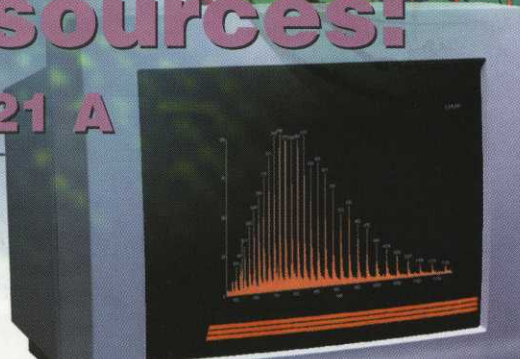
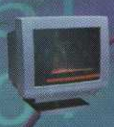
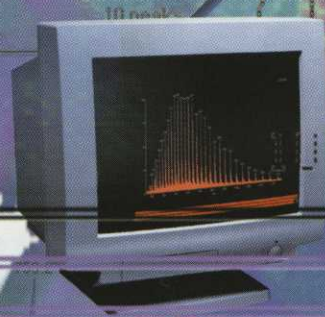


ANALYTICAL CHEMISTRY

Includes News & Features and AC Research DECEMBER 1, 1996



151 VEAEEARI
 201 GDDDSAD
 251 KLSEVFKG
 301 NFITETG
 351 TLTIEQV
 401 FLDLIQEG
 451 RIPVHMIE
 501 AKEPISME
 551 AREAKV
 601 SRSEV



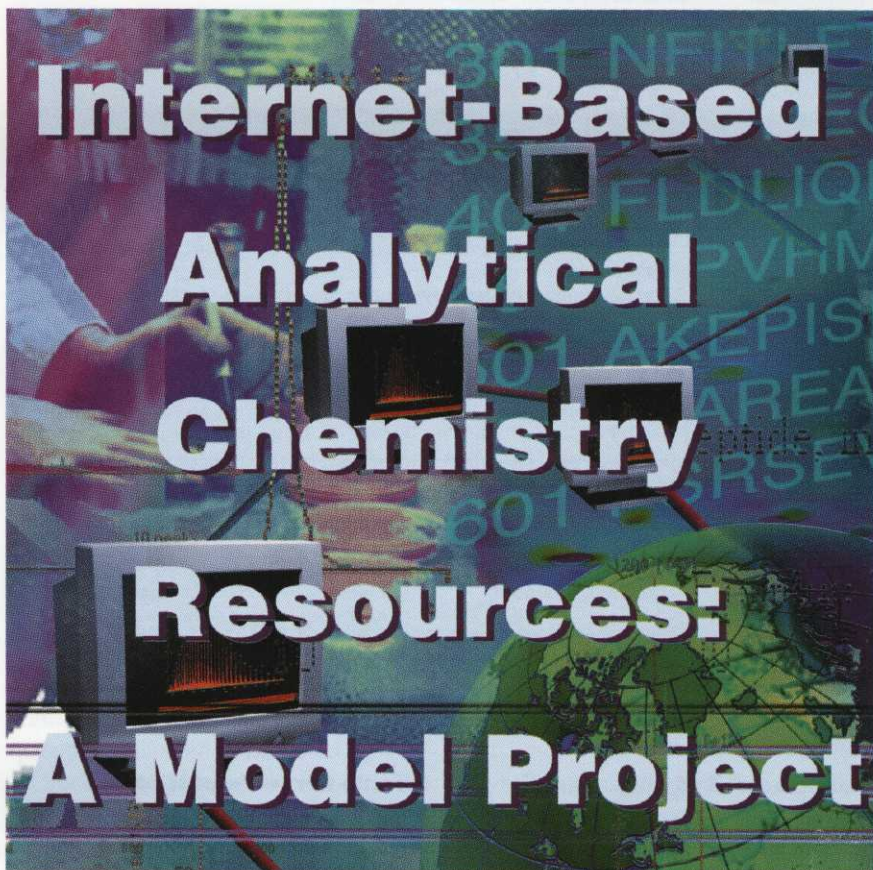
Internet-Based Analytical Chemistry Resources: A Model Project 721 A



Man has always sought to organize knowledge in his attempts to comprehend the complex world. We have seen many times during history how technological improvements prompted radical changes in the way information has been disseminated. One of the most dramatic examples was Gutenberg's invention some 500 hundred years ago that allowed the mass production of printed books. Today, with the popularity of the Internet, we may be seeing the beginning of a revolution equally influential. New tools have become available that promise to completely change the way information is disseminated. The Internet is already being used for a large number of projects involving, for example, global public libraries and teaching in a variety of fields (1).

The storage and dissemination of scientific information are also being radically reorganized. Within five years of their introduction, World Wide Web browsers have become the most commonly used software on personal computers. Organizations of all types are rushing to create the most attractive Web sites possible in order to stay competitive with other groups in the same field. Cryptic incantations such as "http://www.acs.org" or "http://nationaldebt.com", once the sole property of computer geeks and wizards, have become prominently displayed on television programs and in magazine advertisements.

Web technology was in fact invented for the exchange of scientific information. It has not, however, replaced more conventional sources of information such as journal articles and books any more than television has replaced radio. (Count how many radios you own compared with the



Web technology can be used to manipulate analytical data and facilitate the exchange of scientific information

number of televisions.) Computer screens are difficult to watch for long periods, and the physical requirements of using a computer make it incompatible with the way most people read long articles.

Web browsing has, however, become the first method of choice for many scientists and students when they want to find information on a subject. Currently, the quality of the information on the Internet is somewhat lower than that obtainable from conventional print resources, but the immediacy of discovering new information seems to outweigh issues of editorial and artistic nicety. Even so, the success of a global scientific "library" depends on the

development of high-quality databases for the relevant scientific data, as well as on the development of means for convenient interconnection of these databases.

For all practical purposes, some types of scientific information exist only as Internet-based resources. Bioanalytical databases containing DNA and protein sequences and three-dimensional protein structures have become invaluable to molecular biology researchers and are normally accessed through Web interfaces (2-7). These databases are based on information from one of the largest analytical chemistry projects in history: the determination of the complete sequence of

David Fenyö
Wenzhu Zhang
Brian T. Chait
Rockefeller University
Ronald C. Beavis
New York University Medical Center

Databases, CGI tools, and helper applications used in PROWL.**Databases**

MassBank	A collection of protein mass spectra
MatrixDepot	Data about known MALDI matrices
Protocols	Recipes and advice for carrying out protein chemistry experiments in a manner compatible with MS
Amino acid information	A collection of tables, figures, and models for understanding amino acid chemistry in proteins
Sequences	Up-to-date copies of SWISS-PROT, GENPEPT (the translation of GENBANK), PIR (Protein Identification Resource), and OWL (a nonredundant protein sequence database)

CGI tools

ProteinInfo	Retrieves protein sequences on the basis of several query mechanisms, using a phylogenically organized set of sequence databases
ProFound	Searches known protein sequences for a pattern of masses obtained from protease digest experiments for protein identification
PepFrag	Searches known protein sequences for a pattern of MS/MS fragments from protease digest-derived peptides for protein identification
Display	An interactive tool for displaying mass spectra on an HTML page

Helper applications

PAWS	For planning and analyzing the results of protein chemistry experiments on the basis of a proposed primary sequence that can be supplied in several standard formats
M/Z	Displays and analyzes mass spectra in a number of different formats, including the highly compressed format used to store raw data in MassBank

For the names and URLs of other helper applications that are suggested for use with PROWL, see the PROWL Software Page at "<http://128.122.10.5/software/contents.htm>".

genomes from selected organisms. The Web interface for these databases is simple: A user queries the database for information, and a file containing the requested data is downloaded by the user's computer.

In the process of constructing a bioinformatic database of protein mass spectra (MassBank), it became clear that the value of a Web-based resource can be greatly enhanced if it supplies a well-chosen set of software tools and basic information along with raw data. These additional features make the database an integrated resource for data retrieval and analysis rather than a simple data repository. Therefore, we created a more versatile resource called PROWL (<http://chait-sgi.rockefeller.edu> or <http://mcphar04.med.nyu.edu>). This resource incorporates a number of ideas about Internet resource design that can serve as an example of how current Web technology can be used for manipulating data derived from analytical chemistry. In order to explain the ideas underlying the design of PROWL, it is necessary to explain some of the basic technological concepts behind Web sites.

Client-server architecture

All Web-based transactions use a "client-server" model of computer interaction. When a user sits down at a computer and runs browser software, that computer becomes a "client". When the client browser wants a piece of information, it sends out over a network a request that details what the client wants. The request is in a highly structured, standardized form (called a "protocol") that depends on the type of information required and carries with it the address of the computer that has the required data (the "server") and some type of pointer that says where that information can be found on the server. The combination of protocol, server address, and pointer to the data is referred to as the universal resource locator (URL) of the information.

Protocols currently supported by browsers include the hypertext transfer protocol (HTTP), the file transfer protocol (FTP), and the sendmail transfer protocol (SMTP). A universal resource locator starts with the abbreviation for the protocol to be used, followed by a colon and

two separator marks; the URL for a piece of hypertext has the general format "<http://...>". The name of the protocol is very important because it tells the server what sort of software to run to retrieve the required piece of information and what format to use when sending it back to the client computer.

The next element in a URL is the Internet address of the computer that will act as the server, in either words or numbers that are separated by periods. The address of the NYU server for PROWL can be written as either "mcphar04.med.nyu.edu" or "128.122.10.5". The address in words doesn't actually contain the information necessary to find a computer directly; it is used to find the numerical address in lookup tables that are scattered around the network in computers called domain name servers (DNS). Once the numerical address has been determined, it is then used to send the request to the appropriate server.

The use of numerical addresses for computers attached to a network in the form of four 8-bit numbers is a fundamental part of the Internet protocol (IP), which is the current standard for the interconnection of computer networks. Entering the numerical address in a URL will usually result in a quicker response from a server than using a word address because it does not require the initial step of consulting a set of lookup tables to translate the address.

The simplest (and most common) response that a server can make when receiving a query is to send back a requested file to the Internet address of the client computer, using the specified protocol. The server includes an additional piece of information along with the file, specifying what type of file it is sending. This additional information is called a multipurpose Internet mail extension (MIME) specification. The browser running on the client computer then interprets data sent from the server in a fashion that is appropriate for that particular MIME specification. For example, a browser will interpret text (MIME type "text/plain") differently from a picture (MIME type "image/gif") or a sound (MIME type "audio/x-wav").

This simple set of behaviors—sending a URL and receiving information with a

MIME type—has led to the success of the Internet style of information exchange, as opposed to older models. A client does not have to “log on” to a remote computer and run software explicitly; the standard protocols perform these operations in the background without user intervention. The server can send back a complicated mixture of different types of data (pictures, text, videos, etc.) and have it interpreted correctly. The server and the client do not need to be running the same type of operating system in order to interact, because neither computer attempts to directly control the other. Instead, they interact with each other in a rather abstract manner using formal expressions that do not refer to the manner of performing an action but only request that the action be carried out.

More complex client-server behavior

During the designing of MassBank, it became clear that the simple set of client-server behaviors outlined above is not flexible enough to produce a really useful resource. Protein mass spectra are usually quite complicated, so it was decided that rather than storing a simple table of masses, it would be necessary to store a representation of the original mass spectrum. The interpretation of these protein mass spectra can only be made within the context of the proposed covalent structure of the protein (i.e., the protein’s amino acid sequence). In turn, the amino acid sequence can only be interpreted with reference to the properties of the individual amino acid residues, known or suspected post-translational modifications, and the methods used to prepare a sample for analysis.

Storing, viewing, and interpreting complex mass spectra and protein sequences require software tools that can examine and compare data from several sources. The resource designer’s aim should be to condense all of the available information to a format that can be readily grasped by a user who is interested in a particular protein, while retaining the possibility of accessing detailed information for in-depth examination. The best format for the reduced information is a set of standard diagrams that can be composed “on-the-fly” by the client-server combination.

The client-server architecture should allow the user to interact with these diagrams, so that features of interest to the user can be highlighted and explored by a sequence of point-and-click operations.

A range of technologies have been developed to make client-server interactions more versatile. The first and most commonly used of these techniques, called the common gateway interface (CGI), provides an interface to the server computer. The CGI allows a client to request a server to run a specified program that must be located on that server. The program is used to generate a stream of information directly back to the client by using the hypertext transfer protocol. The browser allows the client to send a set of parameters to the CGI, which will be supplied to the requested program and used to determine what output will come from the program. Therefore, the server can respond to a client by composing a piece of hypertext that did not originally exist but was composed to resolve the question asked by the client. Almost all WWW-based databases use CGI programs to do searches on the database and return the entries that correspond with the search parameters entered by the user.

Another well-developed method involves the server requesting that a client computer run a specific program to read the data that it is transmitting. The server accomplishes this task by specifying a MIME type for the data, in response to which the browser starts up another program on the client computer that interprets the information from the server. For the browser to start the correct program, the browser must be configured so that when it receives a particular MIME type (such as “image/gif”), it starts up a program capable of displaying GIF-type graphics files. The program that gets started on the client computer is called a “helper application”. The browser actually receives all of the information from the server and stores it in a temporary file; the helper application is given the name of the temporary file when it starts, which gives it access to the information without it having to deal with receiving network information. Therefore, a helper application can be any software on the client machine that accepts a filename as a parameter when it is started. A somewhat more sophisticated type of helper ap-

plication called a “plug-in” can be used to display graphics within a browser’s window. Plug-ins use the same idea as helper applications: A program that already exists on the client computer is activated with a downloaded temporary file as one of its start-up parameters.

The most recently developed form of enhanced client-server interaction involves programs that exist on the server and are downloaded and run on the client computer in response to a request from the client’s browser. The client computer never has a permanent copy of the program; a new copy is downloaded and run every time it is required. The program is not run directly on the client computer’s operating system, and it has no direct access to the client computer’s hard disks or memory. Instead, the browser behaves like an operating system, managing the downloaded program’s use of the client machine’s resources.

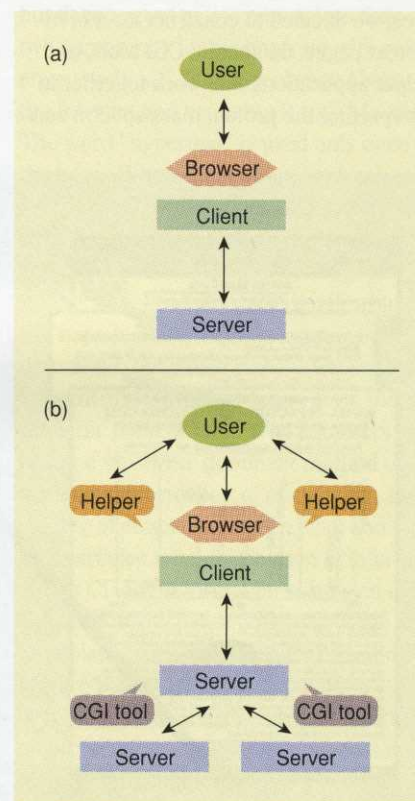


Figure 1. Comparison of (a) simple client-server architecture and (b) the more complex model chosen for PROWL.

The arrows indicate the main channels of information flow between elements of the resource. Some lesser channels of information flow, such as swapping information between helper applications, have been omitted for clarity.

