The Digital Reference Fallacy

R. David Lankes

SUMMARY. This article discusses the fallacy that real-time and asynchronous digital reference software are fundamentally different. Instead the author argues that the only real difference is lag time, and that this difference does not support the separation of digital reference functions. An attempt is made to create a unified model for digital reference and digital reference functions. Lastly, the author presents some practical considerations for libraries seeking to purchase digital reference software. *[Article copies available for a fee from The Haworth Document Delivery Service: 1-800-HAWORTH. E-mail address: <docdelivery@haworthpress.com> Website: <http://www.HaworthPress.com> © 2002/2003 by The Haworth Press, Inc. All rights reserved.]*

KEYWORDS. Digital reference, digital reference software

INTRODUCTION

Digital reference is a rapidly evolving domain, with new issues being identified on nearly a monthly basis. As with all active and growing fields, certain issues tend to polarize discussion, or create camps. These camps are often shifting, and created for expediency, to seek partnerships, and/or to create a dialectic for discussion and exploration (much

---

R. David Lankes is Director, Information Institute of Syracuse University, Syracuse, NY 13244.

[Haworth co-indexing entry note]: "The Digital Reference Fallacy." Lankes, R. David. Co-published simultaneously in *The Reference Librarian* (The Haworth Information Press, an imprint of The Haworth Press, Inc.) No. 79/80, 2002/2003, pp. 35-44; and: *Digital Reference Services* (ed: Bill Katz) The Haworth Information Press, an imprint of The Haworth Press, Inc., 2002/2003, pp. 35-44. Single or multiple copies of this article are available for a fee from The Haworth Document Delivery Service [1-800-HAWORTH, 9:00 a.m. - 5:00 p.m. (EST). E-mail address: docdelivery@haworthpress.com].

http://www.haworthpress.com/store/product.asp?sku=J120

10.1300/J120v38n79_03

like playing the devil's advocate in a conversation). However, these dialectic groups can also inhibit cross-group discussion, and are often proved fallacious as more information is gained within a field.

The author argues such a polarization is occurring in digital reference, and further that this polarization is based on a fallacy; namely that synchronous and asynchronous digital reference are in some way two distinct approaches to digital reference. This belief in the fallacy of synchronous versus asynchronous systems comes out of the Digital Reference Research Symposium (http://quartz.syr.edu/symposium) held at Harvard in 2002 to develop a research agenda in digital reference.

This article is also meant to further refine thoughts and data put forth in Lankes and Shostack (2002) that argue for the continued relevance of asynchronous systems. In that article the authors state, "in the authors' opinion, real-time systems and asynchronous systems will need to coexist (or rather digital reference systems will need to support both forms of interactions)" (Lankes and Shostack, p. 354). In light of work in the research symposium, and subsequent thinking, the author believes that there is only one significant difference between synchronous and asynchronous systems, lag time, and that this difference should not lead to completely incompatible systems, planning or even staffing. The author will highlight practical consequences of such a new conceptualization of digital reference system.

THE NEED TO RECONCILE SYNCHRONOUS AND ASYNCHRONOUS SYSTEMS

The continued division between real-time and asynchronous systems has very real consequences for the library profession. It often creates a division between proponents of existing asynchronous (e.g., e-mail based) systems and staff that wish to go to real-time systems. This divide often masks the most important question, "what do our users want?" This lack of user involvement was cited in Gross et al.'s (2001) analysis of the literature. Unfortunately, without this user information digital reference committees and libraries are often left basing decisions on stereotypes (e.g., "the younger audience we want to reach always use chat and instant messaging, therefore we must use the same technologies"–ignoring this same population is a heavy user of e-mail), peer pressure (e.g., "every other library is using real-time systems, therefore we must use real-time"), or vendor marketing materials (e.g., "the only digital reference systems available on the market are real-time systems"–of course web and e-mail systems do not market themselves as digital reference products, though they work well for digital reference applications).

What many libraries have discovered is that both real-time and asynchronous features are needed. Longtime asynchronous systems such as AskERIC, even before they adopted real-time software, would use phone calls to get real-time data from patrons. It was simply faster and more efficient with certain types of users and questions. Many real-time services will follow-up with e-mail for additional information, or use the real-time software only to provide a reference interview.

The point is, that by artificially dividing these two communication modes into system requirements, we have no systems that serve the complete range of digital reference needs. Even so-called "real-time features" such as co-browsing have a place in asynchronous interactions.

While at first this may seem counter-intuitive, consider the problem of proprietary database access. This has been a persistent problem in current real-time systems, and has been overcome by the use of proxy servers. Librarians can "push" information from proprietary databases to a patron's browser, keeping all licensing intact (because it is done by a librarian in the normal course of their jobs). The asynchronous solution has been to violate license and send a copy of a licensed resource to a patron.[1] What is needed is a single solution that can queue on-demand licensed resources requested by a librarian into a "safe-haven" where a patron can gain access to the resource within a licensing agreement, for a pre-determined purpose, and/or a pre-determined time. That way, a patron can access that resource either with a librarian, or at a later time (in an asynchronous setting).

Other features that are currently handled differently, but are in fact needed by both approaches include: queuing of patrons, and routing of patrons to a qualified expert or librarian. Table 1 lists some key digital reference functions, how they are currently implemented in both real-time systems and asynchronous systems, and an example of how they might work in a unified approach.

This list is based on the assumption that real-time and asynchronous approaches can be unified–that they are, in fact, the same thing. This assumption comes from a detailed examination of the General Digital Reference Model (Pommerantz et al.).

TABLE 1

Feature and Need	Real-time Approach	Asynchronous Solution	Advantage of Unified Approach	Examples
Co-Browsing: To share licensed resources to identified patrons remotely	Proxy Servers	Item Forwarding	Maintain license agreements and serve patrons legally	Much like electronic reserves, a librarian places a licensed article or resource in an electronic "holding tank" with an expiring password. This holding tank can either be accessed by escorting the patron into the holding tank or by sending the password to the user through e-mail (or the web).
Queuing: To queue patrons in line for an available resource based on some priority measure	Queues or "Waiting Rooms"	Inbox Management	Be able to shift users to their mode of choice, or eliminate patron wait-time for real-time librarians when a question can be asked asynchronously	A patron clicks on an "Ask a Question" button. They are alerted to an estimated wait time, and given the option to leave a question, and possibly a time to interact in real-time.
Screen Sharing: To manipulate user resources at the desktop level	Applet Installation	Providing detailed set of step-by-step instructions	Allow a librarian to manage a patron's computer at the desktop level	The user downloads an applet allowing a limited time (and limited access) to a patron's machine. This access can either be initiated immediately, or at some later time.
Expert Routing: To send a question to the right expert based on some criteria	Creating different queues	Forwarding, or creating differentiated list of subjects	Experts create a single profile and establish a single mechanism for receiving questions	A librarian enters a system profile that identifies what type of questions s/he is willing to answer, and when. If a user's question matches at a given time, chat is initiated, if not, a question is put in a queue to be picked up later.
User Evaporation: To identify when a session was ended by patron choice rather than wondering if there was a technical problem, or if the user is simply taking a long time to respond	Having the patron affirmatively close a session (normally by hitting a button)	Receiving a thank you message	Creating a unified session management system for improved statistics and resource management	Using the web, cookies can be dropped for each digital reference question (versus per user).

THE GENERAL DIGITAL REFERENCE MODEL

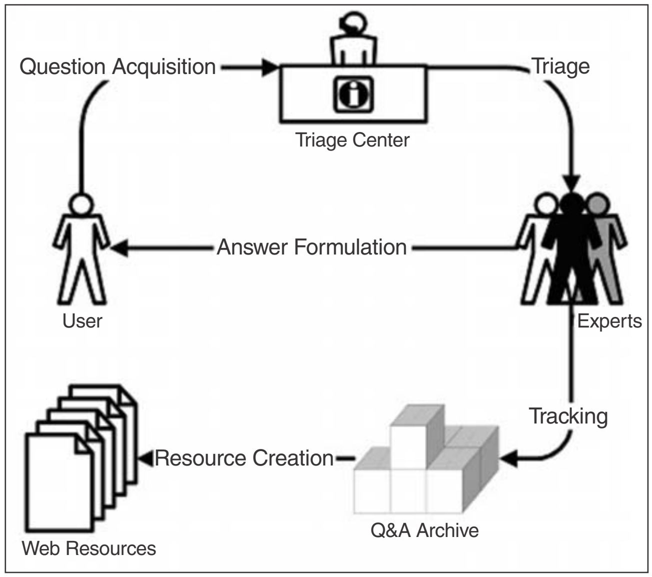
The digital reference model pictured in Figure 1 is a general process model developed through an empirical study of high-capacity digital reference services, primarily in the math/science area (Lankes, 1998).

The model consists of 5 steps:

1. *Question Acquisition* is a means of taking a user's questions from e-mail, web forms, chat, or embedded applications. This area of the model concerns best practice in "online reference interviews" and user interface issues.
2. *Triage* is the assignment of a question to a process or topic expert. This step may be automated or conducted via human decision support. Triage also includes the filtering of repeat questions or out of scope questions.
3. *Experts Answer Formulation* details factors for creating "good" answers such as age and cultural appropriateness. Answers are also sent to the user at this point.
4. *Tracking* is the quantitative and qualitative monitoring of repeat questions for trends. Tracking allows the creation of "hot topics," and may indicate where gaps exist in the collection(s).
5. *Resource Creation* concerns the use of tracking data to build or expand collections and better meet users' information needs within and outside of the digital reference process.

This model was developed originally to capture the workings of asynchronous digital reference services. However, it seems to capture the workings of synchronous systems as well. In systems such as 24/7 Reference, online librarians choose users from a list of waiting patrons in a queue. This is exactly the same process (at this level of abstraction) as an expert (the online librarian) choosing a question in a list of waiting questions. Both are examples of triage. Analogies can be made at every step of the processes. In real-time systems answer formulation happens in real-time interactions with patrons through chat and co-browsing. In web and e-mail systems this takes place either in the form of a constructed response normally e-mailed to the user, or through a serial exchange of messages (also, normally over e-mail). Tracking is nearly identical in all digital reference systems and involves looking at the transcripts of exchanges (almost always a semi-structured text file). Resource creation is the last step, and nearly as unexplored in any mode of interaction to date.

FIGURE 1. General Digital Reference Model



INTRODUCING THE CONCEPT OF LAG TIME INTO THE GENERAL MODEL

The author argues, once again, that there are more than simple analogies between real-time and asynchronous services: they are the same activities. Question acquisition seeks to get a user to identify a question as much as possible before the involvement of an intermediary. This is often done through web forms regardless of what system lies behind the form. What is important is that this model represents a real-time system. All digital reference services can be seen as real-time systems. When a user inputs a question, s/he receives feedback. This is often a web form followed by a web page stating the question was received, and what will happen next ("we'll get back to you within two days," or "wait, a librarian will be with you shortly"). The point is that at different times in interacting with a digital reference system, the user must wait for a system resource. That may be a web page loading, a librarian coming into a chat environment, or a proprietary database returning a result set. This waiting on system resource, the author will refer to as "lag time."

Lag time is a well-known concept in the information retrieval world. It is assumed that there is a balance between what a system provides to a user, and how long the user must wait to get that response. When you type a query into Google, you expect good results, and you expect not to wait "too long" to get those results. However, it is also assumed a user will wait longer for better results. So metasearch engines (web sites that simply run a query on several existing search engines then combine the results), often take longer to get a result set, and will often "time out" a given search engine if it takes too long. However, it is assumed the user will wait longer to get a more complete set of results.

There are two factors identified in this subtle (and often experimental) equation between a user's willingness to wait, and a system's ability to produce good results. The first is an interface issue; the second is a performance issue. In the interface, if an information system makes clear what it is capable of (and what is required of the user . . . like waiting a specified amount of time) the user feels informed and can make an informed choice ("I am willing to wait for that result"). Switching back to a digital reference context, if a system says that it will take up to two days to respond, the user is informed and can choose whether to wait or not. They are more likely to be satisfied if you meet or exceed expectations, than if you let the user determine their expectations alone. This is clearly seen in the AskERIC customer service survey data (Lankes and Shostack, 2002) where users were informed of a two-day turn-around time and they were highly satisfied. Imagine if a user was put into a queue for a real-time system and had to wait three hours for a librarian to come online. The user would hardly be happy if they were expecting a minute or two wait time.

The second part of this equation (balancing wait and results) is the performance of the system. If the user's expectation of the result are met or exceeded, the user will be highly satisfied. So even if the user had to wait, but they received as much, if not more, relevant information than they needed, they will be satisfied. So if a librarian feels rushed to get information in a real-time setting and sends just adequate information to a patron they will be less satisfied than if the librarian took a day to reply, but sent excellent information to the patron. Once again, the point is that lag time is only one factor in meeting user expectation, and, the author would argue, less important than (or at least ameliorated by) the performance of the system (i.e., was the user's question answered well).

So, the author has postulated that the only significant different between current real-time and asynchronous systems is lag time, and that lag time exists in all digital reference systems, and has the same positive or negative effect regardless of the means of librarian/patron interaction. In other words, you must set and meet user expectations for waiting for answers. Further these expectations can be set to any time or any mode within digital reference systems.

The means of setting users' expectations are two fold: clearly delineating the lag time in a digital reference system in the interface to the user, and clearly meeting users' expectations in the answers the system provides.

CONSEQUENCES FOR SYSTEM DEVELOPERS AND PRACTITIONERS

This may all seem like a rather academic discussion, but it has very real and current application to system designers. Digital reference systems must stop trying to develop separate real-time and asynchronous systems. Rather they should build unified solutions that can account for varying lag times and interactions. In order to build such unified systems, a unified feature set needs to be constructed. Certainly this feature set includes the items listed in Table 1 with routing, queuing and co-browsing.

For practitioners this means demanding more from vendors and software developers. Until these two modes are married, tools will be grafted together (often with varying licensing schemes), and will favor one approach over another. While interoperability standards, such as those being developed by NISO, may provide some relief in integrating systems, nothing can replace bottom-up system construction of a unified digital reference system.

Of course, this unified approach leads inevitably to one more level of integration, system integration across all reference modes (i.e., desk, phone, correspondence, digital). Seeing digital reference as a core service, and part of reference, it makes sense that integrated reference management software will emerge. In fact, it would seem to follow that such integrated systems may well become part of integrated library solutions. While the architecture for such an integration of reference into the complete technical infrastructure of libraries is well beyond the scope of this paper, it is well worth thinking about. Certainly, one step in the absence of such a reference-wide and library-wide architecture is adherence to common software development and acquisition practice.

A library considering the purchase (or construction) of a digital reference package should look for:

- Adherence to known and widely accepted technical standards: The most obvious of these standards is the NSIO standards for networked reference. The problem is, as of this writing, it is not yet even released, so it holds future promise. Instead, look for more general compliance with Internet standards such as XML, SOAP and web services that are seeking to make applications interoperable by sharing database information and functions over the web.
- Willingness of vendors to place source code in an escrow account in case of later discontinuation of the product or vendor: If the .com boom of the late '90s taught libraries anything it is that promising software might disappear if the company providing the software does not have a sound business practice. Already we have seen digital reference software packages arrive and disappear. To mitigate the negative effects of this, libraries should seek some assurance such as direct access to source code, or the holding of source code by a trusted third party in the event a product goes out of support.
- Ability of software to interoperate in an open method: This is related to the first bullet, but speaks to a broader willingness of the vendor to make their products work with in-place systems such as OPACs. Will the vendor allow connections directly to an underlying database? Will the vendor share high-level architectures? The point is to assess how open the software provider is to working with you and your local implementation.
- Willingness on the part of the vendor (and ability) to negotiate pricing: Many digital reference packages are rapidly evolving. They are fixing problems in the field. That means that some of what is promised is not yet available, or not available at time of installation. Vendors and software producers should take this into account, and understand that libraries are often underwriting ongoing development of market share. Vendors should be willing to absorb some of the costs of this development process and be flexible in licensing. The less flexible a vendor is in licensing, the more stable their product should be.

This is a small list, and very much incomplete, but it is a starting point. Libraries must act as responsible and learned consumers in this market. They must also take the lessons they have learned from licensing other software and information products (e.g., databases, integrated library systems, etc.) and apply it to the domain of digital reference. If digital reference is to become a core service, we must approach it with maturity, and not as a "cool new" tool with heightened expectations for usage, and lowered expectations on the part of software producers.

NOTE

1. The author acknowledges this is far from a legal opinion, and fair-use does factor into this discussion.

REFERENCES

Gross, M. & McClure, C. & Lankes, R. (2001). Assessing Quality in Digital Reference Services: Overview of Key Literature on Digital Reference. Available http://quartz.syr.edu/quality/VRDphaseIILitReviw.pdf.

Lankes, R. D. & Shostack, P. (2002). "The Necessity of Real-Time: Fact and Fiction in Digital Reference Systems." *Reference and User Services Quarterly* 41(4).

Lankes, R. David (1998). *Building & Maintaining Internet Information Services: K-12 Digital Reference Services*. ERIC Clearinghouse on Information & Technology; Syracuse, NY.

Pomerantz, J., Nicholson, S., Belanger, Y., & Lankes, R. D. (Forthcoming). "The Current State of Digital Reference: Validation of a General Digital Reference Model through a Survey of Digital Reference Services." *Information Processing & Management*.