Book Reviews

Tom Zillner, Editor

In the Beginning . . . Was the Command Line

by Neal Stephenson. New York: Avon Books, Inc., 1999. 151p. $10 (ISBN 0-380-81593-1)

Neal Stephenson is best known for his cyberfiction, including *Snow Crash* and most recently *Cryptonomicon. In the Beginning . . . Was the Command Line* is a quite different kettle of fish. *Command Line* is a short book with a succinct message: the command line is a good thing, because the full power of the computer is only available to those who can access the command line and type in the magic commands that make things happen. Stephenson learned this lesson the hard way, after first spending much time as a Macintosh-devoted GUI-head. The revelation came when he lost a document he was editing on his PowerBook, completely and without a trace, forever irretrievable. Actually, I say the book has a succinct message, but it has many messages and many metaphors, all artfully constructed by a master of prose.

Stephenson constructs his arguments along multiple lines, providing a discursive tour through Windows, Macintosh, and UNIX history, offering personal history as well as his own take on the economics of the software industry. For example, he believes that Microsoft would be better off as an applications company rather than carrying the millstone of a family of operating systems. As for Apple, he suggests that they have been doing their best to destroy themselves for years, so far unsuccessfully (but give them time).

The real meat of the book is whether, in fact, it is better to offer to people the flash of metaphor with the recognition that power and certain levels of choice are lost, as with graphical user interfaces exemplified by Windows and the Macintosh, or whether it is better to have at least some access to the command line interface, which MS/DOS offered and members of the UNIX family (e.g., Linux) afford. This is, in fact, both a silly and important question at the same time. Silly because many people would wonder why anyone would want command line access to any software. Silly because others might wonder why you couldn't have both. Important, or at least apparently important, because we seem to have become, without much warning, a world wrapped in GUIs of one sort or another. Important in the library automation world, because end-user tools are moving increasingly toward GUI-based or Web-based interfaces without text-based alternatives (except, perhaps, Lynx or similar Web browsers, which have their own problems). For much of the book, Stephenson dances around the question, among others, of why not both GUI and text-based interfaces, and finally finds the answer in the Be operating system. My question is, why not as many interfaces as it takes, of whatever sort? To repeat the trite saw, there are two kinds of people in the world, those who divide the world into two kinds of people and those who don't. Stephenson has a lot of fun trying to make the division in this case, then ultimately comes out from behind the posturing and admits that he believes in the availability of both worlds.

There are many people who do, indeed, want hard things hidden from them, at least some of the time. When I am dealing with an automated teller machine, I don't want to have to use mechanical levers or pedals as I might have needed were ATMs invented in an earlier age, nor do I want to type in commands, although I am comfortable using a command line environment in my workplace. I just want to be prompted through a minimal number of steps to walk away with some cash from my checking account. The world is a complicated and challenging place to navigate. Some people would like to be helped by other people in this navigation, although many have found that they would far rather deal with the dumbed-down interface of an ATM machine than to interact with not-so-friendly, underpaid bank tellers.

Similarly, many people want to accomplish a particular task requiring the use of a computer and don't mind having the details hidden from them, no matter how much power knowing the details would provide. Or, they want to do that at least some of the time. As an example in the library world, let's consider a naïve patron who enters the library desiring to perform a known-item search. Such a user might be quite comfortable with an interface with a single type-in box and a set of clickable buttons labeled Title, Author and Subject. Or maybe just a single button "click to start search." Although naïve users may consult library staff, who are most often more friendly than bank tellers, many people want to find their own materials. At the same time, more sophisticated users want more sophisticated capabilities and interfaces from the same catalogs. Although vendors have gotten better at providing a couple of levels of complexity and corresponding user interfaces, why not go further?

There aren't just two kinds of people. There are lots of kinds of people, with lots of kinds of information needs, representing lots of experience levels. Why the restrictions at the user interface? In the history of microcomputing, Stephenson points to the evolution of two major players, Microsoft and Apple, with Linux coming on strong and Be representing an interesting offshoot. I think the important insight implicit in what Stephenson discusses is that much of the appearance and behavior of Windows and the Macintosh desktop are historically based artifacts. In order to maintain backward compatibility with existing applications, the Windows and Macintosh operating systems have picked up a great deal of "cruft," computer code that allows multitasking and other improvements cobbled on to the fragile inner shell of ancient code required for compatibility with older applications. At the same time, Stephenson invokes the familiar refrain that the user interfaces of both platforms are tied to a tired set of metaphors that attempt to mimic the real-world office (e.g., desktop, folder) but do not do so with any kind of useful fidelity. In the library world, I think a similar kind of lineage might be traced from command line interfaces to the current Windows- and Web-based front-ends. Although many libraries and librarians have faced painful conversion processes over the years in moving through generations of automated systems, it might be interesting to see if there are still traces of underlying code that owe their existence to backward compatibility.

Where does Stephenson turn in the face of the inelegance of the Windows and Macintosh worlds? He finds solace in the power and integrity of Linux. It may take a long time to successfully install the operating system and get it to function with all of the hardware components of a particular computer configuration, but it has all that power, and all of those cool applications carefully constructed by people who care. Bugs are fixed quickly. It's a community effort. That's all very appealing, particularly when compared to the appalling response (or lack of it) to Windows or Macintosh bugs. The problem is that so far most of us aren't equipped to deal with the steep curve required to install Linux on personal computers, and the corporate or library environment usually isn't politically prepared for Linux to be adopted as an institutionwide standard. So, while Linux boxes are frequent choices for servers, they are not widespread personal PC choices. Nor should they be until easy installation tools are available.

Again, Stephenson is ambivalent. On the one hand, he recognizes that there are many people who don't want the kind of power offered by being so close to the machine if it means becoming experts in arcane commands and codes. Even though he wants the power and simplicity, and decries the limitations imposed by the GUI, he recognizes that Linux is not for everyone. He's right. Most people use computers to get some work done (or to play). To the extent that the software gets in the way, it isn't operating properly. By that criterion, none of the three environments described are particularly useful in a desktop world.

In spite of the fact that the old metaphors have been rightly criticized for years for their tiredness, there doesn't seem to be much movement beyond them, except in limited research operating environments and applications. Similarly, it seems, in the library and information world, at least in most people's routine interactions with OPACs and databases. Yes, I am waffling, because I'm sure that someone could point out the "Snarfle™ Virtual Reality interface to the LC catalog that affords a walkthrough browsing experience," but of course only six computer science researchers have actually experienced the Snarfle™ interface, and it requires a $25,000 workstation and $10,000 in virtual reality gear to work, plus it is s-l-o-w. Pardon the sarcastic riff, but there is a lot of wonderful user interface work that is certainly not finding its way onto mainstream computer users' desktops, or to the library or information center.

So what's the answer? Criticism is fun, because critics don't necessarily have to provide a positive account to match their nay-saying function. If things are bleak in the world of the user interface, both on the average user desktop and on the library desktop as well, what is to be done? For a taste of what is to come in the library world, take a look at MyLibrary (http://my.lib.ncsu.edu/), which allows profiling of user preferences and customization based on academic discipline. Similarly, there are a number of Web portals and other sites that allow customization for users (e.g., My Yahoo, My Excite, etc.). Suppose that these first steps in customization are carried further, so that each user's unique profile generates a unique user interface experience across all databases he or she deals with in a session.

The interface unification could be accomplished across heterogeneous databases in a couple of different ways. A simple initial step that many libraries already employ is to obtain databases from a single aggregator, so that a uniform interface is presented to the user. For example, OCLC's First Search offers a single interface to a number of commercial databases. This type of solution is not possible for libraries that need access to a diverse array of databases not available through a single aggregator or vendor. Of course, this situation can present patrons and staff with a bewildering array of interfaces and search methods. A more elaborate solution is to employ Z39.50 to access the databases and build a single interface at the front end. There may be aggregators that already use this strategy with the databases they provide, but in the future perhaps there would be an incentive to offer unified interfaces with fine-grain customization possible by users.

Getting back to Stephenson's more generalized view of the user interface, I think there are also opportunities here for more fine-grained customization. Stephenson points to the BeOS, which apparently allows both command-line and GUI-based interactions, as an example of what can be done when an operating system is constructed anew, from the bottom up, with no pre-existing audience to satisfy. At the same time, and in contrast, Stephenson extols the power of open software development, which he believes is most apparent in operating systems, the production of which he describes as money-losing propositions. Yet, Linux is tremendously successful without, for the most part, commercial gain for developers. Can this same model be applied to interface and other development in the library world? In this example, might not some group of librarian coders (or coder librarians) work together to put MyLibrary together with Z39.50 capabilities and customization of interfaces to produce a little slice of paradise for library patrons? Promising moves are being made within the library community to get open source efforts off the ground. This could be one of many especially useful and fruitful projects to come out of open software development for libraries.

Although his book is ostensibly about a few issues that elicit yawns from most of the world, Stephenson is really using *In the Beginning . . . Was the Command Line* to look at a much bigger picture than simply the command line versus the GUI at its microscopic level. Stephenson looks at the cloaking, obfuscation or replacement of underlying text by images and multimedia as contributing to the decline of civilization. That seems like a radical claim, but at heart it is the one that Stephenson makes in his discussion of the Disney-ification of the world—that visual metaphors and explanations oversimplify and obscure the truth. In fact, Stephenson goes further, discussing this trend toward anti-word as our attempt at an antidote for the kind of intellectualism that resulted in a lot of death, pain, and suffering for people in the twentieth century. He, as a person who lives by words and loves the intellectual life, thinks we've gone too far, reaching a state of cultural relativism where there is neither good nor bad remaining. This discussion includes my favorite quote of the book:

> The problem is that once you have done away with the ability to make judgments as to right and wrong, true and false, etc., there's no real culture left. All that remains is clog dancing and macrame. The ability to make judgments, to believe things, is the entire point of having a culture. I think this is why guys with machine guns sometimes pop up in places like Luxor and begin pumping bullets into Westerners. . . . When their sons come home wearing Chicago Bulls caps with the bills turned sideways, the dads go out of their minds. (p. 56)

It's a pretty startling move to try to connect up the decline in use of the command line to an anti-intellectualism following World War II that resulted in cultural relativism. I think it actually has some merit, although in the case of visual interfaces versus the command line the ethical import is minimal, i.e., I don't believe my decision to accomplish certain tasks using visual metaphors contributes to the decline of civilization, and I think the fact that I like to work on other tasks utilizing a command line won't serve to save our written culture. It's too much of a stretch.

I think that something Stephenson misses in his discussion of the replacement of the written word by visual images is that there is still a creative force and judgment involved in the creation of the images. There is still script writing. Isn't this, after all, what a writer does in any case, creating images, metaphorically, through his or her work? Certainly, we are moving through a perilous time, when the world really is changing from a reliance on the written word to more dependence on the visual. There will be many things lost in this transition. Plato had some major, well-founded doubts about the transition from Greece's oral cultural tradition to a written one. The change happened anyway.

Civilization has been declining for a long time. My fearless prediction is that it will continue to decline for a long time. I think Stephenson has done a masterful job of writing a brief glimpse of the overall picture that represents the state of culture and intellectual life in the world today, and has also made some important points about the economics and character of the world of software and operating environments. His writing skills make this fairly short book a pleasurable read and a worthwhile one. As I did, I think you might find this long essay a useful starting point for thoughts about issues large and small.—*Tom Zillner, WILS*

The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary

by Eric S. Raymond, Sebastopol, Calif.: O'Reilly, 1999. 288p. $19.95 (ISBN 1-56592-724-9)

This short essay examines, in the guise of a book review, the concept of a "gift culture" and how it may or may not be related to librarianship. As a result of this examination, and with a few qualifications, I believe my judgements about open source software and librarianship are true: open source software development and librarianship have a number of similarities—both are examples of gift cultures.

I have recently read a book about open source software development by Eric Raymond. *The Cathedral & the Bazaar* describes the environment of free software and tries to explain why some programmers are willing to give away the products of their labors. It describes the "hacker milieu" as a "gift culture":

> Gift cultures are adaptations not to scarcity but to abundance. They arise in populations that do not have significant material scarcity problems with survival goods. We can observe gift cultures in action among aboriginal cultures living in ecozones with mild climates and abundant food. We can also observe them in certain strata of our own society, especially in show business and among the very wealthy.[1]

Raymond alludes to the definition of "gift cultures," but not enough to satisfy my curiosity. Being the good librarian, I was off to the reference department for more specific answers. More often than not, I found information about "gift exchange" and "gift economies" as opposed to "gift cultures." (Yes, I did look on the Internet but found little.)

Probably one of the earliest and more comprehensive studies of gift exchange was written by Marcell Mauss.[2] In his analysis he says gifts, with their three obligations of giving, receiving, and repaying, are in aspects of almost all societies. The process of gift giving strengthens cooperation, competitiveness, and antagonism. It reveals itself in religious, legal, moral, economic, aesthetic, morphological, and mythological aspects of life.[3]

As Gregory states, for the industrial capitalist economies, gifts are nothing but presents or things given, and "that is all that needs to be said on the matter." Ironically for economists, gifts have value and consequently have implications for commodity exchange.[4] He goes on to review studies about gift giving from an anthropological view, studies focusing on tribal communities of various American Indians, cultures from New Guinea and Melanesia, and even ancient Roman, Hindu, and Germanic societies:

> The key to understanding gift giving is apprehension of the fact that things in tribal economics are produced by non-alienated labor. This creates a special bond between a producer and his/her product, a bond that is broken in a capitalistic society based on alienated wage-labor.[5]

Ingold, in "Introduction To Social Life," echoes many of the things summarized by Gregory when he states that industrialization is concerned

> exclusively with the dynamics of commodity production. . . . Clearly in non-industrial societies, where these conditions do not obtain, the significance of work will be very different. For one thing, people retain control over their own capacity to work and over other productive means, and their activities are carried on in the context of their relationships with kin and community. Indeed their work may have the strengthening or regeneration of these relationships as its principle objective.[6]

In short, the exchange of gifts forges relationships between partners and emphasizes qualitative as opposed to quantitative terms. The producer of the product (or service) takes a personal interest in production, and when the product is given away as a gift it is difficult to quantify the value of the item. Therefore, along with the product or service, less tangible elements—such as obligations, promises, respect, and interpersonal relationships—are exchanged.

As I read Raymond and others I continually saw similarities between librarianship and gift cultures, and therefore similarities between librarianship and open source software development. While the summaries outlined above do not necessarily mention the "abundance" alluded to by Raymond, the existence of abundance is more than mere speculation. Potlatch, "a ceremonial feast of the American Indians of the northwest coast marked by the host's lavish distribution of gifts or sometimes destruction of property to demonstrate wealth and generosity with the expectation of eventual reciprocation," is an excellent example.[7]

Libraries have an abundance of data and information. (I won't go into whether or not they have an abundance of knowledge or wisdom of the ages. That is another essay.) Libraries do not exchange this data and information for money; you don't have to have your credit card ready as you leave the door. Libraries don't accept checks. Instead the exchange is much less tangible. First of all, based on my experience, most librarians simply take pride in their ability to collect, organize, and disseminate data and information in an effective manner. They are curious. They enjoy learning things for learning's sake. It is a sort of Platonic end in itself. Librarians, generally speaking, just like what they do and they certainly aren't in it for the money. You won't get rich by becoming a librarian.

Information is not free. It requires time and energy to create, collect, and share, but when an information exchange does take place, it is usually intangible, not monetary, in nature. Information is intangible. It is difficult to assign it a monetary value, especially in a digital environment where it can be duplicated effortlessly:

> An exchange process is a process whereby two or more individuals (or groups) exchange goods or services for items of value. In Library Land, one of these individuals is almost always a librarian. The other individuals include tax payers, students, faculty, or in the case of special libraries, fellow employees. The items of value are information and information services exchanged for a perception of worth—a rating valuing the services rendered. This perception of worth, a highly intangible and difficult thing to measure, is something the user of library services "pays," not to libraries and librarians, but to administrators and decision-makers. Ultimately, these payments

> manifest themselves as tax dollars or other administrative support. As the perception of worth decreases so do tax dollars and support.[8]

Therefore, when information exchanges take place in libraries, librarians hope their clientele will support the goals of the library to administrators when issues of funding arise. Librarians believe that "free" information ("think free speech, not free beer") will improve society. It will allow people to grow spiritually and intellectually. It will improve humankind's situation in the world. Libraries are only perceived as beneficial when they give away this data and information. That is their purpose, and they, generally speaking, do this without regard to fees or tangible exchanges.

In many ways I believe open source software development, as articulated by Raymond, is very similar to the principles of librarianship. First and foremost they are similar in the idea of sharing information. Both camps put a premium on open access. Both camps are gift cultures and gain reputation by the amount of "stuff" they give away. What people do with the information, whether it be source code or journal articles, is up to them. Both camps hope the shared information will be used to improve our place in the world. Just as Jefferson's informed public is necessary for democracy, open source software is necessary for the improvement of computer applications.

Second, human interactions are a necessary part of the mixture in both librarianship and open source development. Open source development requires people skills by source code maintainers. It requires an understanding of the problem the computer application is intended to solve, since the maintainer must be able to "patch" the software, both to add functionality and to repair bugs. This, in turn, requires interactions both with other developers and with users who request repairs or enhancements. Similarly, librarians understand that information-seeking behavior is a human process. While databases and many "digital libraries" house information, these collections are really "data stores" and are only manifested as information after the assignment of value is given to the data and interrelations between data are created.

Third, it has been stated that open source development will remove the necessity for programmers. Yet Raymond posits that no such thing will happen. If anything, there will be an increased need for programmers. Similarly, many librarians feared the advent of the Web because they believed their jobs would be in jeopardy. Ironically, librarianship is flowering under new rubrics such as information architects and knowledge managers. It has also been brought to my attention by Kevin Clarke (kevin_clarke@unc.edu) that both institutions use peer-review:

> Your cultural take (gift culture) on "open source" is interesting. I've been mostly thinking in material terms but you are right, I think, in your assessment. One thing you didn't mention is that, like academic librarians, open source folks participate in a peer-review type process.

All of this is happening because of an information economy. It sure is an exciting time to be a librarian, especially a librarian who can build relational databases and program on a Unix computer.

Acknowledgements

Thank you to Art Rhyno (arhyno@server.uwindsor.ca) who encouraged me to post the original version of this text.—*Eric Lease Morgan, North Carolina State University, Raleigh, North Carolina*

References

1. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, 99.
2. M. Mauss, *The Gift: Forms and Functions of Exchange in Archaic Societies* (New York: Norton, 1967).
3. S. Lukes, "Mauss, Marcel," in *International Encyclopedia of the Social Sciences*, D. L. Sills, ed. (New York: Macmillian), vol 10, 80.
4. C. A. Gregory, "Gifts," in *The New Palgrave: A Dictionary of Eeconomics*, J. Eatwell and others, eds. (New York: Stockton Pr., 1987), vol. 4, 524.
5. Ibid.
6. T. Ingold, "Introduction to Social Life," in *Companion Encyclopedia of Anthropology*, T. Ingold, ed (New York: Routledge, 1984), 747.
7. The Merriam-Webster Online Dictionary, http://search.eb.com/cgi-bin/dictionary?va=potlatch
8. E. L. Morgan, "Marketing Future Libraries." Accessed Apr. 27, 2000, www.lib.ncsu.edu/staff/morgan/cil/marketing.

Index to Advertisers

Info USA	cover 2
Library Technologies, Inc.	cover 3
LITA	58, 69, cover 4
MIT Press	95