many tasks in interlibrary loan work, but requests from patrons to borrow an article from another library or purchase from document delivery providers have not been fully automated due to a variety of obstacles. One obstacle is that even though OpenURL linking systems have been commonplace for several years, users still often manually enter requests, and sometimes data is not correctly imported through the link resolver. In Leykam’s (2013, p. 106) study of four years’ worth of ILL borrowing requests, he found that 59% of requests were manually entered rather than through the library’s link resolver, SFX. Surprisingly, students manually entered ILL requests more often than faculty and did not use the direct linking capabilities of the OpenURL resolver (Leykam, 2013, p. 110). In addition to incomplete or inaccurate citations due to OpenURL mapping or manually entered information, the issue of checking copyright status of the request is another barrier that typically requires staff to review a resource-sharing request. The need to have staff review all requests for copyright is even more necessary with increased usage of article purchase-on-demand options, which are often less costly than paying a copyright fee to borrow the article from another library. Heather L. Brown (2012, p. 101), in studying the comparison of copyright payments with purchase on demand of articles, found that over $500 could have been saved in the course of a semester, with purchases being filled in almost half the time of traditional ILL. In addition to article purchasing via interlibrary loan, libraries are also considering whether pay-per-view unmediated services such as CCC’s Get It Now are cost-effective options for delivering access to journals.

Many libraries are exploring multiple models to deliver research material that includes article purchase on demand, subscriptions, and resource sharing. In the case of Loyola Health Sciences Library, the use of a hybrid approach using CCC’s Get It Now to deliver journal content saved the library over $640,000 in related access costs (Hendler & Gudenas, 2016, p. 368). In addition, Hendler and Gudenas see the value of ILL to deliver material in a cost effective and very quick manner (for Loyola under 11 hours), but also acknowledge that article purchasing has its place in the merged world of access and collection management (Hendler & Gudenas, 2016, p. 369). For the users in a health sciences library, “if it is the weekend or if the user wants the article immediately, they might not elect
to use ILL. Get It Now fulfills that need for immediacy and prevents customers going outside of the library to meet their needs” (Hendler & Gudenas, 2016, p. 369). As libraries move toward different models for providing access to research material, there will certainly be a need for more complex configuration of delivery options based on user need and user status.

**IDS Technology Development**

**Methodology**

The development of technology to provide extended access to patrons has always been the focus of the IDS Project community and the interlibrary loan community as a whole. One of the major IDS Project developments was the Getting It System Toolkit (GIST) that allowed staff to "easily route requests between ILL and acquisitions depending on a number of factors, such as user recommendations, the borrowing cost versus the purchase price, regional library holdings, and more" (Pitcher, Bowersox, Oberlander, & Sullivan, p. 224, 2010). Another development focusing on user needs for ease of discoverability across siloed member catalogs was the consortial catalog IDS search, providing users with an "intuitive search experience which enables libraries to easily customize the search interface and add geographic search limits" and a tool allowing for an almost instant submission of ILL requests for items held at regional libraries (Oberlander & Rivenburgh, 2012). The focus of Article Gateway has been to take the approach of GIST and IDS search in which different options for purchasing or borrowing are weighed and firmly connect the role of resource sharing workflows and methods of delivering journal access that is not provided through subscriptions.

As IDS Logic serves as the platform that integrates multiple systems, determining a way to streamline maintenance of electronic holdings information and license information with the resource sharing workflow was necessary. This need led to the development of the Article License Information Availability Service (ALIAS), which is now a component of the IDS Logic platform. ALIAS harvests data regarding electronic holdings that libraries already have in their knowledgebase software, such as EBSCO’s Full Text Finder or Serials Solutions 360 Link, and combines this data for all 100 IDS Project member libraries to provide a basic local article availability lookup and an unmediated request system. Thus, libraries have less maintenance responsibility, benefit from identifying items they own, and are able to send requests only to libraries who can deliver the article they need with extremely high success rates at an accuracy typically over 97%. Rather than create a separate system, ALIAS uses OCLC resource sharing for sending these requests, which keeps more transactions in one familiar workflow.

One major positive of IDS Logic is that, on a nightly basis, all the data for each of the 100 member libraries (except for unique patron information such as name, ID number, and username) is pulled from each library’s ILL system. This data is then aggregated so that reports can quickly be run across the consortia to find areas in ILL workflows that are affecting services. In many cases, the data and analysis provide insights into how much effect specific steps or aspects of ILL workflows have on patron services. In October 2016, using ILLiad data for all 100 IDS Project libraries from January 1, 2016, through October 24, 2016, we found that ILL borrowing requests remained in an ILLiad queue “awaiting copyright clearance” for an average of 8 hours. The awaiting copyright clearance queue is typically designated for requests that need staff review to determine if license fees need to be paid, or whether the CONTU limits have been met. The total number of IDS Project requests analyzed were 228,019, with a total of 1.824 million hours (or 76,000 days) that ILL requests sat waiting for staff to review copyright. Clearly, copyright clearance has a major effect on ILL delivery time.

**Article Gateway**

Article Gateway (AG) features the depth of what is possible with the complex middleware software platform IDS Logic combined with configurable workflows and options. The AG workflow streamlines and automates fixing of citations, checking for copyright clearance and compliance, and, when needed, checking multiple article vendors for best prices, ultimately leading to an unmediated delivery for most requests. To ensure that copyright checking is as accurate as possible, borrowing articles all must have ISSNIs and have fairly consistent citations. To achieve an all ISSN process without forcing staff to open many requests, citation data is sent to the PubMed Web service and to the OCLC Worldcat and xID Web service to harvest ISSNIs,
PMIDs, and other citation information that is then ultimately inserted back into the transaction for a verified and standardized citation. In addition, the date, volume, issue, and other citation information are then run through custom scripts that standardize citation information so that years and other citation data can be clearly compared. Additional queries such as the “rule of 5” query, which checks to see if five requests from the same ISSN have been filled within the past year, is run to prevent the need for staff to review requests. If copyright limitations have been reached, then the Article Gateway platform checks copyright licensing fees, pricing from CCC’s Get It Now service, and Reprints Desk Article Galaxy service. Whichever option offers the best value is then selected by Article Gateway, and the request is fulfilled with no staff intervention or delays. Whether a request with an incomplete citation, no ISSN, or a request for an article where a copyright payment is needed, IDS Logic and Article Gateway work to facilitate almost instant delivery.

The Effect of Article Gateway

Two of the major goals of the Article Gateway platform is to significantly reduce the average turnaround time for ILL articles while also reducing the number of requests that staff need to manually process. One of the major users of the Article Gateway system is the University at Albany which saw major benefits of Article Gateway, including a much improved turnaround time in addition to a decreased need for staff intervention for processing, as outlined in Table 1:

Although SUNY Polytechnic Institute is a much smaller institution than the University at Albany, the effect of Article Gateway on article delivery was significant in improving service for a large percentage of the roughly 2,250 article requests placed from Fiscal Year 2015–2016. Table 2 shows the total percentage of requests at SUNY Polytechnic Institute in Fiscal Year 2015–2016 that were delivered in 4 hours or less, indicating that by having Article Gateway remove obstacles in the ILL borrowing article workflow, ILL can become a near instant option.

Through the case studies at SUNY Polytechnic Institute and University at Albany, the effect of Article Gateway in making ILL a nearly instant option that requires significantly less staff time is clear. With more configuration and by enabling Article Gateway to trigger purchases during certain days or times, ILL can become an increasingly favorable option for patrons to access articles not held by their institution.

Table 1. Effect of Article Gateway.

<table>
<thead>
<tr>
<th>ILL Workflow Issue</th>
<th>Effect of Article Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turnaround time for articles.</td>
<td>Improvement of 16 hours, from 1.58 days in the fall of 2016 (before use of Article Gateway) to .91 days in the spring 2016 (after Article Gateway was implemented).</td>
</tr>
<tr>
<td>Requests with incomplete or citations needing standardized information such as full years.</td>
<td>67% (5,720) of article requests had citations fixed or standardized.</td>
</tr>
<tr>
<td>Number of requests requiring copyright review.</td>
<td>2,637 requests identified as copyright not required and automatically sent. 589 identified as needing copyright payment or article purchase.</td>
</tr>
</tbody>
</table>

Table 2. Article Gateway turnaround time at SUNY Polytechnic Institute.

<table>
<thead>
<tr>
<th>% of Total Requests</th>
<th>Delivery Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Under 10 minutes</td>
</tr>
<tr>
<td>4</td>
<td>15 minutes and under</td>
</tr>
<tr>
<td>10</td>
<td>30 minutes and under</td>
</tr>
<tr>
<td>17</td>
<td>60 minutes and under</td>
</tr>
<tr>
<td>24</td>
<td>2 hours and under</td>
</tr>
<tr>
<td>30</td>
<td>4 hours and under</td>
</tr>
</tbody>
</table>
Through the case studies at SUNY Polytechnic Institute and University at Albany, the effect of Article Gateway in making ILL a nearly instant option that requires significantly less staff time is clear. With more configuration and by enabling Article Gateway to trigger purchases during certain days or times, ILL can become an increasingly favorable option for patrons to access articles not held by their institution.

**Current State of Article Gateway**

As of November 2016, beta testing has been underway for over nine months at three libraries, with 15 more libraries currently in production. Additional libraries are being added weekly, with the goal of having 30 libraries using IDS Article Gateway by the end of calendar year 2016 and 60 libraries using Article Gateway fully functional by the end of Fiscal Year 2016. As more libraries adopt IDS Article Gateway, new features and functionality will be added, such as open access filtering, expanded delivery configurations, and a customizable analytics dashboard.

**References**


Moving From Reclaiming to Reclaimed: The Big Picture and a Case Study of a Trending Initiative

Bobby L. Hollandsworth, Learning Commons and Digital Studio Coordinator, Clemson University

Abstract

Library space has become more and more valuable over time, especially at universities where gate counts increase 3% to 4% each academic year. With circulation numbers and reference questions trending down over the past 10 years, librarians at Clemson University take solace in the ever increasing number of visitors. With this increase in numbers comes overcrowding and insufficient student space. Decisions must be made to alleviate the cramped and inadequate spaces that go along with a 50-year-old library building, the technology of today, and the demands of a savvy student body that will soon be competing in a tight job market. Cooper Library has transformed five spaces over the past six years by repurposing areas designed to hold print reference collections, stacks of bound journals, former staff work areas, and a low technology meeting room. By partnering with campus information technology (IT) on these projects, Cooper Library has brought cutting-edge technology and spaces that equip students with the tools they will need to succeed.

Background

Academic libraries have gone through tremendous change over the past 10 to 15 years trying to transition from print warehouses to a leaner model that relies more on digital access and online resources. Gone are the days of the quiet, sterile, and uncomfortable library as they have morphed into a much more welcoming atmosphere due to the influence of Barnes & Nobel and the “library as place” movement. National trends point to circulation stats and reference interactions that have fallen, but at some academic libraries, patron counts have risen to staggering heights. Logic would dictate that making collections accessible online would also cause a drop in visits, but just the opposite has happened at libraries such as the main library on the campus of Clemson University. Cooper Library turned 50 years old this fall and looks the part of a mid-1960s library, but a closer look reveals pockets of cutting-edge technology, space revitalization, and areas with the potential to grow gate counts even higher. Six years ago, Cooper Library began a transformation by repurposing five little or improperly used areas into spaces that could challenge students to learn, make, explore, and grow.

Why Repurpose?

The act of changing one area or space into something for a different use or intention is a very basic definition of repurposing. The reason for repurposing was simple: The main library was overflowing with people and not meeting the needs of students, and a space conversion was the only viable solution after a renovation plan was not an option due to expense after the 2008 recession. Cooper Library gate counts or visitor counts were the main impetus behind the space improvement and repurposing projects. The library was experiencing tremendous growth in yearly gate counts, which made it harder for students to find space to study, conduct research, and collaborate for group projects. This growth has actually risen with each completed project over the past several years, which magnifies the need to repurpose more spaces in the library as university full-time enrollment has also increased. The table below shows the gate count numbers by fiscal year over the past three years.

Table 1. Cooper library yearly gate counts.

<table>
<thead>
<tr>
<th>Cooper Library Yearly Gate Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>FY 2014</td>
</tr>
<tr>
<td>FY 2015</td>
</tr>
<tr>
<td>FY 2016</td>
</tr>
</tbody>
</table>

Learning Commons

In 2010, the Clemson University Libraries set out to repurpose the fourth floor reference area of Cooper Library, the main library on campus, by converting the space to a learning commons. This conversion had several parts: Relocation of the reference print collection, moving the reference desk to a more central location, removing index tables and other out-of-date furniture, new paint, new carpet, new
furniture, and better signage (see Figure 1, Appendix A). The physical project would start in the spring, with the main thrust happening during the summer. The reference print collection was moved from the east side of the fourth floor to the west side after some weeding. This collection eventually went through an extensive weeding and relocation that reduced 1,620 linear feet of shelving down to two low shelves, now located in the Learning Commons West. This was the first of many collaborative efforts between the library and Clemson Computing and Information Technology (CCIT).

The project was completed in the summer of 2010, just in time for the start of the fall semester. The new furniture arrangement allowed more students to occupy the space of the former reference area. Gone were the 13-high shelves of reference books and the index tables. The new configuration allowed students to physically move furniture to accommodate group projects and group study. This new arrangement also offered nice sight lines from the lobby to the back of the Learning Commons and natural light, something that was missing with the high shelves. The project was a success as students flocked to the new Learning Commons, but limitations were apparent early as student laptop cords plugged into wall outlets blocked walking paths and created trip hazards throughout the area. It was also obvious that there were not enough seats to accommodate the crowds as turnaways became an all too familiar sight during the afternoon and evening hours in the library. During this time, a visit to the Learning Commons at the D.H. Hill Library on the campus of North Carolina State University showed that a denser concentration of seating could be achieved. In 2011, a committee was formed to tackle the issues plaguing the new Learning Commons and come up with suggestions for improvement.

The committee consisted of the Dean of Libraries, one library faculty member, two library staff members, and two undergraduate students. A survey was constructed giving students the opportunity to voice their opinions on what the library should do to improve on the first iteration of the Learning Commons. The survey results confirmed what the committee had already viewed as the major faults of the space that needed attention. Nearly 80% of the survey responses called for more furniture and more power outlets. These were needs that could be met with a Learning Commons phase two implementation.

Phase two of the project started in 2012 with more power and data installed along the walls of the Learning Commons and pushing existing furniture out to the walls. The additional furniture would not only provide more seating for students, but it would also bring power from preexisting columns in the space out to tables and new seating arrangements (see Figure 2, Appendix A). The phase two implementation provided 107 new seats, bringing the seat count in the Learning Commons East to 195 seats. Along with the new furniture, phase two also brought in technological improvements to Cooper Library. Eight iMacs were installed in the Learning Commons, and four dual monitor Mac Pros with Adobe CS6 and Final Cut Pro were set up in the Learning Commons West. This was our first step in the direction for a digital studio in the library. The library also purchased new technology for check out, including 10 digital single-lens reflex (DSLR) cameras, two high-end camcorders, tripods, microphones, and 10 iPads. On the low-tech side, 10 mobile white boards were also purchased and placed in the Commons.

Around the same time, the other side of the fourth floor, now known as the Learning Commons West, was receiving some cosmetic updates with new popular reading shelving, popular magazine shelving, new furniture, and eventually four Dell PCs. This side of the Commons is smaller and for reasons unknown has always been quieter than the east Commons.

Brown Room

A former library staff and university meeting room was renovated in 2013. This space is conveniently located in the lobby to the right just at the main entrance to Cooper Library and was chosen as a repurposed space because of its ability to attract immediate attention to all who enter the library (see Figure 3, Appendix B). A joint collaboration between the library, CCIT, and Dell Computers, this space features three projectors and a visualization wall that is made of a bank of 15 46-inch monitors that span 17 feet wide and 6 feet high of wall space. Just outside the door of the Brown Room is the Hiperwall that displays library and campus events on a continual rotation. This room is designed for faculty to teach in a multimedia setting that offers data visualization in a collaborative environment. This quickly became the best small presentation space on campus, owing large part to the visualization wall and the convenient location (see Figure 4, Appendix B).
Adobe Digital Studio

From humble beginnings, the Adobe Digital Studio has become the go-to place for students looking for help and inspiration for anything dealing with audio and video production. As mentioned earlier, the first attempt at a digital studio consisted of four dual monitor Mac Pros placed in the Learning Commons West during the phase two project. From there, the studio found a one-year temporary home in a repurposed cataloging staff work area while plans were made to construct the Adobe Digital Studio on the fifth floor of Cooper Library in 2015. This project once again involved a collaborative effort from the Library, CCIT, and this time Adobe Systems. From the beginning, this project set out to transform not only the seldom-used corner of the fifth floor (see Figure 5, Appendix C), and engage everyone who entered the library. Two portions of a huge wall that separated the fifth floor from the lobby were removed to showcase the space. If that wasn’t enough, the space extended out 5 feet into the large lobby area of the library, beckoning all to engage with the activity and innovation going on upstairs (see Figure 6, Appendix C).

The key attractions of the Adobe Digital Studio are the audio production room and the video production room, the latter equipped with one-button technology to record presentations and a full green screen. Throughout the Adobe Digital Studio there are various types of comfortable seating and collaborative stations. There are seven large screen plug-and-work areas, four dual screen iMacs for post-production work, access to the full Adobe Creative Cloud, and expert staff to assist students with all audio/video projects. The 2,457-square foot space opened in October 2015 and remains one of the most occupied spaces in the library.

GIS

The Geographic Information System (GIS) area opened in the fall of 2015 and consists of two areas, the 662-square foot classroom (see Figure 8, Appendix D) and the 1,643-square foot work space (see Figure 10, Appendix E). The classroom seats 20, while the work space has 11 dual monitor iMacs/PCs and one plug-and-work station. The work space was the former temporary location of the digital studio before it moved upstairs (see Figure 9, Appendix E), and the classroom was a repurposed cataloging staff work space (see Figure 7, Appendix D). Once again, the project was a library and CCIT dual collaboration. The space is staffed by three full-time and several part-time GIS experts who teach multiple classes, conduct workshops throughout the year, and hold a yearly GIS day each fall consisting of numerous speakers.

What’s Next?

There are no definite plans at the moment, but the third floor is already being weeded of the government documents that have resided there for decades. This is an ideal location to repurpose because it is a large space that currently has limited seating, stacks that obscure natural lighting, and unlimited potential. Another full commons could be implemented here, easily adding 200 to 250 seats. At the moment, Starbucks is tearing out a wall on the fifth floor opposite the Adobe Digital Studio to mimic the design of that project.

There are also opportunities to repurpose areas of the fifth floor and reconfigure the second floor to make it more soundproof and user friendly. Moving bound journals to offsite storage, which was mentioned previously, would free countless feet of space throughout the library and would consolidate the collection. This has been done in certain areas in the library, but a full-scale removal of all bound journals would open up so many possibilities across the six floors of the library. It takes time and patience to see these projects through, but the payoff is tremendous once the work is done and the students move into the spaces.
Appendices

Appendix A: Images of Learning Commons Before and After Repurposing

Figure 1. Old reference area with index tables and print reference collection.

Figure 2. Learning Commons East with new furniture and power coming from the columns out to the tables.
Appendix B: Images of Brown Room Before and After Repurposing

Figure 3. Workers taking out the wall of the Brown Room to allow more natural light and also invite visitors into the space.

Figure 4. A student displays her work on the visualization wall in the newly renovated Brown Room.
Appendix C: Images of the Adobe Digital Studio Before and After Repurposing

Figure 5. The corner of the fifth floor after the removal of bound journals and shelving.

Figure 6. The finished Adobe Digital Studio from the sixth floor. Notice the portions of the wall that have been removed and the extension out into the lobby.
Appendix D: Images of the GIS classroom Before and After Repurposing

Figure 7. The former staff cataloging area being stripped down during the transformation.

Figure 8. The GIS classroom equipped with dual presentation capability and 20 student seats.
Appendix E: Images of the GIS Work Space

Figure 9. Staff cataloging area before transformation.

Figure 10. New GIS work space.