A Task-Oriented Interface to a Digital Library

Steve B. Cousins

Stanford University
Computer Science, Room 396
Gates Computer Science Bldg. 3B
Stanford, CA 94305
+1-415-723-7784
cousins@cs.stanford.edu

Abstract

In this paper we describe an interface to a heterogeneous digital library. The interface is designed with the following goals in mind: to support user tasks, to smoothly integrate the results of many services, to handle services of widely- varying time scales, to be extensible, and to support sharing and reuse. We discuss each of these goals, and then describe a working prototype interface.

Keywords

Digital Libraries, tasks, direct manipulation

Introduction

This paper describes the design of an interface to a heterogeneous digital library [5]. Our project is one of six digital library initiatives funded by NSF, ARPA, and NASA. The goal of the larger project is to "glue together" many different digital library sources and services. The goal of our interface is to hide the "joints" from the end users so that they can focus on their tasks. We discuss our design goals and their motivations, and then describe and critique the current implementation of our digital library interface.

Our interface is based on scenarios and published studies of library use. The most important lesson is that library use is part of a larger task context [1]. Library users have goals that they want to achieve, and individual library activities are only important as a means of achieving those goals. Another lesson is that the problem is often not solved in a single session. Studies of library use almost uniformly conclude that systems should save result sets automatically for use later [3, 4]. Finally, there is much more to digital libraries than search. Libraries, and especially digital libraries, are made up of many services. They range from search and retrieval, to services that help us understand what we have found, to mechanisms that help us manage our results, to services that help us pass on our newly-acquired conclusions to others.

Goals

Our primary goal is to support user tasks. The tasks we have in mind are composite entities, which are not instantaneously completed. For example, a user might want to buy a color printer. The corresponding library task

would involve searching for information about color printers, retrieving promising articles, sifting through those articles, annotating relevant passages, compiling new documents such as lists of desirable features, and perhaps sharing the results with colleagues or the world.

A larger task would be a professor preparing a course. Her work would involve accessing materials in the digital library, and potentially adding new materials such as an annotated bibliography or a syllabus, which other professors could access. This task might be divided into sub-tasks for the various topics covered in the course. A digital library interface needs to provide affordances for the various components of each task. Each instance of a task should persist across time, since the task is unlikely to be completed in a single session. Based on our reading of library-use studies, we believe that user tasks require a tool that falls between a "scrapbook" and a completely-automated, custom application.

Since user tasks involve an increasingly rich variety of services, our next goal is to design the interface to integrate the results from a broad array of services. In our example of the professor teaching a course, relevant services include document summarization, bibliography creation, and "sense-making" (understanding the results of broad searches). We use the term "service" to refer to computational objects that take digital library objects, such as documents, queries, and collections, as input. We are working with a list of about 30 types of services, ranging from complex information visualization services to (conceptually) straightforward translation services.

Library services differ widely in the amount of time they require, so another goal is to design the interface to handle widely varying time scales. The interface needs to let the user know before initiating a service whether it will take milliseconds or hours to complete. While a service is working, the interface needs to provide feedback on the progress of the service, and a means for interrupting the service. If the user has moved on to another task, the interface should continue to accept results from running services and compile them into a meaningful form for when the user returns to this task.

Our next goal is to make the system extensible. The number of available services is constantly growing. Ideally, adding a new service to a task needs to be as easy as dropping a "service card" onto the interface for a task. The service card would describe the parameters needed to invoke the service. It would either contain the service (for example as a Java applet) or would point to a network object that would perform the service. When appropriate, it would also include a fee schedule. Service cards would be exchanged via electronic mail or retrieved from catalogs of services.

Finally, the interface needs to support sharing and reuse of information processing knowledge. Bonnie Nardi has described how "local developers" of spreadsheet macros pop up in many different organizations [2]. We expect that with a well-designed task-based interface, individuals could share expertise in informal and semi-formal ways. An individual who spent a lot of time configuring her "color printer" research task might want to share that with a colleague who is looking to buy a new ethernet card. More formally, a digital librarian's job description could include the creation of specialized task templates for use by his patrons. Task representations also facilitate reuse by an individual, and could be used to manage a history of digital library activities.

In the next section we describe how our prototype digital library interface attempts to achieve these goals. We emphasize the interface issues that the goals raise, and ignore other issues due to space considerations.

Description of current prototype

To stress the ability to plug new services in and use them right away, our Digital Library infrastructure is called the "InfoBus," based on an analogy to a hardware bus. For the interface to the InfoBus, we have chosen to use direct-manipulation, with a relatively straightforward mapping between library objects and screen representations. The basic types of objects in our interface are queries, documents, collections, and services. Services are activated by dropping queries, documents, or collections onto them. Collections support multiple views; we currently have implemented a tabular view and a simple graphical view. Clicking on documents causes them to become activated; they currently respond by instructing a Netscape web browser to display their contents.

Our current prototype provides an interface to the InfoBus search services and to services for sense-making, for summarizing documents, for doing copy detection and for creating bibliographies from collections of document descriptions. The figure shows the interface instantiated for a demonstration task based on the scenario of a professor teaching a course.

Status

The current prototype has emphasized building an interface to a diverse set of services. We are working on improving the support for task sharing and reuse. At the present time, we handle varying time scales as long as they are in the same session, but have not yet implemented a mechanism that will allow services to continue to run across sessions. Extensibility also needs work in the current prototype. Once a new service has been implemented, it is easy for a programmer to add it to a task template, but that has not yet been elevated to something a user can do easily. Finally, our prototype runs as an X Window process alongside of a browser, but we recognize that ultimately it will need to run in something like Java/Netscape in order to be widely available.

Librarians from our project like the task-orientation, but caution us that the real world is a messy place. As the interface prototype and the digital library testbed become more robust, we will be performing user testing.

Conclusions

There is more to a digital library than search, and there needs to be more to a digital library interface than a query form and a list of results. Libraries, and library users, vary widely, so task-based interfaces are required.

Acknowledgments

This work is supported by the NSF under Cooperative Agreement IRI-9411306. Funding for this cooperative agreement is also provided by ARPA, NASA, and the industrial partners of the Stanford Digital Libraries Project. Thanks to Terry Winograd, Andreas Paepcke, Michelle Baldonado, and Scott Hassan for comments on this paper.

References

- [1] E.A. Fox, D. Hix, L.T. Nowell, D.J. Brueni, W.C. Wake, L.S. Heath, and D. Rao. Users, user interfaces, and objects: Envision, a digital library. *Journal of the American Society for Information Science*, 44(8):480-91, 1993.
- [2] Bonnie A. Nardi and James R. Miller. An Ethnographic Study of Distributed Problem Solving in Spreadsheet Development. CSCW '90, pp. 197-208.1990.
- [3] Vicki L. O'Day and Robin Jeffries. Orienteering in an Information Landscape: How Information Seekers Get from Here to There. CHI '93, pp. 438-445. 1993.
- [4] J. Ray Scott. Library Information Access Client. CHI '94 Interactive Posters, vol. 2, pp. 143-144. 1994.
- [5] Stanford Digital Library Project Homepage: <http://www-diglib.stanford.edu/diglib/>