JOURNAL OF MAGNETIC RESONANCE **131,** 362–366 (1998)
ARTICLE NO. MN981377

Internet Technology in Magnetic Resonance: A Common Gateway Interface Program for the World-Wide Web NMR Spectrometer

Marian L. Buszko,* Dominik Buszko, and Daniel C. Wang†

Department of Microbiology and Cell Science, University of Florida, Gainesville, Florida 32611

Received October 2, 1997; revised December 31, 1997

A custom-written Common Gateway Interface (CGI) program for remote control of an NMR spectrometer using a World Wide Web browser has been described. The program, running on a UNIX workstation, uses multiple processes to handle concurrent tasks of interacting with the user and with the spectrometer. The program's parent process communicates with the browser and sends out commands to the spectrometer; the child process is mainly responsible for data acquisition. Communication between the processes is via the shared memory mechanism. The WWW pages that have been developed for the system make use of the frames feature of web browsers. The CGI program provides an intuitive user interface to the NMR spectrometer, making, in effect, a complex system an easy-to-use Web appliance. © 1998 Academic Press

Key Words: **WWW Interface; CGI program; Web-operated NMR spectrometer.**

INTRODUCTION

The World Wide Web (WWW) (*1*) has been designed as a universe of global, network-accessible information. This broad definition can certainly span information generated in real time. Indeed, many real-time devices, including telescopes and other scientific instruments (*2*), have been connected to the Web. Once connected, the instruments can be controlled via the familiar graphical interface of a Web browser. The ease of use of Web browsers has been one of the hallmarks of the World Wide Web. Recently, with the development of the new features for Web browsers, e.g., frames and Java support, it has become possible to provide not only intuitive but also feature-rich user interfaces that are of particular value to interfacing instrumentation.

In the area of magnetic resonance, a Web-operated MRI imager (*3*) and a WWW NMR spectrometer (*4*) have been demonstrated. On both systems, users can start acquisition with their own acquisition parameters, and the results are displayed by Web browsers. On the WWW NMR spectrometer, FID signals are displayed. After evaluating the time domain signals, users launch an off-line data processing package by bringing the acquired data to their local computers. This operation is done automatically in the Web environment when the data processing package is recognized by the Web browser as a helper application. Local processing of the data does not tie-up the time of the NMR console which, at that time, is free to service other users' requests for acquisition.

As described in our recent paper (*5*), the Web interface of the WWW NMR Spectrometer runs on a dedicated workstation. The workstation, Sun Microsystems' SUN IPX (*6*), is additionally equipped with an S-Bus digital input/output (I/O) adapter (*7*) and a set of two analog–digital converters (*8*). Together, they provide the necessary *hardware link* to the Nicolet NT-300 NMR console (*9*). The *software link* is provided by a custom-written program. The main elements of this Common Gateway Interface (CGI) (*10*) program are described in this paper.

RESULTS

The fundamental concept of Web-enabled instruments is the utilization of a World-Wide Web server and a CGI script or program written in a high-level language. For this development, unmodified NCSA HTTPd was used. NCSA HTTPd is a general purpose Web server developed and available at the National Center for Supercomputing Applications, University of Illinois at Urbana–Champaign. The CGI program of the WWW NMR Spectrometer was written in-house in the C language. In operation, the CGI program is launched by the Web server upon request from a user over the network. To provide an interface between the user and the CGI program, a set of hypertext markup language (HTML) (*11*) pages is used. The pages that we provide allow the user to input the required acquisition parameters and to access the acquired data. The CGI program controls the spectrometer via the serial port of the host computer. During acquisition, the CGI program reads the time domain data from the digital

* To whom correspondence should be addressed.
† Present address: Actel Corporation, Sunnyvale, CA 94086, USA.

1090-7807/98 $25.00

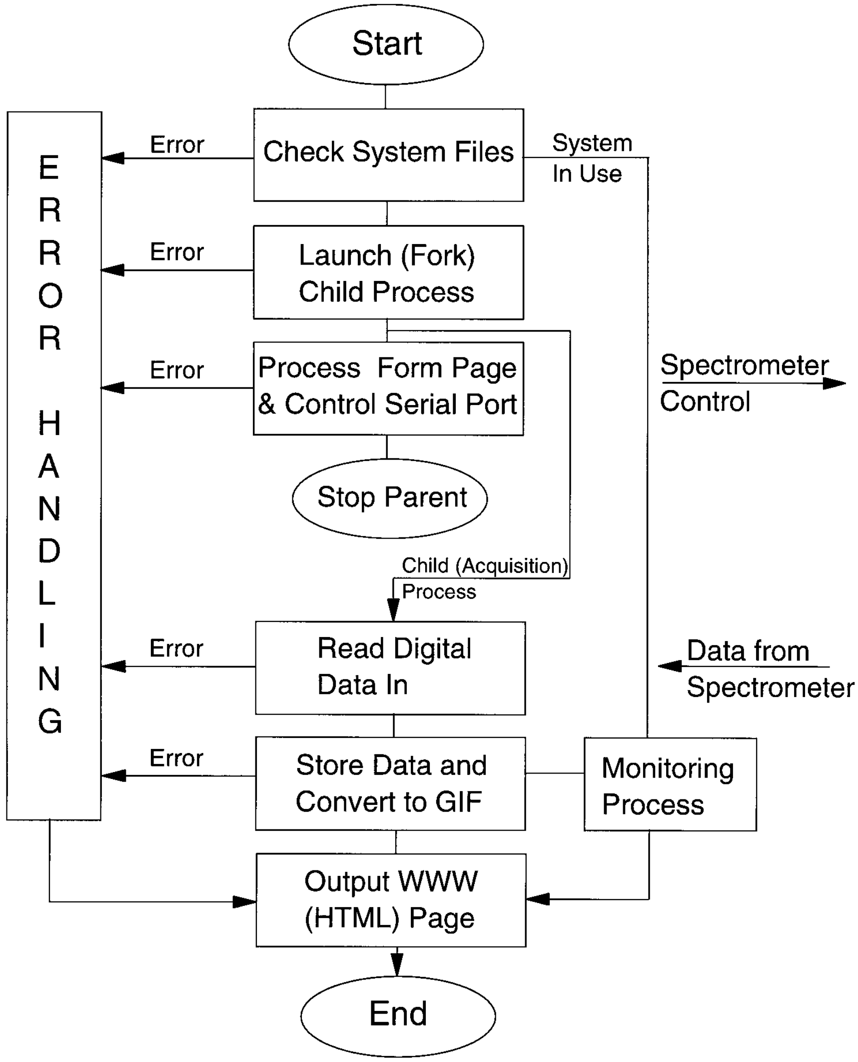

FIG. 1. The block diagram of the Common Gateway Interface (CGI) program for the World Wide Web NMR spectrometer.

I/O board for processing and output to the user. Figure 1 presents the block diagram of the CGI program.

One of the key design elements of this CGI program is the application of the *fork* system call to generate multiple processes. The UNIX operating system itself relies extensively on the use of multiple processes to handle multiple users and/or multiple tasks (*12*). Similarly, the CGI program can use multiple processes to handle simultaneous tasks of interacting with the user and with the spectrometer. In the current implementation of the CGI program, the parent process primarily communicates with the network browser and sends out commands to the spectrometer via the serial port; the child process is mainly responsible for data acquisition. The communication between the two processes is handled through a shared memory (*12*). To handle unexpected situations, including alarm clock, software termination signal, etc., the *signal* system call is employed.

Upon launching the CGI program, a program identification file is created to be present throughout the progress of the acquisition. This is to ensure that only one set of acquisition parameters is executed at a time. By checking the existence of this file, the CGI program can send a busy signal back to the browser and can allow for monitoring of the acquisition in progress.

After checking the program identification file, the *fork* system call generates both the parent and the child processes. Specifically, the parent process contains initialization, parameter extraction, and pulse program execution. At the beginning of the initialization, a short delay is introduced to guarantee that the shared memory has been created by the child process. After reading the system information file, the parameters used for the main program are extracted. The serial port and the signal handler are also initialized during the initialization.

The parameter extraction procedure includes the reading of parameters from the Web browser, the decoding of parameters, and the examining of the parameters' validity. Data, posted by the Web browser using the FORM POST method (*11*), is decoded back to their original ASCII format in this procedure. All parameters are checked by a parameter-examining procedure to make sure the values are valid for execution of the pulse program. After all parameters are verified, their values are stored in specific structures for use in other parts of the program.

The child process contains two main tasks: initialization and acquisition. During the initialization, the shared memory is created and the digital I/O board is opened. The signal handler is set up at this time to handle the software termination.

For signal acquisition, the child process gets the parameters from the shared memory and sets up a procedure loop that corresponds to the requested number of acquisitions. A procedure within the child process actually reads the digital data from the I/O board. After each single scan, a graph of the FID and the actual data file are created and stored for output to the user. This procedure repeats until the end of the acquisition. At the end of the child process, the program identification file is removed.

When an unexpected situation is encountered during the program execution, the error procedure is called. Since the parent process interacts with the network browser, an error message can be sent directly to the user. Depending on the error condition, different descriptions will be stored in a log file for reference and debugging. The log file can be opened by the parent process and displayed in the browser. The program identification file is always removed at the end of the error procedure.

A short signal-handling procedure was also incorporated into the CGI program. This procedure executes only when the system signals are initiated. The most common signals are alarm and software termination signals. The procedure can close all input–output ports and terminate the current acquisition on the spectrometer console.

DISCUSSION

Figure 2 presents two WWW pages that have been developed to facilitate communication with the WWW NMR Spectrometer and its CGI program. Both pages make use of the frames feature of modern browsers. On both pages, the top frames provide links to pages that facilitate learning, feedback, access to data repository, and execution of NMR experiments. They remain visible throughout the entire interaction of the user with the system, and they are purposely blended with the typical elements of the Web browser, like the reload, print, or other buttons of the browser. The large bottom frame is used to either input acquisition parameters (Fig. 2a) or to view the acquired data (Fig. 2b). The form page in Fig. 2a presents to the user the relevant input parameters for a typical 1D acquisition of NMR signals. Within limits imposed by the hardware of the spectrometer, users have full control of the acquisition parameters.

The CGI program and the Web environment can facilitate either remote or local operation of the NMR spectrometer, much like the X-Windows environment (*11*) would do. Both the WWW and the X-Windows interfaces can share the same look and feel. However, several distinct features are perhaps worth mentioning. First, the WWW environment makes the Web interface widely available, platform-independent, and easy to use. Second, of particular importance for remote operation, the WWW interface does not require a high-speed communication channel; a Web-operated spectrometer can efficiently be run over slow communication links, including analog telephone lines. Third, since no user accounts are required on our Web-operated system no administrative task is necessary to set individual passwords, user quotas, etc.; those tasks can be extensive and time consuming in large multiuser sites.

The child process of the CGI program performs only a minimum amount of handling of the spectrometer data; i.e., it only reads the data off the digitizer and stores them in a file. However, more handling can be performed and actual digital signal processing (DSP) functions can be added to the child process. In particular, digital filtering and decimation of oversampled data could be implemented, thus adding functionality unavailable on the original NMR console.

CONCLUSIONS

The CGI program provides intuitive user interface to the NMR spectrometer, making, in effect, a complex system an easy-to-use Web appliance. Although the program is designed around the specific hardware, the fundamental concepts of the program are general and can be utilized in programming other systems for Web operation. The program, currently running on a UNIX workstation, could also be deployed on a PC platform running the Windows 95 or Windows NT operating systems. Comparable software tools are available on both platforms, and the price and/or performance differences between UNIX workstations and PC/Windows computers are narrowing.

While the performance of the Web-operated NMR system is improved, compared to the original spectrometer, it could be enhanced even further. For instance, since all the elements of the TCP/IP networking are available to the CGI program, it would be possible to distribute the tasks of acquiring data, advanced data processing, and user interfacing not only among different processes but also among multiple, even heterogeneous computers. The resulting distributed architec-

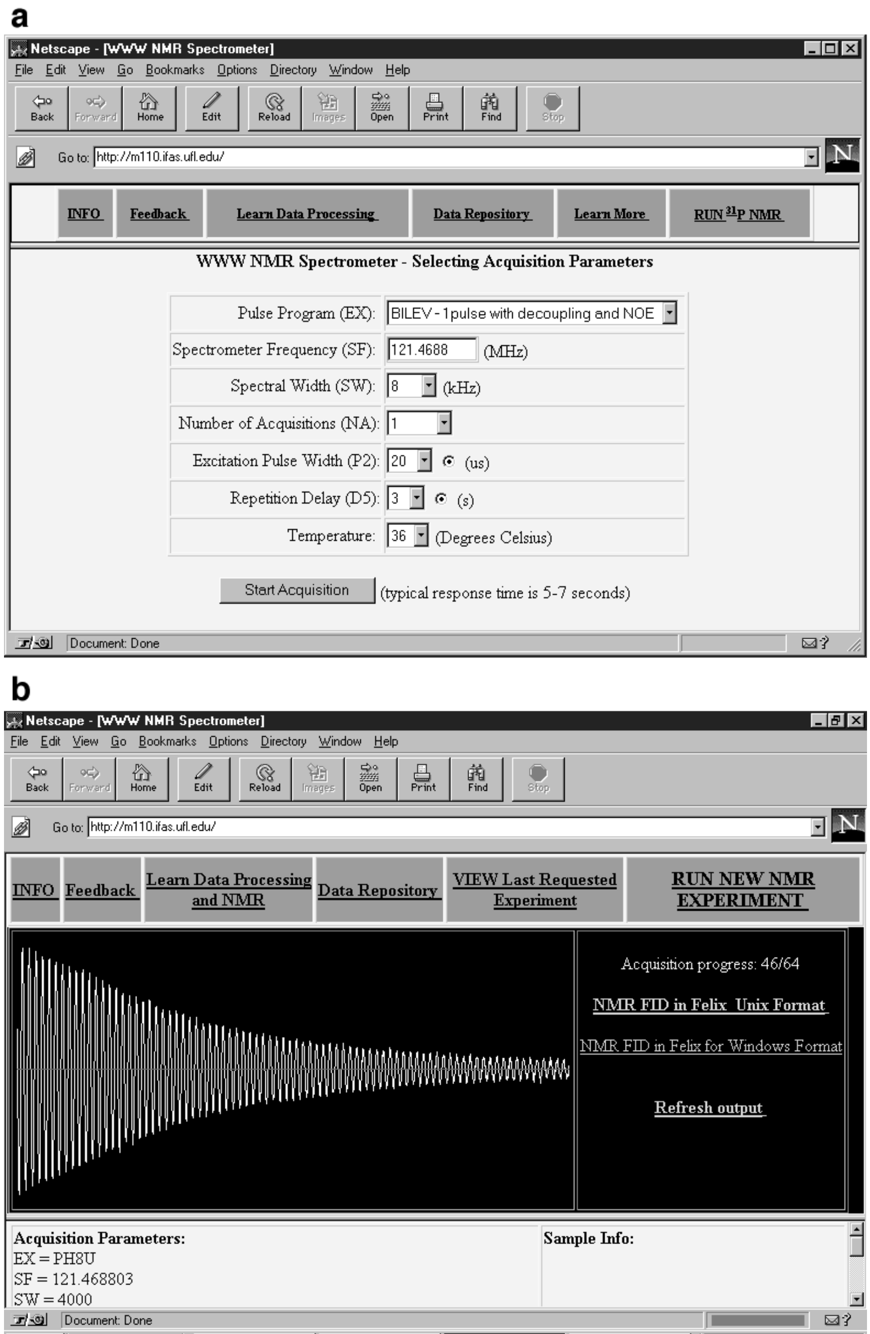

FIG. 2. Two WWW pages designed to facilitate the communication with the CGI program for operation of the WWW NMR spectrometer. The WWW NMR Spectrometer available to the general public can be accessed at the Internet address http://m110.ifas.ufl.edu.

ture could greatly enhance the overall performance of the system and make it suitable for more demanding applications, e.g., real-time *in vivo* imaging.

Future enhancements of the system could also include a Java applet approach (*14*). In the Java applet environment, the Web server sends out a complete data processing package to the user. Currently, the performance and the features of Java applets are rather limited. For example, since a Java applet cannot access certain hardware elements, printing of the spectra is not permitted on the user's workstation. However, future developments, including the ActiveX technology (*15*), could make this approach very attractive.

ACKNOWLEDGMENT

The authors thank Drs. J. N. Wilson and Yann-Hang Lee for helpful discussions. The work was partially sponsored by the Dean of Academic Programs of the Institute of Food and Agricultural Sciences, University of Florida, Gainesville, Florida.

REFERENCES

1. T. J. Berners-Lee, WWW: Past, present, and future, *Computer* **29,** 69–77 (1996).
2. (*a*) M. J. Cox and J. E. F. Baruch, Robotic telescopes: An interactive exhibit on the World Wide Web, *in* ''Second International Conference of the World-Wide Web,'' Chicago (1994). (*b*) R. Maturo, G. Kath, R. Zeigler, and P. Meechan, Control of a remote microscope over the Internet, *BioTechniques* **22,** 1154–1157 (1997).
3. C. S. Potter, C. D. Gregory, and P. C. Lauterbur, *in* ''Proceedings, Fourth Scientific Meeting of the Society of Magnetic Resonance, New York, April 27–May 3, 1996.''
4. M. L. Buszko, D. C. Wang, D. Buszko, J. E. Gander, and K. N. Scott, Is NMR spectrometer still a single-user instrument? A World-Wide Web interface, *in* ''28th Congress Ampere, Extended Abstracts'' (M. E. Smith, and J. H. Strange, Eds.) pp. 216–217, University of Kent at Canterbury (1996).
5. M. L. Buszko and D. C. Wang, A World-Wide Web interface to an NMR spectrometer, *Appl. Magn. Reson.* **12,** 95–101 (1997).
6. Manufactured by Sun Microsystems, Inc., Mountain View, California, USA (http://www.sun.com, August 28, 1997).
7. Manufactured by Engineering Design Team, Inc., Beaverton, Oregon, USA (http://www.edt.com, August 28, 1997).
8. Manufactured by Burr-Brown Corporation, Tucson, Arizona, USA (http://www.burr-brown.com, August 28, 1997).
9. Manufactured by Nicolet Magnetics Corporation, Fremont, California, USA; Bruker Instruments, Inc., Fremont, California, USA (http://www.bruker.com, August 28, 1997).
10. T. Boutell, ''CGI Programming in C & Perl,'' Addison-Wesley, Reading, MA (1996).
11. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, ''Hypertext Transfer Protocol—HTTP/1.1,'' RFC 2068, U.C. Irvine, DEC, MIT/LCS (1997), *available* http://www.cis.ohio-state.edu/htbin/rfc/rfc2068.html.
12. W. R. Stevens, ''UNIX Network Programming,'' Prentice-Hall, Englewood Cliffs, NJ (1991).
13. T. C. McFarland, ''X Windows on the World: Developing Internationalized Software with X, Motif and CDE,'' Prentice-Hall, Upper Saddle River, NJ (1996).
14. M. Grand, ''Java Language Reference,'' O'Reilly, Cambridge/Sebastopol, CA (1997).
15. A. Blum, ''ActiveX Web Progreamming: ISAPI, Controls, and Scripting,'' Wiley New York (1997).