A Web Gateway to a Virtual Mathematical Software Repository*

Ronald F. Boisvert
National Institute of Standards and Technology

Abstract

Embedding robust and efficient algorithms in high-quality reusable software is an effective means of transferring the expertise of mathematicians to researchers in other disciplines, and a wealth of such software is already available. Unfortunately, its volume and distribution over labyrinthine computer networks makes it difficult for users to locate appropriate tools.

The Guide to Available Mathematical Software (GAMS) is network-based cross-index and virtual repository that provides scientists and engineers with convenient access to such software. GAMS currently indexes some 9000 problem-solving software modules from some 80 packages distributed among four Internet-accessible software repositories. This software is cross-indexed using the widely-adopted GAMS problem classification system. In addition, GAMS provides on-demand redistribution of such objects as abstracts, documentation, and source code of software that it catalogs. The system is based on a client-server architecture utilizing a specialized high-level repository protocol; communication is based on Unix sockets. Recently, a gateway to the World Wide Web was built using the Common Gateway Interface (CGI) of NCSA's httpd, enabling access to GAMS services by Web browsers like Mosaic. This has added more than 32,000 "pages" to the Web, many of which are constructed on demand by the gateway.

In this paper I outline how the GAMS system is structured and describe the challenges involved in using CGI, Mosaic, and the Web to provide public access to a large virtual software repository.

1 Introduction

We have come to rely on a wide variety of tools to help us mine large research libraries for useful nuggets of information. Among these tools are cross-indexes, such as the card catalog, Mathematical Reviews, Current Index to Statistics, the ACM Guide to Computing Literature, Science Citation Index, and the various classification systems on which these are based. The explosive growth of the World Wide Web (WWW) has already provided a network-based resource that may already surpass any existing library in size and complexity. The Web, however, still lacks many of the navigational aids common in library settings.

As a concrete example, consider an average computer user who would like to find software for solving a particular mathematical problem. There already exists a wealth of good software packages for solving such problems. They might be commercial packages installed for use at a local computing site or public-domain packages available for downloading on the Internet. Unfortunately, a typical user has a difficult time dealing with this bounty. Even for users who know where and how to traverse the labyrinthine computer network to a software repository, determining what is actually applicable to the problem at hand is still a tedious and error-prone process. Each repository may have a different access mechanism, and directory services are usually primitive at best. Our user would be greatly helped by an easily accessible, centralized domain-oriented information resource like a cross-index journal found in the library, with effective directory services and facilities for retrieving software and related items once they are located. Fortunately, systems with these characteristics are beginning to emerge.

*Presented at the *Second International WWW Conference '94 : Mosaic and the Web*, Chicago, IL, October 18–20, 1994.
Contribution of the NIST, not subject to copyright. Commercial products are identified in this article in order to facilitate understanding. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the products identified are necessarily the best available for the purpose.

2 Math Software Cross-Index

The Guide to Available Mathematical Software[1](GAMS) is one such service. GAMS currently contains information on some 9000 problem-solving software modules from about 80 packages found in four physically distributed software repositories (three at the National Institute for Standards and Technology[2] (NIST), and *netlib*[3]). The system supports software searches by problem classification, keyword, or name, as well as retrieval of such items as abstracts, documentation, examples, and source code. (Some 32,000 such objects can be retrieved.)

GAMS performs the function of an interrepository and interpackage cross-index. As such, it provides a *data mapping* service for its specialized domain [6]; that is, it collects data about software available from external repositories and combines it into a homogeneous whole. GAMS also provides the functions of a repository itself (i.e., retrieval). However, instead of maintaining the cataloged software itself, GAMS provides an *operation mapping* [6], i.e., transparent on-demand access to multiple repositories managed by others. We use the term *virtual repository* to describe systems of this type.

The GAMS project started in the early 1980s as an effort to catalog software supported for use on NIST's central computing facility (several printed catalogs were produced, the last in 1990 [5]). This software, which includes some commercial products, remains a large subset of GAMS. (Note that source code of commercial software is not redistributed by GAMS, although items such as documentation and examples are often available thanks to the support of vendors, such as Visual Numerics, NAG, and Cray Research.) More recently the scope of the system has been broadened to include external repositories of publically available software, although only one, *netlib*, has yet been indexed. In addition, NIST now supports public access to GAMS to demonstrate the possibilities afforded by community resources of this type. This service was officially announced to the public in the spring of 1994, and there have been some 3000 users each month since then. (See the Appendix for information about access to GAMS.)

Cross-indexing in GAMS is based on a 736-node tree-structured taxonomy of mathematical and statistical problems developed by the GAMS project [3] [4]. (See Figure 1). This taxonomy has been adopted for use by many library developers and scientific computing centers (see, for example, [8] [9] [10]). Each software module within GAMS is assigned one or more problem classifications. The taxonomy can then be used as a rudimentary decision tree to help focus in on a group of software modules for a particular problem. We also maintain an index of related mathematical terminology. These terms are mapped to the problem taxonomy, making it possible to translate alternative terminology into that used by the taxonomy. This provides the basis for keyword searching in GAMS.

Another useful concept supported by GAMS is that of *filters*. Partitioning more than 9000 problem-solving software modules using a 736-node taxonomy necessarily leads to classes populated by a large number of modules. Thus, even differentiating among the modules in a single class can still be quite tedious. Filters allow the user to specify additional preferences in a number of areas: language (e.g., Fortran, C), precision (e.g., single, double, multiple), access (e.g., free, proprietary), package, and repository. When presenting the user with modules in a given class, GAMS screens out those that do not satisfy the current set of filters.

3 The GAMS Software Repository Model

The operation of GAMS is based on a simple software repository model. We define a virtual software repository to be a collection of modules. A *module* represents a reusable software part, such as a Fortran subroutine, a C procedure, a self-contained program, or a command in a self-contained system. Modules may have alternate *versions*, identical except for the precision with which the computation is done. In addition, modules are characterized by a set of predefined *attributes*, such as precision and language, which can be used to established filters for module selection. One such attribute is a list of problem classifications from the taxonomy, and a tree-structured taxonomy is an important part of the repository model. *Packages* are collections of modules that are related in some way; they may solve related problems or be distributed and maintained as a unit.

[1] http://gams.nist.gov/
[2] http://www.nist.gov/
[3] http://www.netlib.org/

Figure 1: Topmost level of the taxonomy, with problem F expanded.

```
A.    Arithmetic, Error Analysis
B.    Number Theory
C.    Elementary and Special Functions
D.    Linear Algebra
E.    Interpolation

F.    Solution of Nonlinear Equations
F1.     Single equation
F1a.      Polynomial
F1a1.       Real coefficients
F1a2.       Complex coefficients
F1b.      Nonpolynomial
F2.     System of equations
F3.     Service routines

G.    Optimization
H.    Differentiation, Integration
I.    Differential and Integral Equations
J.    Integral Transforms
K.    Approximation
L.    Statistics, probability
M.    Simulation, stochastic modeling
N.    Data handling
O.    Symbolic computation
P.    Computational geometry
Q.    Graphics
R.    Service routines
S.    Software development tools
Z.    Other
```

Associated with each module and package is a set of *components*. These are retrievable objects such as abstracts, documentation, source code, examples, and tests. Component types are not restricted to a predefined set, so that retrievable objects unique to a particular module or package are easily accommodated. Module and package components physically reside at one or more *repositories*. Each repository may have a different access mechanism, such as electronic mail, anonymous FTP, or a repository server; any access mechanism that can be encapsulated into a stand-alone retrieval tool can be accommodated.

A user's goal in using GAMS is to locate appropriate modules. Once found, the user downloads one or more module components. The system hides the details of the component access mechanisms from the user.

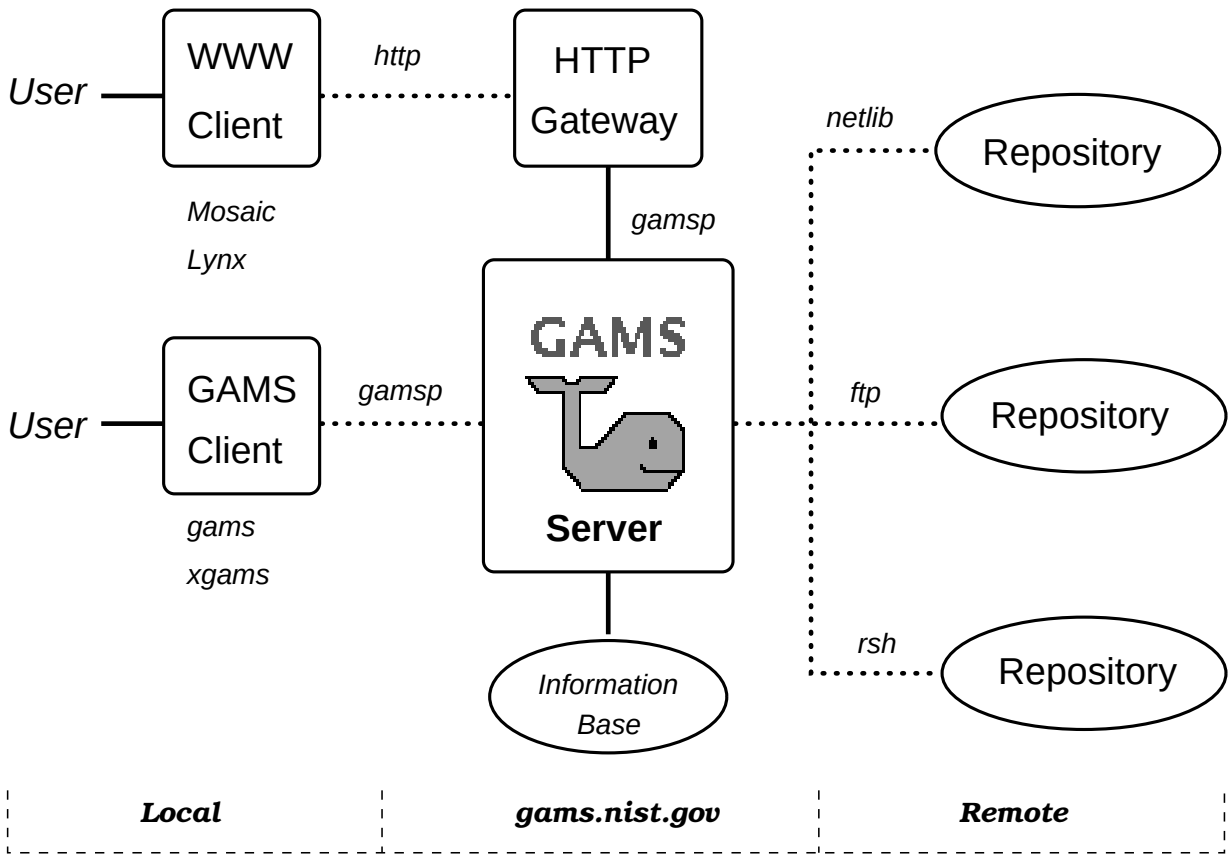
4 Architecture

GAMS is based on a client-server architecture (see Figure 2). The GAMS server runs continuously at NIST, listening for TCP/IP-based socket connections from remote clients. When a client connects to the server, it uses a private ASCII protocol (*gamsp*) [1] to request and receive information from the server. The server is stateless, and is based on one developed for the *xnetlib* project [7].

The interactions between a client program and the server are at a fairly high level, corresponding to operations derived from the GAMS repository model. Thus, typical requests have semantics such as : What classes are related to the keyword "regression"? What are the subclasses of class D2? What modules are classified at class D2a1? What are the retrievable components of module 7754? Send Source of module 7754.

Server operation is illustrated in Figure 3 where we simulate the interaction between a client program

Figure 2: The architecture of the GAMS system.



and the server using telnet. Note that each server request is preceded by two lines; the first contains the client's email address and the second contains the basic service name (only one, `gams-serve`, is available). The third line contains the actual client service request. In this example the client requests information about the problem class with tag F. The response is in a fixed, easily parsed, format. The first line returned indicates the status of the request (`OK` or `BAD`), provides a count of the number of characters in the response, and echoes the request. A complete set of server requests is listed in Figure 4. Some 30,000 transactions of this type are currently being processed each month.

The server has a small database that contains such items as module and package names, brief descriptions, classifications, and other data. In addition, it knows what retrievable components are associated with each module and package and how to retrieve them from external repositories. When asked by a client to retrieve a component, the GAMS server connects to the external repository in real time (i.e., it too becomes a client), downloads the requested object, and forwards it to the GAMS client.

The GAMS clients manage the interface with the user. They run on the user's workstation, connecting to the GAMS server to obtain data as needed. Several such clients, including a simple command-line client (*gams*) and a Motif-based graphical user interface for the X Window System (*xgams*), have been developed for use on Unix systems. (See Figure 5.) A WWW gateway to the GAMS server was developed recently, allowing browsers like Mosaic and Lynx to be used to access GAMS. This provides a means of access for PC and Macintosh users.

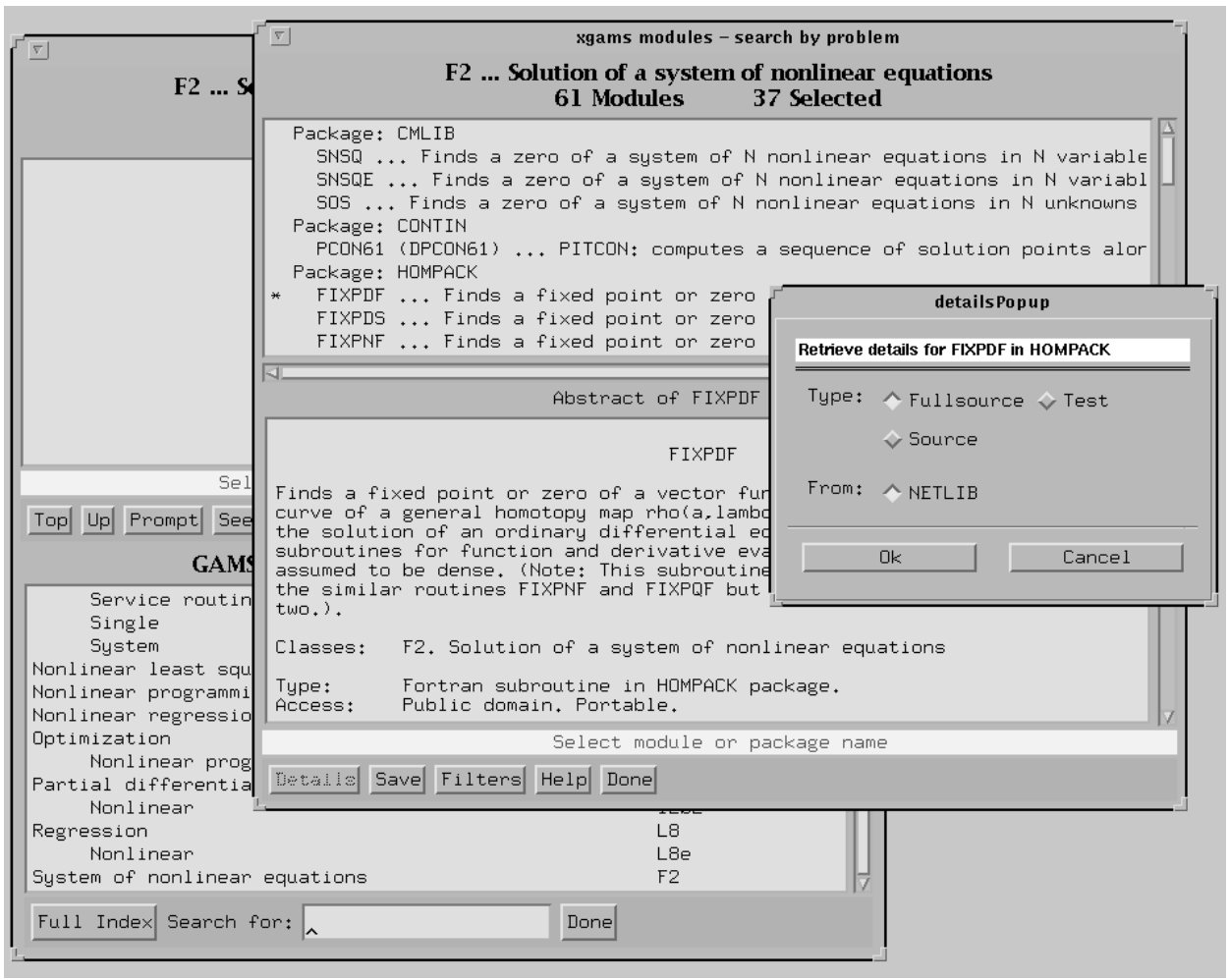
Figure 3: Example of GAMS server interaction. Lines representing server requests start with >, those representing responses start with <.

```
> telnet gams.nist.gov 5555
Trying 129.6.80.11 ...
Connected to doc.
Escape character is '^]'.
> boisvert@nist.gov
> gams-serve
> class f
< OK 140 :  class f
< F
< Solution of nonlinear equations
< 3
< F1
< Single equation
< F2
< System of equations
< F3
< Service routines (e.g., check user-supplied derivatives)
< 0
Connection closed by foreign host.
```

Figure 4: GAMS server requests.

```
class [<tag>]
class-index [<keyphrase>]
explain-filter <filter> [= <value>]
list-services
get-module-component <component> <mid> [<site>]
get-news-item [<item>]
get-package-component <component> <package> [<site>]
list-classes
list-filters
list-module-components <mid>
list-modules-in-class[+] <tag> [<filter-spec> ...]
list-modules-in-package <package>
list-modules-with-name <module> [<package>]
list-news-items
list-package-components <package>
list-packages
parent-class <tag>
register
```

Figure 5: The native *xgams* client.



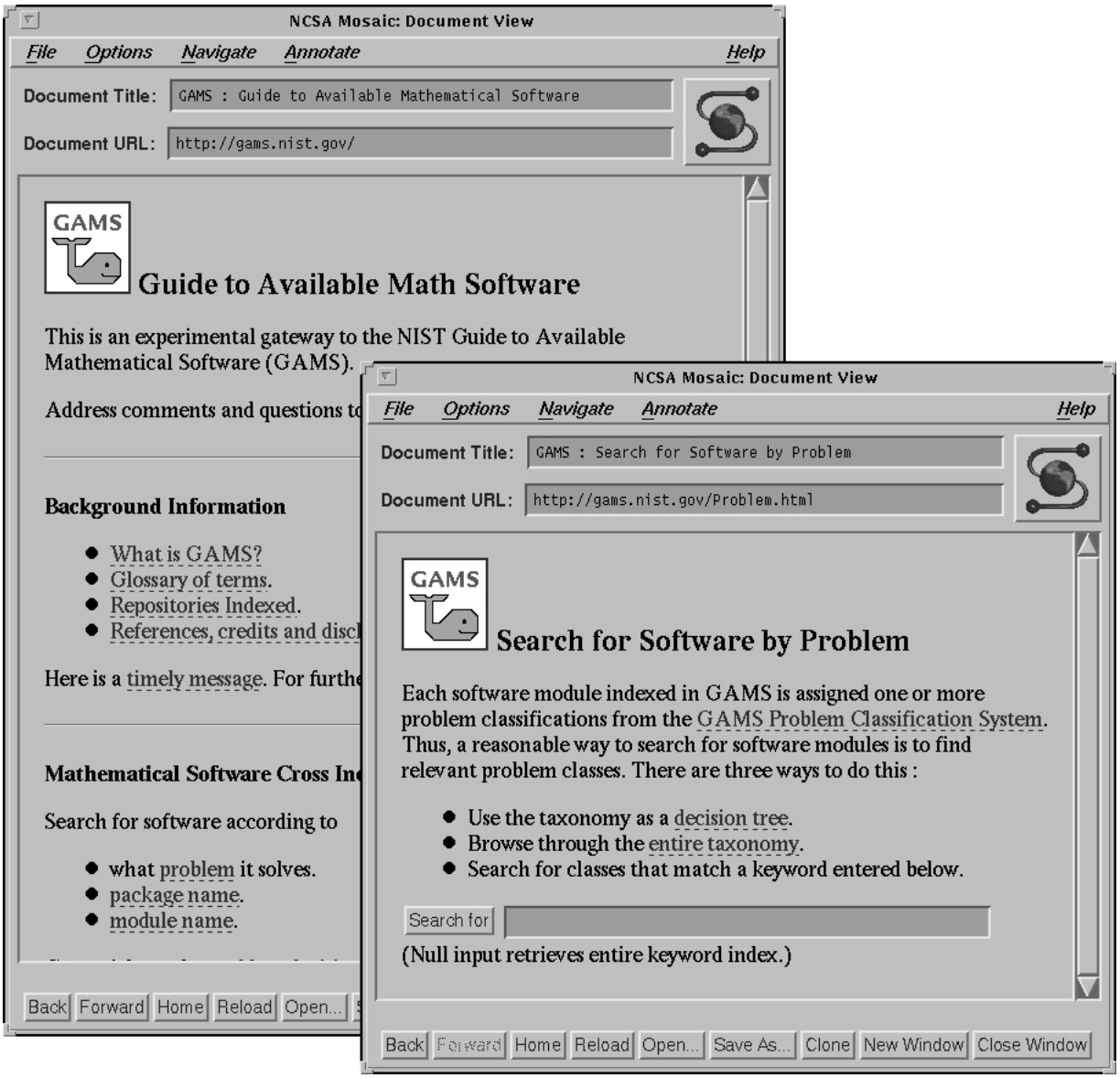
5 The Web Gateway

The WWW gateway to GAMS supports hypertext-style user interfaces to GAMS provided by WWW browsers such as Mosaic and Lynx. The gateway is implemented using an NCSA httpd server[4] installed on gams.nist.gov, a Sun SPARC 10 Model 51 workstation which is also host to the native GAMS server. The GAMS home page provides pointers to background information on the project, and an introduction to the cross-indexing and repository services provided by GAMS. (See Figure 6.) These services include search for software by problem, by package name, and by module name. Searching by problem can be done by using the taxonomy as a decision tree, browsing the taxonomy as a single file, or searching an index to the taxonomy. Browsing in this hypertext space will lead users to lists of appropriate software modules, and then to lists of components, which can then be downloaded by referencing their hypertext links.

No static replication of data is required to support the WWW gateway to GAMS. Instead, most hypertext links in the GAMS Web are invocations of a gateway program, `gams-serve`, using the Common Gateway Interface[5] (CGI) supported by NCSA httpd. `gams-serve` translates the input URL to one or more requests in the *gamsp* protocol, submits them to the native GAMS server, and reformats the response into HTML. (See Figure 2.) In effect, the gateway adds a carefully organized space of more than 32,000 documents to the

[4] http://hoohoo.ncsa.uiuc.edu/docs/Overview.html
[5] http://hoohoo.ncsa.uiuc.edu/cgi/

Figure 6: The Mosaic interface to GAMS.



Web, most of which are either generated on demand or mapped on demand from some other network service. Although this arrangement requires the participation of two servers, with the gateway providing translation in both directions, we have not observed a significant degradation in performance over more direct access such as that provided by the native *xgams* client.

I next present some observations on the use of WWW browsers and CGI in this context.

- WWW browsers such as Mosaic are a good first step toward the development of universal network-based client programs. Reuse of such a universal client eliminates many of the difficulties associated with building a user interface, such as programming directly with the Motif toolkit. The hypermedia interaction model is quite flexible, allowing instructions on how to use GAMS, as well as pointers to related information, to be interspersed with the active parts of the system in a natural way.

- CGI is flexible enough to provide basic gateways to stateless servers such as GAMS. Most of the facilities provided by the *gams* and *xgams* clients were easily accommodated. At the same time, CGI is

a simple enough mechanism to permit rapid prototyping. A prototype gateway was developed in about one week's time, and a second week was required to complete the addition of services, and polishing the system.

- Although Mosaic is quite flexible, implementing sophisticated user interactions is often awkward, and sometimes impossible. For example, the *xgams* client requests information from the GAMS server about the existing query filters and their possible values on startup. *xgams* can then present these to the user on demand, without any additional server interaction. By setting these filters, the user provides a profile that *xgams* uses to qualify each subsequent query. While the forms extensions to HTML provide a useful facility for presenting such choices to the user, implementing a persistent user profile in the context of Mosaic and HTML would be, at best, awkward and inefficient. Because of this we have not yet implemented GAMS filters in the GAMS WWW interface.

- The GAMS server requires that the user's email address be supplied as a part of its protocol, and the native *gams* and *xgams* clients attempt to generate this information automatically. This provides a mechanism to record who has downloaded a piece of software so that they can be contacted with notifications of updates, bug fixes, etc. Since the HTTP protocol only provides the hostname, this information is lost when the WWW gateway is used.

6 Looking Ahead

The next step for systems like GAMS is to provide expert-level advice on software selection, and within the context of GAMS we are concentrating on more intelligent schemes for selecting software in a given class. To do so, we are focusing on extensions to the filtering mechanism that support both simple knowledge representations and effective user interactions [2]. Plans for more extensive problem-dependent module filtering requires a type of user interaction based on many experimental filtering operations with immediate response. This is best done within the client program itself rather than by repeated interactions with the server. This could be accomplished by the addition of a scripting language to systems like Mosaic. We think that this is a necessary step in the evolution of a universal network client.

We are also beginning work on a substantial revision to the GAMS problem classification system. Much of the current taxonomy is more than 10 years old, and the emergence of software for new problems, accumulated experience with the system, and our expert advisory system goals all provide compelling reasons to start anew. We feel strongly that the building of subject-area taxonomies, thesauruses, and other classification mechanisms are crucial to the provision of effective directory services on the WWW.

Finally, we believe that experiences in the research sector with systems like GAMS demonstrate that the WWW indeed provides a promising means for disseminating useful information that can increase the productivity of its users.

References

[1] R. F. Boisvert. Guide to Available Mathematical Software network server. GAMS Working Note, 17 pages, July 1993.

[2] R. F. Boisvert. The architecture of an intelligent virtual mathematical software repository system. *Math. & Comp. in Simul.*, 1994. to appear.

[3] R. F. Boisvert, S. E. Howe, and D. K. Kahaner. GAMS—a framework for the management of scientific software. *ACM Trans. Math. Softw.*, 11:313–355, 1985.

[4] R. F. Boisvert, S. E. Howe, and D. K. Kahaner. The Guide to Available Mathematical Software problem classification system. *Comm. Stat. – Simul. Comp.*, 20(4):811–842, 1991.

[5] R. F. Boisvert, S. E. Howe, D. K. Kahaner, and J. L. Springmann. Guide to Available Mathematical Software. NISTIR 90-4237, National Institute of Standards and Technology, 1990. Also available as PB90-216508/AS, National Technical Information Service, Springfield, VA 22161.

[6] C. M. Bowman, P. B. Danzig, U. Manber, and M. Schwartz. Scalable internet resource discovery : Research problems and approaches. *Comm. ACM*, 37(8):98–107,, 1994.

[7] J. Dongarra, T. Rowan, and R. Wade. Software distribution using XNETLIB. *ACM Trans. Math. Softw.*, 21, 1995.

[8] eLib is the electronic library of the Konrad-Zuse-Zentrum fuer Informationstechnik Berlin (ZIB). To use eLib, telnet to elib.zib-berlin.de (130.73.108.11) and login as user elib. eLib can also be reached via the ZIB Web page at URL http://elib.zib-berlin.de/.

[9] The Numerical Algorithms Group, Ltd., Oxford, U.K. *The NAG Fortran Library Manual*, June 1991. Mark 15.

[10] Visual Numerics Inc., Houston, TX. *IMSL MATH/LIBRARY User's Manual, FORTRAN Subroutines for Mathematical Applications*, September 1991. Version 2.0.

Appendix : Access to GAMS

- *Local clients.*
  The GAMS Unix clients (*gams* and *xgams*) can be obtained by anonymous FTP from enh.nist.gov (129.6.16.1) in the directory [GAMS].
- *Guest login.*[6]
  Guest logins are available for those who wish to try out the GAMS clients. To do this, telnet to gams.nist.gov (129.6.80.11) and log in as user gams (for the command line interface) or xgams (for the X Window System interface).
- *Gopher.*[7]
  A gopher gateway to the GAMS guest logins is provided on gams.nist.gov.
- *World Wide Web.*[8]
  GAMS can be accessed by World Wide Web browsers like Mosaic or Lynx by specifying the URL http://gams.nist.gov/.

Author Biography

Ronald F. Boisvert received a B.S. in Mathematics from Keene State College in 1973, a M.S. in Applied Science from the College of William and Mary in 1975, and a Ph.D. in Computer Science from Purdue University in 1979. Since then he has been at the National Institute of Standards and Technology (formerly the National Bureau of Standards), where he is now leader of the Mathematical Software Group in the Computing and Applied Mathematics Laboratory. His research interests include mathematical software and scientific computing. He was one of the original developers of the ELLPACK system for elliptic boundary value problems at Purdue and now heads the Guide to Available Mathematical Software (GAMS) project at NIST. He is author of more than 40 technical publications and serves as Editor-in-Chief of the ACM Transactions on Mathematical Software. In 1992 he received the U.S. Department of Commerce Silver Medal for meritorious federal service. For more information, see his personal home page[9].

Author's Electronic Address : `boisvert@nist.gov`
Author's Postal Address : NIST, Gaithersburg, MD 20899, USA.

[6] telnet://gams.nist.gov
[7] gopher://gams.nist.gov
[8] http://gams.nist.gov
[9] http://larch.cam.nist.gov/~boisvert/