Open Source Software for Libraries: from Greenstone to the Virtual Data Center and Beyond

by Micah Altman[*]

Introduction: The Growth of Open Source Software

Open Source Software (OSS) has grown tremendously in scope and popularity over the last several years, and is now in widespread use. OSS has a long history of supporting technology infrastructure – the fundamental tasks of managing host names and addresses, sending information across the Internet, delivering web pages, and relaying electronic mail are all primarily based on OSS, and have been for many years. Some of the most well-known technology businesses, like Amazon and Yahoo, are based on OSS, and other technology companies like IBM have made heavy investments (O'Reilly 1999; Sandred 2001, chap. 11; Lerner and Tirole 2002, secs. 1 and 3). Last year, the OSS operating system Linux was used on one third of all servers (making it the second most popular server operating system)), and its use is expected to continue to grow rapidly (Broersma 2002).

OSS is not limited to basic infrastructure. There are tens of thousands of OSS projects providing everything from games to statistical packages to digital photography editing. Directories of OSS projects, such as Gnu (<http://www.gnu.org/>) and Sourceforge (<http://www.sourceforge.net/>) now list over 50,000 projects, and the numbers continue to grow.

The growth of OSS has gained the attention of research librarians (Frumkin 2002) and created new opportunities for libraries. We might well ask, What distinguishes OSS from commercial software? What are the advantages and disadvantages of OSS software? Out of the thousands of packages available, which are most useful in a library environment?

In this essay, I first discuss the primary features of OSS, and where these features particularly benefit libraries. I also provide capsule summaries of OSS projects and resources that may be of particular interest to the library community.

What is Open Source Software?

Open Source Software's distinguishing feature is the broad *rights* it awards the consumer. Usage of software, as intellectual property, is restricted by copyright law and patent law.[2] Both commercial software and OSS award the consumer certain rights to use it. Most commercial software licenses give the consumer only limited use-rights – such as the right for a single user to run the software on a single system for a limited period. In contrast, OSS provides broad rights to *use*, *modify*, and *distribute* the software.

Although the exact details of the rights will vary by license, all OSS provides a number of broad rights (see Perens 1999 and the Open Source Initiative definition at: <http://www.opensource.org/docs/definition.php>). These rights fall into three broad categories:

1. *Rights to use without discrimination*. Unlike commercial software (and even some 'academically licensed' software), OSS may be used for any purpose, by anyone, at any time. For example, the same OSS used to run an academic website can also be used to run an e-commerce business. There are no annual license fees, restrictions on the numbers of users or systems, restrictions for non-commercial use, restriction to a particular country, expiration dates, or other artificial limits on use.

2. *Full rights to create derived works*. OSS not only permits one to use the software, but permits one to create new software from it.

   a. *Source Code Availability.* The source code for the software is made on the same terms as the binaries used to run it.

   b. *Free modification and redistribution.* Consumers not only have the right to examine the source, but to freely modify and redistribute modified (or unmodified) copies.

   c. *Integrity of authorship*. OSS may require that previous authors be acknowledged, and that modifications be clearly labeled, and separately packaged and named from the original software when redistributed. This maintains the integrity of software.

3. *No traps*. Modified copies of OSS must be redistributable under the same license as the original. The license cannot be restricted to a single product, and it must not restrict the distribution of other independently licensed software. (For example, the license must not insist that only OSS software appears on a distribution CD-ROM.)

All major open source licenses honor these rights, including the Gnu General Public License (GPL), Apple Public Source License, W3C license, Mozilla license, and BSD license. Although OSS offers broad rights, users of both OSS and commercial software user may still be under some restrictions:

- Government restrictions (such as the U.S. export controls on encryption software) may prevent the distribution of software, although the license allows it.

- Software (both commercial and non-commercial) that runs afoul of commercial patents may be restricted by the patent holder -- regardless of the license between the consumer and the author.

In addition, some open source licenses (most famously the GPL) prohibit the merging of open source and commercial softwareThis limits the ability of the consumer to intermingle open source and commercial source code in the same piece of software, but does not otherwise limit the combined distribution and use of commercial and open source software. Still, in practice, even under the most restrictive of current OSS licenses, OSS products can still be sold (as long as the source is also made available for free), and one is free to sell documentation, support, installation and other services for OSS.

Advantages and Disadvantages of OSS

General Advantages and Disadvantages

Many librarians are now considering OSS because of its low purchase costs. Unlike commercial software, there are no initial purchase fees, licensing fees, or upgrade fees. Furthermore, OSS is generally not tied to proprietary hardware, so the hardware costs associated with OSS tend to be lower. Other direct costs for OSS are often lower than those for comparable commercial software. Since the original supplier of the software has no monopoly on the information, the market for support and maintenance of OSS is more competitive than that for commercial software with comparable user bases. Thus support and maintenance costs are often lower (Kenwood 2001).

Many advocates[3] of OSS development argue that it leads to faster software development and more reliable software. Raymond (1999, 39) argues that successful OSS projects update their software quickly and frequently, paying close attention to users' bug reports and responding rapidly. Moreover, bugs are fixed quickly because of the exposure OSS provides; that is, "given enough eyeballs, all bugs are shallow" (p. 41). The Open Source Initiative (<http://www.opensource.org/>) states this particularly clearly:

> "When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing."

A recent Mitre study offers some support for these practical claims, finding that OSS often has advantages in reliability, frequency of bug fixes, extensibility and support (Kenwood 2001).

OSS comes with a number of risks as well. First, when considering any software for long-term use, one must consider longevity, and OSS is no exception. OSS projects may fragment into incompatible versions or stagnate (particularly after losing a lead developer). One should consider the size of the user base and the number and activity level of the developers before incorporating software into long term plans. It is important to note, however, that commercial software suffers similar risks, and that the relative longevity of commercial versus OSS software is an open question (Lerner and Tirole 2002). OSS offers the opportunity for users to continue to contribute to a project even after the original developers leave. In contrast, development of commercial software (especially specialized software) is frequently abandoned completely when a business goes out of business or is acquired, or even when the business creates a new product.

Second, OSS is often not as user-friendly as commercial counterparts. Many open source projects (with notable exceptions, such as Gnome, Greenstone, and the Virtual Data Center) are not cognizant of usability (Hovater 2002), and the incentives for producing OSS, while emphasizing utility, may de-emphasize user-interface development (Lerner and Tirole 2002).

Library-Specific Advantages of OSS

In addition to these general advantages, there are a number of reasons that libraries, in particular, may prefer to use OSS over commercial software: preservation, privacy and auditing, community resources, and open standards.

As librarians, we are sometimes the stewards of unique collections. The preservation of digital objects is currently intimately tied to software that presents those objects. Complete preservation of complex digital objects, especially, is likely to require preservation of the software needed to use those objects (Granger 2000). Since commercial software is usually distributed only as a binary that will run only on a single hardware platform (and often only under a single version of a particular operating system),

commercial software is very difficult to preserve over the long run without developing hardware emulation (and possibly Operating System 'emulation' as well). OSS, in contrast, can often be recompiled, or at least ported, to new hardware and operating systems.

Librarians have a strong tradition of defending the privacy of users. Increasing numbers of commercial packages, by both major and minor vendors, quietly collect information on the systems and habits of people who use them. Whether as part of 'adware,' 'spyware,' 'live' updates, or digital rights management, these applications send usage information back to the vendor (Millman 2001). Furthermore, similar features are increasingly added by vendors in order to 'transparently' (and in many cases, silently) install new software on the client's system, or to disable software that is the subject of payment or intellectual property disputes. Although this behavior can be limited with placement of network firewalls, it can be challenging to audit commercial software for this type of behavior. In contrast, OSS is easily audited. Moreover, since the source code is available, it is relatively easy for a community of users to check for and disable 'spyware' and other remote reporting and installation functions.

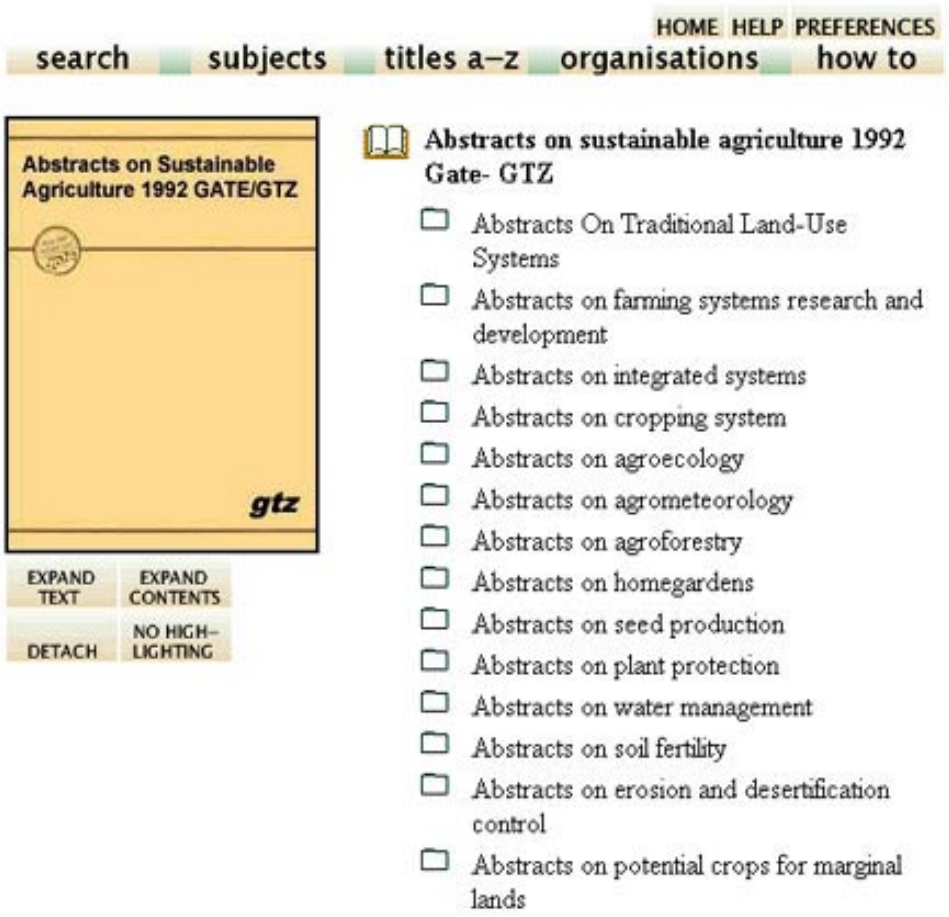

Figure 1: Example Results Screen from Greenstone Digital Library

Scholarly Standards and Exchange

Both the library and the academic community have a history of sharing information and of using open standards. For example, both citations and cataloging, two of the mainstays of librarianship and academics, are based upon standards that are fundamentally open. The use of OSS ensures that all software standards, both explicit and implicit, will continue to be open to inspection, and allows others to build upon previous work done in the community. As Lessig (1999) has made clear, software infrastructure has important implications for the types of community values that software can support and encourage. Open academies and open libraries demand open source infrastructures.

An Overview of Stand-Alone Solutions for Libraries

There are several packages that aim to offer stand-alone catalogs or complete digital library solutions: Greenstone, Koha, RIB, Sitesearch (which is, unfortunately, not really open source) and the Virtual Data Center. In this section, I draw thumbnail sketches of each. In the next section, I discuss other software tools and resources that are likely to be useful to librarians who wish to construct their own library applications or digital libraries.

Greenstone

Greenstone is a package for creating, managing and distributing collections of documents. It runs on Linux, Unix and Windows platforms. Collections created through Greenstone can be used on-line or distributed on CD-ROM. Among the features it provides are multi-lingual interface support, full-text and fielded searching, browsable indexes, customized formatting, metadata extraction and a Z39.50 client.

Greenstone was developed as part of the New Zealand Digital Library Project, run by the Department of Computer

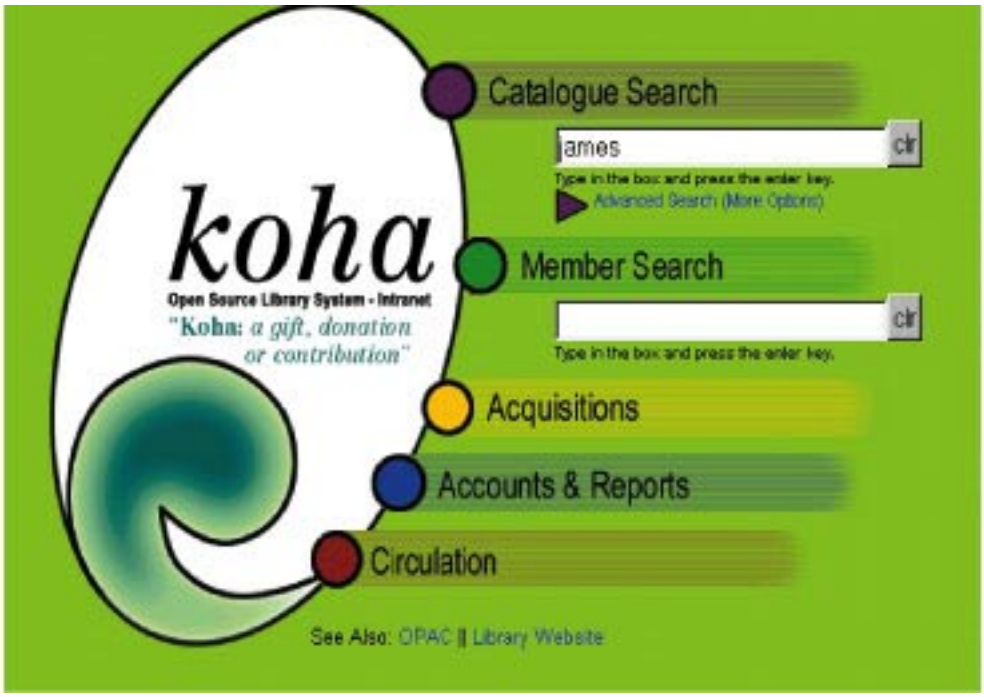

Figure 2: Example Splash Screen from Koha Catalogue

Science, University of Waikato, New Zealand. The software is in its first major production release, and is actively maintained and updated (through Sourceforge). It is available from < http://www .greenstone.org >.

Koha

The Koha system is a full catalogue, OPAC, management, and acquisitions package. It does not, however, support document distribution and indexing. It runs on Linux and is accessed primarily through a web-based interface. Among the features it provides are simple and fielded searches, reading lists, acquisitions management (including budgets and pricing information), circulation management, and patron management.

Koha was made in New Zealand by the Horowhenua Library Trust and Katipo Communications Ltd. The software is in its first major production release, and is actively maintained and updated (through Sourceforge). It is available from <http://www.koha.org/>.

RIB

Repository in a Box (RIB) is a software package for creating web-browsable metadata collections. It runs on Linux, Unix and Windows. Collections created through RIB are accessed through the web, and can be configured to interoperate with other similar remote RIB collections. Among the features RIB provides are searching, browsable indexes, repository federation, and a user-friendly Java-based management GUI.

Figure 3: Example Navigation, Results and Management Screens from RIB

RIB was created by the RIB Development Team at the University of Tennessee under direction by the National HPCC Software Exchange (NHSE). The software is mature (in its second full production release) and seems to be actively maintained, although updates are infrequent. It is used in a number of NASA, DOE, DOD and NSF research centers. It is available from <http://www.nhse.org/RIB/>.

Sitesearch
Sitesearch is an enhanced OPAC and distributed catalog search system. It runs on both Unix and Windows but is not documented to run on Linux. Among the features it provides are cross-catalog searching over world wide web and Z39.50 sources, Z39.50 Client and Server, interoperability hooks for interlibrary loan and document delivery services, relatively advanced search history and result set handling, and MARC support.

Sitesearch was developed by OCLC. The software is mature, and has been actively maintained, although the maintenance model is being changed. It is available from <http://www.sitesearch.oclc.org/>.

Sitesearch offers many of the benefits of OSS within an academic library environment, but, unfortunately, Sitesearch is not fully open source. The license under which it is distributed by OCLC prohibits commercial use, and may limit the availability of third-party support. This license is incompatible with many popular OSS licenses, which would hinder integration of Sitesearch with other OSS solutions.

Virtual Data Center
The Virtual Data Center (VDC) software is a comprehensive, open-source digital library system. The VDC software provides a complete system for the management and dissemination of federated collections of quantitative data. It runs on Linux. Collections created through VDC are accessed through the web, and can be distributed across multiple servers, or virtually include selected parts of other data archives.

Among the features it provides are simple and fielded searching (at all levels of granularity – collection, study and data), data and documentation delivery, data extraction (variable and row selection), data format conversion (Data Documentation Initiative, SAS, SPSS, Stata, Splus, CSV), on-line data analysis (descriptive statistics, exploratory data analysis graphs, crosstabs), archival format and filesystem-independent storage, Open Archives Initiative service, Z39.50 service, persistent naming, distributed operation, distributed virtual collections, metadata harvesting, federated authentication and authorization, and on-line GUI management tools.(See Figure 5)

VDC was developed by the Harvard-MIT Data Center and Harvard University Library as part of the Digital Libraries Initiative, sponsored by the National Science Foundation and other agencies. The software is in beta release, and is actively maintained and updated (through Sourceforge). It is available from < http://thedata.org >.

Other Resources for OSS in Libraries
The preceding projects provide stand-alone OSS catalogs or complete digital libraries. In addition to these, there are a host of open source resources that are useful to libraries, including website toolkits, indexing engines, databases, and clients, servers and software libraries for specialized technologies such as Z39.50, USMARC, and Ariel.

OSS Meta-sites. These sites provide directories of open-source projects.

- The **"Open Source Software for Libraries"** (< http://www.oss4lib.org/ >) uniquely specializes in library applications, and is particularly useful.

- **Sourceforge (<** http://sourceforge.net/ >), **Freshmeat** (< http://freshmeat.net >), and **The Free Software Foundation** (FSF; < http://gnu.org/ >) provide huge catalogs of open source projects. FSF is the oldest OSS site, and hosts thousands of projects. Sourceforge hosts nearly 50,000 OSS projects.

- "The Impoverished Social Scientist's Guide to Free Statistical Software and Resources" (< http://data.fas.harvard.edu/micah_altman/socsci.shtm l>) is a collection of pointers to OSS tools for data analysis and manipulation collected by the author.

- The "FreeGIS Site" (< http://www.freegis.org/ >) is a collection of pointers to free GIS tools and toolkits.

- A large collection of searching, harvesting and indexing tools is cataloged at < http://www.searchtools.com/tools/tools-opensource.html >.

Specific tools and toolkits. A number of tools and toolkits may be of specific interest to libraries that are building their own tools, or who wish to supplement current tools. Full-text searching of material in web-based catalogs can be provided by using indexers such as Swish-e or HT://dig. More structured catalogs can be built using XML and XML databases such as DbXML, Xindice, or Cheshire, or using SQL databases such as Postgressql or MySql. Information portals can easily be built with toolkits such as Slash and Jetspeed.[4] Finding aids, pathfinders, and other dynamic, organized web applications can be built with open source application servers such as Zope and Gist.

Conclusions
For the last several years, Open Source Software has dominated the infrastructure of Internet and web services. OSS continues to grow in this and other areas, and there are now

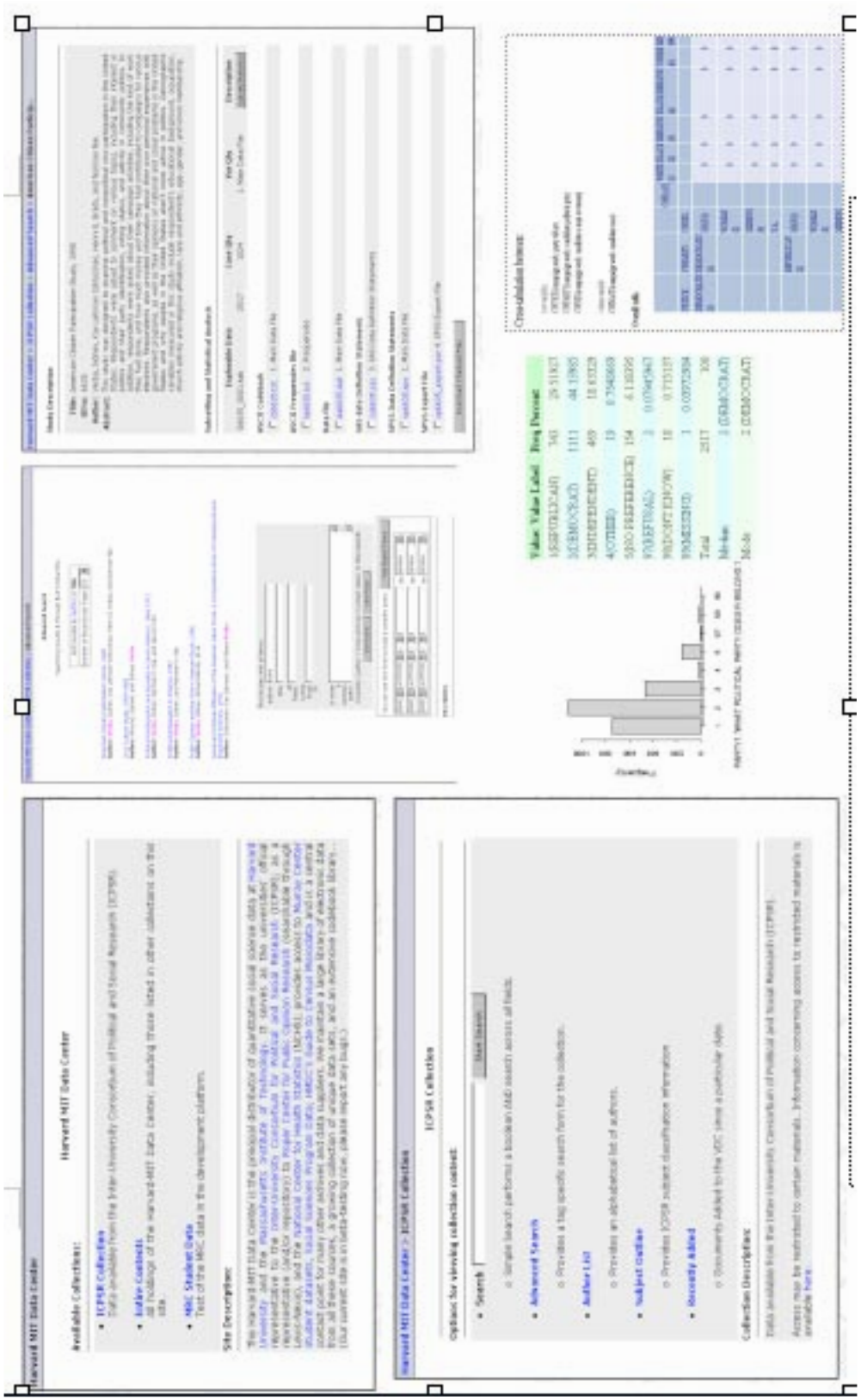

Figure 5: Example Navigation and Results Screen from VDC

over 50,000 open source software applications available for instant download. Among these are a number of high-quality packages that provide stand-alone digital library and OPAC functionality, as well as a host of other applications and toolkits that would be of great use in the development or enhancement of library services.

The most popular Open Source Software projects produce software that is quite often more stable, secure, auditable, and extensible than commercial alternatives. Using OSS also makes the preservation of digital objects easier and less risky. Moreover, using OSS guarantees that the standards and protocols used in the library will always be open to examination, and helps the library community to build upon previous successes.

Bibliography

[1] Broersma, Matthew, 2002, "Will Linux Survive the Dot-com Crash?," *ZDNET (UK)*, January 2, 2002, < http://www.zdnet.com/filters/printerfriendly/0,6061,2835454-92,00.html>.

[2] Frumkin, Jeremy, ed., 2002, "Special Issue: Open Source Software," *Information Technology and Libraries* 21(1) <http://www.lita.org/ital/ital2101.html>.

[3] Granger, Stewart, 2000, "Emulation as a Digital Preservation Strategy," *D-Lib Magazine* 6(10). < http://www.dlib.org/dlib/october00/granger/10granger.html>

[4] Hovater, J., D. Kiskis, M. Krot, I. Holland, and M. Altman, 2002, "Usability Testing of the Virtual Data Center," in *Proceedings of the Joint Conference on Digital Libraries '02*, ACM Press: New York.

[5] Kenwood, Carolyn A., 2001, "A Business Case Study of Open Source Software," Report # M P 0 1 B 0 0 0 0 0 4 8, MITRE Corporation: Bedford, MA. < http://www.mitre.org/support/papers/tech_papers_01/kenwood_software/>

[6] Lerner, Josh, and Jean Tirole, 2002, "Some Simple Economics of Open Source," Journal of Industrial Economics, 50 (2002) forthcoming

[7] Lessig, Lawrence, 1999, *Code and Other Laws of Cyberspace,* Basic Books: New York.

[8] Millman, Howard, 2001, "How to Keep Vendors From Quietly Violating Your Privacy," *New York Times,* January 18, Late Edition - Final, Section G, Page 9, Column 1.

[9] O'Reilly, Tim, "Hardware, Software and Infoware," in *Open Sources,* edited by Chris DiBona, Sam Ockman and Mark Stone, O'Reilly and Sons: Sebastapol, CAq

[10] Perens, Bruce, 1999, "The Open Source Definition", in *Open Sources,* edited by Chris DiBona, Sam Ockman and Mark Stone, O'Reilly and Sons: Sebastapol, CA.

[11] Raymond, Eric, 1999, *Cathedral and the Bazaar,* O'Reilly and Sons: Sebastapol, CA.

[12] Sandred, Jan, 2001, *Managing Open Source Projects*, John Wiley & Sons: New York.

[13] Stallman, Richard, "The Gnu Operating System and the Free Software Movement," in *Open Sources,* edited by Chris DiBona, Sam Ockman and Mark Stone, O'Reilly and Sons: Sebastapol, CA.

Footnotes

[1] This material is based upon work supported by the National Science Foundation under Grant No. 9874747.

[2] To a much lesser extent, some pieces of software may additionally be governed by trademark law.

[3] Some advocates, most notably Richard Stallman of the Free Software Foundation, argue for OSS on ethical grounds. For Stallman, free software is a "stark moral choice" (Stallman 1999, 55), and restrictions on the distribution of information in general, and software in particular are harmful to society. (See <http://www.gnu.org/philosophy/why-free.html>.)

[4] The Scout toolkit, which was released in beta as this article was going to press, also looks promising because of its attention to metadata. (<http://scout.cs.wisc.edu/research/SPT/>)

* Paper presented at the IASSIST Conference, June 2002, in Storrs, CT, USA.

Micah Altman, Harvard University, micah_altman@harvard.edu.