BRUCE R. SCHATZ

Associate Professor
Graduate School of Library and Information Science

Research Scientist
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

Electronic Libraries and Electronic Librarians: Who Does What in a National Electronic Community*

INTRODUCTION

I'm an information systems architect who worked at Bell Labs and Bellcore and IBM for many years, so it's my great privilege to be talking about electronic libraries at this Clinic. Actually, starting in the fall, I'll be a professor in the School of Library and Information Science here at the University of Illinois, and you'll see why that is potentially a very good match, although many of the things I talk about might seem very odd. I hope you'll realize that a lot of things I'm talking about are, in fact, mainline topics for library and information science. I'm honored to be giving this talk, and I hope I can give you a practical taste of "what the future will be like" and also what the information professional's role might actually be in this.

This talk will have two parts: First, I'm going to describe very briefly what this new kind of library technology is like through a discussion of the Worm Community System (WCS), why it is going to be very important, and why it will involve a lot of money. What I want to emphasize at the start is that while WCS may seem like an esoteric research project, in fact it is one of the flagship information projects funded by the National Science Foundation. In addition, the National Information Infrastructure Act looms in the immediate future, authorizing an enormous amount of money to be spent in the development of digital libraries in specialized areas. Digital libraries will require information systems like WCS. This project has become a national model of this new kind of information system, but its primary content is really just a special collection, in the same sense you already know. It is an important national effort, but there will be lots of other efforts like this in many different subject areas.

*This paper is an edited transcript of the author's presentation at the Clinic.

Second, I'm going to discuss in more detail what kinds of people are required to do this kind of activity. The roles range from traditional librarians all the way to systems architects. Similarly, the roles range from those that involve no computer knowledge at all to those that involve very intensive computing. My expectation is that people who call themselves "librarians" in the foreseeable future will actually span this entire range, even though now they are significantly skewed towards the traditional end.

ELECTRONIC LIBRARIES

What is a digital or electronic library? It's something like a physical library except that it's got different materials in it. It's dynamic so that people are not only navigating archival collections, but they are publishing their own materials as well. It's a repository of all sorts of things—different levels of quality and different types of information. Finally, all the items are linked together, and it's knowledge in some very profound sense. In effect, you get what I usually call an *electronic community.* The particular ones that I will discuss are in science, but you could imagine very similar communities in other subject domains.

In an electronic community, you have both formal and informal information, from both literature and data. The formal literature is a traditional domain of librarians, e.g., abstracts and full text. But, in science, large data collections are also important. For example, scientific databases are a very big area in genome projects and physics projects. The people who manage these collections are called librarians, but often they are actually trained first in a science and then in library schools. The informal literature and data include the communications services across the networks, such as electronic mail and bulletin boards; however, in an electronic community, the generated messages do not exist in isolation but are interlinked together back to the formal archives they discuss. This all is like a hypertext system, except that there is a whole library of items from many sources spread across the network. Therefore, it is really more like a hyperlibrary, where related items are linked together and references can be followed back to the source. This moves towards one system that lets you sit at your computer and navigate through all these different kinds of knowledge.

In order to see what such an interconnected space would be like and to understand what's involved in building the collection and writing the software, I've been concentrating on specialized collections in a subdomain of science in molecular biology. In particular, the library is for the "worm community," the molecular biologists who study the nematode worm *C. elegans,* and the project is called the Worm Community System (WCS).

This particular worm is a major organism in molecular biology. It has become the model for the human genome project because it's sort of bite sized—it's only got a thousand cells. It's big enough to be a real animal but small enough so you can learn everything about it. Its community is similarly appropriate for building a library, because it's big enough to be interesting but small enough to be doable. The knowledge base is fairly small in amount

(about 30,000 total units of information) and in size (mostly text and line graphics). The people are mostly at big universities that are hooked up to the Internet, so you can think seriously about building an interactive electronic library across national networks.

There are about 500 people in the United States and Europe who participate in the worm community. It's big enough so if you say, "Is this really a model of a national library?" the answer is, "Yes, it's sort of like a national library." It isn't really a full one, since it's small enough so that a modest research project can build it and do all the steps. And we've actually built the library, gathered the collection, and implemented the technology, so that we can study the social and organizational needs for effective system use. Remember that before the country spends 100 billion dollars to build a national information infrastructure and make a universal library, it might be prudent to build a large-scale model to understand what this would actually involve and which things would work and which things wouldn't work. I'm not going to tell you now which things would work and which wouldn't work (although I do have a little bit of information about that); I'm just going to indicate what the problems are, so you can understand what kind of people are needed to solve them.

The kind of knowledge in WCS covers both data and literature. There is a wide range of biology data, which is very specialized, genomic data (like genes and maps and sequences), and cellular data (like lineages). Then there are things that look like traditional literature, not just journals, but also newsletters and conferences. In biology communities, this latter informal literature is a very important source of information. For this community, we took the primary newsletter—about 10 back years of one-page articles—scanned it to get searchable text and figures, and proofread it so we could build automatic links from the references in the text. It turned out that this was one of the main activities that sold the system. The people who did this literature encoding are now calling themselves data librarians, although they needed a little bit of specialized biology knowledge. Lastly, there are informal materials, both data (like methods) and literature (like notes), that complement the formal archives.

Next I'm going to give you just a brief taste of the system functionality. WCS is a custom-written piece of software. It's running in 25 worm labs across the United States and a few in Europe. The system has an internal representation called an *information space*—not any of the traditional data representations but little pieces of information that could be text or pictures or graphics are all interconnected. The user can point to one piece and display it, then hop to the next one and display it, even though the first might be text and the second might be a picture, and the first might be in one physical location and the second might be in another. Internally, the system has something like a "federated distributed heterogeneous object-oriented database," that keeps track of where the information objects are and what type they are, so that different software can be invoked at the appropriate time. If you have seen demonstrations of WCS and other network-based information software, you know that the Internet is now fast enough for transparent access to be practical for the types of data needed for biologists.

The basic stages of system functionality within the information space are browsing, filtering, and sharing. In browsing, you locate items using information retrieval techniques like keyword searching and by navigation through interconnection links. In filtering, you examine the returned items to select those of current interest. In this domain, with biology data, it's not like text where you can look at it and say, "Yes, that's interesting." It's often something long and dense like a sequence that you want to feed into an analysis program. So this kind of scientific environment has to let you pass objects into other programs without much effort. Finally, in sharing, there's what could be called a publishing system to let you compose new items and propagate them to other people. For a more complete description of the system and how it's used, see Schatz (1991/92). For a general portrayal of the role of information technology in the context of science, see the National Research Council (1993).

Though we've built this nationwide information space in a specialized area, our goal is to work on building the Interspace, which is the information manipulation analogue of the Internet for data transmission. You build an information space for worms and then expand through molecular biology into another one for flies and another one for humans. Then you do neuroscience (which we've already started on), then physics, then humanities, and so on. What you get is all these specialized communities, special libraries that together across the national network will make up the grand national library. And so, if you keep connecting information spaces into the Interspace, finally you get the WorldNet. Thus the electronic community strategy is, "Today the Worm. Tomorrow the World."

ELECTRONIC LIBRARIANS

Well, the WorldNet is a great thing, but where is it going to come from? It's going to come from lots and lots of hard work and smart people who have a wide range of interests and skills. So what I really want to discuss is, "Where do those people come from?" The answer will explain why someone like me, who is in some sense only a systems designer, is a professor in a library school. It's because there is an important set of people who already exist, called "librarians," who don't quite have the right orientation yet to do this kind of project, and there's another important set of people who don't exist at all, namely those who design these kind of systems. These latter people, called "architects," aren't getting trained anywhere despite the crying need in significant national efforts. As part of their training, these systems architects need to understand traditional library and information science in order to be able to build the kinds of systems that are very much needed in the future. This is why systems architects and systems architecture belong in their natural home—library schools—to prepare people for these new roles.

Now that you know that there's a need for new librarians and that there's money in it and that it might be interesting to you, what do you have to do to be an *electronic librarian?* Well, here are the parts of the electronic library the way I've been defining it. There are users (who are the people who use the library), there's knowledge, and then there are things called systems that

are supposed to connect users to knowledge without getting in the way. Knowledge is the material coming from the community, and the question is, "How do you get it into the system, how do you encode it, how do you interconnect it?" The users don't care at all about that—the encoded representation. They care about the interactive navigation, or "How do I find what I'm looking for?" The systems designers, conversely, care mostly about building the information environments, or "How do I make everything transparent?" Transparency is a technical term that means that when I point to something, the system finds and displays it—I don't care where it is in the network or what type of data it is. There might be a lot going on in order to accomplish transparency, and the design represents very hard questions in information systems and computer science. Let me emphasize that you need to address all these (users—systems—knowledge) to have a complete and functional electronic library.

Before discussing roles per se, let's view the functions of electronic libraries in a slightly different way. There are three pieces: data—environment—programs. Each of these pieces is critical to a complete library, and each requires appropriate librarians to support the desired functionality. The "data librarian" is involved with supporting the electronic materials, i.e., "Where does the data come from? How is it gathered and connected?" The "program librarian" is involved with supporting the semantic relationships, i.e., "What are all the ways these data can or should be related? How are the relationships recorded?" This person must evaluate both the situation, e.g., standard places or exhaustive search, and the user, e.g., casual or serious interest, to be prepared to match the users to the knowledge. In some cases, there may be analysis programs that will help with this process, whereas in others, only personal experience will help. Finally, the "environment librarian" is involved with supporting the uniform interaction for the data and the programs, i.e., "What system is necessary to provide appropriate transparency?" This person is like an architect. If you only have books, you want somebody who builds a building for books. If you have a computer system, you have to have someone who builds the software, lays the networks, worries about the data.

All of these kinds of librarians are necessary to build electronic libraries in the future. As I discuss the roles of each in more detail, please note that all of these roles already exist in traditional physical libraries. What is different is the degree of programming needs and computer expertise required. In the discussion, I try to lay out a range of different levels of programming activity, to emphasize that there are important roles requiring very little computer expertise and important roles demanding very much computer expertise—that there is a role for everyone.

The problem with the future is that it's different from the past so it seems scary. But, on the other hand, it rarely is fundamentally different. In some significant sense, the same problems exist now as when the Greeks were trying to build the library in Alexandria. What happens is that the technology changes. Whether there are scrolls or books or disks, you have to worry about how to collect the materials, how to locate desired items, and how to retrieve the located items. So the same topics in library science, information science, and information systems recur in each generation of technology.

The title of the school here at Illinois—the Graduate School of Library and Information Science—is very nice because it includes both library science and information science as integral parts that are actually very closely related but that also have their own domains. The third domain—information systems—is beginning to come into its own as a separate entity with the current proliferation of computers and communications technology. I'm going to emphasize this third domain a bit more heavily because it's the one you are least familiar with and because it's the one that is my particular specialty.

Traditionally, librarians have simply bought information systems, primarily for automating circulation and card catalogs. Everybody knows that online card catalogs are really bad, and even if they weren't, they certainly do not do this community systems stuff. You need new, custom software to perform this new functionality, and it's got to be developed by somebody. Those somebodies are not people in computer science who are only interested in the technology itself. Those somebodies will be people in library and information science who are interested in building libraries to serve traditional needs with the new technologies.

To summarize the roles for electronic librarians, I propose that these new people provide new solutions for old problems. Those in "library science" are like "collection librarians," who perform the encoding and classification for electronic materials. Those in "information science" are like "reference librarians," who provide paths and analysis for electronic navigation. Those in "information systems" are like "systems architects," who design plugs and transparency for electronic environments. All these together are needed to build and maintain electronic libraries. Remember that for each role, there is a whole spectrum of people ranging from those who don't know anything about computing but now happen to deal with data instead of books to people who are expert programmers.

What I want to do now is go through each one of these roles and describe a range of sample tasks and real-life jobs that are going to be important in the future. I hope to give a concrete impression so that you can decide for yourselves if you would like to do this kind of activity or where you'd like to position yourself. The old activities will still exist, but they will become decreasingly important, and these new activities will become increasingly important. So you should think about how much training you need in order to be ready for the future. I've chosen stereotyped ways of discussing each one of these roles, which I know aren't the only ways, and I'm not a professional librarian, so I hope you'll bear with me if they seem narrow-minded. However, they should be illustrative of what kind of activities might be possible.

Library Science

A library scientist deals with these collections of interconnected knowledge; the corresponding role might be termed a data librarian. Library scientists have three primary tasks in dealing with the knowledge: collecting the materials, transforming the formats, and connecting related items within the materials.

The collecting task is very much like that performed by librarians in relation to traditional collections. For example, there are people in genome centers,

more than 20 in the United States, who maintain electronic collections of biological data without the need to know much about computing. They are database administrators, who basically know how to enter files and do word processing. But what they do for a living is just what librarians do. They locate a lot of sources, they figure out which ones are reasonable quality and which ones aren't, they classify the items, they make sure the items all have a name, and they update the collection periodically. It's just like maintaining a collection except that it is a database, and it involves a little bit of computing knowledge but no programming at all. It's really just getting a file from somebody, putting it in a specified place, and running a program on it that somebody else wrote. The skill here is making sure it's current, and if there are 10 possible sources for this piece of material, choosing the one that best meets your users' needs. There's a lot of people skills here, which librarians have, and a lot of economics, too, which is very important.

Transforming data, the middle stage of collection management in the electronic library, approaches territory that is new for most librarians. The problem is that almost none of the collected databases are in the right format for this grand universal system, and they have to be changed. Now, typically, these transformation programs are very simple to write and execute. They are like two-page C or awk programs that change the formats of data by changing the names of the fields and a little bit of the values. This is the sort of program that someone who has taken a single programming course in a library school can write. These are very easy programs—if you are at all facile with writing programs, you can write one of these in an hour or two. This makes you enormously more valuable because once you can do that, every time you want to change or add a database, you don't have to run over to the programmers and bother them and say, "I don't know how to change the names of the fields." That's so easy for them, they don't even want to talk to you. On the other hand, if you can take a programming course and write a few of these very simple programs, you now have an immensely valuable skill. It means you can traverse the network, grab these sources, and start adding them to the databases all by yourself. That's a very reasonable thing to want to learn, even for people who swear that they are not electronic librarians. Such a skill means you can move right into one of the big science projects, for example, and be the data librarian as a stand-alone, independent person. My guess is that while most of the current positions are at the collecting level, most of the future positions are going to be at this transforming level. Therefore, in the future, the data librarians are going to have to learn some programming and do their own database transformations.

Connecting, the final level for collections, is much harder and involves more extensive programming skills. It requires writing software to automatically build links between related items, by parsing out embedded names of objects and standard syntax for names. For example, in molecular biology, the programs parse text from many sources for gene names and connect them to referencing sources. These are somewhat harder programs than the transformation ones, though still within the reach of someone with a programming course or two,

and require some biology knowledge to implement properly. But, again, it's not terribly hard, so the more sophisticated people who know programming will tend to operate at this even more valuable level.

Information Science

An information scientist deals with the navigation of interconnected knowledge; the corresponding role might be termed a user coordinator (program librarian). Information scientists have three primary tasks in dealing with navigation: assisting the user in operating the system, scripting standard paths through the materials, and analyzing significant patterns between related items in the information space.

The assisting function is very much like the function performed by traditional reference librarians. Their primary task is helping people use the system to find desired items within the available sources. So they must understand the system as well as the knowledge, from a usage standpoint rather than from a system standpoint. Since the users typically run the system from their own computers, these librarians have the additional role of community systems administrators, ensuring that the users' sites have correctly operating machines and systems. These librarians answer questions such as, "How do I install the system? What software do I need? I want to do this search, how do I do it? What kind of words do I use?" They also write the online help and the tutorials by working with the users and the programmers. So, they understand how to listen to people, but they don't actually have much computing knowledge.

Every project that succeeds has a number of people who essentially provide user assistance and training. It's computer assistance, but their knowledge of programming isn't very great. When they get a little more knowledgeable, they can find standard paths. One of the problems with having this grand interconnected space is that you can't find anything. Anyone who has used Gopher, for example, knows that this is a real problem if there are hundreds of thousands of sources. It's like having a library without a card catalog, and you have to read a book then jump to all the things that some person who didn't know what he was doing and didn't understand the subject very well connected to it. Well, there are facilities in these community systems for recording navigation, so if you have found a valuable path through the space, you can record it by either recording an actual session or by doing a sort of meta-classification by specifying a set of useful items about, say, molecular biology, even though some of them are in a physics database.

Scripting is thus like the work of a reference librarian, who can write programs available to users to satisfy simple requests. So, if you can do a little programming, really just specifying sequences of commands, then you can be a more effective reference librarian because you provide scripts that can automatically handle some common user queries. This mechanism is not as good as a person, but it serves more users. In slightly more general implementations, such scripts become an encoding of reference works about basic information sources. This type of program is becoming popular on the

Internet as a solution to the resource discovery problem of "knowing where to look."

Analyzing is the final level for navigating the information space. This consists of writing interactive software to perform sophisticated pattern finding and real semantic matching. Finding nontrivial patterns of related items will likely require both deep semantic parsing and flexible contextual display of the resulting connection graphs. For example, in the worm space, you might say, "I've done this traversal through genes and maps and literature, and I think this uncovers the mechanism for fertilization of eggs in worms. Find me some other navigation graph that's very similar in biology space, which represents some similar pattern in some higher organism, so I can compare the mechanisms." Then the analysis software suggests related hyperbooks or related subcollections, which is very sophisticated programming that doesn't work very well at present. On the other hand, my guess is that a lot of people who at one time might have been reference librarians, and who are now sailing around the Internet and the information spaces, will want to write sophisticated programs to help them find patterns more efficiently so that they can become real trailblazers.

Information Systems

An information systems designer deals with environments for interconnected knowledge; the corresponding role might be termed an information specialist (environment librarian) at the low level and a systems architect at the high level. Information systems designers have three primary tasks in dealing with the environments: customizing existing systems, designing new systems to match user needs, and implementing new designs to provide functional electronic libraries.

Customizing is very much like what is done by a traditional information specialist. Such a person is a technical staff member in a library, who interviews vendors of existing information systems and chooses the most suitable system for the needs of the users. If the specialists are lucky, they can customize the system a little bit and change the data to their taste. Usually, however, the system does what it does, and the library must cope with the functionality provided. The specialists have to know a little computing, but mostly they just select from given choices.

The problem is that existing information systems do a poor job of satisfying the needs of many users. For example, all the big science projects that have tried to use commercial databases find they just don't suffice. The systems don't lose the data, but they don't provide any help in navigating and analyzing, in finding out what really is connected to what. That is, existing systems don't really allow the scientists to ask the kind of questions that they want to ask in order to make good use of databases. To have an effective system, you really need an architect.

Designing is what an architect does. A building architect designs buildings (physical structures), and a systems architect designs systems (logical structures). An architect finds out from the users such vital sociological specifications as what kind of searching they want to do, what kind of navigation they need,

what kind of sharing they want to do, and what kind of analysis they need. Given these specifications, the architect lays out the entire functionality of the system, then knows enough about the technology to estimate what can be implemented, what cannot be implemented, how much it costs, how long it's going to take to build, and so on, through the entire process of creating an electronic library.

Implementing, on the other hand, is what a builder does—the realization of a design into an actual structure. In established fields of architecture, such as those for physical structures, there are formal disciplines for architects and builders with different organizations specialized to the different tasks. For less-established fields of architecture, such as that involving logical structures like systems design, the tasks merge. Typically, there is a small architecture team formed at the beginning of the project, which then expands to become the complete development team. The original architects then become the supervisors of the programmers doing the implementation. Some organization must stay in place to maintain the system and help it evolve until it stabilizes to fulfill the needs of the users.

Information Systems Architect

What emerges from these observations is that there must be a true profession of information systems architects. Just as the world needs people who create buildings, namely, architects who design buildings for particular needs, the world needs people who create systems, namely, architects who design systems for particular needs. Architects have to understand a little bit about everything. They are really artists, if you think about it, but also like engineers. What they do is match user needs to feasible technology. Or, restated, an architect matches a set of knowledge and navigation needs to what environments you can actually build now. And what you can do now varies dramatically over time. The computer industry is growing very fast, while the users' needs are relatively static. An information systems architect designs and implements electronic libraries, in this case, for specialized communities that have a particular set of knowledge, a particular set of needs. They are special librarians, who can create all of the components necessary to build and maintain an electronic community library.

Professors in library school can't actually build large commercial systems; they build models of future systems. In that sense, I'm no longer a commercial systems architect. Instead, I do research in information systems architecture by designing and implementing large-scale models in scientific domains. What I try to do is to design protocols for information manipulation and to build frameworks of underlying software to increase the technological understanding and sociological analysis of electronic community systems. This might be thought of as constructing toolkits for knowledge environments, to learn how to effectively construct complete community library toolkits by implementing model community library systems in the sciences. To be successful at architectural design of toolkits, you have to design a lot of real systems and see how they play in the living world—at least small ones, if not huge commercial ones.

The conclusion is that if you want to construct these large-scale electronic libraries that are special collections for particular communities, you need the

magic triangle of users—systems—knowledge. This is actually the same triangle that appears in library school brochures as the core of the subject. And everybody likes triangles because they look complete. What is at the apex—I am immodestly putting myself at the apex—is the environment, the system actually running, but in order to make it work at the base, on one hand you need to have all the knowledge stuff, the real data, and on the other hand you need to have all the user stuff, the real people. So to make electronic libraries happen, you need systems architects, data librarians, and reference librarians.

The problem is, and the reason I'm at a library school now, is that there are people who train data librarians that are sort of like library scientists, although they need to be pushed more to higher levels of more electronic and computing skills. And there are people who train reference librarians who are sort of like information scientists, but, again, if you're really going to search around these huge information spaces, you need people with higher levels of skills in computing and information analysis programs. But right now, there's a large hole in the training of the requisite systems architects.

Where do new information systems come from? The answer is that right now they really don't come from anywhere. The hardware is growing on an almost infinite upwards curve, and the software is sort of sneaking along but getting increasingly more sophisticated. But the information systems, not just fancy displays but what people can really do, are very little changed from what they were 25 years ago. There's a big national crisis here. If you have lots of people who live in cities, you really need architects or you can't have a functional city. Well, if you have lots of people who are going to live in information spaces and live in these electronic worlds, then you really need information systems architects or you can't have functional systems. The revolution of the WorldNet will never reach its genuine potential without fundamentally new information systems, which must be designed by these missing architects. All of you can see that the revolution should and must come, or you wouldn't be at this conference on networked communities. Systems architects must be trained, like other architects, through an apprenticeship, where they build increasingly larger systems and learn the complexities required to design functioning electronic libraries.

So, if anyone in the audience would like to learn to be a systems architect, I'd be very pleased to talk with you afterwards. Thank you for your patience and attention—it is you who will invent the future.

REFERENCES

Schatz, B. R. (1991/92). Building an electronic community system. *Journal of Management Information Systems, 8*(Winter), 87-107. (Reprinted in R. Baecker (Ed.), *Readings in groupware and computer supported cooperative work.* Los Altos, CA: Morgan Kaufmann.)

National Research Council. Committee on a National Collaboratory. (1993). *National collaboratories: Applying information technology for scientific research* (Computer Science and Telecommunications Board study report). Washington, DC: National Academy Press.