Prospero: A Base for Building Information Infrastructure

B. Clifford Neuman Steven Seger Augart

Information Sciences Institute
University of Southern California

Abstract

The recent introduction of new network information services has brought with it the need for an information architecture to integrate information from diverse sources. This paper describes how Prospero provides a framework within which such services can be interconnected. The functions of several existing information storage and retrieval tools are described and we show how they fit the framework. Prospero has been used since 1991 by the archie service and work is underway to develop application interfaces similar to those provided by other popular information tools.

I. Introduction

The past several years has brought the introduction of a large number of information services to the Internet. These services collectively provide huge stores of information, if only one knows what to ask for, and where to look. Unfortunately, knowing what and where to ask is a major problem; available information is scattered across many services, and different applications are needed to access the data in each.

While there have been many attempts to create gateways between these services, such gateways have not been as useful as needed. Gateways have typically taken one of two forms. The first is a portal between two services: one service is used to find an instance of a second service, to which the user is then connected, leaving the user to figure out how to query the second service. The second form of gateway translates queries from one service into those of another, and returns the result of the query in a form that a user of the first service can understand. The second kind of gateway is easier for the user, but it is still limited since, as typically implemented, such gateways do not allow complete integration of the two information spaces.

This paper shows how the Prospero Directory Service provides a framework for interconnecting information services. The paper begins by dividing the functions of information systems into four categories: storage, retrieval, organization, and search. The role of existing services in this taxonomy is described, and the role of Prospero in interconnecting existing information services using this taxonomy is presented.

II. The Four Functions

The services of existing Internet information retrieval tools can be broken into four functions: storage, access, search, and organization.

The storage function is the maintenance of the data that may subsequently be provided to and interpreted by remote applications. File systems support storage, providing a repository where data may be stored and subsequently retrieved. Document servers including the Wide Area Information Service (WAIS) [5], menu servers including Gopher [7], and hypertext servers including World Wide Web [1] also provide the storage function since they manage documents that are subsequently retrieved and displayed by their clients.

Whereas storage is primarily a function of the server, access involves both the client and the server. The access function is the method by which the client reads and possibly writes the data stored on a server. The access method is typically defined by network protocols including the File Transfer Protocol (FTP) [11], Sun's Network File System (NFS) [12], the Andrew File System (AFS) [4], and the application specific protocols used by WAIS, Gopher, and World Wide Web. The client and the server must use the same protocol.

The search function involves the iterative or recursive retrieval and analysis of information about data (meta-information), possibly across multiple repositories, with a specific goal of identifying or locating data that satisfies the search criteria. A search heuristic defines the method and strategy used to conduct the search. Examples of search tools include Knowbots [6] and NetFind [13]. Though searches are initiated by a client when the need for specific data is realized, parts of the search may execute remotely. For example, the lookup of an entry in a remote index or directory on a WAIS, Gopher, WWW, or Prospero server constitutes a search. The Dynamic WAIS interface to NetFind [3] takes this a step further; the server initiates and manages, in real-time, a search across multiple hosts on behalf of the client.

Figure 1: A framework for network information services

Brute force search in a system as large as the Internet is not practical. To be effective, search heuristics must be applied. These heuristics rely on meta-information describing the data that is available. The more structured this meta-information, the more useful it is in directing searches. The organization function involves the collection, maintenance, and structuring of such meta-information. Examples of organization mechanisms include directories in Prospero, menus in Gopher, links in hypertext documents, indices of data local to a WAIS server, and the file name index maintained by archie [2] for files scattered across the Internet.

The difference between search and organization is that a search is initiated when specific data is needed, whereas data is organized in advance to support subsequent searches. Search mechanisms play a role in the organization of data when the results of possibly complex or expensive searches are recorded for subsequent use by applications.

III. A Flexible Framework

Most information services presently available on the Internet provide their own mechanisms for each of the four functions, whereas their novel features are usually confined to their user interfaces and/or at most one of the four functions. As a result, gateways are required to allow applications from one service to use information maintained for another.

A well defined interface is needed that will allow information maintained by one service to be used by another. This interface would appear between search and organization, providing a common method for applications to query the meta-information maintained by other services. A similar interface would separate storage and access, providing access by all applications to data maintained by different services.

Figure 1 shows the location of existing services in such a framework. Meta-information available from archie, WAIS, World Wide Web, Prospero, and Gopher menus are available through a common query mechanism. Menu browsers, file system tools, hypertext browsers, and dedicated applications such as the command line archie client and NetFind use this mechanism to query services. Update of meta-information is also supported, providing a common interface between hypertext and menu editors and repositories. Search engines can use the query mechanism to identify objects of interest. They can then record the result of the search for subsequent use by using the update mechanism to add new links to the information fabric. Section IV discusses how Prospero provides this functionality.

The separation of storage and access is a bit more complicated. There are already numerous access methods in use, and many of the files of interest in the Internet exist on servers that will only support a single protocol. The problem can

Figure 2: The Prospero naming network

initially be addressed by supporting a common application library that accepts a reference to an object obtained from the meta-information services, and automatically invokes the appropriate access method to retrieve the object. Prospero currently uses this method. While not described in this paper, we are also working on a common data access protocol for information services. That protocol will support gateway services to existing data access protocols.

IV. The Model

The information services available on the Internet each have their advantages. In order to accommodate the needs of each, a mechanism to integrate these services must be flexible. However, to be practical and efficient, such a mechanism must not become bogged down providing functionality that will only be used occasionally. The solution to these competing goals is to provide a simple but extensible framework on which the mechanisms specific to individual systems can be built.

These characteristics are present in the Prospero Directory Service [8]. We chose to concentrate initially on integrating tools for organizing information, applying Prospero as a common protocol for access to meta-information from multiple information services.

Meta-information is represented by Prospero in a *naming network*, a directed graph with labeled edges (shown in figure 2). This naming network provides the fabric through which applications navigate. Each node in the graph represents an object. An object can be a file, a directory, both, or neither (if neither, it only has attributes). Each object has associated with it a set of user extensible attributes. These attributes can have intrinsic meaning, such as the length of a file, or they can be application specific.

If an object serves as a container for data, then it is a file and has one or more ACCESS-METHOD attributes associated with it. Each ACCESS-METHOD attribute encodes the information needed to retrieve the data from a data repository using a particular access method.

If an object is a directory it contains a set of named links to other objects. The labeled edges in the naming network correspond to these links. The names on the links serve as sign posts to help the user, or the user's application, navigate through the naming network in search of the desired information. Applications that treat the naming network as a filesystem use the names of the links as components of filenames, allowing files in subdirectories to be named by the concatenation of the names of the links that are followed to reach them.

When searching for information, users and applications also use attributes. In addition to object attributes, attributes can be stored with links. Such link attributes either store additional information about the link such as annotations, or they cache information about the object to which the link refers. Cached attributes allow an application to retrieve attributes for the objects in a directory in a single query, rather than making individual requests for the attributes of each object referenced from the directory.

A filter can also be associated with a link. A filter, and a second kind of link called a union link, allow a view of a directory to be created that is a function of another view. Filters and union links provide a mechanism for users to encode, on a link, methods to be applied when searching. Because they are applied when a directory is listed, the resulting view is kept up-to-date, even when the underlying information changes. Filters and union links are described in greater detail elsewhere [9].

V. Implementation Notes

Prospero was designed to be simple but extensible. The protocol is simple, supporting stateless queries for attributes and the contents of directories. The protocol is layered on top of a reliable delivery mechanism implemented using UDP. In the normal case, a query requires a single packet for the request and a single packet response, eliminating the cost associated with setting up and breaking down a connection. This low cost is particularly important for clients that contact multiple servers in series as is often the case when resolving names.

Despite the simplicity of the protocol, Prospero is extremely flexible. Functions specific to an application rely on the extensible attribute mechanism. Applications not needing that functionality ignore the additional attributes, while applications that know about it can take action based on the value of the attributes.

VI. Integrating Services

This section describes how the meta-information maintained by several Internet information services can be represented using the model from section IV.

Archie [2] constructs an index of files from Internet anonymous FTP sites and makes the index available to applications using the Prospero protocol. A query to the archie database corresponds to a Prospero directory query for a directory whose name includes the arguments of the query. These arguments include a string that is matched against the filenames in the index. The result of a query is represented as the contents of the Prospero directory. Upon receipt of the directory query, the archie server extracts the arguments, performs a query on the archie database, and returns the results to the client. Each file matching the query is represented as a link in the directory to the file on the remote FTP site. The information needed to retrieve the matching file is returned as part of the ACCESS-METHOD attribute associated with the link. Other attributes are also returned, including file size and the time of last modification. One can apply a Prospero filter to a directory query to restrict the links that are returned to those for files on hosts in a particular part of the network.

Gopher [7] allows users to browse menus configured by Gopher service providers. By selecting menu items, users are able to run applications, search local databases, select and display files and sub-menus, and connect to other services available on the Internet. A Gopher menu can be represented as a Prospero directory. The individual items in the menu correspond to links in the directory. Submenus are links to other directories. Links to files have an associated ACCESS-METHOD attribute that provides the information needed to retrieve the file. Attributes associated with links also specify how the data in a file should be presented, for example how to display images or play back an audio recording. Links to Internet services reachable by telnet have an ACCESS-METHOD attribute of type TELNET; this ACCESS-METHOD contains the information needed to telnet to the service.

The WAIS server [5] maintains a full text index to a set of documents on a single system, allows that index to be queried remotely, and allows remote read access to the documents. The WAIS client provides a common front end for queries and access to documents on multiple WAIS servers. A WAIS query can be represented as a Prospero directory in much the same manner as an archie query. Like archie, the result of the query would be represented by the links in the directory. Each of these links would be a reference to a document on the server itself and the ACCESS-METHOD associated with the link would specify the WAIS access method. Other attributes of the link could include the relative confidence in the match. To retrieve a matched document the application would pass the selected link to the Prospero library function `pfs_open()` which would automatically invoke the WAIS access method to retrieve the document from the WAIS server.

World Wide Web is a hypertext browser that supports the interconnection of collections of documents with links that originate within the documents themselves. Links to non-hypertext documents are also supported. A hypertext document may be represented in Prospero as a node that is both a file and a directory. The document that is displayed to the user would be retrieved according to the ACCESS-METHOD attribute associated with the node. The links in the directory would represent links to other documents. The name of each link would contain the text of a tag within the document from which the link is to originate. Offsets in the target document can be represented as attributes of the link.

Prospero allows users to create their own directory hierarchies from which they can make links to directories and objects of interest. Since queries to other services and the objects stored by them are represented as nodes in the Prospero naming network, users can create directories in which the linked objects reside in different services, but can be accessed using a common protocol, thus supporting the integration of information across services.

VII. Security

Security is an important feature that is missing from most information systems on the Internet. Without fine grained control for access to information, the owners of certain information will not make it available using such systems. Since our goal is to allow the integration of information from all sources on the Internet, security must be an important consideration. Security is especially important in our model since we support the remote modification of information.

In Prospero, access control lists (ACLs) may be associated with directories in the naming network, and with individual links within a directory. The permissions supported include read, modify, insert, delete, list, and administer. Prospero requests are authenticated and the identity of the client is used to determine the access permissions that apply. Several authentication methods are supported including authentication based on the client's Internet address, the use of passwords, and Version 5 of Kerberos.

Hooks are also present to support distributed authorization and accounting mechanisms [10]. Support for accounting will become critical as new for-hire information services are introduced.

VIII. Status

Prospero has been available since December 1990. Recent revisions to the protocol allow more flexible integration of information services. Prospero has been used since Spring of 1991 to make available meta-information maintained by archie. Work is underway to develop application interfaces for Prospero that provide the functionality of Gopher, WAIS, and World Wide Web. To find out more about Prospero, or for directions on retrieving the latest distribution, send a message to info-prospero@isi.edu.

IX. Summary

A huge amount of information is available on the Internet. Unfortunately, this information is scattered across many services and different applications are needed to access the data in each. A framework was defined within which such information services can interoperate. By providing a common method for applications to query the meta-information maintained by the services that organize data on the Internet, information maintained by one service can used by all. Prospero provides such an interface. A service that uses Prospero to export meta-information about the data it provides will be usable by many applications. Similarly, an application that uses Prospero to find files will be able to access information from many services.

Acknowledgments

Many individuals contributed to the design and implementation of Prospero. Ed Lazowska, John Zahorjan, David Notkin, Hank Levy, and Alfred Spector helped refine the ideas that ultimately led to the development of Prospero. Kwynn Buess, Steve Cliffe, Alan Emtage, George Ferguson, Bill Griswold, Sanjay Joshi, Brendan Kehoe, Dan King, and Prasad Upasani helped with the implementation of Prospero and Prospero-based applications. Gennady Medvinsky and Stuart Stubblebine commented on drafts of this paper.

References

[1] Tim Berners-Lee, Robert Cailliau, Jean-Francois Groff, and Bernd Pollermann. World-wide web: The information universe. *Electronic Networking: Research, Applications and Policy*, 2(1), Spring 1992.

[2] Alan Emtage and Peter Deutsch. archie: An electronic directory service for the Internet. In *Proceedings of the Winter 1992 Usenix Conference*, pages 93–110, January 1992.

[3] Darren R. Hardy. Scalable internet resource discovery among diverse information. Technical Report CU-CS-650-93, Department of Computer Science, University of Colorado, Boulder, April 1993. M.S. Thesis.

[4] John H. Howard, Michael L. Kazar, Sherri G. Menees, David A. Nichols, M. Satyanarayanan, Robert N. Sidebotham, and Michael J. West. Scale and performance in a distributed file system. *ACM Transactions on Computer Systems*, 6(1):51–81, February 1988.

[5] Brewster Kahle and Art Medlar. An information system for corporate users: Wide area information systems. Technical Report TMC-199, Thinking Machines Corporation, April 1991.

[6] Robert E. Kahn and Vinton G. Cerf. The Digital Library Project; Volume 1: The world of Knowbots (draft). Corporation for National Research Initiatives, 1988.

[7] Mark McCahill. The Internet gopher: A distributed server information system. *ConneXions – The Interoperability Report*, 6(7):10–14, July 1992.

[8] B. Clifford Neuman. Prospero: A tool for organizing Internet resources. *Electronic Networking: Research, Applications and Policy*, 2(1):30–37, Spring 1992.

[9] B. Clifford Neuman. The Prospero File System: A global file system based on the Virtual System Model. *Computing Systems*, 5(4):407–432, Fall 1992.

[10] B. Clifford Neuman. Proxy-based authorization and accounting for distributed systems. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pages 283–291, May 1993.

[11] Jon B. Postel and J. K. Reynolds. File transfer protocol. DARPA Internet RFC 959, October 1985.

[12] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon. Design and implementation of the Sun Network File System. In *Proceedings of the Summer 1985 Usenix Conference*, pages 119–130, June 1985.

[13] Michael F. Schwartz and P. G. Tsirigotis. Experience with a semantically cognizant internet white pages directory tool. *Journal of Internetworking: Research and Experience*, 2(1):23–50, 1991.

Author Information

Clifford Neuman is a scientist at the Information Sciences Institute of the University of Southern California. After receiving a Bachelor's degree from the Massachusetts Institute of Technology in 1985 he spent a year working for Project Athena where he was one of the principal designers of the Kerberos authentication system. He holds M.S. and Ph.D. degrees from the University of Washington, where he initially developed Prospero as part of his dissertation. His research focuses on problems of system organization and security in distributed systems.

Steven Seger Augart is member of the research staff at the Information Sciences Institute of the University of Southern California. He received a Bachelor's degree from Harvard University in 1989 and an M.S. from the University of California at Irvine in 1992, with various bouts working as a systems programmer along the way. His work at ISI has focused on the development of Prospero as a scalable information infrastructure for large distributed systems.

This research was supported in part by the National Science Foundation (Grant No. CCR-8619663), the Washington Technology Centers, Digital Equipment Corporation, and the Advanced Research Projects Agency under NASA Cooperative Agreement NCC-2-539. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any of the funding agencies. Figures and descriptions in this paper were provided by the authors and are used with permission. The authors may be reached at USC/ISI, 4676 Admiralty Way, Marina del Rey, CA 90292-6695, USA. Telephone +1 (310) 822-1511, email bcn@isi.edu, swa@isi.edu.