# OSS and libraries: A current SWOT analysis

After more than ten years of listening and watching the library-related open source software, a number of things have changed. This presentation outlines some of those changes as well as outlines some of the strengths, weaknesses, opportunities, and threats of open source software. The presentation ends some ideas for a "next generation" library catalog – services against texts.

### Much of my history

In one way or another, I have been writing computer programs since 1976. When I say my first computing experience was with punch cards it is the truth, but not in the way most people envision. We were calculating the solutions to quadratic equations.

A year or so later I took a BASIC programming class at the neighboring college. We were expected to write a dice game. I was a miserable failure. I was the first person at my college to hand in a word processed paper.

In 1983 I was a taxi driver, and I didn't know how much I was earning. Thus was born my first real computing need – my first "itch". Using my driver's log as input, I wrote a computer program calculating all sorts of strange statistics. It was also during this time when I was really into astronomy, and a book came out called Practical Astronomy With Your Calculator. It described how to predict things like the position of the Moon, stars, and planets for any given date.

By 1984 I needed a real job. I decided to become a librarian, and three things happened to me in library school. First I heard about something called "copyleft". Second, while using my astronomy program I pressed the wrong button at the wrong time, and my program was completely erased! I spent the next eight to ten hours rewriting my program from scratch. Third, I was in charge of interlibrary loan lending. Such was born my second big "itch". I resolved to write a computer program that generated my annual report daily.

My first professional library job, in 1988, I was using a program called UltraCard to print my catalog cards, and I proceeded to write a computer program that read the data and implement an online catalog. I got an Apple Library of Tomorrow grant. I would do online searches at the hospitals, send back requests via fax, get back the articles via fax, and reduce my turn-around time to hours instead of days.

By 1991 I out grew that job and moved on to North Carolina State University. I created all sort of hacks. Most importantly, I learned about the "serials pricing crisis" and Mr. Serials was born. The folks from Apple Computer came by again. This time they asked me to write a book describing how to use a Macintosh to implement World Wide Web servers. Around 1997 I wrote the first version of MyLibrary. By this time the term "open source" had been coined, and I was asked to sign a legal agreement with the University making the content distributable via the GNU Public License.

I out grew the NC State Libraries too and in 2001 I moved on to be more of an administrator and less of a hacker at the University of Notre Dame. There MyLibrary was transformed from a turnkey application into a set of object-oriented Perl modules. Which brings me to the present day. I still write a lot computer programs, but more for my personal use than for work. My Alex Catalogue of Electronic Texts has been a labor of love for more than fifteen years. Now-a-days I'm into digital humanities computing.

### Open source software & SWOT

As you know, the definition of "free" in the context of open source software should be equated with liberty and not necessarily gratis. At the same time, open source software is only as "free as a free kitten." There are financial costs associated with it.

Just as importantly, open source software is about community. While some of its roots may be bound to the ideas of a "gift culture", its branches are supported by a community of hackers. In many ways building an open source software community and providing technical support is more difficult than actually writing code.

- **Strengths** - Open source software benefits from the numbers game. When it comes to open source software, there is usually plenty of choice.

- **Weaknesses** - Probably the biggest weakness of open source software is support. Specialized skills are needed to create and maintain open source software. People see software for what it is – a means to an end – and sometimes this means is seen as an impediment as opposed to enabler.

- **Opportunities** - Many of the opportunities of open source software have their roots in the low barrier to entry. A computer is a tool and similar to an unshaped hunk of clay. What a person does with it is only limited by one's time, imagination, and ability to think systematically.

- **Threats** - Open source software is threatened by established institutions. The library profession's past experience works against the adoption open source software.

Open source software is not a panacea. It will exist side-by-side with commercial "closed source" software. Each serves a different need for different people and institutions.

### "Next generation" library catalogs

The next big challenge for library-related software, whether it be open source or not, is to figure out how to make content more useful. Find is not the problem that needs to be solved. Instead, the problem to solve is figuring out ways to do things with the information a person identifies as interesting – services against texts.

software
source
computer
open
things
library
people
time
number
one
text
information
word
program
content
works
person
list
first
document
libraries
words
ideas
use
times
search
score
Open
great
work
write
results
called
set
made
items
years
community
need
given
another
free
process
like
well
learn
needed
etc
example
services
go
comes
used
INSERT
want
experience
documents
full
way
sort
modules
common
book
People
get
lot
Using
MyLibrary
data
know
wrote
interest
different
make
via
count
profession
opportunities
access
Perl
thing
TFIDF
relevancy
computing
scores
cat
now
analysis
application
paper
end
grade
skills
think
environment
buy
created
many
tool
idea
length
difficult
ways
really
Coefficient
asked
limited
kitten
also
texts
done
next
new
changes
working
total
find
Ideas
came
generation
based
corpus
college
brings
Great
Boolean
Internet
support
exist
ranking
using
user
Suppose
much
means
read
back
Walden
Instead
else
often
axis
whole
year
job
available
sorts
result
per
everybody

search within, translate, trace idea, trace author, trace citation, etc.

By parsing out the words in text a person can be begin to count those words. Once things are counted they can be mathematically analyzed, and once they are analyzed new observations and perceptions can be articulated.

Chart and graph are two more action verbs that ought to be added to the list of things readers do with texts. Using just about any scripting language it is easy to count the words in any text. The most modern form is called word clouds.

Here is a more complicated idea, one of my own design, intended to illustrate a work's use of "great ideas". I call it the "Great Ideas Coefficient". Suppose a person were to count the number of times "great ideas" appear in a document. If done, then a person could get an idea of how much a work discusses the "great ideas". If done, then such a weighting would help a person determine whether or not the work was one discussing the human condition versus instructing someone on how to grow roses. There are other ways to assign numeric scores to texts. Readability scores and grade levels are additional examples. Things like a works's length, readability score, grade level, and Great Ideas Coefficient can be used as metadata describing a work, and consequently, they could be incorporated into to a search interface.

All of the above are merely examples of what can be done to do computational analysis against full text content. All a person needs is the full text of one or more works, the ability to quantify characteristics describing them, and the ability to ask meaningful questions against the results.

Eric Lease Morgan
University of Notre Dame

April 23, 2010

With the increasing availability of full text content, it is time to go beyond find and access. It is time to move towards use and analysis. Ask yourself, once you acquire an item of interest what do you do with it? The answers can all be expressed with all manner of action verbs. Examples include but are not limited to: read, print, summarize, annotate, review, rank, compare & contrast, save, delete,