

Toward an Expert System for Reference Service: A Research Agenda for the 1990s

John Richardson Jr.

Reference service exists to maximize access to data contained in library material. Yet reference librarians have not achieved this goal in several areas of reference work. While an expert system has possibilities, formidable research and development obstacles exist. In the form of a tutorial, this paper posits an explicit research agenda: (1) to define the fact base and articulate the heuristics necessary to build the requisite knowledge base, (2) to select the appropriate programming language or shell, (3) to design an effective user interface, and (4) to develop an expert system capable of operating in a real-time, reference environment. This paper also specifically addresses system testing, describes what has been done, evaluates the existing systems, and identifies work in progress. Finally, this paper raises seven critical questions which must be answered along the way.

No one yet has succeeded in inventing an automation to answer all the wise and foolish questions asked by the American public.—Louis Shores, 1937.



Reference service developed before the turn of the century to provide readers advice on how to retrieve relevant and pertinent sources with which to satisfy their information needs.¹ Its goal is to maximize access to the information contained in library collections. Today, either explicitly or implicitly, many reference departments have adopted the American Library Association's Reference and Adult Services Division standards of service.²

Most departments wish to provide the

best possible service. Yet substantial evidence suggests that, for a variety of reasons, the quality of reference service is not high. Extensive studies of the quality of reference service have consistently found that the accuracy of answers to questions is very low because, among other reasons, librarians use outdated sources and make only infrequent referral to more knowledgeable staff.³ Unfortunately, researchers do not know how many times library users' questions simply go unasked.

In attempting to answer questions, reference librarians face "several alternative courses of action but [have] only incomplete information about the true state of

John Richardson Jr. is Associate Professor in the Graduate School of Library and Information Science at the University of California, Los Angeles 90024. The author particularly wishes to thank Boyd Sutherland, his summer 1986 CRL Research Assistant; Bob Tennant from his fall 1986 GSLIS 420 "Information Resources and Services I" class; all nineteen students in his fall 1987 GSLIS 420 class; Michael Chung, an Anderson Graduate School of Management doctoral student, on whose expert systems dissertation committee he serves; and especially his spring 1988 GSLIS 596 "Expert Systems Group" students—Deborah Henderson, Kayla Landesman, Patti Martin, Lauren Mayer, Pamela Monaster, Maloy Moore, and Edward Pai—for leading him into deeper thought about this subject and encouraging him to provide clearer explanations.

affairs and the consequences of each possible action. The [general] problem is to choose an action that is optimal or rational with some definite criteria of optimality or rationality."⁴

PROBLEM STATEMENT

It is not only possible but desirable to build an expert system, i.e., a decision support system for answering reference questions (see figure 1). Of course, the domain of reference service encompasses more than answering questions.⁵ Several alternatives exist to improve the quality of answers. Library administrators could spend more to attract higher quality staff or to improve reference collections. In-house staff training could emphasize the importance of referral to other library departments that contain specialized information, such as government publications. Similarly, public relations efforts could more effectively advertise the existence of ILL.

"... a study of the intelligence required in reference service, specifically that of answering questions, could significantly improve user access to the information in library collections."

Expert systems are known to work well in narrow domains. Yet the knowledge base, consisting of the facts and rules necessary to build such a system, is still not well understood. Donald Waterman points out that "if the task is so new or poorly understood that it requires basic re-

search to find solutions, knowledge engineering will not work."⁶ Such work involves some risk. Nevertheless, a study of the intelligence required in reference service, specifically that of answering questions, could significantly improve user access to the information in library collections. I am confident that the payoff justifies the risk. Thus I wish to propose a research agenda for the next five to ten years on seven critical questions in this area. These questions must be answered if we are to have a truly expert system for reference service.

JUSTIFICATION

Hypotheses of justification for work on expert systems posited until now address the economic or technological reasons for proceeding with the development of particular systems.⁷ An expert system in reference is desirable primarily because it can preserve the corporate memory within reference departments and can increase the individual's success in answering questions.

The groups that stand to benefit most immediately from better answers and/or an expert system are end users and, of course, librarians (see appendix A). Potentially, an expert system could teach reference, so library school faculty and their students could have a stake in this venture as well. Finally, reference book authors and publishers have a vested interest in this field because such systems may suggest the need for new sources. At the very least, existing sources will be recommended and publishers may also wish to finance new ventures in this profitable area. Even though the advantages appear to outweigh the disadvantages, researchers should weigh the pros and cons and

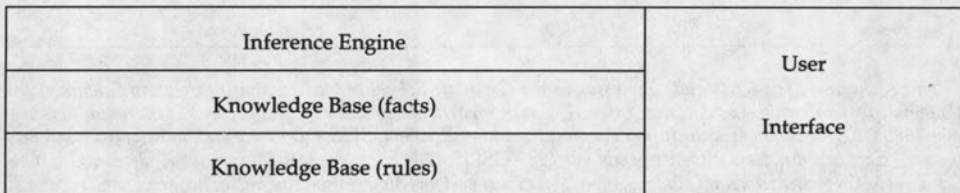


FIGURE 1

Components of an Expert System (suggested by N. Shahla Yaghmai)

their effects upon the implementation and operation of an expert system.

Researchers interested in these developments must address several moral and ethical questions before proceeding. A central question concerns the proper role of an expert system: what can it do and how much should it do? In other words, how much responsibility should it be given? Should end users or only reference librarians have access? Is the system an adviser, an associate, or simply an assistant?⁸ What are the consequences of a wrong answer? Who is responsible for wrong answers given by an expert system? How does it mesh with what librarians do now? Who owns this expertise, that is, the knowledge base.⁹ Should researchers limit themselves to data currently available or is a more fundamental study needed of how reference librarians actually answer questions?¹⁰

THE RESEARCH AGENDA

What Is the Proper Scope of an Expert System for Answering Reference Questions?

Fundamental theoretical issues about the knowledge base have not been resolved, although development of a system is technologically feasible. The essential question is: what must an advice-giving system in reference know? To begin with, the relevant knowledge domain of an expert reference system includes the fact base and the rule base.

What Is the Fact Base? The fact base is the explicit and declarative knowledge within the domain. In reference, the fact base is largely "public knowledge," in Patrick Wilson's phrase. It contains the reference resources, i.e., the basic tools of reference work. Besides including traditional print-based sources, their call numbers, and/or their locations, should not an expert system's domain also include in-house information files, CD-ROM products, interlibrary loan or even online databases?¹¹ Does it include knowledge of how to use the catalog; library policy; the physical layout of the main reference collection; and location of other collections or facilities, such as buildings, photocopy machines and restrooms? Does it include in-

formation necessary to refer the user? The fact base must be operationally defined and at the very least must contain the print-based resources, but even this requirement is problematic. How many titles should it contain? The same number as a reference librarian?

Mary Biggs and Victor Biggs (1987) found that collection size in the main reference collection of academic libraries varied from 35,000 titles for a college to 82,000 titles for Association of Research Libraries (ARL) libraries.¹² Must a truly expert system base recommendations on the entire collection? Alternatively, the fact base could be defined as all the titles in the tenth edition of Eugene Sheehy's *Guide to Reference Books*. Over the course of its development, this source has grown from only 100 titles under Kroeger's 1902 editorship to approximately 14,000 in October 1986; apparently, it will continue to grow. Even expert librarians must find this a daunting number; and consider the poor novice. Of course, the system may never use some of these titles or may use them infrequently. Nevertheless, *human* experts will still have a limit to the number of sources that they can remember to recommend.

Seeking informed opinion represents yet another way to limit the fact base. In 1960, Wallace J. Bonk at the University of Michigan found that library school faculty teaching reference courses in twenty-five schools collectively cited more than 1,200 different titles in their syllabi.¹³ He lists 352 titles, identifying 115 core works that have at least 50 percent overlap. Notably, only five titles appeared on all twenty-five of the library school's lists.

In a subsequent study of reported use in 1,078 secondary school, public, and academic libraries, Bonk asked reference librarians to identify titles as vital, recommended, or peripheral.¹⁴ Although he reports on individual titles, in ranked order by format, he found that the vital category consisted of handbooks first, then geographical sources, biographical sources, government publications, yearbooks, dictionaries, serials, encyclopedias, indexes, bibliographies, and directories, followed by audiovisuals.

In 1979, *RQ* published Larsen's replication of Bonk's study of reference instructors.¹⁵ This time thirty-one schools responded, but only sixteen provided usable syllabi. Nevertheless, schools listed many more reference titles: 2,014 different sources. The range was from a high of 615 titles to a low of 229. By format they presented encyclopedias most often, followed by yearbooks, biographical sources, indexes, bibliographies, geographical sources, dictionaries, directories, audiovisuals, government publications, and lastly, handbooks. Two encyclopedias, two biographical sources, two indexes, and one yearbook emerged as core titles.

The fact base can also be more narrowly prescribed by studying how many titles librarians actually use. The Enoch Pratt Library listed the top ten most frequently used titles in a 1968 survey of telephone reference. Their Telephone Reference Service collection contains 750 titles, which are used to answer about 80 percent of questions asked.¹⁶ More recently, a statewide study in Maryland found that as few as seven titles were used to answer about 87.5 percent of questions asked.¹⁷ Interestingly, a single title—the *World Almanac*—was used to answer 57.5 percent those questions.

Should the system contain a limited number of titles, such as those that the library owns? Does merely increasing the size of the fact base result in a better system? Should the system recommend more than a single title, for educational purposes? Might the fact base become prescriptive, i.e., leading users to think these are the only approved tools? Determining the appropriate number of titles is a critical design issue because the fact base must be manageable yet large enough to satisfy user requests.

Can Experts Articulate Their Heuristics?

What Is the Rule Base? Expert knowledge can be represented by rules. These rules, or information about courses of action, constitute the procedural knowledge of a field. Such heuristics work best when no algorithmic solution exists, but rules offer

no guarantee of a solution to the problem every time. In contrast to the fact base, which is public, librarians' implicit rule base for solving reference problems appears to consist of nearly entirely private knowledge. That knowledge that is public and contributes to the rule base, however, should be discernable in the professional literature, especially in texts on the proper way to perform reference work.

Having examined the textbook experts on reference work—specifically, Wyer (1930); Shores (1937, 1939 and 1954); Hutchins (1944); Cheney (1971); Katz (1969, 1974, 1978, 1982, and 1987); Cheney and Williams (1980); and Thomas, Hinckley, and Eisenbach (1981)—the author is pessimistic about finding there all but the simplest rules.¹⁸ Furthermore, very few library schools teach the explicit rules of reference. (See figure 2 for an example of such surface rules for dictionaries). The heuristics—the rules of thumb—must be learned indirectly by students during class lectures or during hands-on assignments. If textbook authors and professors are not revealing the rules, who else can?

One approach is to conduct interviews with the other experts, the practitioners. Among reference librarians, how can we determine who is the most expert? Should they be given a version of the now familiar twenty questions used in studies of reference quality and see how well they do? Or should we search for the one answer upon which a number of practitioners agree? Once the expert has been identified, how can we learn how they perform reference work? How does one obtain the best expert's best opinion? Should they be interviewed *in situ*? They may not be able to articulate the process; many will simply answer, "I just know," or that they make educated guesses. They will be able to identify the tools, but only a few of the simplest rules? For example, "IF the client wants to know the meaning of a word, THEN recommend a dictionary" is a simple rule. Deeper rules address understanding, for example, "IF there is a business or professional address associated with a person's name, THEN it may help establish the credibility of that person."

Some answers can be found in related

<i>IF (condition)</i>	<i>THEN (conclusion)</i>
Spelling	Webster's 3d
Definitions	Webster's 2d
Pronunciation	Webster's 2d
Etymology	Oxford English Dictionary 2d
Levels of usage	Fowler's <i>Dictionary of Modern English Usage</i>
Pictures or illustrations	American Heritage
Synonyms or antonyms	Roget's International Thesaurus; Webster's Collegiate Thesaurus
Neologisms	World Book Dictionary; Barnhardt's; RHD 2d; Webster's 9th Desk; OED Supplement
Slang	Partridge's <i>Dictionary of Slang</i> ; <i>Dictionary of American Slang</i>
Dirty words	American Heritage; RHD 2d
Dialect	Dictionary of American Regional English
Grammar	Strunk and White's <i>Elements of Style</i>
Abbreviations	De Sola's Abb. Dictionary
Nonlexical	RHD 2d
Translations	
German	Langenscheidt's <i>Deutsch/English</i>
French	Cassell's <i>French/English</i>
Italian	Cambridge Italian Dictionary
Spanish	Appleton's New Cuvas
Russian	Miuller's <i>English/Russian</i>

Source 3: Author's research in progress; Shores, *Basic Reference Sources* (1954), p.9; Katz, *Basic Information Sources* (1969), p.14; Cheney, *Fundamental Reference Sources* (1971), p.112.

FIGURE 2
Production Rules for Selecting Dictionaries

fields. A review of psychological research suggests that

when people attempt to report on their cognitive processes, that is, on the processes mediating the effects of a stimulus on a response, they do not do so on the basis of any true introspection. Instead, their reports are based on a priori, implicit causal theories, or judgments about the extent to which a particular stimulus is a plausible cause of a given response. This suggests that though people may not be able to observe directly their cognitive processes, they will sometimes be able to report accurately about them. Accurate reports will occur when influential stimuli are salient and are plausible causes of the responses they produce, and will not occur when stimuli are not salient or are not plausible causes.¹⁹

Interviewers should ask librarians what they would do in a given scenario. The risk is that the experts might rationalize what they do rather than say what they really do.²⁰ An alternative method might have an expert and a novice discuss a problem reference question scenario; the recorded exchange could reveal important differences. Is there an appropriate method for extracting the librarian's cognitive model? Researchers must under-

take further exploration of reference librarians' cognitive models if we are to have truly expert systems.

"How does one move the expert's domain-specific information into the machine?"

Having answered the questions about the knowledge base, researchers or so-called applications engineers can next direct their attention to building a system that can choose a resource based on a rational set of facts and rules. How does one move the expert's domain-specific information into the machine?

Is there a good way to capture or acquire this knowledge? Some promising work by George Kelly suggests that experts identify relevant information as cues in their work environment.²¹ By combining cues, these experts construct decision-making patterns. Hence, cue identification is critical. Researchers have generated a variety of inductive learning methods to elicit the

cues or rules from the environment with or without the help of an expert.²² Yet we may still be left with the question, Does merely increasing the number of rules, simple or deep, in the knowledge base make for a better system?

Simple rules or surface knowledge can probably be articulated easily, and these should be taught in library schools. Library schools would produce better-prepared practitioners who would at least know the fundamental logic of answering questions. However, these simple rules occasionally fail. Certainly, expert systems based solely on simple rules could frustrate a user. Thus, we will need deep knowledge of the reference process. Such first principles (axioms, definitions, laws) may be harder to identify, but truly expert systems will need to know these as well; symbolic logic can play a role.

METHODOLOGICAL QUESTIONS

What is the Best Approach to Implementing an Expert System?

Several approaches can lead to an expert system. These approaches have been grouped into three categories: the custom approach, the semicustom approach, and the off-the-shelf approach.

Custom Approach.

A custom development route which starts from scratch using AI development languages and highly skilled AI professionals to build a system to meet specific needs [is one approach]. After the two professionals (the knowledge engineer and the expert) create the outline of rules and data which comprise the expertise, the knowledge engineer translates it into computer code, usually LISP. He then builds a [software] structure known as the inference engine, which can correlate the outline's general rules to more specific pieces of knowledge that will be added to the system later. Combined, the rules and data of the knowledge base, and the inference engine form the complete expert system.²³

An argument advanced for the custom approach is that it is cheaper than other options because it only requires time, not money. Yet a dedicated LISP machine costs \$50,000-\$100,000, and even a dedicated artificial intelligence (AI) personal computer can cost \$20,000. With the increased power of Intel's 80386 chip, amateurs can explore PC-based languages

such as LISP or Prolog, or another, more conventional language.²⁴

1. LISP (*LISt Processing*). John McCarthy invented the AI language of choice in the United States. This declarative language in which the computer, told what to do, does it, is the second-oldest high-level computer language after FORTRAN. It processes symbolic data (knowledge bases are symbolic data structures) represented as linked list structures, and it can handle nested subroutines. The *de facto* standard is CommonLISP.²⁵ Such an approach would characterize each and every reference book by topic, frequency, types of indexes, etc., much like ALA's *Booklist* guidelines. Sheehy's guide, for example, contains some of the declarative knowledge about reference books. This approach has been tried for a government documents expert system, described and evaluated below. A variety of PC implementations exist, but novices may wish to peruse the literature and experiment first with XLISP, a public-domain version.^{26,27,28}

2. Prolog (*Programming in Logic*). In contrast to LISP, Prolog is usually described as a procedural language (that is, you tell the computer how to do it and it does it), although its statements can be either declarative or procedural. "In its declarative form, it proves something is true by searching through a database of facts and rules."²⁹ As a symbolic language, it is useful too for solving problems that involve relationships between objects. Prolog is based on predicate calculus, especially Horn Clause axioms, which are used to structure the program and guide its execution. Invented in France in the 1960s, Prolog has been selected by the Japanese government for their Fifth Generation Computer Project. A variety of PC implementations exist, including Marseille and Edinburgh (or Mellish), two different syntaxes.³⁰ Novices may wish to peruse the literature and experiment first with PD Prolog, a public-domain version.^{31,32}

Supporters claim Prolog is more efficient than LISP in that the same task entails less coding. Others claim it offers increased program accuracy and better organization of modules, and handles relationships between symbols better than

LISP. The most compelling argument, however, is that a procedural language more closely resembles the way experts actually think. On the negative side, Prolog detractors claim it lacks control constructs, does not handle lists well, and may not be as readable as other languages.

In fact, other languages, including C or Hypercard, exist that could be used to create an inference engine and its surrounding structure.³³ James R. Parrott wrote REFSIM in PASCAL (described and evaluated below), and Karen Smith rewrote her POINTER system from LISP into BASIC (see below).

Semicustom Approaches. A second approach is a semicustom development route, beginning with a commercially available "generic" expert system shell which the institution adapts to its specific needs by building a base of knowledge around it. Few individuals outside university laboratories and AI-specific companies are able to create expert systems from scratch, and thus vendors offer development tools, variously known as shells, inference engines, framework, and structures, that allow users "to test the waters without investing hundreds of thousands of dollars in custom development."³⁴

Followers of this approach must consider the two primary ways of representing knowledge. Most shells follow production rules, i.e., if some condition exists, then some conclusion follows, based on Newell and Simon's early work in modeling human cognition.³⁵ This approach has obvious utility in answering print-based, fact-type questions, for example, "IF the person is living AND the person is American AND the person is male AND the user only wants vital statistics, THEN recommend *Who's Who in America*." A collection of such if-then rules appears capable of representing a substantial body of information, but the question may still be asked, Can knowledge be represented by the rules of formal logic?³⁶

1. Forward Chaining Shells. Forward chaining starts from the facts and works forward in the direction of the conclusions they imply. Waterman states, "If your goal is to infer one particular fact, forward chaining could waste both time and

money."³⁷ Samuel Waters recommends this approach without specifying why he thinks it is the way of the future.³⁸

2. Backward Chaining Shells. Backward chaining, an inference method, operates on a set of given rules. The process works from the hypothesis or conclusion through the rules back to the set of facts that would lead the user to one of these conclusions. Essentially, it embodies the elimination of conclusions for which there are no supporting facts. Here, one is left with the question of how the rules are generated.

3. Example- or Frame-Based. Yet another approach is a network of nodes connected by relations and organized into a hierarchy. Hence one might have a framework of concepts with attributes (often called "slots"). For example, each frame might contain a specific reference tool with slots filled by all its identified attributes. When a particular request matches this pattern, the result is a specific recommendation. Some of these shells actually induce the rules but do not allow the designer to control the order in which they fire.

Shells have both advantages and disadvantages. They are readily available from vendors.³⁹ Because little or no programming is required, this approach can drastically cut system development time, perhaps by one-third to one-half. Shells can save time by pre-packaging an expert system's inference engine, thus obviating the need for the knowledge engineer to create this structure from raw code. "The knowledge engineer need only add a specific knowledge base to the generic shell structure" to develop a complete expert system.⁴⁰ Because most of the effort in shells goes into the logic and interface design, ambitious reference librarians may prefer this approach. Initially,

shells seem like appropriate tools for nontechnical users, but most are beyond the technical proficiency of the average user. While several vendors claim to offer expert systems that don't require users to know arcane AI languages, such as Prolog or LISP, more than just a beginner's knowledge of computers and computer languages is required.⁴¹

Parenthetically, a number of UCLA

Graduate School of Library and Information Science students had no PC experience and yet created very good systems (see below for description and evaluation).

Off-the-Shelf Systems. These systems offer "a packaged route whereby the organization installs a prewritten application and makes minor adjustments to fit its exact needs. . . . [This results in] 'off-the-shelf' expert systems that are, quite literally, ready to run."⁴² To the best of my knowledge, none exist as yet; however, Karen Smith is selling her POINTER system, although it will need substantial modification to work in other libraries. At the 1988 ASIS Mid-Year meeting, Tome Associates demonstrated their TOME-SEARCHERS, derived from PLEXUS, which constructs online searches for the end user.

Should Expert Systems Model the Reference Process?

If system designers have a model of reference services, why not use it? A validated model would be best but even an idealized model could be used to structure the expert system's human-computer interaction. In the mid-1960s, Jesse Shera observed that "the machine problems *per se* are well on the way to solution; the great unsolved problems are those which are fundamental to the reference situation itself."⁴³ Since then, researchers have posited a variety of models of the reference interview, question negotiation, and the reference process. The problem is that we do not know which of the competing models is optimal. Such information is important because a viable expert system must contain a sophisticated model, especially one based on the user.

Extant systems appear to be responding to simple, fact-type questions, for instance, "Tell me more about (a person)" or "Do you have the SuDoc classification number?" This suggests that these systems contain an implicit model of the type of person asking questions. While many reference librarians keep *statistics* on the number of questions asked, few have studied their true nature or the characteristics of persons asking those questions. In

fact, the percentage of fact-type questions asked is largely unknown. Several researchers (Rees and Saracevic, 1963; Shera, 1964; Taylor, 1968; Crum, 1969; Bunge, 1970; Jahoda and Olson, 1972; Lynch, 1978; Rich, 1979; and Daniels, 1986) have studied the reference process and the user in particular, but few have consulted with librarians or otherwise tested their models.

In 1963, Allan Rees and Tefko Saracevic introduced one of the earliest models of the reference process.⁴⁴ They detail a ten-step process focusing on the searcher's analysis and the translation of search concepts into the appropriate indexing language. They omit the characteristics of the inquirer and the librarian.

Shera adopted several aspects of this model in his own model of the reference process.⁴⁵ He, too, believes that the process is self-evident, and must include the need, the inquiry, and searcher's analysis, but he adds the inquirer and the librarian's characteristics, plus the organizational structure, information store, response, and output language. Notably, he also includes an evaluation of the response based on pertinence to the information need and relevance to the inquiry.

In 1968, Robert Taylor identified five filters by interviewing special librarians.⁴⁶ Although each filter had already been covered in the previous models, his articulation of the user's need represents a significant contribution. Norman Crum recognized Taylor's contribution regarding users' needs or motivation, and posited his own four explicit elements in a user model: personal frames of reference, information use behavior, profession, and work group.⁴⁷ In addition, Crum includes time of use as an important motivating factor in the reference process. Charles Bunge's work in 1970 makes a minor advance, explicating some feedback channels.⁴⁸ Interested readers might consult two review articles on this topic for additional information.⁴⁹

A closely related question concerns what constitutes an effective user interface. Any system must adopt some method to structure the interaction.⁵⁰ Thus far, the flow of information in most

expert systems is controlled by the system; in some expert applications the system takes control immediately, or shortly after the user poses the initial question, the advisor takes over. Should systems allow for shared control? The user task in most systems is either binary or multiple-choice. The Socratic mode, usually a series of closed-ended questions requiring either a yes or no answer, has a long and popular history; but little research exists to support this method of interaction. More often than not, menus can conveniently collect closed-ended questions into a multiple-choice task. Menus have several advantages: (1) typing is not required; (2) correct spelling is not necessary; (3) they are relatively flexible; and (4) interfacing with other programs is relatively straightforward. The primary disadvantage is that a menu requires the user to read each possible selection. Direct manipulation interfaces such as a mouse are common in some microcomputers and windows are increasingly popular.

Finally, the issue of natural language interaction must be considered. Successful expert systems will process natural language; its obvious importance and utility have been acknowledged by researchers who wish to use open-ended questions in the interface to capture a maximum amount of information. Once again, however, there are few research findings to guide us.

What Has Already Been Done?

The following section describes the reported work in expert systems for reference service and evaluates work completed. That four or five systems already exist offers us substantial proof of the soundness of the concept. Some adopt the custom approach while others utilize a shell.

Which Systems Use the Custom Approach?
In 1983, the British Library Research and Development Department (BLRD) awarded A. Vickery and H. M. Brooks, at the University of London's Central Information Service, a grant to design a demonstration prototype expert referral system called PLEXUS.⁵¹ After abandoning microProlog because it lacked a compiler,

the designers wrote the software in PASCAL. It currently runs to some 10,000 lines of code.⁵² Operational in February 1986 on a SIRIUS I microcomputer, the prototype performs in the narrow domain of gardening and recommends resources, i.e., it refers the users to publications, organizations, databases, and experts.

The four functional modules of the system consist of a user model (GETUM), the user's problem (GETSTAT), a search strategy (SEARCH), and the outcome and user's evaluation (EVALUAT). The GETUM module characterizes the user in six different ways: familiarity with the system, job-related interest, length of experience, familiarity with existing resources, prior advice-seeking activities, and geographical location. The system then presents the user with an open-ended question, "Please tell me about your problem," and the user responds in natural language. This module uses frames to represent its knowledge of the user's stated problem. When it has enough information, the precompiled problem-solving SEARCH module takes over using production-rule sets and Boolean statements to query the database. It then returns with a proposed resolution to the stated problem. PLEXUS may not be portable, but it has adopted several good strategies to resolve the preceding theoretical questions.

With the Courseware Authoring System, "a much-extended subset of PASCAL" that runs on Digital Equipment under VAX, James Parrott wrote REFSIM for the IBM PC. Described as a reference tutor, REFSIM can be used by either client or librarian. Adopting a menu system, his system forward chains but appears capable of some backward chaining. The system asks the user the field of the person about whom information is sought and whether s/he is dead or alive and living in the U.S. or not; then it responds by suggesting sources. In the tutor mode, the system specifies a person and asks, "What should I look in?"; eventually it gives the student a list of sources.

In a newer and much larger implementation, Parrott rewrote REFSIM in Prolog. He envisions a bimodal system capable of

training and consultation. The latter, called the reference dialogue module, handles simple English, approaching natural language.⁵³ Apparently, it contains a module that helps the user make interlibrary loan requests.

Custom-tailored to SUNY-Buffalo's Lockwood Library's Documents and Microforms Department, Karen Smith's POINTER required 6,064 lines of code (about thirty-nine pages) in BASIC and runs on an IBM PC.⁵⁴ In 1984, she and Stuart C. Shaprio received a grant from the Council on Library Resources, and a SUNY Buffalo computer science graduate student wrote the original program in LISP.

"POINTER's new first screen welcomes the user by suggesting that it 'will help you find U.S. government documents by directing you to appropriate reference books.' "

POINTER's new first screen welcomes the user by suggesting that it "will help you find U.S. government documents by directing you to appropriate reference books." Next a screen appears containing information that stresses the importance of the SuDoc number for finding items in the collection. The system then asks whether the user has such a number; if not it will ask if more information is desired and, if so, will give a brief description of these numbers. If the user already has a SuDoc number, the system will direct him or her to the shelves or a nearby handout, and provide information concerning the location of the circulation department and the loan policy.

If the user does not have a SuDoc number and still wants help, the system offers a menu containing four choices: title, number, subject, or maps. Selecting title or numbered document generates menus of five more questions that require responses before a specific source is recommended. A subject request leads to fifteen

questions, and if the user is not satisfied, the system allows him to leave his request, name and telephone number for further assistance. Selecting maps refers the user to the map collection, one reference book and a brief SuDoc explanation.

POINTER offers several positive features. First, the system covers physical facilities and policies, besides fifty basic sources and their call numbers, and even directs the user in one instance to the structure of the source itself. Second, the system allows for uncertainty at one point. Third, the user can leave his request on the system if he is not satisfied.

Unfortunately, the systems disadvantages may outweigh the advantages. First, POINTER has a primitive user model; it assumes the user either has or does not have a SuDoc call number. If the user says he is unsure, it gives examples, but never asks if the user has determined that he has such a number. Second, POINTER uses forced, closed-ended questions. Third, the screen design is inconsistent and poorly laid out. Fourth, at least one screen moves too fast; the system should allow the user to hit a key to indicate he has finished reading each screen. The beginning screen should require the user to strike any key to continue rather than selecting yes or no and then pressing the return key. Most importantly, however, POINTER does not follow the established paradigm in the field;⁵⁵ consequently, one wonders how effective it really is and whether another implementation which does follow the paradigm would not be more efficient. In other words, is it just a superficial, "quick-and-dirty" system, or does it encompass a deeper understanding of how such a system should be designed for government information requests?

Do Any Systems Use the Shell Approach? Designers of the more recent expert systems are adopting shells. For instance, in April 1986, Howard White and Diana Woodward received Drexel University's Research Scholar Award to carry out their work. Using the Personal Consultant Series, EASY shell, to design their system, they constructed the "Expert System for General Library Reference." Conceptu-

ally they borrowed heavily from White's work on Joseph C. Meredith's RE-FSEARCH at Berkeley.⁵⁶ However, their early version of Texas Instruments' shell did not have a database interface, and consequently they adopted another shell, INSIGHT, to weigh recommendations according to the sureness of a source's information. At present, they use 144 common, frequently used sources. Their system uses memo fields to provide the user with call numbers and other relevant information. It may also have graphic capabilities, but this is uncertain, as the designers have not yet published the findings from their project.⁵⁷

At the National Agricultural Library, Samuel T. Waters has created Answerman to run on a 256K IBM PC using First-Class, a menu-driven, example-based shell.⁵⁸ Answerman's advantages include its ability to indicate specific page numbers of reference sources. Unfortunately, it assumes that the user knows which reference format (e.g., dictionary, encyclopedia) is appropriate. Finally, Answerman recommends only thirty-one different sources.

At UCLA we are using the Expert System Inference Engine (ESIE), a rule-based backward chaining shell written in PASCAL.⁵⁹ In early 1987 using ESIE, I wrote a modest Socratic prototype for selecting twenty-three dictionaries. Later, I revised it to use menus because it played a tiresome version of "twenty questions." I also created a biographical source module and most recently a module for bibliographies and indexes. At the 1988 ASIS Mid-Year Conference my students present a demonstration module called the Searchin' General.

During the 1987 fall quarter, students in my course on Information Resources and Services wrote production-rule modules for the reference formats we covered.⁶⁰ Edward Pai wrote a FORMAT-module for selecting more than a dozen formats or types of reference sources. Pai's module asks the user to indicate one of three levels of familiarity with the topic before presenting a menu with six additional options.⁶¹ Others, notably Deborah Henderson, Patti Martin, Lauren Mayer, and

Pamela Monaster, wrote linking modules for specific formats such as biographical sources; their "Searchin' General" module recommends about twenty-five titles.⁶² For the future, we have contemplated linking these modules seamlessly to a master module and performing field tests of ESIE's effectiveness.

ESIE is valuable as a pedagogical exercise. It teaches students the difference between facts and rules in a reference situation. They learn the characteristics of select sources, and by writing explicit rules they progress quickly from novices to advanced beginners. Although the limitations of this shell frustrate the best students, it does show them the potential of an expert system in this field.

Several significant efforts are as yet unreported in the literature. Lloyd A. Davidson is working on a menu-based dBASE III expert system for automated reference service at Northwestern University's Seeley G. Mudd Library for Science and Engineering. Brian Nielsen and Gilbert Krulee at Northwestern University won a 1987 Council on Library Resources grant to develop a natural language support system for reference librarians. Alex Vrenios, a doctoral student at the University of Texas, is developing a Prolog program on the Apple IIe to interpret natural language queries on business reference. Goucher College has developed a biographical expert system, Joseph Cavanaugh has worked on PISCES, and William E. McGrath has been teaching science and technology reference sources using First-Class.

The following summarizes the state of affairs concerning existing expert systems: unvalidated and/or primitive user models; potentially spurious assumptions that the user pool is homogeneous; modest natural language capability; small fact bases that make these systems little more than *idiots savants*; and simple if-then production rules. By comparing these systems, however, the knowledge base in reference service could be substantiated.

What System Validation Has Been Undertaken?

According to the published literature,

no system validation has as yet been attempted. All the previously discussed systems appear to be research prototypes, although the engineers of PLEXUS and POINTER appear to be planning some system testing and evaluation. In testing any of these systems, researchers could query regarding user satisfaction or create test questions. Does an expert system perform as well as a human? Existence proof or sufficiency examinations or a kind of Turing test could be useful. Can anyone tell which answer is human as opposed to machine generated?

How Shall We Evaluate Future Efforts?

We need something deeper than a mere checklist of subjective or normative guidelines. Lacking these, however, the Rand Corporation has offered some criteria which may serve some useful duty until the others appear.⁶³ Engineers could base their design specifications upon this list as well.

CONCLUSION

In summary, seven critical questions must be answered before expert systems can be adopted for use in libraries. First, what is the proper scope of an expert system for answering reference questions? Thus far, we know substantially more about the declarative knowledge of reference (e.g., the information about the titles in Sheehy's *Guide to Reference Books*) than about the procedural knowledge. Furthermore, the ethical questions have not been addressed, and yet existing systems are operationally using a core of printed reference sources. They have not included CD-ROM or online databases to any large degree.

Can experts articulate their heuristics? This is the second critical question. I believe they can, but researchers have not

systematically tried to identify the heuristics involved in general reference work. Third, what is the best approach to implementing an expert system? If a procedural language reflects how experts actually think, then Prolog seems the most promising, assuming one wants to adopt a programming language. Alternatively, if one assumes that reference work is done by matching a request to the characteristics of known sources, then a declarative language such as LISP makes more sense. If saving time is a major consideration, then there are numerous shells; at the moment, First-Class has the most adherents.

The fourth question is whether expert systems should model the reference process. Rather than answer this question directly, expert system designers have implemented systems that do appear to be modeling the process. Fifth, what has been done already? A handful of prototype systems exist. The Council on Library Resources has been most instrumental in advancing the work through funding. Sixth, what system validation has been undertaken? Unfortunately, nothing formal has been presented in the literature. Rather than simply create an expert system, we need to determine if it is any better than the half-right reference service we already have. Seventh, how shall we evaluate future efforts? At best, we have only ad hoc evaluations and must develop evaluative criteria. Something similar to ALA's *Booklist* guidelines would help librarians evaluate potential systems for their library.

Finally, I believe that it is imperative that a variety of groups, including library directors and reference librarians, become involved with this new technology in order that our libraries retain their competitive edge and to ensure that expert systems are truly expert.

REFERENCES AND NOTES

1. Samuel Rothstein, "The Development of the Concept of Reference Service in American Libraries, 1850-1900," *Library Quarterly* 23:1-15 (January 1953) and "Across the Reference Desk: A Hundred Years of Reference Encounters," in *The Reference Interview; Proceedings of the CACUL Symposium* (Ottawa: Canadian Library Association, 1979), p.27-53. The concepts of relevance and pertinence are important in their own right. See Don R. Swanson, "Historical Notes: Information Retrieval and the Future of an Illusion," *JASIS* 39:92-98 (March 1988).

2. American Library Association, Reference and Adult Services Division, Standards Committee, "A Commitment to Information Services: Developmental Guidelines," *RQ* 18:275-78 (Spring 1979).
3. The best review of this literature is Kenneth D. Crews, "The Accuracy of Reference Service: Variables for Research and Implementation," *Library and Information Science Research* 10:331-55 (July-September 1988). Nice M. DeFigueiredo, "A Conceptual Methodology for Error Prevention in Reference Work" (Ph.D. Diss., Florida State University, 1975) and Ian Douglas, "Reducing Failures in Reference Service," *RQ* 28:94-101 (Fall 1988) go a long way toward suggesting how to avoid errors.
4. Patrick Suppes, "Decision-Theory," *Encyclopedia of Philosophy*, v.2, p.310-14.
5. Expert systems in reference challenge us to rethink our definition of such services. Some writers define reference service narrowly to mean "the interpersonal communication process." See James Rettig, "A Theoretical Model and Definition of the Reference Process," *RQ* 18:19-29 (Fall 1978).
6. Donald Waterman, *A Guide to Expert Systems*, Foreword by Frederick Hayes-Roth (Reading, Mass.: Addison-Wesley, 1986), p.128-29.
7. Based on heuristics presented in chapter 13 of Waterman's book, Professor R. Clay Sprows in UCLA's Anderson Graduate School of Management has created ESGUIDE, an expert system to determine whether an expert system is justified in a given domain.
8. These are some of the central philosophical issues that should be debated. According to the *Random House Dictionary of the English Language* (2d ed.), an "assistant" provides aid and support; "aide" and "adjutant" are synonyms. An "associate" (literally, "to connect or bring into relationship") can mean having equal or nearly equal responsibility. An "adviser," on the other hand, is informed (literally, one who gives advice). Perhaps the existing, first-generation expert systems fall into the first category. For the future, I envision a truly expert reference adviser. If, however, reference service is defined as face-to-face, then such a system should only support the librarian's decision making. On this point, Hubert and Stuart Dreyfus argue in their *Mind Over Machine; The Power of Human Intuition and Expertise in the Era of the Computer* (New York: Free Press, 1986) that it is philosophically and practically impossible to solve problems the same way people do. Although computers have a role, ultimately they will only aid reference librarians. For a succinct statement of their views, see "Making a Mind Versus Modeling the Brain: Artificial Intelligence Back at a Branchpoint," *Daedalus* 117:15-44 (Winter 1988).
9. The issue is not clear-cut. If the research is undertaken in a university, then that institution may claim ownership. On the other hand, work supported by government contracts may not be copyrightable at all. Should the experts, the knowledge engineers, or the vendors claim copyright? The creator of Answerman, Samuel T. Waters, who works for the federal government and thus cannot make a copyright claim, advocates making the software "freely available to others so that the knowledge bases are readily accessible to all." Letter to author, 15 April 1988. Those interested should read Diana Woodward's "Proprietary Expert Systems: A Threat to Intellectual Freedom," ASIS Mid-Year Meeting, Ann Arbor, Mich. May 18, 1988.
10. The latter means substantially more work. For instance, Jim Parrott has "come to the conclusion that deep knowledge representation will be necessary in order to facilitate the recognition of deep cognitive errors by the student." Letter to author, 14 April 1988.
11. The rules related to interlibrary loan have been articulated in Jaime Pontigo, Guillermo Rodriguez, and Sergio O. Gama's "Generation of Decision Rules for An Expert System Used in Document Supply," ASIS Mid-Year Meeting, Ann Arbor, 17 May 1988. Studying under Linda C. Smith at the University of Illinois, Gail Thornburg designed an expert system to allow optimum choice of a database from among nineteen different databases. See her "LOOK: Implementation of an Expert System in Information Retrieval" (Ph.D. Diss., University of Illinois, August 1987), which operates in the Aurora environment. Rule-based, LOOK is capable of learning as it advises a human expert. To better understand the concept of a reference source, see Marcia Bates, "What is a Reference Book? A Theoretical and Empirical Analysis," *RQ* 26:37-57 (Fall 1986).
12. Mary Biggs and Victor Biggs, "Reference Collection Development in Academic Libraries: Report of a Survey," *RQ* 27:67-69 (Fall 1987).
13. Wallace J. Bonk, *Composite List of Titles Taught in Basic Reference by 25 of the Accredited Library Schools*. Ann Arbor, Mich.: University of Michigan Department of Library Science, 1961. 66p.
14. Idem, *Use of Basic Reference Sources in Libraries*, Cooperative Research Projects, No. 1584 (Ann Arbor, Mich.: University of Michigan Department of Library Science, 1963). Researchers adopting this approach should modify it in light of Marcia Bates' work cited above.
15. John C. Larsen, "Information Sources Currently Studied in General Reference Courses," *RQ* 18:341-48 (Summer 1979).
16. "Books in Order of Frequency of Use," Enoch Pratt Free Public Library, Fall 1968; the top three

- sources were *World Almanac and Information Please*, *World Book Encyclopedia*, and *American College Dictionary*.
17. R. Gers and L. J. Seward, "Improving Reference Performance: Results of a Statewide Survey" *Library Journal* 110:32-35 (November 1, 1985).
 18. According to a report in the August 17, 1987, issue of *Infoworld*, Arthur Andersen and Co. estimates that "only 20 percent of the real information in [their] paperwork is in the words, 80 percent is the non-explicit invoking of known relationships." Similarly, a paradigm is at work in the reference process. For a detailed discussion and critique of that process as well as the historical development of the teaching of reference, see John V. Richardson, Jr., "Teaching General Reference Work: The Essential Paradigm, 1890-1987" (in progress).
 19. Richard E. Nisbett and Timothy D. Wilson, "Telling More Than We Know: Verbal Reports on Mental Processes," *Psychological Review* 84:231-59 (May 1977). The field of complex information processing may offer insights. See, for example, K. Anders Ericsson and Herbert A. Simon, *Protocol Analysis: Verbal Reports as Data* (Cambridge, Mass.: MIT Press, 1984).
 20. A promising approach to understanding expert behavior might be modeled upon David Hawkins, "An Analysis of Expert Thinking," *International Journal of Man-Machine Studies* 18:1-47 (1983). What are the differences among novices, advanced beginners, competent reference librarians, and expert librarians? Do the experts simply remember more details? Is this a function of experience? Two useful background works are John R. Anderson, *The Architecture of Cognition* (Cambridge, Mass.: Harvard Univ. Pr., 1983) and Howard Gardner, *The Mind's New Science: A History of the Cognitive Revolution* (New York: Basic Books, 1985).
 21. George A. Kelly, *The Psychology of Personal Constructs* (New York: Norton, 1955).
 22. Earl B. Hunt, Janet Mavin, and Phillip J. Stone, *Experiments in Induction* (New York: Academic Press, 1966); Ryszard S. Michalski and R. L. Chilausky, "Knowledge Acquisition by Encoding Expert Rules Versus Computer Induction from Examples," *International Journal of Man-Machine Studies* 12:63-87 (1980); P. Langley, "Data-Driven Discovery of Physical Laws," *Cognitive Science* 5:31-54 (1981); J. R. Quinlan, "Discovering Rules by Induction from Large Collection of Examples" in *Expert Systems in the Micro-electronic Age*, ed. Donald Michie (Edinburgh: Edinburgh University, 1979), p.169-201, and Quinlan, "Learning Efficient Classification Procedures and Their Application to Chess-End Games," in *Machine Learning: An Artificial Intelligence Approach*, ed. Ryszard Michalski, Jaime Carbonell, and Tom Mitchell (Palo Alto, Calif.: Morgan Kaufmann Publishers, 1983), p.463-82.
 23. Harvey P. Newquist III, "Expert Systems: The Promise of a Smart Machine," *Computerworld* 20:43-46, 50-58, 60 (January 13, 1986).
 24. For a discussion of the pros and cons of several languages, see Richard Wexelblat, *History of Programming Languages* (New York: Academic Press, 1981). Mac users should read Allen Munro, "Choosing a Programming Language," *MacWorld* 4:142-49 (October 1987), which covers forty-eight implementations including LISP and Prolog.
 25. Mark Bridger and John Frampton, "Creating a Standard LISP," *PC Tech Journal* 3:98-117 (December 1985).
 26. IBM's CommonLISP retails for \$10,000, Golden CommonLISP for \$495, and Mac implementations from \$399 to \$600 for Allegro to \$995 for Expert CommonLISP or ExperLISP. PC Scheme (Texas Instruments) retails for \$95, WaltzLISP for \$169, muLISP-86 for \$250, IQLISP for \$270, and Star Shappliance LISP for \$495.
 27. For a brief article on its history, advantages, and structure, see Steven Cherry, "The World According to LISP," *Micro: The 6502/6809 Journal* 57:65-69 (February 1983). The first part of Patrick H. Winston, *Artificial Intelligence* (Reading, Mass.: Addison-Wesley, 1984) contains a nonmathematical introduction to AI followed by a competent discussion of LISP. Steven L. Tanimoto, *The Elements of Artificial Intelligence: An Introduction Using LISP* (Rockville, Md.: Computer Science Press, 1987), contains exercises. Readers may prefer Robert Wilensky's *Common LISPcraft* (New York: Norton, 1986).
 28. See David Betz, "An XLISP Tutorial: This Public-Domain Language Lets You Experiment with Artificial Intelligence," *BYTE* 10:221-36 (March 1985).
 29. Boyd Sutherland, *Selection of a Reference Service Expert System Shell*, supervised by John Richardson as a CLR Supporting Study under the Long Range Strategic Planning for Libraries and Information Resources in the Research University; Robert M. Hayes, Principal Investigator (Los Angeles: UCLA Graduate School of Library and Information Science, Summer 1986), p.8. A good discussion of the difference between procedural versus declarative knowledge can be found in Ralph Alberico, "More on Knowledge Representation," *Small Computers in Libraries* 7:10-16 (December 1987).

30. IBM implementations include Arity/Prolog for \$95-\$795, Turbo Prolog for \$99.95, M Prolog for \$195-\$495, ALS Prolog for \$199-\$499, and Prolog-2 for \$895.
31. For a brief article, see William F. Clocksin, "A Prolog Primer: An Introduction and Tutorial to the Popular Artificial Intelligence Language," *BYTE* 12:147-58 (August 1987). Coauthored with C. S. Mellish, the third edition of his book-length treatment, *Programming in Prolog*, was published by Springer-Verlag in 1987. Beginners interested in the Edinburgh syntax may consult Ivan Bratko, *Prolog Programming for Artificial Intelligence*, (Reading, Mass.: Addison-Wesley, 1986). Feliks Kluzniak and Stanislaw Scapakowicz, *Prolog for Programmers* (London: Academic Press, 1985) comes highly recommended for its practical information for intermediate programmers. Serious programmers may want Leon Sterling and Ehud Shapiro, *The Art of Prolog: Advanced Programming Techniques*, (Cambridge, Mass.: MIT Press, 1986).
32. Robert Morein, "PD PROLOG: A Public-Domain Version of the Fifth-Generation Language," *BYTE* 11:155-65 (October 1986).
33. For a discussion in the context of our own field, see Thomas R. Kochtanek, "Procedural Logic versus Object-Oriented Logic in Library Automation Instruction," *Journal of Education for Library and Information Science* 28:55-57 (Summer 1987). Because Hypercard is the popular approach on the Macintosh it should be explored. Henry Newquest argues that we should debate LISP versus C in "Will the Real Artificial Language Please Stand Up?" *Computer Language* 4:58-59 (July 1987), and Thomas Hill compares the two in his article, "Expert Systems Shells May be the Key to Artificial Intelligence," *PC Week* 4:47-54 (July 28, 1987).
34. Newquist, "Expert Systems," 45-46. Ralph Alberico's "Software for Expert Systems: Languages versus Shells," *Small Computers in Libraries* 8:4-12 (July/August 1988), neatly summarizes the language-versus-shell debate using Prolog and VP-Expert as examples.
35. Allen Newell and Herbert Simon, *Human Problem Solving* (Englewood Cliffs, N.J.: Prentice-Hall, 1972). For a profile of Simon, see Constance Holden's "The Rational Optimist," *Psychology Today* (October 1986), 20:55-60.
36. James Parrott writes that "so far, I have discovered that one of REFSIM's superficial rules requires over 10 of these deep rules, in order to be properly deduced!" Letter to author, 14 April 1988.
37. Waterman, *Guide to Expert Systems*, p.67.
38. Samuel T. Waters, "Answerman: The Expert Information Specialist, an Expert System for Retrieval of Information from Library Reference Books," *Information Technology and Libraries* 5:204-12 (September 1986).
39. At the 1988 ASIS Mid-Year Conference in Ann Arbor, Mich., N. Shahla Yaghmai briefly reviewed "Expert System Development Tools." Thomas Hill offers an extensive buyer's guide to twenty-six shells in his article, "Expert Systems Shells May Be the Key to Artificial Intelligence," *PC Week* 4:47-54 (July 28, 1987), and Boyd Sutherland's *Selection of a Reference Service Expert System Shell* discusses AI languages including LISP and PROLOG and twelve shells, citing reviews.
40. Newquist, p.46.
41. Jordan Gold, "Shellware: Do-It Yourself Expert Systems," *Computer Decisions* 18:76-81 (January 14, 1986).
42. Newquist, p.45-46.
43. Jesse Shera, "Automation and the Reference Librarian," *RQ* 3:5 (July 1964).
44. Allan Rees and Tefko Saracevic, "Conceptual Analysis of Questions in Information Retrieval Systems," *Annual Meeting of the American Documentation Institute* 1963, Part II, p.175-77.
45. Shera, "Automation," p.3-7.
46. Robert S. Taylor, "Question-Negotiation and Information Seeking in Libraries," *College and Research Libraries* 29:178-94 (May 1968). Lehigh University, Center for the Information Sciences, *Studies in the Man-System Interface in Libraries: Question-Negotiation and Information-Seeking in Libraries*, Report No. 3, Washington, D.C.: Air Force Office of Aerospace Research Grant AF-AFOSE-724-66, July 1967.
47. Norman J. Crum, "The Librarian-Customer Relationship: Dynamics of Filling Requests for Information," *Special Libraries* 60:269-77 (May-June 1969).
48. Charles A. Bunge, "Reference Service in the Information Network," paper presented to the Interlibrary Communications and Information Networks Conference, 1970, and published in *Interlibrary Communication and Information Networks*, ed. Joseph Becker (Chicago: American Library Assn., 1971), p.109-10.
49. See Gerald Jahoda and Paul E. Olson, "Analyzing the Reference Process," *RQ* 12:148-156 (Winter 1972), and James Parrott, "Implementation of Reference Models in Expert Systems," in *Expert Systems in Libraries*, ed. Rao Aluri and Don Riggs (Norwood, N.J.: Ablex Publishing Corporation, in press).

50. Interested readers should consult: Wilbert O. Galitz, *Handbook of Screen Format Design* (Wellesley, Mass.: Q.E.D. Information Sciences, 1981); Alina Vickery, "An Intelligent Interface for Online Interaction," *Journal of Information Science* 9:7-18 (1984); Cynthia A. Kehoe, "Interfaces and Expert Systems for Online Retrieval," *Online Review* 9:489-505 (1985); Ben Shneiderman, "Designing Menu Selection Systems," *JASIS* 37:57-70 (1986); and John M. Carroll and Jean McKendree, "Interface Design Issues for Advice-Giving Expert Systems," *Communications of the ACM* 30:14-31 (January 1987).
51. A. Vickery and H. M. Brooks, "Expert System for Referral: Project Proposal to BL R & D" (London: University of London, 1983); see also A. Vickery, H. M. Brooks, B. Robinson, and B. C. Vickery, "Expert System for Referral: Phase I, Final Report," (London: British Library Research and Development Department, 1986).
52. H. M. Brooks, "Expert Systems in Reference Work," in *Expert Systems in Libraries: Proceedings of a Conference of the Library Association Information Technology Group and the Library and Information Research Group, November 1985*, ed. Forbes Gibb (London: Taylor Graham, 1986); A. Vickery and H. M. Brooks, "Plexus—The Expert System for Referral," *Information Processing & Management* 23:99-117 (1987).
53. James R. Parrott, "Expert Systems for Reference Work," *Microcomputers for Information Management* 3:155-71 (September 1986); "REFSIM: A Bimodal Knowledge-Based Reference Training and Consultation System," *Reference Services Review* 16:61-68 (1988); Simulation of the Reference Process, Part II; REFSIM, an Implementation with Expert System and ICAI Modes," *Reference Librarian* 21:153-76 (Spring 1988).
54. Karen F. Smith and Stuart C. Shapiro, "Final Report on the Development of a Computer Assisted Government Documents Reference Capability: First Phase" (Buffalo, N.Y.: State University of New York at Buffalo, 1984). For a discussion of POINTER, see Karen Smith's "Robot at the Reference Desk," *College and Research Libraries* 47:486-90 (September 1986). My description and evaluation are based on a first-hand examination of the system and "POINTER: User's Guide and Reference Manual," February 1987.
55. John V. Richardson Jr., "Paradigmatic Shifts in the Teaching of Government Publications: 1895-1985," *Journal of Education for Library and Information Science* 26:249-66 (Spring 1986); reprinted in *Encyclopedia of Library and Information Science*, v.44 (in press).
56. Joseph C. Meredith, "Machine-Assisted Approach to General Reference Materials," *Journal of the American Society for Information Science* 22:176-86 (May/June 1971).
57. Diana Woodward, interview with author, Ann Arbor, Mich. 16 May 1988. See also Joseph C. Meredith, *Reference Search System: (REFSEARCH) Users' Manual, Final Report*, Project No. 7-1085, Grant No. OEG 1-7-071085-4286 (Washington, D.C.: DHEW Office of Education Bureau of Research, April 1971).
58. Waters, "Answerman," p.204-12.
59. ESIE's first screen claims copyright by Lightwave Consultants, Tampa, Florida.
60. John V. Richardson Jr., "Expert System Assignment" (Los Angeles, Calif: UCLA GSLIS, Fall Quarter 1987).
61. Edward Pai loosely modeled his system on Gerald Jahoda and Judith S. Braunagel, *The Librarian and Reference Queries—A Systematic Approach* (New York: Academic Press, 1978). Incidentally, his ASIS demonstration module is a much more sophisticated version that may have reached ESIE's Version 2.0 (1986) limits.
62. Deborah Henderson, Patti Martin, Lauren Mayer, and Pamela Monaster, "Rules and Tools in Library Schools," *Journal of Education for Library and Information Science* 30:226-27 (Winter 1989).
63. Jeffrey Rothenberg and others, *Evaluating Expert System Tools: A Framework and Methodology* (Santa Monica, Calif.: Rand Corp., July 1987), R-3542-DARPA.

**APPENDIX A: PROS AND CONS OF AN EXPERT SYSTEM
(OR INTERACTIVE DIALOGUES IN GENERAL)
WITHIN THE CONTEXT OF VESTED INTERESTS**

End users

Advantages

1. Service is always readily available (Brooks, 1985)
2. Independent, self-help situation (Brooks, 1985; Waters, 1986)
3. Can leave messages for librarians (Smith & Hutton, 1984)

Disadvantages

1. "People may prefer people"—warmth and touch
2. One user per time; need several machines
3. Uninspired; it follows rules; familiar users can predict responses

Reference librarians

Advantages

1. Frees one from routine questions (Brooks, 1985; Waters, 1986)
2. Librarians cannot remember the best sources for answering questions at typical reference desk (Waters, 1986)
3. Relief during high-demand periods (Parrott, 1986)
4. Results in higher-level questions, hence greater job satisfaction
5. Lower risk of job burnout (Parrott, 1986; Smith, 1986)
6. Relief from overwork, boredom, and frustration (Smith, 1986)

Disadvantages

1. Potential threat to job security—elimination of position
2. Less pay for professional services
3. Might forget basic reference work

Reference department paraprofessionals

Advantages

1. Supports their work
2. Teaching role in their training

Disadvantages

1. Staff most likely to be replaced by expert system

Reference department heads

Advantages

1. High-quality, expert "librarians"
2. Consistent answers to questions
3. Staff shortages covered (Parrott, 1986)
4. Scarce resources (Smith, 1986)
5. Cost savings, if staff is replaced
6. Stems the "brain drain" due to turnover (Waters, 1986)
7. Minimal level of service always available
8. Relatively affordable

Disadvantages

1. Potential threat to job security
2. Protect the intellectual property of reference staff
3. Staff time devoted to development and maintenance

Library directors

Advantages

1. Potential cost savings
2. Consistent with policy/mission statements
3. Utilize existing computer equipment (additional/new use)

Disadvantages

1. Development time of staff if custom approach is adopted
2. Increased demand for computer equipment
3. Additional costs of LAN if expert system is placed on file server

Library school faculty

Advantages

1. Frees them from routine instruction (use valuable class time for important material)

Disadvantages

1. Work on curriculum implications
2. Changes the current content of the course
3. Faculty have to learn new material

Library school students

Advantages

1. Tutor
2. Tireless—endless repetition, if necessary
3. Explicit learning of tools and rules

Disadvantages

1. "People prefer people"

Reference book authors and publishers

Advantages

1. Identify need for new tools that do not exist
2. Potential profit from commercial introduction of such a system

Source: Unless otherwise noted, these advantages and disadvantages are original ideas of the author.

By the end of this century, if the present growth rates continues, the Library of Congress will have 23,000,000 volumes and Harvard will have more than 12,000,000.

—William H. Carlson, January 1952

Most librarians approach the library by way of the book (form) while the user, often unconsciously, approaches the library by way of information (content).

—Robert S. Taylor, July 1957