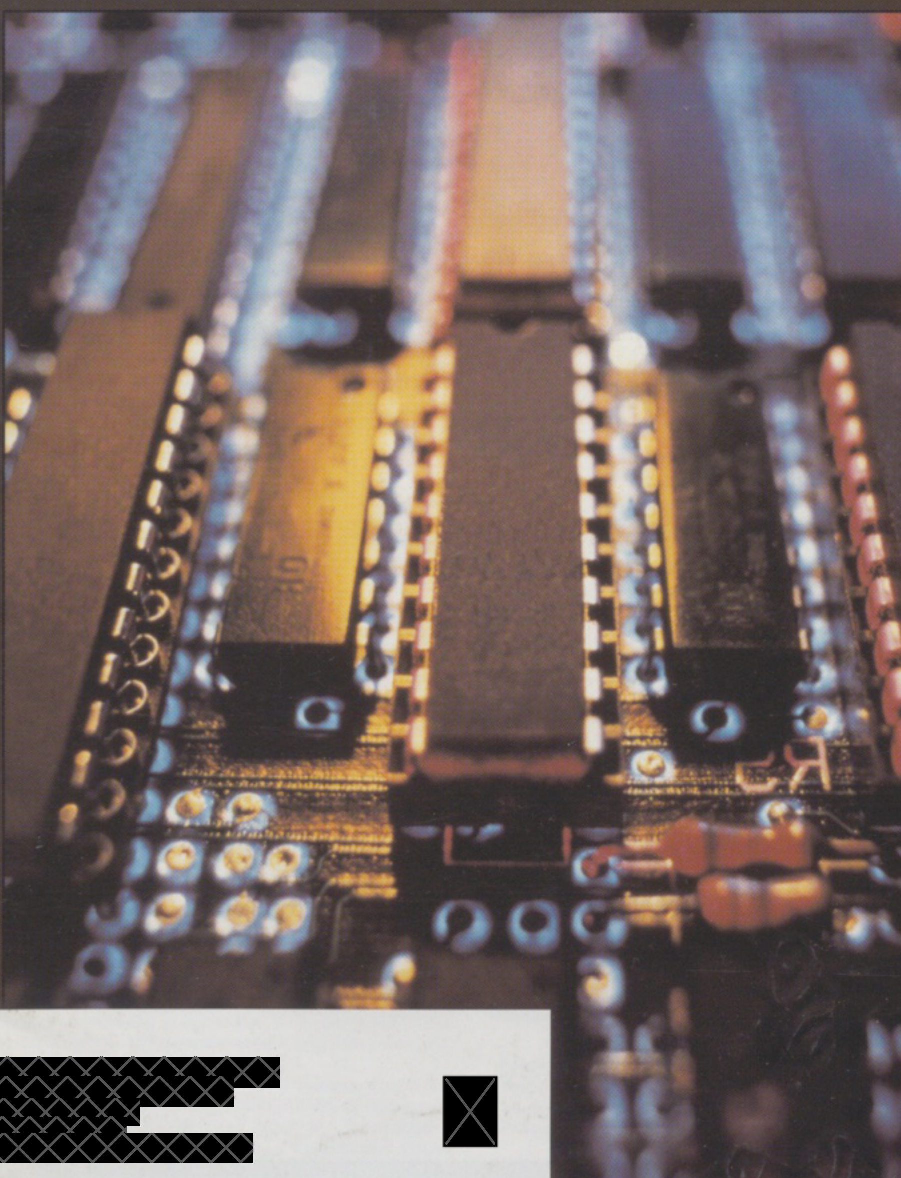
Information Technology AND Libraries

ISSN 0730-9295

Volume 21 Number 3

SEPTEMBER 2002



IN THIS ISSUE

- GIS and Mapping Technologies on the Web
- PHP: An Open Source Solution
- Designing Library Instruction Tutorials for the Web

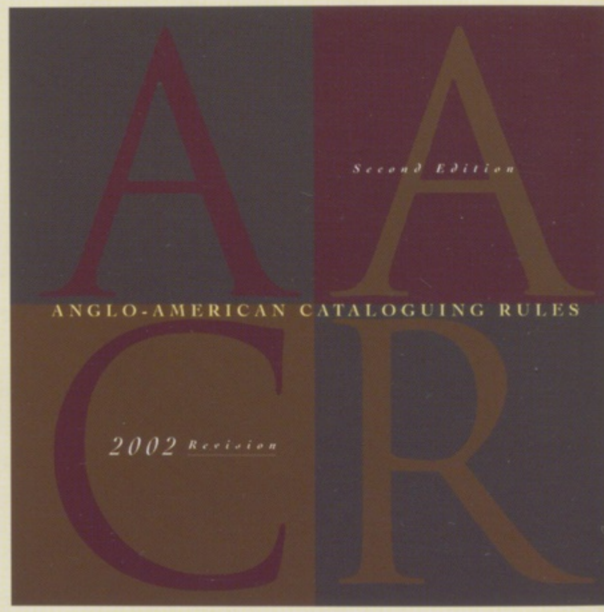
AACR2—now more than ever.

Anglo-American Cataloguing Rules

2nd Edition

2002 REVISION

Canadian Library Association
Chartered Institute of Library and Information Professionals
American Library Association



You've got an online catalog search station in your library and want to include the best and most up-to-date Websites for your customers, but what you learned in Cataloging 101 has not prepared you to create the record. What is the site's title? When was it published? In the Web environment, the answers to the standard questions are not always obvious. What do you do?

You can now turn to the 2002 Revision of AACR2!

In the digital world of 2002, with the process of cataloging more complex than ever, the Joint Steering Committee has come together to present an up-to-the-minute, forward-looking revision that will equip you to catalog any type of resource, print or electronic.

Now in a completely redesigned loose-leaf format, *AACR2* is more user-friendly than ever. With 8½" x 11" pages (that fit a standard 3-ring binder), separately numbered chapters (for easy integration of future updates), and brand new text design (that clearly distinguishes the rules from the examples), the *2002 Revision* is the powerhouse for resource description and access.

Why are the Rules as important as ever?

- They are the brains behind quality catalogs.
- They can be adapted to the full range of media formats, including digital content and Websites.
- Their tests for consistency make global information sharing possible.
- They have been developed with the changing needs of the end-user in mind.

What's new in 2002?

- Fully integrated 1999 and 2001 Amendments and revisions approved and finalized through 2002
- More than 100 substantive changes throughout the text and completely updated Index
- Three completely revised and updated chapters (Cartographic Materials, Electronic Resources, Continuing Resources)

AACR2-2002 KIT
(Loose-leaf pages with AACR Binder and Tabs)
Price: $87.00
ISBN: 0-8389-3529-X

AACR2-2002 PAGES
(Loose-leaf pages only)
Price: $62.00
ISBN: 0-8389-3530-3

Code: 2292

phone **866-SHOP-ALA** ■ fax **770-442-9742** ■ online **www.alastore.ala.org**

Information Technology and Libraries (ISSN 0730-9295) is published quarterly in March, June, September, and December by the American Library Association, 50 E. Huron St., Chicago, IL 60611. It is the official publication of the Library and Information Technology Association, a division of the American Library Association. SUBSCRIPTIONS: Subscription price, $30, is included in membership dues. Nonmembers may subscribe for $55 per year in the U.S.; $60 in Canada, Mexico, Spain, and other PUAS countries; $65 in other foreign countries. Send subscription requests to *Information Technology and Libraries*, Subscription Department, American Library Association, 50 E. Huron St., Chicago, IL 60611; 800-545-2433; fax: (312) 944-2641; subscription@ala.org. Single copies, $20. Periodical-class postage paid at Chicago, Illinois, and at additional mailing offices. POSTMASTER: Send address changes to *Information Technology and Libraries*, 50 E. Huron St., Chicago, IL 60611.

Information Technology and Libraries publishes material related to all aspects of libraries and information technology, including digital libraries, metadata, authorization and authentication, electronic journals and electronic publishing, telecommunications, distributed systems and networks, computer security and intellectual property rights, technical standards, geographic information systems, desktop applications, online catalogs and bibliographic systems, optical information systems, software engineering, universal access to technology, futuristic forecasting, library consortia, vendor relations, and technology and the arts. *ITAL* welcomes unsolicited manuscripts. Submissions should follow the guidelines stated under "Instructions to Authors" on page 92 of the June 2002 issue or on the LITA Web site at www.lita.org/ital/infoauth.htm. Manuscripts of feature articles, communications, and tutorials should be addressed to Dan Marmion, 221 Hesburgh Library, The University of Notre Dame, Notre Dame, IN 46556-5629. Books for review should be addressed to Tom Zillner, WILS, 464 Memorial Library, 728 State St., Madison, WI 53706. Software for review should be addressed to Andy Boze, Hesburgh Library, The University of Notre Dame, Notre Dame, IN 46556-5629.

As a matter of policy, *ITAL*, as the scholarly organ of LITA, does not review LITA publications.

PRODUCTION: ALA Production Services (Troy D. Linker, Kevin Heubusch; Ellie Barta-Moran, Angela Hanshaw, Kristen McKulski, and Karen Sheets), American Library Association, 50 E. Huron St., Chicago, IL 60611.

Publication of material in *Information Technology and Libraries* does not constitute official endorsement by LITA or the ALA.

Abstracted in *Computer & Information Systems, Computing Reviews, Information Science Abstracts, Library & Information Science Abstracts, Referativnyi Zhurnal, Nauchnaya i Tekhnicheskaya Informatsiya, Otdyelnyi Vypusk,* and *Science Abstracts Publications.* Indexed in *CompuMath Citation Index, Computer Contents, Computer Literature Index, Current Contents/Health Services Administration, Current Contents/Social Behavioral Sciences, Current Index to Journals in Education, Education, Library Literature, Magazine Index, NewSearch,* and *Social Sciences Citation Index.* Microfilm copies available to subscribers from Proquest/Bell and Howard, Ann Arbor, Michigan.

The paper used in this publication meets the minimum requirements of American National Standard for Information Sciences—Permanence of Paper for Printed Library Materials, ANSI Z39.48-1992.∞

Information Technology AND Libraries

Volume 21, Number 3 September 2002 ISSN 0730-9295

Abstracts and contents pages of recent issues of *ITAL* can be found on the LITA Web site at www.lita.org/ital/index.htm.

cover design Kevin Heubusch ● **interior design** Dianne M. Rooney

Editorial: Chin Up!

Dan Marmion

I want to step away from the world of library technology for the moment and tell you about an incredible experience I was privileged to undergo this summer. It's called the Frye Institute, and it's awesome.

The full name is the Frye Leadership Institute, and the following is on their Web site: "The challenges and changes of contemporary higher education have created the need for campus leaders with new competencies and perspectives—people who can bring a new framework to our historic enterprise. The purpose of the Frye Leadership Institute is to bring to tomorrow's higher education leadership the insights and understanding of the issues that will inform this framework, including academic, technology, economic, public-policy, student, and constituent-relations dynamics."[1] The Council on Library and Information Resources (CLIR), EDUCAUSE, and Emory University sponsor the institute, and the Robert W. Woodruff Foundation provides funding. The class of 2002 was the third class in Frye history.

One Sunday afternoon in early June, approximately forty people from universities all over the country, plus three outside the United States, converged on Emory University in Atlanta and began what would be for many a defining period in their career. We were mostly librarians and academic IT folks, with a few teaching faculty for good measure. Some were from big universities, others from tiny colleges, and one attendee came from a community college.

Our leaders were the co-deans: Richard Detweiler, president of Hartwick College, and Deanna Marcum, president of CLIR. The faculty consisted of about thirty-five veterans of various aspects of higher education, starting with Billy Frye, Chancellor of Emory University and the institute's namesake. We were the grateful recipients of wisdom from, among many distinguished faculty and listed here in no particular order: an associate provost from Brown University, the president of EDUCAUSE, the president of the University of Central Florida, the senior vice president for Government and Public Affairs at the American Council on Education, the president and CEO of the J. Paul Getty Trust, a professor of classics from Rhodes College, an associate vice president for IT from Wheaton College (and alumnus of the first Frye Institute), the provost at the University of Kansas, and the college librarian from the University of Virginia.

They started us off easy, with a reception followed by a dinner to help us begin getting acquainted. Little did we suspect what was in store for us. For the next two weeks, we were occupied almost twelve hours a day with organized activities. We listened and talked to the experts. We broke up into small groups and planned hypothetical institutions of higher education. We learned a lot about scholarly publication, copyright law, and metadata harvesting. We visited Emory University's Woodruff Library, where they are doing some really interesting things. We learned new words like Sunoikisis and deterritorialization, and new catch phrases like "re-entry strategy" and "chin up!" We Meyers-Briggs'd ourselves (I'm an ISTP, for those who care to know). We watched movie snippets to glean managerial nuggets. And we ate—boy, did we eat! Mostly, however, we interacted, which was the main point of the institute.

One of the intended outcomes of the Frye experience is the forging of a network of people who stay in touch and who become resources for each other. We have our own discussion list, FRYE2002, and we have space in Emory's Blackboard environment. It really is amazing how the forty of us bonded together and became a unit. We even discovered we have our own poet laureate, not to mention Graham, the toastmaster.

Finally it came to an end, and we had another big dinner, complete with presentations of certificates and Frye pins that we can wear with pride. But those are just mementos; what we really took away from Atlanta the summer of 2002 was an enlightenment, a sense that we can see the big picture, and we can lead the way. Chin up, everyone.

Finally, on an entirely different note, I call your attention to the article by Rachel Mendez that leads off the issue. Rachel was the winner of the second annual LITA/Endeavor Student Writing Award for her article "Hanging Indents and the Reference Librarian: Offering Productivity Software in the Public Library."

Dan Marmion (dmarmion@nd.edu) is Associate Director for Information Systems and Access at the University of Notre Dame Libraries, Notre Dame, Indiana.

Note

1. Frye Institute attendees are chosen based upon a number of criteria. For more information see the institute's Web site at www.fryeinstitute.org/about.asp. Nominations for the class of 2003 are due by Novermber 15, 2002.

Hanging Indents and the Reference Librarian: Offering Productivity Software in the Public Library

Rachel Mendez

[Editor's note: The following article is winner of the second annual LITA/Endeavor Student Writing Award.]

This article explores ways to expand the public library's mission, and that of the reference librarian, to include offering word processing on public access computers. The author defines access to and the ability to use word processing software as a form of literacy and links this to the library's established role in promoting literacy. This article also provides anecdotal information about introducing this software at a public library system.

In the last decade, much has been written about the digital divide that separates the technological "haves" from the "have-nots." As more information became available digitally, first on CD-ROM databases and then via the Internet, concerns grew over lack of access to valuable information sources for those without computers and Internet access. Public libraries took it as their duty to address this inequity by connecting public-access computers (PACs) to the Internet. This decision is relatively easy to justify; libraries are in the business, after all, of providing access to information regardless of format and providing access to the Internet is, in essence, providing access to information.

The Digital Divide: Take Two

There is another aspect of the digital divide, however, that is not so widely discussed nor so easily resolved; those without computers are unable to use today's powerful productivity software for word-processing, spreadsheets, and other applications. Without these tools, it is difficult to compete in today's working and academic worlds. Since these tools are now so widely used, there is an expectation that documents have a professional appearance. Documents such as letters to lawyers, school papers, story manuscripts for submission, résumés, and cover letters will not be taken as seriously if not produced on a computer and printed out neatly.

Libraries can address this need by providing productivity software on PACs, along with the Internet. In fact, many libraries do. However, there is some debate about this practice. "I did not go to library school to end up teaching people how to make hanging indents," said one indignant librarian, as her library prepared to add productivity software to PACs. Productivity software, the argument goes, is not information. If the mission of the public library is to make information accessible to all, does providing this software fit into that mission? This paper defines the ability to use word processing software as a form of literacy, and provides an argument for offering this software on PACs.

The Need: Literacy

> Literacy should be seen as a practice that enables people to understand and change their lives.
>
> —Paolo Freire, *Literacy: Reading, the Word, and the World*

Traditionally, literacy has meant the ability to read and write. In recent decades, this definition has evolved along with society's technological sophistication. The Workforce Investment Act of 1998 defines literacy as "an individual's ability to read, write, speak in English, compute and solve problems at levels of proficiency necessary to function on the job, in the family of the individual, and in society" (National Institute for Literacy 1999). The definition from the U.S. National Adult Literacy Survey is even broader: ". . . using printed and written information to function in society, to achieve one's goals, and to develop one's knowledge and potential" (Fitzgibbons 2000).

In 1999 the Clinton Administration expressed that those without the skills to "communicate effectively, solve problems, and use technology" will be unable to "succeed in the workplace, family, and community" (National Institute for Literacy 1999). The number of adults in the United States without these skills is estimated at 20 million, higher than the number in Australia, Belgium, Canada, Germany, Ireland, the Netherlands, New Zealand, Sweden, Switzerland, and the United Kingdom, according to a 1997 survey (National Institute for Literacy n.d. a).

According to Earl Shorris (2000), author of several books about poverty in the United States, there is another kind of illiteracy that also keeps people from leaving the "culture of poverty." The lack of cultural literacy is one of the deficiencies that keeps the poorest from becoming part of society and experiencing the power that all educated citizens possess. To help bring this form of literacy to those entrenched in poverty, Shorris initiated the Clemente Course. Participants in the Clemente Course are exposed to the humanities, such as great literature, concerts, and fine art. While learning about the humanities, students also learn to "think reflectively, to enjoy their own innate humanity to its fullest" (48).

Rachel Mendez (rockyshoe@earthlink.net) is a 2002 graduate of the School of Library and Information Sciences, Emporia State University, Kansas, and works as reference staff at the Multnomah County Library, Portland, Oregon.

Yet another form of literacy is computer literacy. In simple terms, to be computer literate is to be able to use computer technology. On a deeper level, computer literacy is "the ability to make use of computer technology for investigation of the world" (Jones 1996). Learning to use technology gives people "a certain level of self-confidence, acquired with an understanding . . . that much of the world is self-explanatory and . . . that [if you can teach yourself to use technology] you can teach yourself most anything you want to." Achieving computer literacy also helps the individual to acquire the skills mentioned previously, such as "using printed and written information to function in society" (Fitzgibbons 2000, 2).

Computer Literacy: More than Chatrooms and eBay

> Literacy is a concept, a process, and a skill that has meaning in relation to the demand of the economy and society or individuals. . . .
>
> —Shirley Fitzgibbons, *Libraries and Literacy: A Preliminary Survey of the Literature*

Understanding computer literacy requires a leap from considering computers as diverting, but ultimately unnecessary, technological luxuries to seeing them as key to an individual's economic and social development. In fact, as pointed out in the Urban League's 1998 report, *Losing Ground Bit by Bit: Low-Income Communities in the Information Age,* the high use of computers among mainstream Americans has intensified the inequities for those in poverty (Benton Foundation 1998). For example, a neatly hand-written high school paper was once acceptable, but now even elementary school children hand in perfect documents formatted on home computers. The proverbial playing field has grown more uneven.

Some scholars (Gallian and Greenfield n.d.) believe that computer literacy skills go beyond basic technological competence; the literate individual understands what technology makes possible. With this knowledge, the individual can use the computer to find and manipulate information as needed. Douglas (1999) agrees, noting that information literacy (the ability to find and use information) and computer literacy are linked. Lack of computer literacy can directly limit an individual's information literacy.

Dertouzos (1998) is among the many who believe computer literacy, combined with access to computers, can truly transform lives. He foresees computers helping the poor become literate, take care of their health, grow food, and sell their services and crafts, but only if access is possible.

Computer Literacy and the Information Agency

Computer literacy is the ability to "not only . . . locate and collect information but also to evaluate and apply it in responsible and significant ways" (Gallian and Greenfield n.d.). In other words, computer literacy can be seen as part of the information transfer cycle (ITC), and firmly established as within the purview of an information agency such as the public library.

ITC consists of the following steps: creation, production, dissemination, diffusion, utilization, organization, and preservation (or destruction) of information (Achleitner 1995). Nearly all of these steps can take place using productivity software. For example:

- Creation: writing a paper, poem, story, or essay
- Production: producing a flyer for a concert, formatting an artist's book or 'zine
- Dissemination: producing a paper, brochure, newsletter, or presentation
- Diffusion/utilization: taking information from research and using it in original or critical work
- Organizing: creating a spreadsheet
- Preservation: writing memoirs or meeting notes

If all steps of ITC occur when using productivity software, this software can be called an information tool, thus strengthening the connection with the purposes of an information agency.

What Does Literacy Have to Do with Libraries?

> Making information available is not enough. Making information useful is the key.
>
> —Karen Quinn, *Information Literacy: Learning How to Learn*

The promotion of literacy is not new to libraries; urban libraries in the 1890s offered classes in English and citizenship to the growing population of immigrants. In the '60s and '70s, librarians increasingly regarded social responsibility, including literacy activism, as a professional duty (Fitzgibbons 2000).

Some library professionals argue that libraries should not be involved in literacy. White describes the idea of literacy as a library goal as a mission foisted on us "by nonlibrarians in the federal beaucracy" (2000, 30). According to White, supporting literacy is "neither an expertise we possess as a result of our master's degrees, nor an expertise worth claiming" (2000, 30).

In a report from a focus group on literacy in libraries, Rodger quotes some librarians from the opposition:

"Libraries are not educators. Education is a discipline that provides ongoing work with somebody. Literacy is a connection to what we do well . . . but we provide resources, not training" (1999).

On the other hand, if we consider literacy as a specialty, akin to youth services, it makes sense to become involved in this training. Teaching finger plays to children, reading stories, and singing "The Noble Duke of York" are activities similar to providing literacy services in that both types of activities work towards ushering patrons into the world of reading. In addition, both create new library customers. On a basic level, if an individual cannot read, what need does that individual have for a library? By promoting literacy, a library expands its customer base, and increased circulation is one factor that can lead to stable or increased funding. Literacy, then, is a self-serving issue for libraries.

Assuming that the chief mission of the library is to provide information, literacy services are still justifiable. If one is unable to use information due to illiteracy, the library is prevented from fulfilling its mission to all members of society. In order to move toward the goal of universal information access, the library must offer not just literature, but also the means to become literate.

If we consider public libraries the university of the people, as many have, we can turn to Ranganathan for inspiration. Among Ranganathan's tenets of the library's mission are helping all patrons in "perpetual self-education" and increasing the economic ability of all people to support their children and maintain lives of comfort, free from want (1996, 171). The promotion of literacy is certainly one way to meet these goals.

Although other institutions, public schools, and community colleges for example, often offer literacy programs, libraries are uniquely positioned to offer these services (Andersen 1998):

- Libraries are an accessible part of the community's infrastructure.
- Libraries have public buildings throughout the community.
- Libraries diagnose community needs.
- Libraries have (or should have) connections with various community institutions.
- Libraries have resources (books, software) to enhance literacy studies.
- Libraries are, to many, more approachable than other institutions. An adult might be embarrassed to take literacy classes at a children's school. An illiterate adult may feel too intimidated to take a literacy class at a college or university.
- Librarians are skilled at repackaging information to meet the needs of customers.
- Library schools, already focused on training librarians in public service, could be encouraged to offer units on basic education.

Today, many libraries tackle the problem of literacy. The National Institute for Literacy estimates that seven thousand United States libraries currently offer such literacy services as English as a second language instruction, family literacy, computer instruction, group and individual tutoring, collections targeted at low-level readers, and information about local literacy programs (National Institute for Literacy n.d. b).The American Library Association (ALA) actively supports literacy in libraries, pronouncing literacy services as essential to achieving the goal of helping people develop the skills needed in today's information society (ALA 2001).

Computers: The Need

The gap between the number of computer owners and those without computers still exists. As seen in table 1, race, education, and income all greatly influence the incidence of computer ownership. Nevertheless, some writers believe that the digital divide is greatly exaggerated. Thierer (2000) points out that most Americans own televisions, and claims that an average television costs more than an entry-level personal computer (PC). If there is a digital divide, why can people afford televisions, but not PCs?

A quick consultation with consumer sources, however, reveal deficiencies in Thierer's argument. The Wal-Mart Home Entertainment Gift Guide for December 2–24, 2001, features a color television with a twenty-five-inch screen for $189 and an entry-level PC for $898, a difference of $709. Additionally, a television is truly "plug and play," requiring no special knowledge to operate. Conversely, a computer, no matter how simple and user friendly, can be intimidating to a novice.

However, there *is* good news. During the last two years of the twentieth century, a temporary golden age of economic optimism, more people became computer owners. In the years 1998 to 2000, the number of computer owners in every type of household grew. For example, the most recent statistics from the U.S. Department of Commerce's Falling through the Net Project (2000) indicate that computer ownership has increased 4.7 percent in households making less than $15,000, the lowest income bracket.

On the other hand, four-fifths of the households from this income bracket are still without a computer. Further, although more Black and Hispanic households had computers in 2000, the 18.4 percent gap between the number of Black households with computers and the national average is not statistically different than the gap existing in 1998. The 17.3 percent gap between Hispanic computer owners and the national average has not changed either. One must also wonder if computer ownership increases will continue, with the United States officially in a recession and unemployment rates rising daily.

Table 1. Statistics from *Falling through the Net* (2000)

	Percentage of Computers by Household
Race	
Asian/Pacific Islander	65.6
White	55.7
Hispanic	33.7
Black	32.6
Annual Income	
$75,000 and above	86.3
$50,000–74,999	73.2
$35,000–49,999	58.6
$25,000–34,999	44.6
$15,000–24,999	30.1
Below $14,999	19.2
Education	
Post graduate	79.0
College graduate	74.0
Some college	60.3
High school graduate	39.6
Less than high school	18.2

In the 1990s, welfare reform legislature limited the amount of time an individual or family could receive welfare benefits. Welfare recipients, expected to find jobs to replace government support, are only allowed twelve months of job skill training. Imagine an individual, finished with one year of computer training, with no home computer on which to practice newly learned skills. The library, by providing PACs loaded with the most commonly used software, assists these individuals in maintaining or developing skills needed to transition from welfare benefits to employment.

If we look to the public to answer the question of why computers are needed, we see a variety of uses. Patrons surveyed at libraries offering word processing PACs list the following computer uses (Gordon, Gordon, and Moore 2001b):

- Research and type term papers to earn Theology degree
- Complete college degree
- Create current résumé and keep computer skills sharp while unemployed
- Complete school assignments
- Satisfy educational needs and explore the computer
- Assist with college assignments and personal projects
- Keep in the social and cultural loop
- Learn Word and Excel
- Write letters and text for stamp club presentations and spreadsheets for stamp collections

Libraries Can Do It!

> An overwhelming 85 percent of Americans think it is important . . . for libraries to provide computers . . . to children and adults who don't have their own.
>
> —Laura Weiss, *Buildings, Books, and Bytes*

Libraries have the buildings, the infrastructure, the reputation, and often already have PACs. They have staff trained to be responsive to the public. Libraries have a commitment to providing literacy services. They have a public that looks to them to help meet their information needs. Libraries have a commitment to their communities to provide services to all. They can take one further step and provide productivity software on PACs.

The Bill and Melinda Gates Foundation's U.S. Libraries Project helps libraries establish computer services for the public. With grants for PC purchase and staff training, the foundation has helped thousands of U.S. libraries offer computers to their patrons (Rava 2000). In order to qualify for a grant, 10 percent of a library's community must live at or below the poverty level.

Some libraries are partnering with Community Technology Network, a nonprofit organization, to offer computer workstations in the libraries, or to bring library staff and resources to community technology centers (Williams n.d.). Other libraries partner with local schools and community colleges to maximize limited funding and resources.

For some libraries, funding is not as much of an obstacle as is resistance by staff and administration. In these libraries there is sometimes a belief that offering productivity software is beyond the scope of the public library. The Public Library Manifesto of the United Nations Educational, Scientific, and Cultural Organization (UNESCO) clearly states, however, that public libraries should fill the public's "need for all relevant materials including modern technology" (Niegaard 1994).

How to Make It Happen: Anecdotal Experience

At City Library System, an anonymous library system with fifteen branches and a large central library upon which the following experiences are based, patrons have had Internet access for five years. One year ago the library opened a computer lab with limited hours in one of its branches, thanks in part to a Gates Foundation grant. Other than this lab, there was no productivity software on the library's PACs. Concerned about the number of requests for these services, and by the fact that many similar-sized libraries were offering these services, City Library decided to offer these services.

A technology committee discussed the issue over the course of several meetings. After discussing the feasibility of adding this software with the computer department and checking with other library systems, the committee decided to offer these services at two branches on a trial basis.

A small committee of branch supervisors and reference staff was formed. This group coordinated staff and public issues related to the new software. Another small committee, made up of computer department staff, worked on the technology issues. The two groups communicated through email, Intranet, and in committee meetings.

The technology issues took months of planning and several days to implement and troubleshoot. Since the library uses the Pharos system to limit the amount of time that patrons have on the PACs, technology staff had to install Microsoft Office software to be compatible with the current set-up and congruent with current security measures. Despite the complications, the technology rollout happened on time.

The staff and public issues committee addressed the following issues:

- Training for staff and volunteers in using and troubleshooting Microsoft Office products
- Reference books for staff and public, printed "how-to" literature for the public
- Staff morale—some staff were worried about giving new duties to already busy librarians while others worried that offering these services distorts the mission of the library and the role of the reference librarian
- PACs are heavily used, and offering new services may force patrons to wait longer
- Evaluating the services, before and after, to assess impact on staff and areas for improvement

The training sessions for staff were based on training classes previously given by the computer lab coordinator. Drawing on her experience of the lab, the training sessions included common troubleshooting skills, and common patron questions, as well as a basic review of Word, Excel, and PowerPoint applications. Volunteers who help patrons with computers were invited to join the training sessions, but, as a whole, already had high computer skills and none signed up.

The committee ordered a variety of manuals for PowerPoint, Excel, and Word, in both English and Spanish. Emphasis was placed on books with visual format for quick reference, but the collection also includes at least one thick, definitive guide for each application. The collection attempted to reflect a variety of learning styles. These non-circulating books were kept at the reference desk. In addition, the committee made available laminated quick guides for each application. These sheets, provided by the Gates Foundation, are extremely simple guides showing the parts of a toolbar and how to open a new document. Several months after implementation of the new software, these collections were rarely used by patrons, although some staff members consulted the books.

Some staff were glad to offer this service to the patrons who had requested it so often. Others were relatively neutral and didn't notice much difference in their workloads. Others, initially distressed about the new service, later claimed that the stress they anticipated never quite materialized.

Those opposed to offering productivity software on PACs were of two, sometimes overlapping, camps. The first camp is made up of reference staff who were not themselves technologically adept. Offering new computer services means offering some level of support to patrons using these services; these staff members did not feel capable of offering this support. Despite group and individual training, these workers felt overwhelmed and unprepared. They felt that suddenly their jobs required skills they did not posses.

The other group of reference staff members opposed to offering productivity software are those who felt the library mission, not to mention the job description of reference staff, does not include offering glorified typing facilities.

There is little that can be said to change the minds of those opposed to offering productivity software. It is useful, though, to recall other technological changes that have rocked libraries in the past. When the library catalog became automated, some staff were dismayed. Offering the Internet was another change that was difficult for library staff. Many libraries initially offered Internet, but prohibited e-mail (some still do). Libraries now recognize legitimate information needs for e-mail. It seems likely that productivity software will follow the path blazed by these other innovations.

Problems and Challenges

To successfully offer productivity software, libraries need to anticipate the challenges that may affect their staff and the public. The main problems are the possible increase in staff stress and workload, the quality of assistance available, and the limited number of PACs available. In addition, funding, as always, can be a problem.

Critics and Malcontents

It is tempting to dismiss the grumbling of those staff members who believe that productivity software does not belong on PACs. We can call them technophobes or accuse them of being elitist in claiming one service more important than another. Nonetheless, we would do well to remember that unhappy staff might not offer the best public service. Worrel points out that "resistance to change is inevitable . . . critics and malcontents must be made an integral part of the planning and implementation process, so that their complaints are dealt with openly. . . . Managers must ensure that employees have the emotional support and the necessary training to adjust to new work arrangements" (1995, 357). Libraries can embrace new technologies and new services, but the feelings and attitudes of staff must not be forgotten (Stapley 1996).

Training is certainly part of staff preparation, but equally important is the emotional reaction of staff members. Management should engage front-line staff in dialogues about the reasons for the proposed changes. Staff should feel safe to express negative opinions and management should be prepared to offer solid explanations and answer tough questions. Although total buy-in is an impossible goal, informed staff who feel their concerns are listened to may be more likely to accept unwanted change with grace. Listening to criticisms from staff members should go deeper than hand-holding. Worrell quotes Kramlinger in noting the value of feedback from staff members: "Everyone can be a source of useful ideas. . . . The people closest to the problem usually have the best ideas about solutions. . . . Library managers can learn much from clerks, paraprofessionals, and lower-level librarians. . . . The process of open dialogue improves ideas" (1995, 354). Figure 1 lists several reasons not to offer word processing (Orr 2001). Even if management decides to implement productivity software, the points raised by the dissenting librarian should be taken seriously. Is there a way to create more seating for patrons not interested in computers? Can more community activities be offered? Can volunteers be recruited to teach patrons clerical skills, leaving the librarians free to offer reference services?

But I Don't Know How!

Trying to help the public with productivity software will be nothing but frustration for staff members unacquainted with the software itself. Although classes and tutoring sessions are necessary, the most valuable training tool is practice. In the evaluation of the Gates Library Project, staff at small or underfunded libraries complain that they themselves have no access to PCs with productivity software (Gordon, Gordon, and Moore 2001). It is clear that staff must be encouraged and enabled to use the software themselves if they are to be able to assist patrons.

- PACs are already full of patrons using chat rooms and playing games on the Internet. There are not enough computers for searching legitimate Internet resources or even the library's catalog.
- Since so much space has been allocated for PACs, there is less room for people to sit and read, less room for displaying books.
- There are many more dynamic ways of using library funds and staff. For example, offering more community activities, supporting local writers and cultural activities.
- Word processing does not promote a love of books and reading.
- Reference staff should not be spending their time helping patrons perform clerical tasks, but rather with offering the best reference service they can.
- Patrons can go to job centers or other locations to type résumés and cover letters.
- It would be more worthy of librarians' time and expertise to work on creating online readers' advisory, search guides, and offering search training to small groups.
- Librarians may not know how to use the software themselves and they are not there to teach clerical skills to patrons.

Figure 1. Some Reasons for Not Offering Productivity Software on PACs

The Long Wait

Most libraries with Internet-connected PACs know that demand usually outstrips supply. Some libraries use waiting lists and often end up policing patrons to ensure that everyone behaves. When possible, volunteers can be used to handle this task, freeing reference staff to concentrate on other duties. Other libraries use time-out software that automatically shuts off the PAC when the patron's time is up.

Whatever method is used, there is no satisfactory way of ensuring that every patron has immediate access to a PAC. If the staff is apprised of this likelihood, plans can be made ahead of time. Staff can brainstorm about how to handle disgruntled patrons and come up with ways to explain the shortage of PACs.

Money, Money, Money

Libraries fortunate enough to receive Gates Foundation money will be able to purchase PCs and software and have access to training and support materials. However, the money is finite, and libraries bear the ultimate responsibilities of maintaining the computers, upgrading them, and, eventually, replacing them. In addition, continuing education for staff and providing adequate staffing to handle the likely increase in library traffic (Gordon, Gordon, and Moore 2001c) will require continuing expenditure.

On the positive side, administrators can point to increased library use and a broader patron base when requesting adequate funds. Administrators surveyed by the Gates Foundation (Gordon, Gordon, and Moore 2001a) report an average 23 percent increase in patron traffic and a 15 percent increase in book circulation after introducing PACs with Internet access and productivity software. Serving more of the community more of the time means that libraries are more necessary and funding them is more important than ever.

Ray Bradbury and Productivity Software

Ray Bradbury's classic book, *Fahrenheit 451*, is a perpetual favorite among high school teachers and library science professors. In Bradbury's imagined future world, books are forbidden, and the job of firemen is to burn hidden stashes of contraband literature. *Fahrenheit 451* is a rallying cry against censorship and a warning about the death of intellectual thought. Bradbury wrote this book on typewriters in the basement of his local library (Charles 2000). Where would today's budding author type the manuscript that might become tomorrow's classic literature?

Melodrama aside, Bradbury's experience is a valuable touchstone for libraries to consider. If we have the ways and means to provide productivity software to our patrons, we must take that step. In doing so, we encounter true difficulties, but we also reinforce the public's belief that the public library is the community's learning center. Included in appendix A are several quotes to inspire the librarian charged with crafting a persuasive justification.

Works Cited

Achleitner, Herbert K. 1995. Information transfer, information technology, and the new information professional. In *The impact of emerging technologies on reference service and bibliographic instruction*, ed. by Gary M Pitkin. Westport, Conn.: Greenwood.

American Library Association. 2001. Leading the way for literacy. ALA Office for Literacy and Outreach Services OLOS. Accessed Dec. 8, 2001, www.ala.org/olos/adultlit.html.

Andersen, Josephine. 1998. Social responsibilities discussion group paper: Literacy in libraries. International Federation of Library Associations and Institutions. Accessed Dec. 8, 2001, www.ifla.org/VII/dg/srdg/srdg3.htm.

Benton Foundation in association with the National Urban League. 1998. Losing ground bit by bit: Low-income communities in the Information Age. Accessed Dec. 8, 2001: www.benton.org/Library/Low-Income.

Charles, Ron. 2000. National Book Award winners announced. *Christian Science Monitor*. Accessed Dec. 8, 2001, www.csmonitor.com/sections/books/p-nba2000.html.

Dertouzos, Michael L. 1998. The rich people's computer? *Technology Review* 102, no. 1: 22.

Douglas, Gretchen V. 1999. Professor librarian: A model of the teaching librarian of the future. *Computers in Libraries* 19, no. 10: 24.

Fitzgibbons, Shirley A. 2000. Libraries and literacy: A preliminary survey of the literature. International Federation of Library Associations and Institutions. Accessed Dec. 8, 2001, www.ifla.org/IV/ifla66/papers/122-139e.htm.

Gallian, Judith M., and Jerry Greenfield. n.d. Benchmarks for computer literacy: An intersection of liberal arts and applied information science. Miyazaki International College, Japan. Accessed Dec. 8, 2001, www.miyazaki-mic.ac.jp/faculty/Jgallian/JALT98/WCALL_Paper.html.

Gordon, Andrew, Margaret Gordon, and Elizabeth Moore. 2001a. Library staff support public access computing: Experience increased workloads, stress, and satisfaction. A report to the Bill and Melinda Gates Foundation U.S. Library Program: On a survey of library staff, volunteers, and administrators in five states. Bill and Melinda Gates Foundation. Accessed Dec. 8, 2001, www.gatesfoundation.org/libraries/uslibraryprogram/evaluation/default1.htm.

———. 2001b. Library patrons heavily use public access computers and other library services, and want more: A report to the Bill and Melinda Gates Foundation U.S. Library Program: On a survey of library patrons in five states. Bill and Melinda Gates Foundation. Accessed Dec. 8, 2001, www.gatesfoundation.org/libraries/uslibraryprogram/evaluation/default1.htm.

———. 2001c. Library administrators confirm value, complexities of public access computing: A report to the Bill and Melinda Gates Foundation U.S. Library Program: On a survey of library administrators in five states. Bill and Melinda Gates Foundation. Accessed Dec. 8, 2001, www.gatesfoundation.org/libraries/uslibraryprogram/evaluation/default1.htm.

Jones, Bruce. 1996. Computer Literacy. Accessed Dec. 8, 2001, http://communication.ucsd.edu/bjones/comp_lit_paper.html.

National Institute for Literacy. n.d. a. Reading facts. Accessed Aug. 20, 2002, www.nifl.gov/facts/reading_facts.html.

National Institute for Literacy. n.d. b. Fact sheet. Accessed Aug. 20, 2001, www.nifl.gov/newworld/LIBRARY2.HTM.

National Institute for Literacy. 1999. Policy update: Vice president announces new literacy effort to prepare workers for the twenty-first century. Accessed Dec. 8, 2001, www.nifl.gov/nifl/policy/updates/99-01-12.html.

Niegaard, Hellen. 1994. UNESCO's 1994 Public Library Manifesto. International Federation of Library Associations and Institutions. Accessed Dec. 8, 2001, www.ifla.org/IV/ifla60/60-nieh.htm.

Orr, Laura. 2001. Unpublished manuscript. Quoted with permission of author.

Ranganathan, S. R. 1986. *Reference service*. Bangalor, India: Sarada Ranganathan Endowment for Library Science.

Rava, Carol. 2000. Partnership with North American libraries gives millions access to the Internet: An independent assessment of the effort shows positive impact for libraries and patrons. Bill and Melinda Gates Foundation. Accessed July 10, 2002, www.gatesfoundation.org/libraries/uslibrary program/announcements/announce-330.htm
Rodger, Joey. 1999. Leadership, libraries, and literacy programs: A report of focus group research. Urban Libraries Council. Accessed Dec. 8, 2001, www.urbanlibraries.org/standards/leadship.html.
Shorris, Earl. 2000. Promoting the humanities, or: How to make the poor dangerous. *American Libraries* 31, no. 5: 46–49.
Stapley, Lionel. 1996. *The personality of the organization*. London: Free Association Books.
Thierer, Adam D. 2000. Is the "digital divide" a virtual reality? *Consumers' Research Magazine* 83, no. 7: 16.
United States Department of Commerce. 2000. Falling through the Net. Accessed Dec. 8, 2001, www.ntia.doc.gov/ntiahome/fttn00/ contents00.html.
White, Herbert S. 2000. *Librarianship—Quo vadis? Opportunities and dangers as we face the new millennium*. Englewood, Colo.: Libraries Unlimited.
Williams, Cary. n.d. Library and Community Technology Access Project. Community Technology Centers Network. Accessed July 10, 2002, www.ctcnet.org/libprop.html.
Worrell, Diane. 1995. The learning organization: Management theory for the information age or new age fad? *Journal of Academic Librarianship* 21, no. 5: 351.

Appendix A. More Food for Inspiration

Literacy

> Literacy is a mode of behavior, which enables individuals and groups to gather, analyze and apply written information to function in society.
> Communities have a responsibility for creating a culture of literacy for their members, if they value development.
>
> —Shirley Fitzgibbons, *Libraries and Literacy: A Preliminary Survey of the Literature*

> Literacy is viewed as an avenue to human liberation and a catalytic force for social change.
>
> —Barbara Kwasnick, *Information Literacies for the Twenty-first Century*

Librarians

> A librarian should be much more than a keeper of books; he should be an educator.
>
> —Otis Robinson, "Librarians and Readers"

Libraries

> The public library as a passive, archival institution has been reshaped into a learning, information hub of the community.
>
> —Renee Tjoumas, *Information Literacies for the Twenty-first Century*

> [Public libraries should be involved in:]
> - supporting both individual and self-conducted education as well as formal education at all levels;
> - providing opportunities for personal creative development;
> - stimulating the imagination and creativity of children and young people; [and]
> - facilitating the development of information and computer literacy skills.
>
> —UNESCO Public Library Manifesto

Computer Literacy

> The microcomputer is an information technology tool and it is the responsibility of libraries to provide information.
>
> —Alan E. Guskin, Carla J. Stoffle, and Barbara E. Baruth, *Library Future Shock*

> For African Americans and other minorities, any lack of technological expertise could create additional barriers and make an already difficult job search more arduous.
>
> —Tariq K. Muhammad, "Career Paths to the Next Millennium"

> Computers and the Internet must be available to all, regardless of ethnicity or geography or income. Not because the technology is somehow special or revolutionary, but exactly because it's so ordinary. Nobody expects the ordinary, familiar features of our national infrastructure—public libraries, telephones, highways, public transportation, immunizations, or post offices—to end poverty or to bestow social and economic equity. They're just simple, basic tools for living in the modern world. Everybody should have them. That's why they're important, and that's why they deserve public funding.
>
> —Albert Fong and Josh Senyak, "Bridging the Digital Divide"

Appendix B. Related Works

Argyris, Chris. 1994. Good communication that blocks learning. *Harvard Business Journal* (July/August).
Baker, Nicholson. 1994. Discards. *New Yorker* 70, no. 7: 64–70.

Bascom, Brooke, and Bill and Melinda Gates Foundation. Libraries are connecting millions to the Internet. Accessed Dec. 8, 2001, www.digitaldividenetwork.org/content/stories/index.cfm?key=118.

Bunge, Charles. 1983. The personal touch: A brief overview of reference services in American libraries. In *Reference service: A perspective*, ed. by Sul Lee. Ann Arbor, Mich.: Pierian Pr.

———. 1986. Potential and reality at the reference desk: Reflections on a "return to the field." In *Reference and information services; a reader for today*, ed. by Bill Katz. London: Scarecrow.

Clayton, Peter. 1992. Japanese management theory and library administration. *Journal of Academic Librarianship* 18, no. 5: 298.

Cullings, Karen S. 1996. The public library as cornerstone of the community. *Computers in Libraries* 16, no. 1: 22.

Eisenberg, Michael, and Doug Johnson. 1996. Computer skills for information problem solving: A curriculum based on the big six skills approach. Clearinghouse on Information and Technology, EDO-IR-96-04. Accessed Dec. 8, 2001, http://ericit.org/digests/computerskills.shtml.

Friedman, Pamela. 1999. Post-secondary education options for low-income adults. *Welfare Information Network Issue Notes* 3, no. 12. Accessed Dec. 8, 2001, www.welfareinfo.org/postseced2.htm.

Godwin, Peter. 1997. IT flagship or word-processing shed? The LPC at South Bank University. *MM/T* 24, no. 3.

Gordon, Margaret, Andrew Gordon, and Elizabeth Moore. New computers bring new patrons. *Library Journal* 126, no. 3: 135–39.

Harvey, Brian. 1983. Stop saying, "computer literacy!" University of California, Berkeley. Accessed Dec. 8, 2001, www.cs.berkeley.edu/~bh/stop.html.

Himmelfarb, Gertrude. 1999. Revolution in the library. *Library Trends* 47, no. 4: 612–20.

Katz, Bill, ed. 2000. *New technologies and reference services*. Binghamton, N.Y.: Haworth Information Pr.

Katz, Bill, and Ruth Fraley, eds. 1982. *Ethics and reference services*. New York: Haworth.

McCook, Kathleen de la Peña. 2001. Poverty, democracy, and public libraries. In Kranach, Nancy. 2001. *Libraries and democracy: The cornerstones of liberty*. Electronic version. School of Library and Information Science, University of South Florida, Tampa. Accessed Dec. 8, 2001.www.cas.usf.edu/lis/faculty/PDandPL.html.

National Center for Family Literacy. 2000. Family literacy policy. Accessed Dec. 8, 2001, www.famlit.org.

National Institute for Literacy Policy Update. 2001. Community Technology Centers can help bridge the digital divide for adult education and literacy programs. Accessed Dec. 8. 2001, www.nifl.gov/nifl/policy/updates/01-01 11.html.

National Literacy Act of 1991. 1991. Public Law 102-73, 102d Congress—1st Session. Accessed Dec. 8, 2001, www.nifl.gov/publiclaw.html.

Nerad, Shava. 1999. Libraries and the analog divide. *Community Technology Review* (summer/fall). Community Technology Centers' Network. Accessed Dec. 8. 2001, www.civicnet.org/comtechreview/analogdivide.htm.

Quittner, Joshua. 1996. The web grows in Brooklyn. *Time* (Aug. 19) 148, no. 9: 53.

Rader, Hannelore B. 1988. Information literacy in the reference environment: Preparing for the future. In Blake, Virgil P., and Renee Tjoumas. 1990. *Information literacies for the twenty-first century*. Boston: G. K. Hall.

Rafaill, William. 2001. Technology literacy at Georgetown College: A strategy to address the SACS Requirement. Georgetown College. Accessed Dec. 8, 2001. http://spider.georgetowncollege.edu/wsr/literacy.htm.

Reynolds, John F. 1988. From orality to electronic technology. In Blake, Virgil P., and Renee Tjoumas. 1990. *Information literacies for the twenty-first century*. Boston: G. K. Hall.

Roach, Ronald. 1997. Virginia Tech spearheads HBCU computer consortium. *Black Issues in Higher Education* 19, no. 11: 30.

Rose, Susan. 1997. The role of community access centers in bridging the technology gap. CTCNet. Accessed Dec. 8, 2001, www.ctcnet.org/rose/00title.htm.

Rothstein, Samuel. 1953. Development of the concept of reference services in American libraries, 1850–1900. In Katz, Bill, and Charles Bunge, 1989. *Rothstein on reference . . . with some help from friends*. New York: Haworth.

Schamber, Linda. 1990. The role of libraries in literacy education. ERIC Digest. Accessed Dec. 8, 2001, http://ericae.net/db/edo/ED330373.htm.

Stapley, Lionel. 1996. *The personality of the organization*. London: Free Association Books.

Thomas, S. E. 2000. The necessary library revolution in community college developmental and remedial programs. *Community and Junior College Libraries* 9, no. 2: 47.

Twelve ways libraries are good for the country. 2000. *American Libraries* online. Accessed Dec. 8. 2001, www.ala.org/alonline/news/12ways.html.

Walker, Robert J. 1997. Computers should be made available to children of low-income families. *USA Today* 126, no. 2,628 (Sept.): 64.

Welfare Information Network. n.d. Education and Training. Accessed Dec. 8, 2001, www.welfareinfo.org/edandtraining.htm.

Wijetunge, Pradeepa. 2000. The role of public libraries in the expansion of literacy and lifelong learning in Sri Lanka. *New Library World* 101, no. 3. Electronic version Accessed Dec. 8, 2001. http://rosina.emeraldlibrary.com/vl=3676625/cl=50/nw=1/rpsv/~1181/v101n3/s2/p104.

Williams, Cary. 1998. Computer centers as library outposts. *Community Technology Center Review* 4, no. 1. Accessed Dec. 8, 2001, www.ctcnet.org/r981cary.htm.

Wooldridge, Adrian. 1997. The backlash begins: Management consultancy survey. *Economist* 342, no. 8,009: 20.

Worrell, Diane. 1995. The learning organization: Management theory for the information age or new age fad? *Journal of Academic Librarianship* 21, no. 5: 351–57.

Tapping the Web for GIS and Mapping Technologies: For All Levels of Libraries and Users

Kimberly C. Kowal

Numerous types of geographic information management technologies are currently available on the Web, ranging from simple tools for viewing maps to more complex systems such as Geographic Information Systems (GIS). The term GIS is commonly applied to all automated mapping. However, it specifically describes a system of computer hardware, software, and geographically referenced data, designed to capture, store, edit, display, and plot spatial information.

Geographic Information Systems (GIS) merge the graphic features of a map with its associated data, and because of this, in-depth analyses of geographic relationships are possible. GIS is often defined by its method of storing data and features in layers, allowing users to overlay various types of information to view simultaneously. The number of automated mapping tools that share some, but not all, of these capabilities is increasing, and more and more information is being made available using geographic technologies.

Understanding the capabilities, benefits, and limitations of these tools can greatly enhance reference service in both traditional and virtual environments. In fact, familiarity with some of the geographical resources on the Web can answer many common reference inquiries. For example:

- Maps effectively illustrate data. For instance, if a researcher is seeking the latest population numbers for a particular neighborhood, a map of that area in the context of the entire city with associated statistics is more helpful. Similarly, a map of Superfund locales or toxic release sites can be more effective than a listing, as a visual display of the highest concentrations of harvested croplands or forest cover types is preferable to a description.
- Maps act as alternative finding aids, particularly when features are more easily identified graphically. GIS technologies commonly have clickable maps that link locational information to pertinent data. For instance, a prospective homebuyer can research what houses in a neighborhood last sold for and when by selecting the properties. A hiker can check the land elevation of the area before setting off on a trip, or a citizen can find details on the aquifer supplying municipal water, all by pointing at a map.
- Geographic technologies can combine several variables in a single, custom-made map. For instance, public officials can not only assess the adequacy of public transport to local health facilities by overlaying city bus lines onto a map of clinics and hospitals, but they can also map their districts according to race and income levels from the latest census. Similarly, a researcher can check mortality rates in the United States by specific types of cancer, and this according to race, gender, or a number of other factors.

GIS Resources: A Profile of Types Available

GIS and mapping technologies can refer to a number of different types of Web tools, all serving different functions. Because of the range of capabilities and limitations, the various mapping technologies currently available on the Web can be roughly separated into three categories according to user requirements, technology, and application capabilities in the library: high level, midlevel, and low level.

The high-level user has specific needs, is proficient with computer use, and is rather well informed about the type of information she or he seeks. Unsurprisingly, the technology required is also the most advanced; GIS software is used off-line, and the Web functions as a means to locate and retrieve data from databases of geospatial metadata. The Internet is also often used at this level to find and download additional GIS applications. The midlevel user is comfortable with using the Web and has some specific questions, but needs assistance locating relevant information sources. Technologically, the tools at this level represent the growing number of dynamic map applications available on the Web. These applications are structurally simplified GIS and have some of its capabilities, but are designed for very limited purposes, such as driving directions, environmental conditions, transportation routes, or any other type of geographical information. The low-level user has a simpler information need that does not require in-depth knowledge of electronic mapping. Databases of static images that include maps constitute this type of technology; though functionality and user-interaction are minimal, it is most appropriate for many map needs. A few examples will be cited for each of the three levels; however, in-depth descriptions and evaluations of the more prominent resources may be found in the library literature and online directories and guides.[1]

High Level

"I want to make a map of outdoor camping facilities in Minnesota by county using this table of data I collected." This request exemplifies a need answered by high-level

Kimberly C. Kowal (kkowal@umn.edu) is the Assistant Map Librarian at the John R. Borchert Map Library of the University of Minnesota Libraries, Minneapolis.

GIS tools. The patron will need (1) georeferenced spatial files of the state, with county boundaries; and (2) access to GIS software, such as ArcView or MapInfo. The former can be found using a database on the Web, likely through a state agency, and then downloaded to a local computer. Once the files are decompressed and opened in GIS software such as ArcView or MapInfo, the user's table could be automatically matched to the map information, and a map combining the information created.

As mentioned earlier, GIS is a group of complex, multi-filed systems with advanced capabilities associated with analyzing and displaying large amounts of data geographically. As a result of increased affordability and technological advances that have moved it to the desktop, more and more people are utilizing it as a resource. The availability of spatial data libraries on the Web has also increased accessibility, creating common access to files that in the past would have been costly as well as difficult to locate and identify. Trends in commercial GIS are toward exploiting the plethora of data via networks and promoting data sharing.

The Internet is invaluable as a source of data for GIS users; online files, with standardized data, may be located and downloaded for incorporation and manipulation within GIS software off-line. The burdens associated with locating suitable data in compatible formats have been eased by the increased documentation, organization, and sharing of geospatial data that occurs via the Web. In 1998, the Federal Geographic Data Committee updated the 1994 Content Standard for Geospatial Metadata, which essentially mandates the construction of metadata, or information about information.[2] U.S. government and many states data therefore reflect this mandated structure, though MARC and Dublin Core standards are commonly applied elsewhere.[3] In all cases, guidelines are designed to ensure that descriptions of data origin, format, and content are sufficient for a user to determine its usability, including where, when, and how data were collected. Librarians have a similar cataloging system, with the significant distinction that these records are created by data producers.

In the past five to ten years, there has been a great deal of collaboration and effort toward sharing metadata, resulting in meta search engines or clearinghouses of geospatial databases. A combination of online geographic data sources can be searched this way—from sites compiled by federal, state, or local government agencies as well as commercial software vendors.[4] A user can search by keyword, location, originating agency, or even by using an interactive spatial browser, which is the electronic equivalent of a print map index. Many records contain live links so one can immediately download files. Use of databases and data from federal and state governments continues to be free of charge, and the commercial geographic community is slowly following suit. The most prominent large-scale example is the Geography Network by Environmental Systems Research Institute (ESRI), a project cataloging data from disparate sources worldwide, much of it free.[5] The Minnesota Geographic Data Clearinghouse is an example of a successful state government organization allowing searching of a network of data libraries. This movement on the part of governments and commercial vendors toward documenting geospatial datasets and making them available in a shared, user-friendly environment has decreased the need for recreating files or purchasing datasets, thereby promoting geographic research and expanding the use of the technology.

Though use of the Internet at this high level is primarily associated with locating digital geospatial datasets, new and less-visible trends in geographic technologies promise even closer ties between GIS and the Internet. In the commercial world, ArcGIS (ArcView 8.x and ArcInfo 8.x), a newly released GIS product from ESRI, combines the functionality of GIS with the data searching and access capabilities of the Internet. In this case, GIS software is used to access data from remote servers using metadata written in eXtensible Markup Language (XML) and is referred to as net-savvy GIS software.[6] Another means by which this tier has a presence on the Web is via the distribution of free and open source software applications, viewers, and extensions, which are feasible alternatives for highly-skilled users.

Applications in the Library

The availability of GIS in libraries is relatively limited and does involve some specialized knowledge, but it is increasing in all types of institutions. Collections of government publications make up perhaps the largest single group with a real demand, as the transition of the Federal Depository Library Program to a primarily electronic environment in the 1990s made familiarity with digitally stored information imperative. Though this trend began with magnetic tapes and CD-ROMs, the release of 2000 census results as datasets and map files on the Web necessitated Internet proficiency for librarians and users alike. Other libraries with a demand for GIS resources include map collections and special libraries for businesses or research institutes, though those within larger institutions are by far the most likely to offer GIS. The St. Louis Public Library and the libraries of the University of Minnesota and Princeton University all maintain separate, fully equipped computer facilities and fileservers devoted to GIS.[7] Others may offer GIS software on a single machine, available with staff assistance or by appointment. In addition to housing electronic data, many of these institutions place data of local interest on their Web servers or simply maintain a page of links to relevant data producers.

While files useable in GIS can sometimes be found free on the Web, the costs associated with offering facilities for full-fledged GIS can be fairly intensive. Until recently, high-level GIS applications were altogether inaccessible to most libraries due to the prohibitively high cost of the specialized hardware and software required, but this has changed. For libraries, a single copy of the ESRI ArcView 8.1 software can be obtained for $250, though hardware demands can still keep the cost prohibitive.[8] Staff time and training is considerable. Libraries with an interest in establishing service in this area should consult the literature.[9]

Midlevel

"I'm researching the socioeconomic makeup of a region across the country and would like to look at a map showing demographic features of the area." The patron with this request can be helped using midlevel geographic technologies. She or he could be pointed to the National Atlas of the United States available from the U.S. Department of the Interior or the U.S. Census Bureau's American Factfinder.[10] Both federally produced tools allow the user to select the features to include on the map from among available data. A good computer with access to the Web via a browser would be needed.

At the midlevel, GIS technologies have been moving to the Web via interactive mapping applications. These are dynamic mapping tools, the most advanced of which have the spatial query and analysis features of GIS while requiring no special knowledge or training to use. In these applications, the end user submits requests via a Web browser, which then communicates with a remote GIS containing all geographic data and feature information. Users interact with a data-inclusive, customizable map application without leaving their Web browser. Data and graphics are organized into thematic layers of geographic features that may be turned on or off (included in or excluded from the display), and multiple spatial layers that may be displayed simultaneously. Individual features have identity and, oftentimes, statistical data attached, and functions such as browsing, panning, identifying features, zooming, and querying are allowed. These mappers are essentially Web applications displaying predetermined, stored data, and the applications may use commercial software, the most common of which is ArcIMS; it may also employ some type of public domain software called freeware.[11]

Software and technology for the creation of online mappers have in recent years become accessible both economically and technologically, and their use has multiplied. Interactive maps are already an important means of disseminating information for federal and state governments. For example, the U.S. Environmental Protection Agency, the U.S. Geological Survey (USGS), and the U.S. Census Bureau are among the many federal government agencies using interactive map generators on the Web to distribute data.[12] These sites are all designed primarily to allow users to create maps inclusive of statistical data and boundaries on the fly, and they use one or more commercial products to power their application and prepare data. Several state governments and agencies—including Arkansas, Maine, and Minnesota—also operate interactive mappers but often in partnership with other institutional and research organizations and using a combination of federal and state data.[13] Other states, such as Kentucky and Virginia, maintain electronic atlases, providing search and data retrieval rather than interactive mapping.[14]

It is no longer prohibitive for smaller projects, such as those operated by educational institutions, local governments, and nongovernmental organizations, to utilize online GIS to disseminate information.[15] Research institutes at universities commonly offer Web mapping. GRASSLinks, a project of the University of California at Berkeley, emerged as one of the first mappers on the Web in the mid-1990s, and offers public access to environmental data.[16] This system uses a freeware called GRASS, developed and produced by the U.S. Army Corps of Engineers for its internal GIS.[17] The Nature Conservancy, Council on the Environment of New York City, and Global Forest Watch are examples of nonprofit organizations using interactive mappers online to educate and inform.[18] More local government agencies are exploiting the capabilities of online mappers. The Twin Cities metropolitan area of Minnesota is served by MetroGIS, a collaborative organization for data and mapping, and the online interactive mapper portion, Datafinder, is an award-winning service using commercial software and offering local data.[19] Portland, Oregon, implemented a neighborhood crime mapping application called CrimeMapper to allow the general public to track locations and types of offenses in 2001.[20] Story County, Iowa, is one among a growing number of county government agencies posting public cadastral information on the Web with the goal of alleviating some of the public service pressure on offices while providing an improved vehicle for public access.[21]

Web users most likely are introduced to online mapping by commercial endeavors. These are usually fairly simple in terms of the amount of data provided. Tools such as Microsoft Expedia, MapQuest, and MapBlast are designed to aid navigation.[22] They provide street maps, directions, and, oftentimes, additional layers of information such as locations of sponsoring restaurants, banks, RV sites, and other commercial endeavors. Most commercial interactive maps are not interactive, but are simply mapmakers: they provide maps of requested areas but there is no user manipulation of the given map or its content. Such is the case with National Geographic Map Machine, Lonely Planet's Worldguide, and many other tools that will be discussed with the low-level categorization.[23]

Applications in the Library

Because of the capabilities of this direct Internet mapping and the vast amount of information available, midlevel mapping resources are particularly useful for reference service in the library. Patrons, even proficient users of the Web, are often simply not aware that these tools exist, but the flexibility, user-friendliness, and customization they offer recommends them for many geographical queries. Most patrons use high-level GIS in libraries for simple mapmaking and do not need its immense capabilities. Thus online mappers often satisfy user needs while mitigating the demand for unnecessarily intensive GIS software. A possible disadvantage for some libraries is that the high functionality and graphics require an up-to-date Web browser and a speedy connection, as well as support for frames or plug-ins like JavaScripts and Java. The recommended minimum screen resolution is often 800 by 600 pixels. Libraries with limited equipment also need to consider that most users will want printing capabilities, particularly in color, for the maps they are downloading.

Low Level

"I need maps of China, both current and from other points in the twentieth century." Using low-level tools, a variety of maps may be easily located and viewed. In this case, private or institutional (though still free) Web sites would be good sources. The David Rumsey Historical Map Collection, a private collection of historical maps offering more than six thousand images online, and the Perry-Castañeda Library Map Collection of the University of Texas at Austin both provide access to excellent archives of maps.[24] A good computer with access to the Web via a browser would be needed.

Low-level technology includes searching and viewing images and maps on the Web. The Web is widely used by libraries, museums, and historical societies to display all sorts of visual collections, including photographs, art, ephemera, illustrations, objets d'art, and even newspapers and books. In all cases, an online database is constructed of descriptive records with links to the scanned-in images as raster files, most often in JPEG, PDF, or GIF format. Users conduct textual searches by querying a database of metadata, much like searching a library OPAC. Once the images are retrieved, however, user interaction ends, since there is no manipulation of the image online. Unlike the previous map tools discussed, images are static graphic files.

For cartographic collections, this level of tool is appropriate for historical maps and remotely sensed imagery, where the object itself is of primary interest. Thus, a low-level user is satisfied by predrawn maps and is usually seeking general information. There are advantages associated with using these tools: because they offer digitized images of paper maps, bibliographic information about the item is usually complete. This type of tool is readily accessible and demands minimal computer skills and standard software. Some databases have been equipped with features to create a degree of interactivity with the images via HTML, Java, and applets. For instance, a user can zoom in on an image by selecting a screen size or file size. Some databases also allow graphical searching with an HTML image map of a cartographic index.

The need for photographic images of antique or historical maps is clear: users wish to examine the original document. Many historical libraries, collections, and museums are increasingly creating digital reproductions of their holdings and posting them to the Web, usually as a part of digital library projects. Hundreds of maps from the Geography and Maps Division of the Library of Congress are presented in the American Memory, part of the National Digital Library Program.[25] Sanborn Fire Insurance Maps are a well-known series of maps showing building footprints in American cities; these were recently digitized, put online, and are accessible by paid subscription.[26] These visual databases are a boon for genealogists, historians, and educators. This is also ideal for those seeking predrawn, contemporary maps of all types, such as general reference maps or road maps, desired as-is for reference or printing without any need for analysis or customization.

Aerial photography and satellite imagery are common information needs handled by map libraries and may often be satisfied using the Web. Various commercial vendor services, such as Microsoft TerraServer, TopoZone, and MapMart store government-prepared maps and imagery using attractive and user-friendly browsers with search engines and viewers.[27] These electronically index a vast data store of USGS imagery and are free of charge. However, if the user desires higher resolution via electronic or hard copy, it must be purchased. Local governments are good sources for aerial photography. The state of Minnesota's Department of Natural Resources has several online tools for viewing current Digital Orthophoto Quadrangles (DOQs) for the state—aerial photographs that have been georectified—as well as satellite imagery, USGS topographic quandrangles—Digital Relief Grids (DRGs), and land use and ownership maps.[28] Often this imagery is integrated into an online mapper (see above) enabling it to be viewed and overlaid with other information while remaining static images. Low-level map tools are also commonly used by location-oriented services, such as the travel industry, conferences, and weather services.

Applications in the Library

Low-level mapping technologies can be beneficial to collections by taking pressure off public service staff and sparing the items themselves from wear and tear. However, disadvantages to this type of tool, even when selected appropriately, lie primarily with limitations of library equipment and facilities. Often, patrons desire a copy of the map and may be disappointed with the quality or size restraints of the printed image. The resolution of graphic files also can be unclear and grainy. Libraries with computers with small server space or Internet connections may find that loading large, high-resolution imagery is cumbersome and time consuming. On the other hand, smaller graphic files are often low-quality, lacking acceptable resolution.

Conclusion

Customized digital mapping is no longer limited solely to GIS users, since the Internet is rapidly becoming a forum for usable and simplified GIS operations. Libraries of all types may find these new technologies useful in reference, and librarians in all fields are encouraged to become acquainted with the appropriate resources applicable to their daily users. Libraries are increasingly incorporating decentralized, virtual resources into their own collections. Accessing nontextual information is a new challenge requiring special effort from librarians. Like most Internet technologies, these can be applied in numerous ways, including distance education delivery, and virtual reference.

The geographic technologies were highlighted in three categories to promote thinking about ways in which geographic information and mapping resources may be incorporated into library service. By explicating how each works and providing brief summaries of the type of information available and its use, it is hoped that the types of tools most feasible and useful for the staff and users of their institutions may be better identified by librarians. It is also meant to encourage further thinking and creativity in the application and use of such technologies.

References and Notes

1. For descriptions and evaluations of some specific tools, see Brenda G. Mathenia, "Maps and Mapping Resources," *College and Research Libraries News* 62, no. 9 (2001): 910–12; Mary Lynette Larsgaard, "The World at Your Fingertips: Map Resources," *Library Journal* 124, no. 12 (1999): 57–60; Luigina Vileno, "Geography Resources on the Internet," *College and Research Libraries News* 7, no. 7 (1997): 471–74. There are many Internet directories with links to online mapping sources, among them: Edgewood College, Free Access Maps and Map Query Sites in Geographic Information Systems: A Tool for Inquiry-Based Education. Accessed Feb. 21, 2002, http://danenet.wicip.org/gisedu/gisreslist.htm; Harvard Map Collection. Resource List, Interactive Maps. Accessed Feb. 21, 2002, http://icg.harvard.edu/~maps/cart/cartintr.htm; University of Delaware Library, Cartographic Information, Maps, and Spatial Data: A Guide to Internet Resources. Accessed Feb. 21, 2002, www2.lib.udel.edu/subj/maps/internet.

2. Federal Geographic Data Committee, "Content Standard for Digital Geospatial Metadata (revised June 1998)," Federal Geographic Data Committee, FGDC-STD-001-1998. Washington, D.C. Accessed Feb. 21, 2002, www.fgdc.gov/metadata/contstan.html.

3. Discussion of standards in David Y. Allen et al., "Metadata Primer for Map Librarians," MAGERT Electronic Publication Series No. 3 (2001). Accessed Feb. 4, 2002, www.sunysb.edu/libmap/metadata.htm.

4. Notably ArcData Online, accessed Feb. 4, 2002, www.esri.com/data/online/browse.html, and Geography Network, accessed Feb. 4, 2002, www.geographynetwork.com, both by ESRI.

5. Environmental Systems Research Institute, Geography Network. Accessed Feb. 4, 2002, www.geographynetwork.com.

6. Brandon Plewe, *GIS Online: Information Retrieval, Mapping and the Internet* (Santa Fe, N.M.: Onword Pr., 1997).

7. All facilities are open to the general public.

8. As of January 2002, library use is restricted to instruction and research purposes. A single commercial copy costs $1,500. The final recommendation for ArcView 8.1 made on the GIS Users' electronic discussion list was: "CPU: As fast as you can afford, 733MHz or faster; RAM: 256MB or as much as you can afford up to 1GB+; Drives: Fast high rpm (10,000+) 160MB throughput SCSI; 64MB video (no less than 32mb opengl)" according to the archived "Minimum Requirements for Hardware vs. Recommended." May 3, 2001. Accessed Feb. 4, 2002, http://library.usask.ca/lists/carta/2001/0013.html. As of 2001, purchasing equipment at this level cost around $3,000. ESRI, in the technical paper Performance Tips and Tricks for ArcGIS Desktop 8.1 (http://arconline.esri.com/arconline/whitepapers/ao_/8.1_Performance_Tips.pdf, accessed Feb. 21, 2002) suggests hardware configuration with 450MHz, 128MB Ram (256MB recommended), SCSI drives.

9. For case studies of GIS services offered in libraries, see Jeanne Boston et al., "The Public Electronic Library: Integrating GIS Resources and Tools," *Library Hi Tech* 63, no. 3/4 (1998): 100–106; Lixin Yu, "Geographic Information Systems in Library Reference Services: Development and Challenge," *The Reference Librarian* 60 (1998): 87–110; Ronald C. Jantz, "Geographical Information Systems at Princeton University: Evaluation and Exploration of a New Service," *Collection Building* 16, no. 3 (1997): 125–36; Denise Stepens, "Managing the Web-enhanced Geographic Information Service," *The Journal of Academic Librarianship* 23, no. 6 (1997): 498–503.

10. U.S. Department of the Interior. Accessed Feb. 21, 2002, www.nationalatlas.gov. U.S. Census AmericanFactFinder. Accessed Feb. 21, 2002, http://factfinder.census.gov.

11. Internet Map Server (IMS) by ESRI.

12. U.S. Environmental Protection Agency EnviroMapper. Accessed Feb. 4, 2002, www.epa.gov/enviro/html/mod; U.S. Geological Survey, National Atlas of the United States. Accessed Feb. 4, 2002, http://nationalmap.usgs.gov.

13. The Arkansas Interactive Mapper, Center for Advanced Spatial Technologies, University of Arkansas. Sponsored by NASA. Accessed Feb. 5, 2002, http://Web.cast.uark.edu/local/mapper. Internet Mapping, Maine Office of GIS. Accessed Feb. 5, 2002, http://apollo.ogis.state.me.us/mapping/mapintro.asp.

LandView, Minnesota Department of Natural Resources. Accessed Feb. 19, 2002, http://maps.dnr.state.mn.us/landview.

14. University of Kentucky. Kentucky Atlas and Gazetteer. Accessed Feb. 21, 2002, www.uky.edu/KentuckyAtlas. The Online Statistical Atlas of Virginia, Geospatial and Statistical Data Center, University of Virginia. Accessed Feb. 19, 2002, http://fisher.lib.virginia.edu/va_statlas.

15. Projects described in the literature include Yafang Su, Joan Slottow, and Avi Mozes, "Distributing Proprietary Geographic Data on the World Wide Web—UCLA GIS Database and Map Server," *Computers and Geosciences* 26, no. 7 (2000): 741–49; R. Kingston et al., "Web-based Public Participation Geographical Information Systems: An Aid to Local Environmental Decision-making," *Computers, Environment, and Urban Systems* 24, no. 2 (2000): 109–25.

16. GRASSLinks. Accessed Feb. 4, 2002, www.regis.berkeley.edu/grasslinks.

17. Other freeware systems include MapServer from the University of Minnesota. Accessed Feb. 4, 2002, http://mapserver.gis.umn.edu. Also GeoTools. Accessed Feb. 21, 2002, http://geotools.sourceforge.net.

18. The Nature Conservancy. Accessed Feb. 4, 2002, http://tncgis.env.duke.edu/data/IMS; Council on the Environment of New York City. Accessed Feb. 4, 2002, www.cenyc.org; Global Forest Watch. Accessed Feb. 4, 2002, www.globalforestwatch.org.

19. Datafinder. Accessed Feb. 4, 2002, www.datafinder.org.

20. Portland, Oregon, CrimeMapper. Accessed Feb. 4, 2002, www.portlandpolicebureau.com.

21. Story County, Iowa, Story Assessor. Accessed Feb. 6, 2002, www.storyassessor.org/pmc.

22. Microsoft Expedia. Accessed Feb. 4, 2002, www.expedia.com; MapQuest. Accessed Feb. 4, 2002, www.mapquest.com; MapBlast. Accessed Feb. 4, 2002, www.mapblast.com.

23. National Geographic Map Machine. Accessed Feb. 4, 2002, http://plasma.nationalgeographic.com/mapmachine; Lonely Planet's Worldguide. Accessed Feb. 4, 2002, www.lonelyplanet.com/destinations.

24. The David Rumsey Historical Map Collection. Accessed Feb. 21, 2002, www.davidrumsey.com. Perry-Castañeda Library Map Collection of the University of Texas at Austin. Accessed Feb. 21, 2002, www.lib.utexas.edu/maps.

25. The American Memory, part of the National Digital Library Program. Accessed Feb. 4, 2002, http://memory.loc.gov.

26. ProQuest Information and Learning, Digital Sanborn Maps, 1867–1970. Accessed Feb. 21, 2002, http://sanborn.umi.com.

27. Microsoft TerraServer. Accessed Feb. 4, 2002, http://terraserver.homeadvisor.msn.com. TopoZone. Accessed Feb. 4, 2002, www.topozone.com. MapMart. Accessed Feb. 4, 2002, www.mapmart.com.

28. The state of Minnesota's Department of Natural Resources. Accessed Feb. 4, 2002, www.dnr.state.mn.us/information_center/maps.html.

PHP: An Open Source Solution for Web Programming and Dynamic Content

Kevin F. Cullen

Managing large amounts of static, traditional HTML pages can be difficult. Dynamic content generated by tools such as common gateway interface (CGI) scripts and databases is the best-known alternative to maintaining a vast amount of information in HTML format. There are numerous alternatives to CGI programming, but technologies and acronyms are bandied about so often that the task of selecting suitable tools becomes quite confusing. Anyone evaluating alternatives to pure HTML or CGI programming should take a close look at PHP.

While we often hear about Perl, ColdFusion, Java, and other programming languages, PHP is the sleeper hit of Web programming. Millions of Web domains have PHP installed and hundreds of thousands of Web sites use it, but this open source product lacks the marketing power of Macromedia or Sun—or the cult status of Perl—to sing its praises.

What Is PHP?

According to its creators, "PHP is a server-side, cross-platform, HTML embedded scripting language."[1] It is essentially a programming language and script-parsing engine that work in unison to provide a way for Web developers to quickly and easily develop dynamic content.[2] A more complete discussion and conceptual schematic of how PHP works can be found later in this article. PHP is an open source project, so the server software and full documentation can be found on the Web for free at www.php.net.

PHP can be used for most, if not all, of the same tasks as technologies like Active Server Pages (ASP), ColdFusion, Java Server Pages (JSP), or CGI programs in languages like Perl or C. In the broadest sense, PHP is similar to technologies like JSP, ASP, and ColdFusion. For the purposes of this article, this type of technology will be referred to as HTML embedded scripting languages (HESLs). Unlike CGI programs, which require the entire document to be generated by commands in the programming language, an HESL allows developers to place program elements within HTML text. Using this method, standard HTML can be typed normally, while dynamic content is generated via the scripting language.

Special tags are used to designate PHP program code within a document. When a Web browser requests the document, the PHP engine parses the code and writes the resultant text into the final version of the document. The Web server then sends this document to the end user. For example, a simple PHP document designed to write today's date and the current time would be:

```
<html>
<head>
<title>PHP Document</title>
</head>
<body>
<?php
$now = date("h:ia");
print($now);
?>
</body>
</html>
```

The PHP code is opened by the *<?php* tag and closed by the ?> tag. None of the PHP code is sent to the end user's Web browser. Instead, it receives the following HTML document:

```
<html>
<head>
<title>PHP Document</title>
</head>
<body>
10:25am
</body>
</html>
```

While this simple task could be duplicated with JavaScript, there are advantages to using PHP. As mentioned above, the end user never sees the PHP source code because it is converted to HTML before being sent. And because the PHP script is run on the server, there is no problem with JavaScript version and browser incompatibilities or users who have turned JavaScript off. PHP, however, is not merely a scripting language. It is a full-featured programming language that allows developers to create full-fledged applications.

To better explain PHP's capabilities, it would help to review its history. PHP was first created by Rasmus Lerdorf in 1994 as a limited set of utilities to carry out common Web-site functions such as a guestbook and hit counter. These utilities started as wrappers of Perl/CGI functions that were later discarded in favor of C functions. In 1995 Lerdorf released these utilities in a kit called Personal Home Page tools (PHP). Within months, a second version named PHP/FI was released with HTML form interpretation functions and support for the mSQL database management system. PHP blossomed into a large-scale open source project in 1997 when PHP3 was released. This third version had a new parsing engine created by Zeev Suraski and Andi Gutmans as well as code written by numerous others. Suraski and Gutmans went on to form a company named Zend that wrote the faster scripting engine at the heart of the latest version, PHP4.[3]

While the first two versions were useful and enjoyed a small following, PHP has evolved so much that the original name has been replaced in favor of a recursive acronym—PHP:

Kevin F. Cullen (kcullen@lib.colostate.edu) is Digital Projects Librarian at Colorado State University.

Hypertext Preprocessor. Even this name fails to do justice to the full-featured language that PHP has become. The language now features interfaces to numerous database management systems as well as a set of unified Open Database Connectivity (ODBC) functions. PHP has a sizeable list of functions, including many for regular expressions, secure socket layer (SSL), arrays, files, directories, session management, and even Extensible Markup Language (XML). Over 1.2 million Apache Web servers have the PHP module installed, making it twice as common as the next most popular module.[4] (This number refers to individual server IP addresses and does not take into account the fact that numerous Web sites may run off the same server.)

Dynamic Content

The term "dynamic Web content," or simply "dynamic content," confuses many who are not involved directly in Web development. Gardner and Pinfield give an explanation of the difference between static and dynamic Web content. As they say, creating hundreds of Web pages by hand and maintaining them as needed "is not an activity that scales well. Once there are several hundred resources . . . inconsistencies begin to develop and maintenance becomes a time-consuming task. The alternative is to use dynamic pages that are created on demand by pulling the requested information from a database."[5]

Simple dynamic content can be created with server-side includes (SSI and XSSI). While SSIs are beyond the scope of this paper, Mach provides a good overview with example code and uses for SSI.[6] PHP can duplicate some of the same features of SSI and XSSI with built-in functions such as *include()* and *date()*, as well as predefined environment and Apache variables.

More often, the term dynamic content refers to taking information out of a database and inserting it into a Web page. Each time a given document is requested by a Web browser, the Web server examines any variables contained in the request (such as an ID number or keyword) and processes the request accordingly. The CGI or HESL script will query a database using these request variables and others contained internally. When the database returns records that match the request, the program manipulates and formats the data.

One example of dynamic content is something that libraries use every day: Web-based abstracting and indexing databases. The citations are stored in a searchable database and lists of results are formatted into HTML pages on the fly—dynamically. An even more common example would be commercial sites like Amazon.com, which cannot write HTML pages for each product they sell. Information about each product—such as name, price, reviews, and images—is stored in a database and used to dynamically create pages for products when users click links in search results.

There are numerous advantages to using dynamic content, especially on very large Web sites. Content can be more easily inserted into templates that contain navigation bars, dates, contact information, and other elements that are consistent across pages. Dynamic content can be searchable, such as a staff directory or a list of databases. By creating dynamic lists of databases, many libraries have eliminated the need to continually edit dozens of HTML pages every time a database's URL or title changes. Gardner and Pinfield provide an excellent discussion of how the University of Nottingham did this with PHP.[7]

Technical Comparison

It is difficult to discuss PHP outside the context of other technologies that can be used for similar tasks. According to Aslam, there are at least three basic means of creating dynamic content with database connectivity: CGI programs, HTML processors (referred to here as HESLs), and Java.[8] Since Java is a compiled, object-oriented programming language and, contrary to popular belief, was not designed specifically for Web-delivered applications, it would be difficult to compare it to PHP. In the following discussion, PHP will be compared to CGI programming and three HESLs: ASP, ColdFusion, and JSP.

CGI

Some people use the terms CGI and Perl interchangeably, though the two are not the same thing. CGI is a protocol that defines a way for a Web server to send a user's request, including variables, to a program. The CGI protocol also provides a way for that program to return data, usually an entire HTML document, to the Web server. The Web server then passes this information on to the end user. CGI programs can be written in C, C++, Perl, and other languages supported by the Web server. CGI programs can be compiled executable files or scripts that are interpreted and run at request time.

CGI can be a difficult technology to use. Compiled languages such as C often do not contain Web-specific features. Writing a simple program that handles Web-form data can be quite complex. And because programs must be compiled after each change, testing and debugging can be time-consuming.[9] Perl and other scripting languages have more Web-specific functions and access to session and environment variables. They do not require program compilation but can be extremely difficult to use.[10] Using CGI also has other drawbacks according to Hartman, who states, "CGIs launch a new process each time the script is run, which imposes a substantial load on the server. Although there's sometimes no avoiding them . . . today's high-traffic Web sites usually try to minimize their use of CGI scripts."[11]

HESLs

Before comparing HESLs, it would be best to discuss the common framework in which they operate. As mentioned previously in the description of PHP, HESL program code can be interspersed with HTML within one document. Special tags set off the code and alert the server that the commands need to be processed or evaluated.

Simply placing PHP or other HESL tags within a document will not have any effect if the necessary scripting server engine is not installed on the machine that houses the Web server. The HESL server must be installed on a compatible Web server and configured correctly before a program will work. Once configured, the Web server will then parse files with the correct extensions—for example, .asp, .cfm, .php, .jsp—to look for blocks of program code. Program code is evaluated and the final document is produced by combining the results of the code and any HTML text. The final document is not written anywhere, except perhaps to a disk cache, but is sent to the end user's Web browser. Blocks of HESL code are not visible to the end user.

To someone unfamiliar with Web programming, distinguishing between CGI programming and an HESL may be confusing. The difference is actually quite simple: CGI uses programs to create documents. With an HESL, Web documents contain programs that write part (or all) of their content dynamically.

Microsoft created its ASP technology some years ago as a Web development technology tightly integrated into its own Web server and operating system software. ASP is not a programming language, but rather an HESL framework that allows scripts in some languages to be embedded within Web documents.

Microsoft's VBScript, based on their Visual Basic, is the most common language used with ASP. Abualsamid criticizes VBScript for its limitations as a language, including a lack of system functions such as those used for manipulating files and directories. He also notes that the ASP server software will only run on Microsoft's Internet Information Server (IIS) Web server, which in turn will only run on Microsoft operating systems.[12] As with other HESL technologies, ASP can be very easy to learn and use, especially for those who already know Visual Basic or VBScript.[13]

ColdFusion is a commercial HESL solution originally created by the Allaire Corporation and now owned by Macromedia. ColdFusion uses its own internal programming language and markup system, known as ColdFusion Markup Language (CFML). Unlike ASP, ColdFusion runs on a number of different Web servers and operating systems.[14] ColdFusion is considered easy to learn, has a wide user base, and is well-integrated with Macromedia's Dreamweaver UltraDev editing software.

JSP uses Java programming constructs within a framework similar to other HESL packages. The Java code is placed inside special tags within an HTML document. The Web server must support JSP and Java servlets, which is not the case with many Internet service providers (ISPs). Java is a difficult language to learn, and writing JSP code can take much longer than a similar project in PHP.[15] Those who master Java may find JSP easier to use than other HESLs because its true object-oriented approach allows developers to recycle code across and within projects.

None of these HESLs is perfect, including PHP. ASP's operating system limitations are a great drawback, especially to those who prefer Unix and Linux platforms, as do many in academia. The fact that ASP is limited to two simple languages seems as if it would limit interest, but so many programmers learn Visual Basic that this might actually be an advantage. The less programmers need to learn, the more time they can spend working. JSP is also hampered by its limited deployment and the difficulty involved with learning the language. While ColdFusion may not attract open-source devotees the way PHP does, it is supported by a major software company, is easy to use, and is available on several platforms. Many users may simply like it because it is not a Microsoft product.

ColdFusion has two good—and purpose-built—integrated development environments (IDEs), something that PHP lacks. ColdFusion works with Macromedia's Dreamweaver UltraDev and HomeSite, which were purchased from Allaire along with ColdFusion itself. Microsoft's ASP also has its own full-featured IDE.

One way to learn more about the capabilities of any given language is to look at Web sites that use the language. File extensions usually indicate which technology is being used. CGI files usually end in .cgi, while some end in .pl, indicating a Perl script. JSP files usually end with .jsp or .jhtml, ASP files with .asp, ColdFusion files with .cfm or .cfml. PHP files appear with a number of extensions, including .php, .php3, .php4, and .phtml.

PHP Specifics

PHP works much the same as the HESLs mentioned previously. Interestingly enough, it can also be used to write CGI programs, though this does not seem to be a common practice. PHP files can be created with any text editor, just as HTML files can. Some what-you-see-is-what-you-get (WYSIWYG) editors, however, will not recognize PHP code and may alter it.

While PHP is closely identified with Apache, it will work with other Web servers, including Microsoft's IIS, Netscape, iPlanet, and Oreilly Website Pro. PHP will also run on

Windows and Macintosh operating systems, as well as most flavors of Unix and Linux. Binary distributions for Unix/Linux versions are not available, but many versions of Linux now ship with PHP. Most Unix administrators are comfortable with compiling their own software and are unlikely to mind doing so with the PHP source code available for download. The adaptability of PHP to multiple platforms gives it a clear advantage over ASP and ColdFusion. ASP, as mentioned previously, will only run under the Windows operating system with Microsoft's IIS Web server. ColdFusion will only run on Windows, Solaris, Linux, and HP/UX.[16]

The PHP Web site makes some frank comparisons with other languages in which PHP does not always come out on top. The site claims that PHP executes faster than ColdFusion and ASP, though it offers no documentation. PHP also says that it is more stable than ASP and ColdFusion, though poor stability is something for which Microsoft products are frequently criticized. ASP is integrated with IIS, which makes configuration simpler than PHP, but many features are not included by default, so programmers have to buy more modules as needed. PHP admits that ColdFusion has better error handling and database abstraction, though it claims ColdFusion is more resource intensive.[17]

Unlike JSP and ASP, PHP has its own language. Its syntax is based on C, Perl, and Java, so programmers who know those will probably pick it up fairly easily. PHP contains a number of control structures, which is to be expected in a language modeled after C, including *if, if/else, while, do while, for, foreach,* and *switch*. Variables are loosely typed, meaning that they need not be declared before use, nor assigned a data type. Types are usually decided by PHP at run time. Available data types are very general and include the primitives *boolean, integer, float,* and *string*. PHP also offers arrays and objects as well as two special types: *NULL* and *resource*.[18] Variables can be passed from one document to another via forms, including hidden tags, with the exception of objects. In fact, form and query variables passed in a URL or via POST and GET are automatically available to the requested PHP document/script. The variables need not be declared and, unlike Perl, are immediately accessible by name without extra programming to parse them.

PHP has an impressive range of built-in functions, probably because it is an open source product in its fourth version. This author has often found functions to complete tasks that would have required complex regular expressions, file manipulation routines, or other extra programming in Perl. And because PHP was built for the Web, many of these functions are Web-related, such as those that deal with cookies, HTTP authentication, session management, mail, and URL parsing.

If a PHP developer needs a function that does not exist, the language can be easily extended with new functions. (This is also true of many other programming languages.) PHP allows user-defined functions within individual programs, something that ColdFusion lacks. PHP4's new session management functions can be used for security, customization, e-commerce, and other purposes. PHP can also do more than just generate text documents. It contains functions that allow developers to generate PDF files and JPEG, GIF, and PNG images as well as Shockwave Flash images on some platforms.[19]

The Colorado State University (CSU) libraries have used PHP for a number of Web development projects of varying size. The CSU Libraries Database of Databases allows users to select abstracting and indexing databases by title or subject. It also has a sophisticated set of administrative functions that allow authorized users to edit subject terms; attach subjects to databases; edit URLs; and locate titles, remote access information, and system status warnings. All data lives in tables within a MySQL database.

CSU has also created several request management systems. Staff in library technology services use such a system to receive and track requests for PC hardware and software support. The Web Implementation Team uses one to track requests for changes to the libraries' Web site and keep statistics on the number, type, and origin of requests. The CSU libraries created a Web interface that allowed teaching and research faculty to view titles lost in the 1997 flood and mark those that should receive priority in the replacement process. The International Poster Collection database of artists was built in PHP and MySQL and works in concert with the poster database built in CONTENTdm.[20]

PHP Problems and Advantages

While some information technology experts prefer to use open source software like PHP, others shy away because there is no company to guarantee support and future updates. PHP, however, has a strong community of support behind it. The online documentation and function reference at the PHP Web site, www.php.net, is superior to that of many commercial software products. There are also numerous online forums and discussion groups to provide answers to even the most arcane questions.

One of PHP's major strengths is the ability to interface with many types of databases, including Oracle, PostgreSQL, MySQL, mSQL, and dBase. It can also interface with some other databases that support the ODBC standard. To do this, PHP has created sets of functions for each supported database. While this allows PHP to take advantage of some very advanced features of each database, it means that code must be rewritten if the data move to a different database platform. Hughes discusses this problem in detail and even comes up with a simple solution: a unified database abstraction layer to handle most common functions.[21] The development team behind PHP has released an experimental extension to the language that creates such an abstraction layer internally, though it is not yet complete.

The lack of any visual IDE is also a problem for those who prefer to create as much content as possible with a WYSIWYG interface. While editors such as Chami's HTML-Kit are PHP-syntax enabled, they do not provide the same design ease as Macromedia's UltraDev and similar tools. Perhaps the biggest advantage of PHP has already been mentioned: the fact that it was built for the Web, and was not ported to it, as were Perl, VBScript, Java, and even C.

The Future of PHP

PHP has been around since 1994, but its popularity has exploded within the last year. In 2000 few books were available on the topic of PHP, but numerous titles have appeared since then. More developers are becoming aware of it and have become enamored of its ease of use and large feature set. As the language catches on, it is likely that PHP developers will start to use it for more than just Web programming. Already, the new PHP-GTK+ tool can be used to create stand-alone graphical user interface applications, and when configured properly, PHP can be used for command-line scripting.

It is difficult to predict PHP's position more than a couple of years ahead. Other tools will appear on the market, and some of them may be better than any Web programming solution available today. In the short term however, the language will continue to be adopted by more and more developers looking for a way to create dynamic content simply and easily.

References

1. The PHP Group, PHP: Hypertext Preprocessor, Dec. 11, 2001. Accessed Dec. 11, 2001, www.php.net.

2. Stig Sæther Bakken, *PHP Manual,* Dec. 11, 2001. Accessed Dec. 11, 2001, www.php.net/manual/en/preface.php.

3. Sterling Hughes and Andrei Zmievski, *PHP Developer's Cookbook* (Indianapolis, Ind.: Sams, 2001), 4–5; and Bakken, *PHP Manual.* Accessed Dec. 11, 2001, www.php.net/manual/en.

4. Security Space, *Apache Module Report,* Dec. 1, 2001. Accessed Dec. 14, 2001, www.securityspace.com/s_survey/data/man.200111/apachemods.html.

5. Mike Gardner and Stephen Pinfield, "Database-backed Library Websites: A Case Study of the Use of PHP and MySQL at the University of Nottingham," *Program* 35, no. 1 (2001): 34–35.

6. Michelle Mach, "The Service of Server-Side Includes," *Information Technology and Libraries* 20, no. 4 (2001): 213–19.

7. Gardner and Pinfield, "Database-backed Library Websites," 34–35.

8. Sohail Aslam, "Web-based Query Processing in a Database Course Project," in *Proceedings of the Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education* (New York: SIGCSE: ACM Special Interest Group on Computer Science Education, 1998), 297.

9. Hughes and Zmievski, *PHP Developer's Cookbook,* xv–xvii.

10. Aslam, "Web-based Query Processing," 297.

11. Hans Hartman, "Tools for Dynamic Web Sites: ASP vs. PHP vs ASP.NET," *Seybold Report Analyzing Publishing Technologies* 1, no. 12 (2001): 8.

12. Ahmad Abualsamid, "So Why PHP?" *Dr. Dobb's Journal* 26, no. 1 (Jan. 2001): 58.

13. Hartman, "Dynamic Web Sites," 8.

14. Dennis Gesker, "Alternatives for Dynamic Web Development Projects," *Linux Journal* 2001 (83es) online (2001).

15. Abualsamid, "So Why PHP?" 58.

16. Bakken. *PHP Manual.* Macromedia, *Macromedia ColdFusion,* 2001. Accessed Jan. 9, 2002, www.macromedia.com/software/coldfusion/productinfo/system_reqs.

17. Bakken, *PHP Manual.* Accessed Dec. 11, 2001, www.php.net/manual/en/faq.languages.php#AEN75483.

18. Bakken, *PHP Manual.* Accessed Dec. 11, 2001, www.php.net/manual/en/language.types.php.

19. Bakken, *PHP Manual.* Accessed Dec. 11, 2001, www.php.net/manual/en/preface.php.

20. Many of these CSU sites are not available to the public for security reasons. The public interface to the CSU Libraries Database of Databases is http://lib.colostate.edu/databases. The International Poster Collection is visible at http://lib.colostate.edu/posters.

21. Hughes and Zmievski, *PHP Developer's Cookbook,* 283–306.

The Syllables in the Haystack: Technical Challenges of Non-Chinese in a Wade-Giles-to-Pinyin Conversion

Gail Thornburg

This paper describes the technical challenges of developing software to convert Wade-Giles to Pinyin in bibliographic records that are not in Chinese.

The Chinese language is different from alphabetical languages to which most Westerners are accustomed. To represent items using such a language in a bibliographic database employing principally roman script requires some form of converting the original to a representation in alphabetic characters and possibly diacritics. Systems of such transliteration for Chinese date from at least 1605, but one prevalent in the last hundred years or so in the United States is the Wade-Giles (WG) system.[1]

Gail Thornburg (thornbug@oclc.org) is Consulting Software Engineer at OCLC Online Computer Library Center, Dublin, Ohio.

Recently the Library of Congress (LC) decided to discontinue use of WG and adopt the newer Pinyin form of transliteration, adopted by the People's Republic of China in the late 1950s. This meant conversion of Chinese records in the Online Computer Library Center (OCLC) authority file and OCLC bibliographic file to Pinyin.

This evolved into a consortial effort among LC, Research Libraries Group (RLG), and OCLC, an effort extending over three years. Earlier efforts by the OCLC Office of Research have been reported elsewhere.[2]

Background

Once requests for comments and discussions with key libraries had taken place, there were major parts to the conversion effort to plan:

1. LC conversion of Chinese authorities by OCLC, scheduled to take place not later than October 1, 2000, "Day One"
2. Conversion of LC bibliographic records (bibliographic records) by RLG
3. Bibliographic records conversion by OCLC and RLG of their respective union catalogs
4. Conversion by OCLC of the non-Chinese records containing Chinese text, and later by RLG of similar records in their databases
5. Conversion efforts by OCLC and RLG of records of institutions from WG to Pinyin

Development of the Specifications

These were developed cooperatively as the project progressed. The specifications can be seen at the LC Web site.[3] Some general points to keep in mind about the conversion:

1. Only fields/subfields specified by LC in the specs were to be converted.
2. Conversions made heavy use of dictionary lookups, not only for conversion of WG syllables to Pinyin counterparts, but also for phrase matching as in place names. The conversion sequences, which were directions for specific types of conversion such as geographic place names or Taiwan names, dealt with special types of translations. The dictionaries for these sequences were generally organized in the form of longest to shortest entries, to allow the most complete phrase matching.

The Dictionaries

The Standard Dictionary (STD) was the complete list of the more than four hundred WG syllables and their Pinyin forms. This was searched after all other conversion sequences had a chance to do special matching.

Anyone familiar with WG and Pinyin romanization schemes knows that while diacritics and initial letters are usually enough to signal WG to the human eye, in fact the Pinyin and WG schemes have considerable overlap. In some cases these syllables are uniquely romanized for WG and Pinyin; in other cases the same syllables are romanized in just the same way; and in yet others, syllables spelled the same way in WG and Pinyin represent different sounds in each scheme. In testing early conversions, it quickly became evident that any automated conversion scheme needed to distinguish WG that could only be WG from WG that could also be Pinyin, or could be a common match that could be either.

As if this were not complex enough, it became apparent that the overlap of WG Chinese with other languages could lead to erroneous conversion of other languages to Pinyin. Systems of safeguards were developed.

One safeguard in the specification broke down the STD into four subdictionaries: Unique WG, Unique Pinyin, Same syllable in both, and Common (same spelling but different sound).

At LC this meant a new conversion sequence called "Mixed Text" to indicate what actions should be taken if troublesome mixes of the four categories of syllables were encountered. The Conversion Sequence Mixed Text attempted to identify cases where WG and other text could be discriminated safely, and whether these cases should be flagged for manual review, converted, or skipped. This was implemented at OCLC largely through what came to be called IsWadeGiles testing. This gatekeeper function is discussed further below.

In addition, special requirements were described in the evolving specifications, for cases in which, due to Board of Geographic Names (BGN) requirements, Taiwan place names were to be excluded from conversion.

The Taiwan conversion sequences started as a short list of examples of what not to convert. Prototype software was rapidly developed to do a sort of learning by example and to elucidate what the requirements of the spec needed to be. LC then reviewed a mini-test of sample conversions of fields likely to contain Taiwan place names and wrote the conversion sequence itself based on the behavior of the software.

Other conversion challenges included the interpretation of generic terms for jurisdictions implemented in the G conversion sequences (G3, G2, G1). Two examples are Feng-hsin hsien to Fengxin Xian, or Ying-hsien (Qualifier) to Ying Xian (Qualifier). The idea was that different types of place names could be predicted to occur generally in specific fields and not in others, and that the software should apply the G conversion sequences only in the former. The challenge was that the generic terms did not necessarily represent place names, so care was needed not to run the sequences where jurisdiction names would not be expected to occur. The software being developed did not, of course, know Chinese.

With testing came the recognition that overlap of syllables between WG and Pinyin was only one problem in discriminating WG safe from WG risky conversions. Even if a record was selected as Chinese, the records being converted usually contained strings of text in English and other languages.

In the realm of non-Chinese interspersed with Chinese in a subfield, even abbreviations such as Pa. and Jan. could be misinterpreted as Chinese if special screening was not done to avoid converting such English strings in the middle of Chinese subfields. The earliest of the tests featured software that readily converted English, Russian, Italian—and even Pinyin—to Pinyin (not a good outcome).

Too-common syllables were one of the key pitfalls. It was soon realized that when the software analyzed a subfield for conversion, it was critical for the software to know if the WG match was a case such as a, an, to, no, Jan, Ka, Jun, lung, sung, I, lo, la, le, so, sun, Juan. These might well be WG syllables, but might also be English, French, Italian, Spanish, or other languages. Converting French to Pinyin is not part of the spec. Nor is converting a subfield from

> A concordance to: Yen tzu ch'un ch'iu

to

> A concordance duo: Yan zi chun qiu

A little English-to-Pinyin conversion goes a long way. Soon, revisions to the spec directed blocking the conversion of "to" in certain subfields, but the software also had to try to guess whether it was the English "to" it had blocked, or the Chinese "to" which should be unblocked and converted to Pinyin.

Software Design Challenges

Early in the project, it occurred to the development team that the loosely defined specifications were going to be an issue. Early teleconferences with LC, RLG, and OCLC led to the suspicion that minds were going to be changed and rechanged, and compromises made. Moreover, both time and staff were limited. Yet it was necessary to expend the time to come up with an easy-to-understand flexible record structure that could be parsed by all the modules.

With an attempt to simulate realism, an early design decision was made that flew in the face of conventional software design. A module would be constructed for each conversion sequence in the spec and the same software run wherever indicated. At the same time, the field/subfield modules would be treated as unique. In this way enough granularity could be retained to allow changes in the order in which conversion sequences were run, the levels of safety checks required, and other very specific considerations.

This seemed to make sense as soon as it was realized that even the indicator values for certain fields determined varying courses of processing. This resulted in a huge number of small subfield modules that duplicated a lot but were easy for even the most junior of the development team to change in a hurry. Programs were developed to run mass compiles to keep all the pieces and parts in sync. Changes to the spec for the authorities and later bibliographic records occurred to the very eleventh hour of the development and testing process.

First Stage—Authorities Software

At OCLC, the first deadline was the approval of the authorities software. Since this was developed first it was most the prototypical. Note that the conversion deadline had been fixed at October 1, 2000. A moratorium on Chinese cataloging by libraries had to be imposed for the interval of the conversion. The authorities records needed to be converted in advance of the bibliographic records. In cases where it was not practical or even theoretically possible to ensure a perfect conversion, the outputs of the authorities conversion were organized in separate groups to aid in staged evaluation of riskier conversions such as the G conversion sequences. For bibliographic records the specs did include a text subfield, the 987$f, to which risk flags were written as was shown useful in testing. For the authorities records, there was no such provision for a special text field, so the records could only be isolated by grouping the outputs of the conversion. The software would flag a given subfield as requiring manual review and leave such subfields alone, but also would (by this sorting) flag riskier cases for later checking, even though the conversion was considered successful.

Implementation Challenges

The Sequences of Conversions

The first conversion modules written dealt with place names (the C1/C2/C3 sequences, the G1/G2/G3 sequences). These handled conversions of phrases such as Hsiao-shan shih (Chekiang Province, China) to Xiaoshan (Zhejiang Sheng, China) or Chin-chou shih (Liaoning Province) to Jinzhou (Liaoning Sheng)

The dictionaries were searched in order by longest to shortest phrase, the substitution text was replaced, and the subfield passed on to the next conversion routine. Each program had to maintain and pass along a shadow subfield that indicated which portions of the text had already been converted. Bear in mind, except for the IsWadeGiles gatekeeper, none of the individual conversion routines were aware of the overlap areas of WG and Pinyin, so it was necessary to avoid possible reconversion by later conversion sequences.

Phrases versus Syllables

What worked well enough for the phrase matching conversion routines was inadequate to process a subfield against the STD. This sequence needed to keep track of potentially hundreds of individual words and syllables and the surrounding punctuation and spacing. It was also necessary to maintain the original spacing and punctuation of the subfield except as explicitly instructed by the spec.

To organize this information, a table-like structure was implemented to represent the contents of the subfield, syllable by syllable, and to store the punctuation before and after each word or syllable. Each row of this table was one word or syllable found; different columns represented different types of matches made. The structure also stored the before-and-after lengths of the WG/Pinyin forms, or a zero length to indicate no match. In some cases the punctuation between syllables would be retained or replaced, and spacing might change, depending on the particular syllable and its neighbors.

So, while the subfield or its major parts were the general unit of scrutiny, some conversion modules needed to focus on phrase matches and be blind to the additional contents. Other modules needed to tokenize—break down—the whole subfield into its smallest component pieces and count every space and punctuation mark, all the while turning a selectively blind eye to things that didn't always matter. Punctuation such as parentheses might matter in matching to the C sequence dictionaries, but not for the STD.

The Segue from Chinese Authorities to Chinese Bibliographic Records

It was initially planned to use the same conversion software on the fields converted in bibliographic records as was run in authorities records; problems were soon encountered in this approach. One was scope: the size of fields in bibliographic records quickly exceeded what had been reasonable bounds in authorities records. For a time there was a processing tension between what was big enough to encompass even monster bibliographic records with hundreds of subfields and subparts of subfields, and what virtual meta-construct might grow so big in running memory that the software would be too big to run at all. One size had to fit all, and it wasn't a small. The software was put on periodic diets, but the development team had to lie in wait and catch it bingeing one interesting evening, to track down the problem.

Another issue in moving to the bibliographic world related to the data. Many rules and assumptions of the software that worked reasonably well in the disciplined environment of authority records were bent or broken in the larger world. The test set furnished by LC included records with notes fields with such irregularities that one wondered if the person entering them had a broken finger or two.

This quickly revealed interesting gaps or levels of "trust" in the authorities-developed software that needed to get very mistrustful in the bibliographic records environment. The conversion sequence for STD was rewritten to tighten up the handling of widely varying punctuation and other practices.

In reviewing CONSER records, problem patterns were noticed. For instance, the usual requirement of manual review for subfields solely of Same/Common syllables could lead to needlessly high review rates for certain subfields. For example, a subfield might consist of

$aTi 1 pan..

or

$cmin kuo 66 (1979)..

The first has two Common syllables and the second has one Same syllable and one Common. In cases where the only WG in the subfield is all Same and Common, the subfield is flagged for review because the software cannot possibly know whether it is really WG or already Pinyin. Yet, in cases like those previously mentioned it seemed safe enough to go ahead and convert the patterns, and save human reviewers needless work. Several examples of these special-case instances were developed in the course of the bibliographic records conversion testing. The challenge was to find a way to describe their occurrence narrowly enough to apply to all subfields, since the IsWadeGiles module making the decision and the conversion sequence modules generally did not know which subfield they were processing.

Non-Chinese Records

The first question many asked of the team was "Why would you want to convert non-Chinese?" Indeed a lot of effort had been devoted early on to avoiding that outcome. Still, it was recognized that many records whose language code was not Chinese did in fact contain Chinese text that would be desirable to convert in the course of the project.

About the time it was recognized that separate conversion software would be needed for non-Chinese records, the testing of non-Chinese conversions was moved to a phase after the completion/approval of the Chinese conversion software. This conversion software was to be applied to the Chinese records in the OCLC Online Union Catalog (WorldCat).

At about the time approval was achieved, it was also recognized that for WorldCat conversions that it would be useful to sort out even those Chinese language records that contained Japanese or Korean, or contained so many languages (more than four language codes in the 041 fields) as to need extra checks. These were screened from the Chinese conversion effort, to be picked up later.

Selection of non-Chinese records to try to convert was the initial issue. Part of the specification developed with LC identified the selection process and convertibility testing that would be used to identify good candidate records for conversion. The first non-Chinese test set submitted to LC consisted solely of a selection/rejection set, to determine if the records identified by the software appeared to be reasonable choices and if those omitted seemed correct. The challenge was to find as many as possible but to throw a lot back in convertibility testing. This paralleled the authorities selection process to a degree, in that authority records have no language code.

The specific criteria for selection are detailed in the specification for non-Chinese posted on LC's Web site. In general, records with no 987 (meaning the record had not been converted already, by software or human cataloger) were examined. In addition to records that would have been excluded from the Chinese conversion as noted above, non-Chinese language codes where the 041 tag, if present, contained "chi" in one subfield, were looked for. Also scrutinized were geographic area codes for countries that seemed likely, the inclusion of Chinese/Japanese/Korean (CJK) diacritics in the record, and even the word "Chinese" in a 500 or 546 field. These criteria were used to generate a preselected set, intended to be broad. The software looked for music records that have no language code.

The second phase of selection was to run a sort of IsWadeGiles test on a list of subfields in the record. If a subfield scrutinized in the record emerged from this testing with a status that indicated convertible text, the record was included in the set of those to convert.

Once the selection phase was complete, the conversion itself was run on the set of records selected. This conversion resembled the Chinese bibliographic records conversion, but was elaborated in several ways.

Tailoring the Conversion Software for Non-Chinese

At this point, the WorldCat conversion of Chinese bibliographic records had been run, and inspections of each conversion set led to the belief that the software was sound. So the specification of non-Chinese based itself largely on the same software, with somewhat fewer fields converted. The intended design was to deviate from use of the Chinese software only as found to be necessary.

By this time it became apparent that the presence of Korean or Japanese text in the selected set was a distinct threat. Yet Japanese or Korean are frequently mixed with Chinese in the same record, the same field, even the same subfield.

From LC two lists of Japanese and Korean romanized syllables that matched WG syllables were obtained. With these dictionaries added to the suite, it was possible, for records coded Japanese/Korean, to check the dictionaries to see if all the WG syllables found were also in the Japanese or Korean list. This would alert the IsWadeGiles software to a higher level of risk of misconversion of Japanese/Korean to Pinyin.

Considering the length of the lists, it did not seem useful to search these dictionaries unless Japanese or Korean was indicated by the language coding; too much legitimate WG could be excluded, especially in short subfields. This pointed to the need to think about discrimination patterns.

The team worked with LC and staff at OCLC to develop lists of characteristic patterns of romanized letters that would occur in Japanese or in Korean, but would never occur in WG Chinese. Some of these patterns (initial letter b, end letter m) proved to be too "noisy" in terms of overlap with English terms or place name terms in the phrase dictionaries.

Some, however, proved more effective in discrimination. The diacritics patterns macron-o or macron-u or breve-o or breve-u could be searched for, generally enabling a subfield to be eliminated from further consideration. The longer pattern matches of letters served as further discriminators, such as the Japanese kyo, ryu, pyan, or terminal aa, ae, au, ea, ee, eo, eu.

Another issue was the general risk of converting personal names in non-Chinese records. There was perceived potential for converting names that were actually not Chinese or too generic to determine effectively.[4] It was decided that personal names fields would be converted only by use of the authority control software run by the OCLC Lacey Product Center, formerly known as OCLC/WLN, and that personal names fields would only be converted if the match was to a subfield a, plus either $c or $d. In cases where no authority control match was made, the subfield a was evaluated by the IsWadeGiles module, and if convertible text was found, the record was flagged in the 987$f to alert catalogers to this nonconversion.

IsWadeGiles Meets Non-Chinese

The processing of subfields in the IsWadeGiles module has a gatekeeper function. This is the software that scrutinizes a subfield, tokenizes it into individual words or syllables, and assesses matches against the dictionaries used in the actual conversion sequences. IsWadeGiles may decide a subfield should be skipped entirely, that it is safe to convert, or that it should not be converted but flagged for manual review and possibly broken down and reevaluated in Mixed Text.

Once IsWadeGiles has searched for place name (phrase) matches and has considered a list of stopwords, it searches the syllable dictionaries. It tallies the counts of unique WG, unique Pinyin, WG Same as Pinyin, and Common (could be either WG or Pinyin). If there are fewer matches than the count of tokens in the subfield these will be considered unknown—Other. At the end of this tally, rules for combinations are tested in order, until one fires, causing a decision to be made about the subfield, and exit from the evaluation program. If a subfield has only unique WG, and nothing else, go ahead and convert it. Most cases are more complex. The Same/Common rule illustrated previously is an example where generally the software would decide that the subfield should be flagged for manual review.

The rules are heuristic and so have a slight potential to make a conversion decision that a human reviewer would not. However, these rules had been developed iteratively over successions of authorities and bibliographic records tests reviewed. It was necessary to tailor them, with some care, to allow for special screening of non-Chinese in a high-risk environment. While the software is fairly conservative in its decisions, it was undesirable to have too much flagged for manual review, or skipped when it could have been converted.

One example was the evaluation of personal names fields not matched under authority control. The program needed to flag cases where WG clearly remained, but also try to avoid flagging the numerous cases of Chinese Westernized names (forms like C. C. Chen or Stephanie Chuang) that needed neither conversion nor human review. This was not hard to implement.

Then the team noticed that the variants implicit in bibliographic records conversion could lead the software to think that the subfield contained WG but also Other (and thus was possibly a Westernized name) in cases where the Other was a transposed letter or variant diacritic in a clear attempt to enter a WG personal name. This was a tricky situation; one cannot ignore the specific diacritics used and identify WG successfully. In this context normalization would be unworkable for an evaluative program. Moreover it would have been an enormous task to attempt any spelling correction. Known variants of the Chinese diacritic ayn were built into the dictionary loading, but there were only limited substitutions practical with the dictionaries, due to overlap.

Dictionary changes or variants were generally agreed to by all the consortium participants, with the exception of the IsWadeGiles stoplist used at OCLC. The stoplist attempted to identify common English descriptive phrases (such as "written by" or "published in") that could be skipped, in order to convert more subfields. Over time, additions of common phrases to the stoplist generally improved the likelihood that Notes fields could be converted.

So how does one attempt to identify almost WG? There were no perfect answers, but one new approach was attempted. Little could be done about typos, but a search was added for diacritics as a rough discriminator of attempted WG entries from likely westernized names. It made the assumption that a westernized Chinese name probably would not feature diacritics.

Extensions

If this was a program report for funded research, there would be a section on future research. For the development team, there were practical limits to time that could be spent on the project. Some of the areas that would be interesting to explore in future software implementation are as follows:

- **Addition of a sort of primitive learning component.** Given the sets of inputs (for example, ten thousand-plus bibliographic records at a time), the system should infer patterns in the data for improved processing of future sets. For instance, seeing repeated strings of text that were categorized as Other causing a subfield to be rejected for review would suggest (given tests) additions to the stoplist to be used in future runs.
- **Patterns of rules firing or not.** Observing and categorizing these could suggest to the system either proposals for rule simplifications or that pruning the rules was needed.
- **External analysis of the rules structure.** Over the course of the project, these expert-derived rules became quite complex. A meta-analysis, perhaps using existing software from other sources, could lead to rules simplification that would make maintenance of the system simpler.
- **Harnessing the implicit network of subfield modules.** Here there is a sea of modules waiting, in a sense, for a chance to fire. In a few cases, the domain experts recognized a need for special treatment of situations arising in specific subfield modules, but the architecture of the software made it more tractable to devise rules abstracted enough to cover all subfield modules. It might be interesting to observe the firing of these rules specifically. The goal would be to see if patterns and filters would naturally evolve. Groupings of types such as notes subfields, controlled access subfields, or publication

date, might generate data and reports. These could be used to make the reason for the rules—now hidden in advice from the experts—more visible in the code.

- **Extending the schedule to allow for testing alternative, looser criteria for selection of non-Chinese records.** This might imply the need for development of other dictionaries, for instance, syllables in French or in Russian mimicking WG.

Lessons Learned

All the development staff at OCLC felt they learned from this project in the sense of extending programming skills. The team compiled some insights about the nature of such a project as this. For example:

- Don't generalize from two examples. Or one.
- Choose battles carefully. (If resources are scarce, don't write software for data that doesn't exist.)
- When necessary, pretend to have a spec. Write software to match it.
- Rapid prototyping can help to flesh out a sketchy spec. The tradeoff is unrapid support.
- Dictionaries have bad days, too.
- Champions—for such a consortial effort to succeed, at least one person in each organization must be determined to make it happen, whatever it takes.
- Rules—conjectures and refutations as World View.
- Allow time in the project schedule for review, re-review, and coordination of new versions of the spec against existing software. The team could have used one full-time empoloyee to cover this.
- Test sets—domain experts can devise very good test sets. Expect gaps anyway.
- Version control—don't write a line without it.
- Archive every e-mail. Make telephone records on every call. Make other people's heads hurt by being able to *find* them later.
- Manual review by human experts is an essential part of the team solution.
- Rules for deciding whether to convert—tough to develop anything like a "covering set." Possible to develop some critical gatekeepers.
- Authorities conversion to bibliographic records conversion: Expect more software revision than originally planned.
- Evolving specs—not all parts of the spec will attain equal robustness. Expect to do triage. (For instance software redundant checks, flagging riskier conversions.)
- Granularity—many small repetitive modules, one per subfield. Sounds like bad design; proved essential to the many changes over the lifetime of the project.
- The status of a subfield seems to suffer from multiple personality disorder
- Promise your superiors you'll never agree to a project like this again. Cross your fingers behind your back.
- Moving targets—never believe all parties are working to the same spec.

Acknowledgements

Any effort on the scale of this project requires the help of a host of professionals behind the scenes. This conversion could not have been accomplished without the patient, tireless efforts of Philip Melzer and his colleagues at LC. Thanks are due also to the OCLC CJK Users Group Pinyin Task Force for their comments on test conversions, and to OCLC staff too numerous to mention, for all their painstaking advice and assistance.

References and Notes

1. Yewang Wang, "A Look into Chinese Persons' Names in Bibliography Practice," *Cataloging and Classification Quarterly,* 31 no. 1 (2000): 51–81.

2. Rick Bennett, "Bringing Chinese Cataloging Records Up-to-Date," *OCLC Newsletter* (Mar./Apr. 2000): 18.

3. Library of Congress, Other U.S. Libraries Join International Community on Use of Pinyin. Accessed July 9, 2002, www.loc.gov/today/pr/2000/00-141.html. Library of Congress New Chinese Romanization Guidelines. Accessed July 9, 2002, http://lcweb.loc.gov/catdir/pinyin/romcover.html.

4. Yewang Wang, "A Look into Chinese Persons' Names in Bibliography Practice." Wang gives a detailed discussion of the many problems in the cataloging treatment of Chinese personal names.

Using Microsoft Access and HTML to Produce Browseable Web Lists

Scott A. Opasik

Microsoft Access is a powerful tool for creating and maintaining databases. One of its many features is the ability to publish reports to the Web. The Schurz Library of Indiana University South Bend (IUSB) found a problem using the Web publishing features of Access. The library found its needs better met by directly marking up an Access report form with HTML. Doing so did not take sophisticated computer skills and provided the ability to output a report as one continuous Web page, saving the time of the reader. It is a method that can benefit other libraries.

One of Ranganathan's five laws of library science is to save the time of the reader. One way in which many libraries have helped save the time of the reader has been to provide a list of periodical holdings. With a citation in hand, these lists give patrons a convenient way to determine whether the library owns a particular issue. Even with the advent of online catalogs, users have found a printed list of periodical holdings preferable to deciphering catalog records of serials.

For some time the Schurz Library has maintained a printed list of periodical holdings. The list provides the user with a call number, years and volumes owned, missing volumes, title change information, and volumes on microfilm for each title. More recently, information has been added for titles available in electronic full text.

In order to increase access to the list while decreasing the cost and time of updating, the library looked for a way to place the list on the Web. The printed list was available only in the library during open hours. This level of access was not meeting the needs of Indiana University–South Bend (IUSB) faculty and students, who increasingly used the library's indexing services online from outside the library. Another problem with the printed list was the expense of keeping it maintained. The library's periodical holdings are dynamic; titles change, cease, are added or canceled continuously. It was determined that this information should be included in the list as soon as possible in order to best serve patrons. Due to the expense of printing, only the master list at the reference desk was kept up-to-date with hand-written changes; multiple copies were published only once or twice a year. If the library published the list on the Web, then updates could be made cheaply whenever needed.

The Problem with Access Static HTML Files

The Schurz Library has found Access to be a powerful tool for maintaining the periodical holdings list. The design features of Access make it easy to create and edit tables, data entry forms, queries, and reports. Access has the tools needed to control the input and output of data, two essential functions of any database program. With Access, data input can be controlled by specifying the data type properties of a field and by creating a validation rule for data input into the field. Using combo boxes to create menus in a data entry form can also control data input. During data entry, an item from the menu is chosen to fill the field. Queries can be used to extract needed data. These data can be custom formatted by the use of report forms. Access not only is flexible, allowing new fields to be added to the database as needed, but it also has a global change feature. The library hoped Access could be used to place the periodical holdings list on the Web.

Microsoft Access offers three options for publishing data to the Web: data access pages, server-generated HTML files, or static HTML files. All three options have advantages. But not all options were appropriate for the needs of the Schurz Library. Data access pages allow one to change data via the Web. However, as with the online catalog, the library did not want patrons to be able to edit or delete information in the periodical holdings list. Server-generated pages, which provide up-to-date information from the database, require a Microsoft server that Schurz Library did not have.

The use of static HTML files to publish the periodical holdings list to the Web looked promising. The use of static HTML files requires a Web server, but not a Microsoft server, and would allow patrons to view the information without the ability to edit data. Nevertheless, the use of static files presented a problem. Access publishes reports to the Web in multiple pages instead of one continuous page. This is not a problem unless the report is long, in which case users must download page after page until they find the page with the needed information.

To illustrate the problem, consider a report containing all 714 records for periodical titles that begin with the letter J. If one used Access to publish the report, the report would consist of 145 separate Web pages. Furthermore, each page would contain only five titles. To use the report, patrons would need to download page after page until they found the page with the title they sought. A procedure such as

Scott A. Opasik (sopasik@iusb.edu) is Assistant Head of Technical Services, Schurz Library, Indiana University–South Bend.

this certainly does not save the time of the reader. If the library produced the Web database in such a way, it was doubtful that patrons would use it.

The problem became getting Access to produce a report in one continuous list without page breaks. Furthermore, the report needed to be read correctly by a Web browser as an HTML file.

The Solution

The library solved these problems by directly marking up a report form in HTML and by saving reports in text-only format. Starting with a blank report form in design view, fields from records with their associated labels were placed onto the form. Next, label boxes were used to insert HTML tags into the report (see figure 1). Basic tags such as document type, header, and body were place in the report header and report footer sections of the report form. A paragraph containing information for search options was also placed in the report header. Since the list includes electronic full-text holdings, hot links to aggregator database Web sites were included. This was accomplished by using the HTML tags for URL.

To produce the reports, the correct query from the record source field of the report properties was selected. The file was then exported, making certain that it was saved as a text file. Saving the file as a text file removed the Access control code for page breaks, allowing the report to be produced as one continuous list that could be browsed. The final step was transferring the file to the Web server and changing the file extension from .txt to .html by renaming the file.

The end product consists of twenty-six separate lists, one for each letter of the alphabet. Each list contains all titles, listed alphabetically, that begin with that letter (see figure 2). One long list consisting of all titles was ruled out because it took too

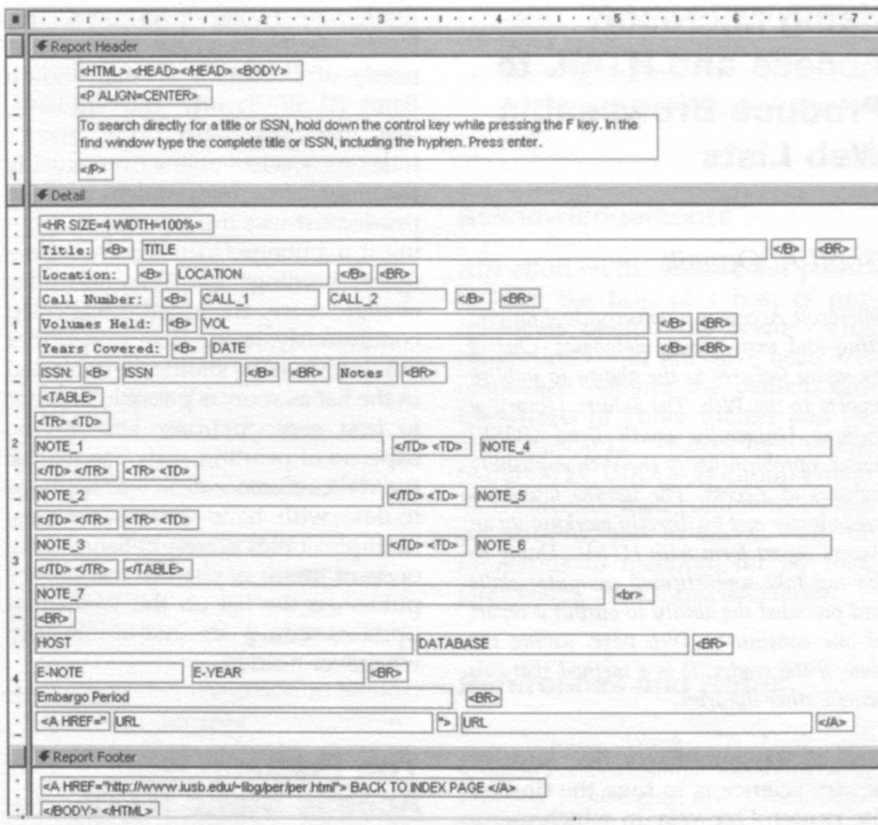


Figure 1. Marked-Up Access Report Form

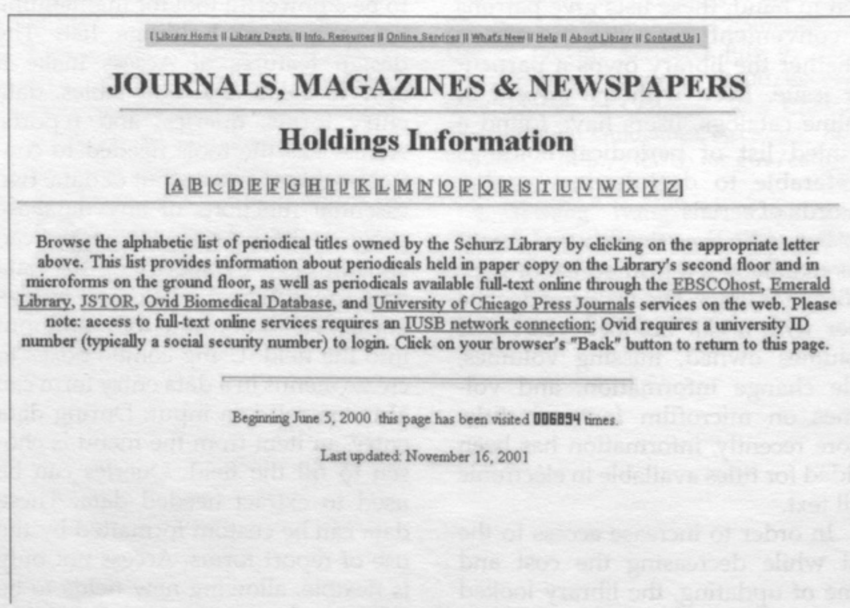


Figure 2. Web Periodical Holdings List

http://search.epnet.com/login.asp

Title: **ACTA PSYCHIATRICA SCANDINAVICA**
Location: **PERIODICALS**
Call Number: **RC 321 .A185**
Volumes Held: **vol.83-**
Years Covered: **1991-present**
ISSN: **0001-690X**
Notes

Full text available online via EBSCOhost's Academic Search FullTEXT Elite
Electronic Holdings 1/1/1999 - present
Latest 12 months not available from EBSCOhost
http://search.epnet.com/login.asp

Title: **ACTA PSYCHOLOGICA**
Location: **PERIODICALS**
Call Number: **BF 1 .A12**
Volumes Held: **vol.50-vol.97**
Years Covered: **1982-1997**
ISSN: **0001-6918**
Notes

Figure 3. Typical Web List Records

long to download and was too difficult to browse.

To use the list, patrons click on a letter to download a file of all journal titles beginning with that letter. Once the file is downloaded, patrons may either scroll the list to find a title or use the Find in Page command of the Web browser to search directly for the title. To use Find in Page, one holds down the Control key while pressing the F key or chooses Find in Page from the Edit menu. This command opens a search window where a search string may be entered. Executing the command will find the next instances of the search string in the page.

The list is browseable and superior to the Web reports created by Access for several reasons. First, it is still possible to use Access to maintain the database. Directly marking up the report form gives more control over the display of data than using one of the format options of Access or saving the print report form as an HTML file. By directly marking up the report form with HTML tags, the library was able to match the display of data in the printed list (see figure 3). This consistency between the forms of the list aids patrons who were accustomed to using the printed list to find periodical information. Because the list is browseable, it saves the time of the reader. Patrons only need to download one file per report. Additionally, using the Find in Page command of a Web browser, patrons are able to search a whole list at once. This command is not useful with the multiple Web page structure created by the Web publishing features of Access. Finally, it is an unsophisticated method of placing a report on the Web, one that does not require any programming knowledge beyond basic HTML.

Other libraries might consider using this method of creating browseable Web lists. Depending upon variables such as size of budget, periodical collection, and staff, this method may be a viable alternative to purchasing a commercial periodical holdings management service such as Serials Solutions. Yet its use is not limited to periodical holdings. This method of creating browseable lists might be employed to place the contents of a vertical file, a collection of ERIC documents, or an in-house index on the Web.

Designing the Web Interface for Library Instruction Tutorials Using Dreamweaver, Fireworks, and Coursebuilder

Doug Suarez

Macromedia's Dreamweaver, Fireworks, and Coursebuilder software were used to create Web tutorials for classes in the Faculty of Applied Health Sciences, Brock University.[1] *This paper offers some important tips that can be shared by all those developing similar pages. Design features rather than specific tutorial content are emphasized on the assumption that librarians need to incorporate these if their pages are to be effective. This software has very powerful features shared by other competitive products. In using them, however, librarians need to be aware that considerable time and effort are required to realize the software's full potential.*

Would you like to be able to produce your library tutorials on the Web but feel somewhat inadequate with all the technology floating around you? Or do you often think it would be quicker to create your own instruction materials instead of relying on your more technically inclined colleagues to create the Web pages? What follows are some experiences with the entire process of conceiving and developing instruction pages from the perspective of a true neophyte lacking much technical know-how and without experience using Web authoring software prior to this project.

Many instruction librarians today are using the Web in addition to their regular teaching tools. Whereas in the

Doug Suarez (dsuarez@spartan.ac.brocku.ca) is Applied Health Sciences Reference Librarian, Brock University, St. Catharines, Ontario, Canada.

not-too-distant past the only tools at one's disposal were overheads, whiteboards, or blackboards, and similar display items, today more use is being made of the Web. By mounting library tutorial material directly onto a Web page it is possible to reach a potentially wider audience and to update the material more quickly if necessary. More graphical features can be used to build pages and thereby create a more exciting visual product for the intended audience.

Instead of creating pages from scratch using direct HTML coding, authoring software like Dreamweaver and Fireworks can provide an environment where pages are produced more efficiently. Authoring software also allows for more creativity. Graphics can be created and used more extensively and as a result, pages will be more appealing for undergraduate students.

Libraries planning to build their own pages or use the Web as a delivery medium to create library tutorial content for existing courses offered on the Web at their universities can benefit from these preliminary pointers:

- Research the literature, the Web, personal contacts to find out what Web authoring tools are on the market, how they perform, what the probable learning curve is, and what they cost.
- Purchase the software in its most current version along with other related software that might be useful, such as graphics software (like Fireworks) that goes with Dreamweaver.
- Do not plan to rely solely on vendor manuals, online tutorials, and online help guides to provide all the expertise necessary to create good pages. Purchase other guidebooks and manuals if possible.
- Do not expect the vendor manuals to be much help in site or page design; most manuals show their shortcomings by assuming that their readers already know how to use authoring software or have experience in site design already. The manuals also presume their readers are producing commercial products so the tutorial examples are usually aimed at advertising a product for sale.
- Spend a lot of time looking at other sites, such as library instruction and nonprofit organizations, to get a feel for what others have done and to get ideas from interesting design elements to try to emulate.
- Make sure to have sufficient hardware to accommodate the software that is purchased. Graphics software tends to use a lot of memory so it is prudent to have at least 128K RAM. Purchase a computer with a CD-RW drive so that backups can be made of large files.
- Allow for the performance vagaries of the local network infrastructure that will be used for the Web site. For example, the system response time will have a great impact on the efficiency of transferring files to the local server, called a file transfer protocol (FTP). If it is slow or intermittent, it will limit the ability to transfer files, check the work on the server, and make corrections to files.
- Remember Rome wasn't built in a day, so think long term, rather than short term, and allow for contingencies.
- Plan to back up all work everyday!

Planning the Tutorial

It would be wise to spend at least one third of the total time planning the Web pages to be produced, whether relatively simple instructional guides or full-blown tutorials. In practical terms, this means that if three months have been allocated for the production of a library instruction tutorial, spend one month planning the actual pages, their probable content, and dealing with related issues. A common mistake that is made by many novice designers is to expect to launch immediately into the software and produce good pages. Note the following:

- Spend a lot of time planning what the instructor wants to say, based on the target audience, what the context of the tutorials will be, and how they will be used.
- Allow time for multiple revisions of the initial plan that will take place as the original ideas begin to develop.
- Use storyboards and sketch out the plan on paper using multicolored pencils and post-it notes to help keep track of ideas or sections of the site.
- Read manuals, articles in the library literature, and talk with colleagues about Web page construction.
- Try some of the tutorials suggested in manuals to get a feel for how Web pages are constructed using certain features demonstrated in the tutorials. However, be aware that the tutorial writers often assume their readers already know more than they really do about the finer points of Web site and page design. Don't expect too much from tutorials.
- Do not skip the manual's chapter on creating the local site. This is very important. Some of the greatest aggravations can be the result of not learning the intricacies of site names, file naming, and site management.
- Become familiar with whatever FTP program will be used to upload the file pages to the local Internet server. Whether the FTP capabilities that come with the Web authoring software (such as Dreamweaver) are used or an external program like WS-FTP is used instead, make sure the peculiarities of the program are recognized and compensated for in pragmatic ways. A great

amount of time will be spent using the program and making corrections to the Web files. Talk to systems or Web server administrators about directories, permission settings, and even Common Gateway Interface-binaries (CGI-bin) programs.

- Remember to include all the files, especially the graphic files, when they are uploaded to the local server. Unless the local server site mirrors the local site in the Web authoring software, there will be broken links on the site. This can be a constant problem when editing and changing links at the local level.
- Pay particular attention to links. When using relative links from files in the local site to files within the same site make sure the links are current and in the proper format.[2] Otherwise, there will be broken links. Similarly, when linking files to other files outside the local site, use the proper format for absolute links or there will be broken links as well.

Site Design

First and foremost, decide on the categories for the home page. Decide how the intended audience is going to navigate to the various parts of the overall site, how they are going to return to those parts when they are on other related pages, and how they are going to get back to the home page. In other words, think long and hard about navigation.

Develop a site structure where the home page acts as an index to the pages containing more specific content. The home page should answer two fundamental questions for the target audience—where they are and what the site purports to do for them—and should then provide the navigational means to get to subsequent interior pages.[3]

The home page should include a readily identifiable logo or heading that will automatically explain what organization the pages represent. This should be consistently used throughout the pages as much as possible. Termed branding, this is very important and allows the audience to identify the pages and prevents confusion once they begin to navigate to the internal pages. If they cannot easily see where they are, chances are they will easily get frustrated, lose interest, and exit the site.

Plan to include some colorful graphical elements on the pages to attract audience attention. Learn how to create graphics from scratch and experiment with shapes, sizes, and colors.[4] Minimize the amount of text on the pages as much as possible in order to keep the audience's attention without compromising the main messages. Users find it difficult to read text on screen and will generally skim over text blocks to see what's most important to them. The use of color, different font sizes, and font emphasis can help to pique interest on the pages.

Page Design

Try to ensure that all the items on a page line up with each other and remain consistent throughout. If headers are left justified, for example, make sure that all similar headings are left justified. If header text is contained within a larger graphic, try and create the text so that it is aligned consistently with the rest of the page and shares similar features with other text on the page.

Design the pages so that the main content items appear on the screen above the fold, or on that part of the screen that first appears on loading. Users do not always scroll down to the bottom of the page, so make sure they see the most important content immediately after the page loads.

Design the pages for users who use current browser versions such as Netscape 4.0 or higher. Doing this may be contrary to most current library practices; at the same time, by always creating for older browsers, graphics are not being used to their full potential. It is a fact, however, that multimedia elements are not going to be widely used until available bandwidth has increased significantly to accommodate software, such as Flash, that requires streaming.

Design pages for a larger screen size but always include a statement on the home page to warn users of the optimal screen size needed to best view the pages. Experiment with basic graphic design principles such as color, texture, graphics, different font sizes, text chunking, and borders to create user interest. The software manuals should be very helpful in this since most have separate chapters on how to create text, graphics, animation, and other design features.

Repetition serves to reinforce the key points. Use synonyms, if necessary, to avoid being too obvious and text elements such as bullets and summaries to consolidate content in appropriate sections. Use plain language as much as possible to avoid coming across as either too pretentious or condescending. Excessive use of specialized jargon can turn users off.

To allow for printing only text content from the Web pages, think about providing a separate print page link to a page that will print only the relevant text on a standard 8.5-by-11-inch page without printing graphics and other design elements.

Flexibility

In the progression from preliminary planning to site development and page creation, it will become evident that the pages will change and evolve in different directions. For example, once expertise is gained in creating graphics there may be specific situations where the page elements may have to shift to accommodate graphic size limitations. Or more pages may have to be added if the text material needs to be broken up into smaller chunks in order to avoid lengthy expanses of text.

Whatever the situation is, it is necessary to be flexible in the learning process. Different designs should be tried to see what works best for each individual situation. Trying to stick too rigidly to a predetermined plan is likely to restrict site effectiveness. This is why it is not always possible to predict what the pages will look like until different layouts are tried, some things are discarded, and others used instead. Don't be afraid to experiment and to ask others for their opinions. Try to get feedback from the potential users, if possible.

Dreamweaver and Fireworks

While both Dreamweaver and Fireworks are great tools for producing Web pages, they can be more than a little daunting to use initially. Pay close attention to the tutorials provided by each software package. Do not skip the preliminary Dreamweaver chapter on local site setup. It is crucial to set the site up in the prescribed manner and stick with naming conventions for files.

Pointers for Using Dreamweaver

The following summary of Dreamweaver features and their applications highlight the software's functionality:

- Use the layer feature to help set up individual graphic and textual segments of the pages. This can be extremely useful but there can also be pitfalls with using layers. Be careful of the sequential order in which the layer was created. Fill each layer with the content sequence that is intended to appear on the pages to avoid loading problems. Allow some room around layers if text is being used outside the layers, since these text boxes are essentially graphics and, as a result, take plenty of invisible room on the pages when they load. This can inadvertently cause some bizarre displays and disheveled pages. Remember, what is viewed on the page once it loads on the student's browser is not necessarily the same page that was viewed when it was created in Dreamweaver. The pages should be checked once uploaded on the local server to see the how they display.
- Using tables as a main page design element can be useful but tables also have their limitations. They can allow for design elements to be properly aligned and displayed on older browsers that have small display screens. The Dreamweaver manual instructs the designer to use layers and then to convert the layers to tables in order to allow uniform page alignment and conversion to version 3.0 browsers. However, this feature does not always work as intended and it restricts the overall size of the Web pages. The result can be very narrow Web pages and best suited for the older, 600-pixel browser screen sizes. Tables can be very tricky to master and best left for specialized uses within the pages.
- Create banners, index sidebars, and footers in Fireworks, in order to make full use of graphic design features, then export them over to Dreamweaver for use in page creation.

  Use the template feature to create pages that share constant design elements intended to be used again and again. The generic tutorial pages for the Brock University library instruction pages have the same structure for each of the subject departments in the Faculty of Applied Health Sciences with the result that a basic design was produced and template created for each tutorial module. This template file can then be called up when necessary, appropriate content changes made, and thus much repetitive input avoided in the creation process.
- To save a lot of time when creating textual content, use the HTML styles feature to make use of consistent font types, sizes, and colors.
- Check the results of the project work frequently by using the browser preview feature. It is prudent to use this feature constantly to see how the pages should actually look when loaded on the local server and viewed on the local browser. Be aware, however, that when this feature is used, the files are loaded into the cache memory on the computer being used to create them. This is why it is necessary to check the files once they are uploaded to the local server to see how they really perform once they are available to others on the Internet.
- Plan to add some interactivity and animation to the pages, but do not overdo it. For example, index buttons and main content headers using behaviors that perform specific tasks can be created in Fireworks. These behaviors alert the target audience to specific parts of the tutorial content when the buttons are activated. Buttons that change color slightly when the mouse rolls over them are more noticeable than those without these added behaviors. On the Brock University library tutorial pages, live effects were also created in Fireworks. These only work on initial page loading and use a short timeline for the animation. The presumption is that the text would be more noticeable if attention were first drawn to the header, instead of possibly being overlooked if users were quickly scanning a page.

- Plan to use alternative (ALT) image tags wherever possible to allow textual reinforcement of what the image is pointing to. For example, if the index buttons are graphics that will link to other pages, ALT tags should be added to indicate where these links will be going. When the mouse rolls over a button, an ALT text message should appear to signal where the user will go if the button is clicked. This is particularly useful for those who may be using older text-only browsers as well as for those who are visually challenged and not able to fully utilize graphics on a page.
- Use the objects palette and properties inspector floating windows to help create pages. For example, the objects palette is useful for importing graphics created in Fireworks. The properties inspector is used extensively for creating the proper links for objects and text, and for creating text properties such as font color and size. It is also useful for tables and other design elements.
- Use Fireworks, or possibly other graphics software like Adobe, to create graphical images. Then export these files to Dreamweaver to use in creating pages.
- Do not rely on importing text from Word documents for the text portions of the tutorial pages. In spite of providing a method of cleaning up HTML codes when the text is transferred, this feature does not always save time and effort.
- Always name each individual page with a page title by using the page properties feature. Each page should be self-evident to anyone who tries to activate a link to it. When the mouse rolls over a page link the page title should be clearly visible to the user. This feature also helps search engine searching, since the page title keywords are the primary page links to the site.
- Use library items whenever possible. Dreamweaver allows the saving of images, text, and other graphics that may be reused or updated frequently on library pages. This feature is very useful for creating navigation buttons because the graphics can be stored as library items and recalled whenever a navigation button is needed.

Pointers for Using Fireworks

Following are some practical tips for using Fireworks graphic software:

- Use the tutorials provided in the manual. These are quite helpful for learning how to create basic graphics. The software interface is presented clearly and provides many good examples.
- Use the toolbox tools and the tool keyboard shortcuts to access the various tools needed to create and modify graphics. The toolbox has many features that are user-friendly and familiar to anyone who has used other similar software products—Adobe Photoshop, for example.
- Creating interactive buttons is relatively easy using Fireworks. The software tutorial shows how to create one button and convert it to a symbol. Then copies or instances of the button can be recalled and edited on other areas of the pages whenever they are required. JavaScript behaviors can also be created for the buttons and easily exported to Dreamweaver for inclusion in the pages.
- Use masks to cover objects so that only the unmasked part of an object appears on the page. This feature is used on the home page of the Brock University library research tutorial pages. Graphics were imported from another source and an oval mask was used to transform a rectangular photograph into an oval-shaped picture.
- Hotspots are another useful feature that can add interactivity to a textual word, phrase, a graphical picture, or an image map to create a link without using a button. This feature is a great alternative to cluttering up the pages by using too many buttons. It also gives some additional design elements to provide a variety of links to various parts of the site.
- Experiment a little with animation. Fireworks allows for the creation of simple animations that might be useful in the tutorial even if the intention is to keep the pages on the conservative side. Animation is used to create the content page main text headers on the Brock University library research tutorial pages. The text appears in a set sequence when the pages first load, and JavaScript live effects provide an additional bit of pizzazz.
- Live effects are easy to create in Fireworks. Text, images, and even paths and strokes that make up graphics can have added effects such as beveled edges, drop shadows, and embossing. Some of these are used for some of the library items on the Brock University pages to give some dramatic effect to what had been plain vanilla text.
- Preview the course pages frequently. As graphics are created in Fireworks, use the F12 command to preview the work on the local browser. This is a tremendous help, enabling the designer to get instant feedback on what is being created and allowing for corrections on the spot.
- Use the image optimization feature to help create the best looking graphical image for a page before it is exported. Follow the instructions in the optimize panel and experiment with the results by looking at the results on the local browser.

- When exporting files to Dreamweaver, pay particular attention to setting the correct version of Dreamweaver that is being used for the site development. Also, choose the correct local site folder the files will be residing in. This is extremely important since all the files on the site must be in the same folder or directory of the local site for links between files to be active. Novice designers will inevitably experience some frustrations with this in the beginning.

Coursebuilder

Coursebuilder is an extension application for Dreamweaver and allows for the creation of some basic tutorial quizzes to complement tutorial pages when testing is desired. The software only works with Dreamweaver and, on installation, resides directly on the Dreamweaver local site. The objects palette is used to activate the software and then the type of interaction exercise is chosen for the type of quiz that is required. Drag-and-drop, multiple-choice, true-and-false, and explore interactions can be selected as templates. On the Brock University tutorial pages, multiple-choice and true-and-false interactions were used to create catalog, database, and general exercise quizzes.

Pointers for Using Coursebuilder

If you decide to use Coursebuilder, you should look over the following list of suggestions that provide effective implemetation:

- Copy the Coursebuilder support files to the local site folder that has been established for the Dreamweaver site.
- Use the object palette in Dreamweaver to activate the Coursebuilder interaction icon to open the software gallery template.
- Select the type of interaction that is required from the interaction gallery and follow through with the interaction instructions. The manual provided with Coursebuilder is fairly adequate for learning the software but, as with most manuals, it can be lacking in details and overall site planning. The basics can be learned pretty well through trial and error.
- Apply the software as a design element on the Web pages. For example, if an interactive quiz is desired at some point in the tutorial, Coursebuilder can be activated and an interaction type chosen. A quiz can be designed and created from the template and then the text can be edited using the Dreamweaver properties inspector. In addition, graphics can be added using Fireworks.
- Custom controls like toggle switches, push buttons, and timers can be created for the quizzes. However, these features were not found to be necessary for the quizzes that were developed for the Brock University pages.
- Coursebuilder allows for the scoring and grading of quiz responses. Called Knowledge Track, this feature requires more than a passing knowledge of computer-managed instruction (CMI) systems like Lotus Pathware, as the manual claims, and knowledge of Microsoft Access, SQL Server, or Oracle database management systems. The Brock University pages do not initially require scoring but this feature may be incorporated in later versions of the tutorial.

Conclusion

Macromedia software was used to build some basic library tutorial pages for the Faculty of Applied Health Sciences, Brock University. Increasingly, instruction librarians are using software like Dreamweaver, Fireworks, and Coursebuilder to build their library instruction pages. Sites with good graphical elements are expected from most of the Brock students since their other more prosaic experiences on the Web have prepared them for graphics and colorful displays. To avoid ignoring this audience, the effort should be made to produce more marketable products in the future.[5] With developments in streaming technology, other software products such as Macromedia's Flash will be used in more and more Web applications and libraries will be expected to keep up with these developments.

References and Notes

1. The author recently purchased a 128 RAM laptop computer, with a CD-RW storage drive, and Macromedia's Dreamweaver/Fireworks Suite (version 3) and Coursebuilder, a software extension application used for building Web interactions like quizzes. An eight- month sabbatical leave was taken to create the Web tutorials. The site can be viewed at http://spartan.ac.brocku.ca/~dsuarez/physeduc.

2. Links to files within the same site are termed relative links, while links to files that reside outside the local site are termed absolute links. The format for linking to each of these varies and must be followed precisely or your links will be broken and not available.

3. J. Nielson, *Designing Web Usability: The Practice of Simplicity* (Indianapolis: New Riders, 2000).

4. C. L. Nicotera, "Information Access by Design: Electronic Guidelines for Librarians," *Information Technology and Libraries* 18, no. 2 (June 1999): 104.

5. C. Ferguson, "Shaking the Conceptual Foundations, Too: Integrating Research and Technology Support for the Next Generation of Information Service," *College and Research Libraries* 61, no. 4 (2000): 300–311.

The Future of Ideas

The Fate of the Commons in a Connected World

by Lawrence Lessig. New York: Random House, 2001. 365p. $24.95 (ISBN 0-375-50578-4).

Lessig has written another important book. His previous effort, *Code*, which has been reviewed in these pages, masterfully examined how the law and cyberspace coexisted and how the architecture of cyberspace constituted its freedom. In his latest book, *The Future of Ideas: The Fate of the Commons in a Connected World*, Lessig claims to have changed his focus to the relationship between the architecture of the Internet and innovation. He does this, but he also reiterates his point that changes to the architecture can curtail freedom. *The Future of Ideas* is worth the time to read because it touches on many ideas that have shaped and will continue to shape the Internet.

As a lawyer, Lessig knows how to lay out his arguments to make points effectively. He is also a good writer, so the prose is clean and clear. Lessig first defines some key terms; in particular, he provides an explanation of "commons" and "layers," which are core elements to his arguments. Most readers are probably familiar with both concepts, so I'll concentrate on Lessig's use of the terms. A commons is a thing shared by all and free for others to take (although there may actually be a charge for access). Many of us are familiar with Hardin's discussion of the tragedy of the commons, wherein resources held in common are depleted by use. Lessig claims that the Internet is a commons and that it, too, is on the verge of experiencing a tragedy of the commons, although of a distinctly different nature.

The term "layers" ought to be familiar in a communications context. There are different layers that can be identified in moving data and information around a communications network. For his purposes, Lessig identifies three: physical, logical or code, and content. In the case of the Internet, the physical layer is the hardware—the wiring. Lessig discusses code in two different ways. The first is as the underlying protocols and software that make Internet communications possible, primarily TCP/IP. He also talks about code as a more general term encompassing all software. Content is the highest form of layer. On the Internet, the clearest example of content is the material seen on Web pages.

There are many lawyers who know little of Internet technology. If it comes to that, most people don't know much about the underlying technology. Lessig is quite knowledgeable and shows this by discussing the character of the Internet as what he calls an end-to-end (e2e) network. What he means by this is that the transport protocol, TCP/IP, is completely neutral as to the data it carries. Additionally, the "smarts" of the network are concentrated at its ends rather than on the network itself. This feature, which was part of the original intent of the Internet's creators, allows for a tremendous amount of creativity in what is built on top of the transport protocol. For example, consider the Web. Building the Web was made easier by the fact that TCP/IP does not care what is contained in the packets it transports. It was also easy to build the Web because there was no need to get permission to run a new set of protocols; the Internet is open to all comers. It was the openness of the Internet to innovation that enabled the revolutionary changes the Web has wrought. In Lessig's mind, innovation is enabled by the freedom of the commons.

In addition to the code layer offered by TCP/IP, Lessig sees the proliferation of free software as another commons. Among his examples are Apache, the free Web server, and Berkeley Internet Name Domain (BIND), the program that resolves addresses such as lessig.org into IP addresses of the form 123.45.67.128. These examples are commons because anyone can use this software and build new or enhanced software using its freely available code. It is not a commons that can suffer from the tragedy of the commons, because use or modification of the software does not deplete the resource. In fact, Lessig claims that the common availability of source code for software encourages innovation, a view not shared by everyone, but one that I think is correct.

Lessig believes that the commons of the code, in both the sense of the neutrality of the Internet and in the sense of free software, spurs innovation in a way that can only come from the fleet, not from older and entrenched companies that upgrade existing products only in incremental steps. There is no installed customer base that these new innovators need to worry about, so they can identify and develop what Christensen calls "disruptive technologies."

This seems like a little slice of heaven. But, of course, there is a darker side to this picture. It's a case of the old battling the new. Lessig examines the threats at the physical level (i.e., wires). Cable companies, many owned by Time Warner and AT&T, have policed their broadband access. Among other things, these companies may place limits on the number of minutes that a customer can use a streaming video connection, disallow the mounting of a server, or filter data and discard packets in an attempt to prevent file sharing. These measures constitute a restriction of the commons of the Internet.

In addition to restrictions on uses of the broadband connection, providers also tend to limit the Internet service providers (ISPs) who can offer services across their connections. For example, AT&T has restricted choices to only @Home or Road Runner, and Time Warner is pushing AOL. This is another kind of constriction of the commons, according to Lessig. It's not intuitive why this is the case, until you consider that the ISP itself may place barriers to full use of the Internet.

Lessig concludes that there are thus two models in competition with each other—the perfectly controlled world of the broadband provider and the freedom of the Internet. It is his claim that the fettered or restricted Internet stifles disruptive innovation. He even goes so far as to claim that such devices as firewalls also impose controls that may make sense locally but have a chilling effect on the Internet commons. I think he goes a bit too far in pointing to firewalls as a problem, but I think it's generally true that the control by near-monopolies of a vast subsection of the Internet pipes is a bad thing.

Lessig delves into the morass of copyright law, patent law, and the efforts of the entrenched old to control all of the layers of cyberspace. In addition to the cable saga, there are other constituencies who want to place controls on the Internet. Prominent among these are copyright holders, who are concerned that the Internet offers a mechanism to quickly and easily disseminate the content that they own. And they're right, of course—the Internet is a perfect medium to seamlessly move content. Corporate and campus attempts to limit peer-to-peer communication (P2P) aside, the Internet doesn't care what's being transported across it.

Lessig focuses on the physical world to discuss the stranglehold that entrenched copyright owners are seeking to place on the marketplace of ideas. Lessig harkens back to the original intent of the framers of the Constitution, who believed that copyright should only last for a limited time. Congress has made a mockery of this, consistently extending the term of copyright so that such valuable properties as Mickey Mouse don't fall into the public domain. These physical world actions impinge, of course, on cyberspace. Copyright holders, particularly those represented by the Recording Industry Association of America (RIAA), have been rabid in protecting their intellectual property. At least the idea of having copyright robots inspect and remove copyrighted material from people's individual PCs died aborning.

Lessig also laments the effects of patent law as currently interpreted. Until the 1980s, courts considered software and business processes to be unpatentable. Then it all changed, as courts opened up patentability of software. One problem is that patent lawyers are ordinarily not computer-savvy, so they may not understand that a particular piece of software is not really novel. Patents are then granted that shouldn't be, and the presence of patents curtails the richness of the Internet. Tim Berners-Lee, the father of the Web, says he has already "noticed its effect on Web development" (213). Richard Stallman, perhaps the earliest proponent of free software, says that such patents may be the worst threat faced by software developers. James Buchanan, a Nobel Laureate in Economics, suggests that there is an anticommons, wherein innovators are reluctant to work in an area in which patent holders could exert a claim on parts of the innovators' work.

There is still another commons and another set of problems that comes with it—spectrum. These are the frequencies on which wireless devices, among many other things, operate. There are huge chunks of spectrum that have been carved out by commercial television and radio. There are also portions reserved for emergency, military, and amateur radio use. No one really owns their little (or big) slice of spectrum, but once they've been granted their slice, they are hard to displace. There are limited portions of the spectrum still available, and the FCC is loathe to hand these out for free. Instead, they will be auctioned to the highest bidder, clearly favoring the large telecommunications providers. Again, this is another potential tragedy of the commons, for it would be better to offer at least some part of the spectrum to be as open as the Internet, some of it offering the same TCP/IP connectivity found in the wired world. Of course, this already happens with wireless networks, but there may be limitations placed on these networks if the for-profit segment has its way. Lessig believes that spectrum is inexhaustible, with all of us able to take little bites out of it for tiny slots of time without harming others' access.

I hope you are not as depressed reading this as I am in writing it. Lessig is persuasive in his argument that there are real and present dangers to the commons that is the Internet. Many of us have been aware of pieces of this problem, but have not seen the breadth and scope of it explored so rigorously and cogently.

Where do we go from here? is the question. Lessig provides a slim but content-packed chapter proposing solutions to the potential reining in of the Internet as commons. First, he proposes we open up portions of spectrum to function as a commons like the Internet. He offers concrete suggestions for how this might be done given the claims of many that spectrum is a scarce resource. Lessig also proposes that government should be helping to build infrastructure, providing as an example the work of Chicago in deploying dark fiber. This would be offered at some point to service providers.

The code layer is "the heart of the Internet" (246), and as such requires the most nurturing and protection. The danger to this layer comes from monopolists in two different contexts: the already discussed context of broadband cable, and the danger from companies like Microsoft that want to protect legacy software. Lessig apparently fears that the effort of Microsoft to protect its Windows operating system may spill over onto the Net, but he also views Windows itself as a barrier to the code commons, given that its source is not reviewable and open to change. What are the solutions? In the case of broadband, a solution Lessig offers is for government to simply require that any of the broadband players must offer Internet services consistent with the principles of e2e. That means unfettered, unfiltered access. As to the threat from Microsoft and possibly other entrenched legacy software vendors, Lessig suggests that the government should encourage development of open code by using it for all governmental computers, and share this code with anyone who's interested. The government should also consider alternatives to commercial operating systems. He sees no need for a coercive approach, although many of us might like to see a more active approach in the case of Microsoft.

Concerning the content layer, Lessig proposes some interesting changes to copyright and patent laws. Currently, some works are protected for as long as 150 years, a mockery of the Constitution framers' intentions. Lessig proposes changing to allow works to be registered and protected for up to fifteen five-year periods. If at any time during this period the copyright is not renewed it would fall into the public domain. Under this scheme copyright would be available for a maximum of seventy-five years. Lessig treats software copyright as a special case, and he has some revolutionary ideas for its copyright protection. Currently copyright for software is for the life of the author plus seventy years, or in the case of a corporation a flat ninety-five years. How can this contribute to the commons? Software would be laughably out of date by the end of the protection term. So Lessig proposes a five-year term of protection, renewable for another five-year period. But this protection comes with a stipulation: the copyright applicant must include a complete source code listing with registration, held in escrow by the Copyright Office. At the end of its protection, this code would be released to the public via the Copyright Office's server.

What of the contentious world of recorded music? Lessig looks back to piano rolls for the answer. When piano roll manufacturers transcribed music onto the rolls, music publishers complained that they were violating copyright. Congress intervened by requiring that the manufacturers pay a compulsory license fee for the piano rolls sold. This fee was not set by the publishers, so it was nominal. Lessig proposes that we do the same with recorded music: have an independent body set a reasonable fee that would be required for peer-to-peer sharing of music.

In the case of patents, Lessig suggests that the patent office study whether the patenting of software is more likely to aid innovation than harm it, and if the balance falls on the side of harm, that Congress should end software patentability. Tim O'Reilly of O'Reilly Publishing and Jeff Bezos of amazon.com jointly propose that business processes should only receive patent protection for a brief period of time, with Bezos proposing five years; Lessig would prefer an even shorter period. Lessig also suggests a closer look at royalty structures for patents, particularly with respect to software patents. Lessig seems less surefooted in his brief exposition of proposed patent policy. It is easily the least helpful of the sets of recommendations that he makes.

This is a first-rate and important book to read. As Thomas Jefferson once said, "The price of liberty is eternal vigilance." It's time for many more of us to be more vigilant. It may be difficult or impossible to make all of the changes Lessig suggests, but there are signs here and there that the entrenchment of the old is being overcome by the new. The Web won't go away, and neither will P2P file sharing, nor the next big thing that comes down the Internet pipe. In Congress, some leaders are fighting against the draconian measures that have and will be pushed by the more rabid organizations representing intellectual property holders. Perhaps the pendulum has moved as far as it will in one direction, followed by the inevitable return. In the meantime, read *The Future of Ideas*. It's well worth your time.


National Forum
Regional Institutes
ALA Conference Programs
Publications
Networking
Best Practices
Web Resources
Awards & Scholarships
Online Job Search
Trend Spotting

Software Reviews

Andy Boze, Editor

For this issue I thought it would be interesting to take a look at some of the network-based multimedia authoring software that's available on the market. The three products reviewed here are RealSlideshow Plus 2.0, ViewletBuilder 2, and Flash 5. They vary widely in purpose, functionality, and cost. About the only thing that they have in common is that they build multimedia presentations primarily accessed via a Web browser.

RealSlideshow Plus and ViewletBuilder are similar in purpose. They both build presentations that are a series of still slides with a soundtrack and added text. Otherwise, they are quite technologically different.

RealSlideshow presentations are generated into Synchronized Multimedia Integration Language (SMIL, pronounced *smile*), which is rather like HTML. If you'd like to know more about SMIL, you'll find plenty of information on the World Wide Web Consortium's Synchronized Multimedia site at www.w3.org/AudioVideo. RealSlideshow presentations are viewed using the RealPlayer. Viewlets created by ViewletBuilder, on the other hand, need nothing more to play back than a Java-enabled Web browser.

Flash is a much more complex product for designing and creating fully animated multimedia projects. It combines drawing tools with sophisticated animation tools and a scripting language to author interactive Web content. The Flash Player is available for all major platforms and operating systems, making Flash content viewable just about anywhere.

ViewletBuilder 2

Qarbon
84 W. Santa Clara St.
Ste. 790
San Jose, CA 95113
(408) 792-3800
fax: (408) 792-3808
www.qarbon.com

Price: $999, quantity and educational pricing available

System requirements: Pentium II 350 MHZ, 64 MB Ram (128+ MB recommended), 16-bit video card with 800x600 resolution or higher for Windows 95/98/ME/NT4/2000/XP and Linux2.x+ (tested on RedHat 7) + Java 1.3. For Solaris 2.5 (32 bits) + Java 1.3, UltraSparc II 360, 64 MB Ram (128+ MB recommended, and 24-bit video card required. MacOSX version in development.

Qarbon's ViewletBuilder 2 is designed to develop multimedia demos and tutorials quickly and easily. A viewlet is a series of slides with additional items that you can insert, such as notes and balloons with text, sound clips, and interactive elements. The compiled viewlets can be viewed by anyone who has a Java-enabled Web browser. The easiest way to explain the functionality of ViewletBuilder 2 is to describe the process of building a viewlet. This review was done using Windows 2000.

Upon opening ViewletBuilder, two options are presented: to create a new project or to open an existing project. A first-time user would choose a new project, which is what is described here. The screenshot wizard opens and prompts you through several steps. Step one is to choose the screen size of the final viewlet. There are four default sizes, ranging from 640x480 to 1280x1024 pixels, or you can fill in your own numbers for custom width and height. Smaller sizes are preferable to accommodate smaller monitors and for faster loading.

In step two, you are prompted to select a capture method, either full screen or a screen area. I preferred the screen-area method, which marks an area on screen that will be captured during the viewlet creation process. The full-screen method switches back and forth between your screen resolution and the size selected for the viewlet.

Step three lets you define a screenshot hotkey or assign a new one. I had no need to change the default from the Pause button, but to do so, you just click the Define button and then press the new hotkey on the keyboard.

Step four is when you take your screen shots. In this step, you start the application from which you want to capture shots, which I'll just call Demo Program. After Demo Program starts, you resize it to fit within the on-screen box you defined in step one. Then you just press the screenshot hotkey every time you want to capture a change to the screen.

Let's say that you wanted to develop a tutorial to demonstrate how to save a file in Demo Program. At every significant step in the process of saving a file, you only need to press the Pause key on your keyboard to capture the screen. When you're done, you click on the ViewletBuilder icon, which has been minimized to the system tray. All of the screen shots are then presented as a series of thumbnail slides.

To view or edit any slide, you just double click on the thumbnail. Once you're in slide view, there are two controls: a player and a toolbar. The player lets you move from slide to slide without having to return to the thumbnail view. The player also lets you play the entire viewlet, much as you would see it after it was compiled. The toolbar lets you quickly insert objects into a slide, although you could use the menus to do this, too. The objects available to place in a slide include balloons, notes, sounds, click zones, text zones, pause zones, hyperlinks, and cursors.

Balloons and notes are very similar. They both contain text that can be formatted in various ways from any of the available fonts installed on the PC. The main difference is that balloons have pointers attached to them, just as in comic strips, so the viewer can easily understand what the balloon text refers to. Notes have a pushpin icon at the top and are reminiscent of a note posted on a bulletin board. You can select from different colors for both notes and balloons, but notes are limited to one of four colors. Each slide can include only one note and one balloon

You can include sounds in a slide only if a balloon or a note has already been added, and each balloon or note in a slide can have its own sound attached. Using the sound tool, you can choose one of three options: no sound, import sound, or record sound. No sound is the default for balloons and notes. If you already have a sound clip, you can import it if it's in WAV or AU format, but there are some limitations to the recording parameters that ViewletBuilder will accept. It's probably easier to record your sound clip directly. Depending upon the functionality of your sound card, you can record from a microphone or various other devices, such as your CD player.

Recording a sound clip is easy enough—just click on the Record button to begin and the Stop button to end. You'll get a message if the recording level was too loud or too soft so that you can make adjustments and rerecord the clip, if necessary. You can also choose to listen to the clip you recorded to judge the quality for yourself.

There are four interactive objects that you can insert into slides: click zone, text zone, pause zone, and hyperlink. The click zone and text zone can be used for a simple quiz or when you want to check that viewers have understood information in your viewlet. In our Demo Program tutorial, you could use a click zone to ask the viewer to click on a particular spot on the slide. Similarly, with a text zone you could ask the viewer to enter a word or short phrase. For each zone, you can define the number of tries the user gets, as well as the response that the viewer gets for correct or incorrect responses. The viewlet waits to move on to the next slide until the viewer provides the correct response or exceeds the number of tries allowed. A pause zone inserted into a slide simply causes the slide to wait until the viewer clicks a button to continue. This can be useful when you want to be sure that the viewer has had sufficient time to look at all the information you have presented in a slide.

The hyperlink object is somewhat similar to a click zone, but when the viewer clicks on a hyperlink object, the Web browser opens a new page pointing to the URL that you embedded. Although there can be only one instance of each type of zone in a slide, you can have multiple hyperlink objects in a slide.

The picture is an additional object, which is not on the toolbar, but it can be inserted from the ViewletBuilder menu. Into any slide you can insert a graphic in any of three formats: GIF, JPEG, and PNG. After a picture object has been inserted, you can resize it and move it to any spot. A picture object can be useful to create customized buttons. For example, you could overlay the picture with a hyperlink object, so that clicking on the picture would open a Web page.

The final object that can be manipulated is the cursor. When you take your screen shots, the cursor is not captured, but the position of the cursor is. As the viewlet is played, a cursor is drawn on the screen, simulating the movement of a real cursor. As you edit slides, you can see the beginning and ending points of the cursor and the path between them. If these points are not quite where you would like, you can simply drag them to different locations, or you can turn off the mouse cursor on particular slides if you'd rather not show it at all. The final effect is quite nice and enhances the flow of the slides.

In addition to the toolbar, the player control lets you make some adjustments to the viewlet. ViewletBuilder automatically adjusts the duration of each slide depending upon its content. The more content, the longer the slide will remain before continuing on. If you want to change the duration of any slide, the player will let you add or subtract time in half-second increments.

When you've finished working with individual slides, you can return to the thumbnail view. In this view, it's easy to add additional slides if you found that you forgot something during the initial capture of screen shots. Just click on a slide, then select Insert from the menu. You can insert blank slides, new screen shots, or slides from previous projects. When you've made your selection, the new slides will be inserted into the position you chose, and the existing slides will be moved. If you have slides you no longer need, you can simply delete their thumbnails, or you can temporarily hide slides to see how your viewlet would look without them. Slides can also be dragged within the thumbnail view if you need to rearrange them.

The last step in creating a viewlet is to compile it. You can do this either from the menu in the thumbnail view, or you can use the Compile tool in the slide view. Dialog boxes will prompt you for information about your viewlet, and then you'll be asked for a location and file name for it.

After the compilation has completed, you are prompted to view or to upload your viewlet. If you choose to view, your Web browser opens with a page that gives you the HTML to insert in a Web page so that people can launch your viewlet. There's also a link you can click on to start the viewlet you just saved so that you can view the finished product. If you select to upload, you'll be prompted for information about where you want the viewlet to be FTP'd.

Using ViewletBuilder, making demos and tutorials could hardly be easier. A couple of times I found myself wishing that I could add more than a single balloon or note to a slide, or wishing I could add sound to a slide without a note or balloon, but after I thought about it, there are good reasons for these restrictions. The restrictions enforce good design principles.

I did run into some problems. ViewletBuilder requires Sun's Java Runtime Environment (JRE) 1.3, which it will install if you don't already have it. I had no problem with the installation, but ViewletBuilder refused to launch. I had previously installed JRE 1.4, which was recently released, and I had a suspicion that it might be the problem. I uninstalled both ViewletBuilder and JRE 1.4. and then I reinstalled ViewletBuilder, which launched without a problem afterward. Another problem I ran into was the inability to record sound clips into my viewlets using my microphone, although I could record clips from my CD player. I guessed that it might have something to do with the fact that I was using a USB microphone. A call to Qarbon's technical support confirmed both of my suspicions. ViewletBuilder doesn't support JRE 1.4, and there is an issue with USB microphones on Windows 2000, but apparently USB microphones will work with Windows 98. When I connected a standard microphone directly to my sound card, I was able to record clips successfully.

One final problem I encountered also related to sound clips. The sound clips played perfectly when I previewed the viewlets within ViewletBuilder, but failed to play in the final compiled viewlets. I tried playing the viewlets on two different computers, neither of which would produce a sound from the viewlets I made. I located a viewlet on Qarbon's Web site that contained sound, and it played perfectly on my office PC. Oddly enough, and for reasons I don't yet understand, my viewlets did subsequently begin to play sounds on my office PC. I was never able to get sound to play on the other PC, either from Qarbon's viewlet or from mine. This just underscores the importance of not using sound clips without having equivalent text in your viewlets.

In any case Qarbon's technical support was always good. I managed to work with the same woman each time I called. She was obviously familiar with the product and always had answers to my questions, or was able to offer appropriate suggestions. She mentioned that the Java and USB microphone problems will be addressed in ViewletBuilder 3, which is currently in beta testing.

ViewletBuilder 2 can be downloaded directly from Qarbon's Web site, www.qarbon.com. In fact, it is only available as a download, not as a boxed product, thus there are no printed manuals or documentation. Some people may find this a drawback, but the Web site has good documentation and tutorials, which are presented as viewlets. ViewletBuilder 2 can be purchased online or using more traditional methods, and the Web site describes other related services Qarbon offers.

The only real drawback I found to ViewletBuilder 2 is its hefty price tag. Even with the 50 percent educational discount, it may be out of some people's price range. However, Qarbon offers a free version of ViewletBuilder. It lacks a couple of minor features of the professional product, but the only drawback is that the compiled viewlets contain a banner ad for Qarbon. The free version will be good for those on a tight budget or for those who would like to try the product before considering a purchase. Viewlets compiled in the free version can be recompiled in the professional version, which will remove the banner ads. The free version offers one feature not available to users of the professional version. Qarbon maintains a viewlet farm on their own servers, to which users of the free version may upload their viewlets. This will be very handy if you don't have your own Web server.

RealSlideshow Plus 2.0

RealNetworks
P.O. Box 91123
Seattle, WA 98111-9223
(206) 674-2700
www.real.com

Price: $99.95

System Requirements: 120MHZ Intel Pentium processor or equivalent, 32 MB RAM, RealPlayer G2 (or later) or RealPlayer G2 Plus (or later), sound card, microphone, and speakers for Windows 95/98/2000/ME/ NT/XP.

RealSlideshow Plus does exactly what its name implies. You can prepare a series of images and add text and sound to build a multimedia presentation. RealSlideshow has a clean, intuitive drag-and-drop interface, so you don't have to spend lots of time learning how to use it. In fact, if you're the least bit adventurous, you can begin building a presentation almost immediately providing you've already collected all the media you want to show. First we'll take a look at the interface, then we'll go through the process of building a simple slide show.

When you open RealSlideshow, you'll see several horizontal bars for images and sounds, a timeline, and buttons along the bottom to configure your project's properties and layout. A standard menu at the top of the window duplicates all the functions within the interface.

The bars for images include one that shows thumbnails of each image, and the thumbnails are sized horizontally along the timeline to show exactly when each slide appears during the presentation and what its duration is. Another bar has a handle for each image that shows how long the image will take to download during playback.

The next two bars show where sounds are inserted in the presentation. The voice bar indicates whether any particular slide has a sound clip attached to it, usually a voice narrative, but it can be anything. The music bar indicates whether a sound track for the entire slide show is present. In both cases, the duration of the sound clips is represented along the timeline.

The last bar is the timeline itself. The timeline can be expanded or contracted, so you can actually see timings down to a fraction of a second, if you need that degree of control over your presentation. Individual slides can be resized along the timeline, and a cursor that runs through the timeline makes it easy to adjust the timing accurately.

Now we'll build a short slide show to illustrate the process. There is more than one way to accomplish most tasks, but RealSlideshow's drag-and-drop interface is likely to be the easiest way for most of them, so I'll concentrate on that.

Let's start by adding images to your presentation. You can open a Windows Explorer window and navigate to the folder that contains your images, then select one or more of them and drag them onto the images bar. Each image will appear in its own thumbnail. After you add your images, they can be reordered by dragging a thumbnail to a different location. Another easy way to add images is to double click on any empty space on the Images bar. Doing so will open a dialog box where you can enter a file name or navigate to a folder to select image files. A useful feature of this method is that the dialog box offers a preview of an image before you add it to your presentation. RealSlideshow accepts images in JPEG (.jpg), GIF (.gif), PNG (.png), or bitmap (.bmp) formats.

After you add at least one image to the Images bar, a title screen thumbnail appears before all the other thumbnails. If you double click on it, the properties dialog box opens. The properties dialog box is where you set information about your presentation, such as author, title, and description. Here also is where you can select the rate at which your presentation will be streamed, thus you can optimize your presentation for people using a dial-up connection or your local network.

After you have set your project's properties, you'll probably want to start editing the individual slides. If you double click on any thumbnail or on the space below it on the Voice bar, you'll get the properties dialog box for that image. The image properties dialog box contains four tabbed sections: Info, Edit, Transition, and Text and Audio Captions.

The Info tab displays general information about the image, such as its size, download time, and duration. If you'd like the image to be clickable during the slide show, you can enter a URL that the viewer's Web browser will open.

The Edit tab, not surprisingly, displays an area where you can do some editing on the image, including cropping, rotating, and flipping. There's also a slider that you can use to adjust image quality against download speed. A preview area displays the effects of any editing you do, and as you use the slider, you get a report of the download time at any particular image quality you select.

The Transition tab provides an area for defining transitions from one image to the next. There are nine transitions to choose from, including a fade, wipes, and pushes. You can set the duration of the transition, and the Play button lets you preview the transition effect.

Text and Audio Captions lets you record a voice caption for an image or select an existing audio file. After you've recorded a clip or selected a file, you can click a play button to preview it. You can also enter text for a caption that will be displayed with the image during the slide show.

If you want to have a background sound for the duration of the slide show, you can just drag any .WAV or .MP3 file onto the Music bar. If you don't have anything prerecorded, double clicking on the Music bar will open the project properties dialog box and allow you to record a track from a CD.

After you have all of the elements of your slide show in place, you can make some final adjustments. Each thumbnail can be resized along the timeline to give you precise control over the timing. RealSlideshow won't let you reduce the timing of any individual slide to less that the amount of time required to download the next slide, otherwise the presentation would become choppy.

You can also define the layout of your slide show. There is a default layout, but you can choose from among the more than thirty custom layout templates included with RealSlideshow, or you can create your own custom templates. Each slide must include a picture, at a minimum, but may include regions for text, text captions, and a logo.

The layout dialog box lets you define the height and width of the finished slide show. Any type of region you want to include is selected from a list, then you can place and resize the region by dragging it within the layout preview. A text region contains text that stays on screen during the entire slide show. Text captions change as each slide changes. The logo region is often used to provide a background for the slide show, and stays on screen during the entire slide show, just as the text region does.

The final step in creating a slide show is to generate it. When you click the Generate button, you are asked to provide a location to save the finished slide show. RealSlideshow automatically resizes any graphics that are larger than the size you defined in the layout so that they will fit in the picture region without being clipped. After the generation is complete, you can click the Play button to play your finished slide show in RealPlayer, which you must install separately. You can continue to make changes in RealSlideshow, and just click on the Generate button again to save them. If you are satisfied with the slide show, you click the Send button, which will let you FTP the entire set of files to your own Web server or to any of several commercial services with whom you might have an account. Your work may also be saved as a project file so that you can come back to it at any time.

There were only a couple of features that I'd like to see included in RealSlideshow. The title screen only produces white text on a black background. It would have been nice to be able to select text and background colors. The same limitation applies to the text and text caption regions. The text region does support RealText formatting, which is an HTML-like language, so it is possible to include some text effects with color and scrolling, for example. Those who are adventurous can open and edit the generated SMIL files with a text editor. With some knowledge of SMIL, you can produce some effects that RealSlideshow won't do.

The background music only holds a single file, and recording from a CD will only get a single track. You need to be sure that your selection is at least as long as your entire slide show, or the music will run out before the slide show does. It would be nice if you could record multiple tracks or use multiple sound files, or at least have the music loop.

I found RealSlideshow Plus very easy to use, especially with its drag-and-drop interface. With a little experimentation, I quickly taught myself how to use it. For those who prefer documentation, the online help is simple and complete. A link on the Help menu to a multimedia tutorial is also included. I used the downloaded version of the program. A boxed version includes a CD-ROM and printed documentation. If you'd like to try the software before deciding to purchase it, a free version is available with a slightly reduced set of features.

Flash 5

Macromedia
600 Townsend St.
San Francisco, CA 94103
(415) 252-2000
fax: (415) 626-0554
www.macromedia.com

Price: $399, educational price $99

System requirements: Pentium 133 MHZ (200 MHZ recommended), 32 MB RAM (64 MB recommended), 40 MB available disk space, color monitor capable of 800x600 resolution, CD-ROM drive for Windows 95/98/ME/2000/NT/XP or Power Macintosh (G3 or higher recommended) running system 8.5 or later.

If you've spent any time on the Web, you've undoubtedly run into Flash animations or movies. I was always curious how they were done, and now I know. It's not as difficult as I thought, and using Flash 5, it's actually fun.

I don't do a lot of work with drawing programs, but I've used several over the years. Flash 5 offers an environment that is immediately familiar and reminiscent of other products, such as Adobe Photoshop. Many options are available from tabbed palettes. Although this makes the options easy to access, I kept feeling a bit claustrophobic, even with a nineteen-inch monitor. Fortunately the palettes can be moved and docked, or closed if not needed. Different layouts for the palettes can be saved and restored later. A launcher bar at the bottom of the main window lets you easily access or hide common tool palettes. Among the items on the launcher bar are icons to change an object's color, change the properties of text, and associate behaviors with objects.

One thing I thought was a clever design feature is how the palettes behave if the Flash window is not maximized. The palettes can be moved outside the window. Many other programs allow you to do this, too, but then you end up with a lot of clutter on your screen, especially if you want to get to something else on your desktop. Clicking on anything outside the main Flash window automatically hides all the palettes. Clicking on the main window brings back all the palettes.

The tools palette contains most of the drawing tools that are available in any drawing program. The tools include line, text, pen, pencil, lasso, circle, square, and several others. Many of the tools have options that can be selected and configured. Oddly, tools for polygons and other common shapes are lacking.

Working with drawn objects takes a little bit of getting used to. The behavior is different than what I am familiar with in other drawing programs. For example, if you draw a filled circle, the fill and the bounding line are separate objects. Dragging the fill to a different location doesn't drag the line with it. You have to use the arrow tool to drag a box around all of the objects you want to select, then dragging any one of the selected objects will drag all of the others along with it. There's nothing wrong with this, and I got the hang of it quickly.

But Flash is much more than a drawing program, and here's where you'll notice the difference. One of the palettes is a timeline. You position objects on the timeline, then you define what you want to occur at points along the timeline. This is easier said than done. Fortunately, Flash comes with several lessons and tutorials to teach you how it all works. They are quite good at explaining the basic functionality and concepts, and while you're working on a lesson, you might even be tempted to think it's easy to create a Flash movie. What it really takes it lots of time and practice. The well written, printed manual is a good learning and reference tool.

Flash does have lots of features to ease the process of making an animated movie. One of these is tweening. Tweening lets you take an object and define beginning and ending points for it. Flash will take care of moving it along all points between. You can also tween shapes, which produces a morphing effect. All you have to do is define the beginning shape and the ending shape, and Flash will tween the location, size, and shape from one to the other.

Flash lets you share and reuse your objects—referred to as symbols—so that multiple authors can access the objects from symbol libraries that are stored separately from a project. When you change a shared symbol, the change automatically takes place in all the projects that use the symbol. Multiple instances of symbols can also be used within a project, in order to ease making global changes and reduce download time.

Because Flash projects can be large and complex, the Movie Explorer is a useful tool to view their structure. In a way, the Movie Explorer is an outline of a project, so that you can see how the project is organized, enabling you to find individual elements easily. This can be particularly useful if you need to analyze a project done by someone else, or if you are returning to a project of your own after a length of time.

An interesting feature of Flash 5 is its ability to create Web-native printing, or as Macromedia calls it, WYPINWYS—what you print is not what you see. This lets someone viewing a Flash movie, let's say a banner ad, to print a much larger page of information without needing to view it first. Another great feature is the ability to incorporate external HTML text in a Flash movie. Then, instead of having to redo the movie in order to change the text, all you need to do is revise the HTML file.

Flash 5 also includes a scripting language called ActionScript. It's similar to JavaScript, although not identical. In expert mode, you can enter scripts directly in an editor. In normal mode, you can select the scripting elements you want from a pop-up menu so that you can't make coding errors. The product includes a 453-page printed ActionScript reference guide.

As this review was being written, a new version of Flash was released. Called Flash MX, it has several exciting new features. One of these is video support. Flash 5 can only import still images, but the new version can import any standard video file supported by QuickTime or Windows Media Player, including MPEG, digital video (DV), MOV (QuickTime), and AVI. You can manipulate, scale, rotate, skew, mask, and animate video objects, and make them interactive using scripting. There is a new Free Transform tool that allows you to make any of several kinds of transformations to graphic and text objects, and the color mixer has been enhanced.

Laudably, the new Macromedia Flash Player 6 supports assistive technologies such as screen readers. Flash MX integrates tools for creating accessible content so that developers can add descriptive text to animations and user interface elements thus enabling users with disabilities to experience the content.

Flash MX is priced at $499.

Index to Advertisers