SPICE based Circuit Analysis using Web Pages

Bogdan Wilamowski, Aleksander Malinowski, John Regnier
University of Wyoming / Bradley University / Micron Technology

Abstract

An application called the Spice Internet Package (SIP) has been developed for use through the Internet and intranet networks. The SIP provides an interface that is operating independent, which allows Spice simulation and analysis to be performed from any computer that has a web browser on the Internet or intranet. One can access the Spice Internet Package and try some examples with the password "**sip**" at the following URL: http://atlantis.uwyo.edu/sip for UNIX server and http://nn.uwyo.edu/sip/ or http://sant.bradley.edu/sip/ for Windows NT server.

The presentation shows how to use the new opportunity created by Internet technologies for the efficient and platform independent usage of CAD tools. The presented SIP is just an example, but it shows the way in which the technology can be implemented. They also illustrate how significant the impact is of a particular data flow on the programming tools and approaches taken to implement the Internet access to server-installed software. The authors are convinced that such an approach is suitable for pay-per-use access and it may revolutionize the general approach toward the development of CAD tools.

I. Introduction

Several versions of the SIPCE program are frequently used in circuit and in electronic classes. The most popular was the PSPICE version by MicroSim, which in the student version has its limitation to 10 transistor circuits. Recently, the SPICE program associated with Electronic Workbench is gaining attention. Although much larger circuits can be analyzed with electronic Workbench the program is designed to be used at university laboratories, and it is not affordable for students. Also relatively simple transistor models make this program not very useful for the assignments that deal with integrated circuits.

With the increase of Internet bandwidth the World Wide Web (WWW) could revolutionize design processes by ushering in an area of pay-per-use tools. With this approach very sophisticated design tools will become accessible for engineers in large and small businesses and for educational and research processes in academia. Currently, such sophisticated design systems are available only for specialized companies with large financial resources. The pay-per-use approach will have tremendous impact on engineering design since the number of engineers and researchers which have an access to sophisticated design tools will increase by a factor of at least of 100. This rapid increase in the number of people involved in sophisticated design processes will significantly accelerate technological development.
The common problem being faced by many electronic engineers in industry is that their design tools often operate on several different platforms such as UNIX, DOS, Windows 95, Windows

NT, or on Macintosh. Another limitation is that the required design software must be installed and a license purchased for each computer where software is used. Only one user interface handled by a network browser would be required [1]. Furthermore instead of purchasing the software license for each computer EDA tools can be used on a pay per use basis. Similar trend exists in electronic publishing where the reader pays per access to the material instead of purchasing journals, books, or abstracts.

II. Most Commonly Used Network Programming Tools

Although it is possible to develop network applications using C++ or other compiled languages the approach presented here is far more efficient. It is much easier to develop networked applications using dedicated software, and there are several very well developed network-programming tools available today [2]. These tools include HTML, JavaScript, VBScript, Java, ActiveX, CGI, and PERL [3-8]. During software development it is important to justify which part of the software should run on the client machine and which part should run on the server. Applets are transferred through a network when requested and execution is performed entirely on the client machine that made the request. In CGI much less information has to be passed to the server and the server executes instructions based on the given information and sends the results back to the local machine that made the request. Fig. 1 shows the program component division and data flow in a network based application.
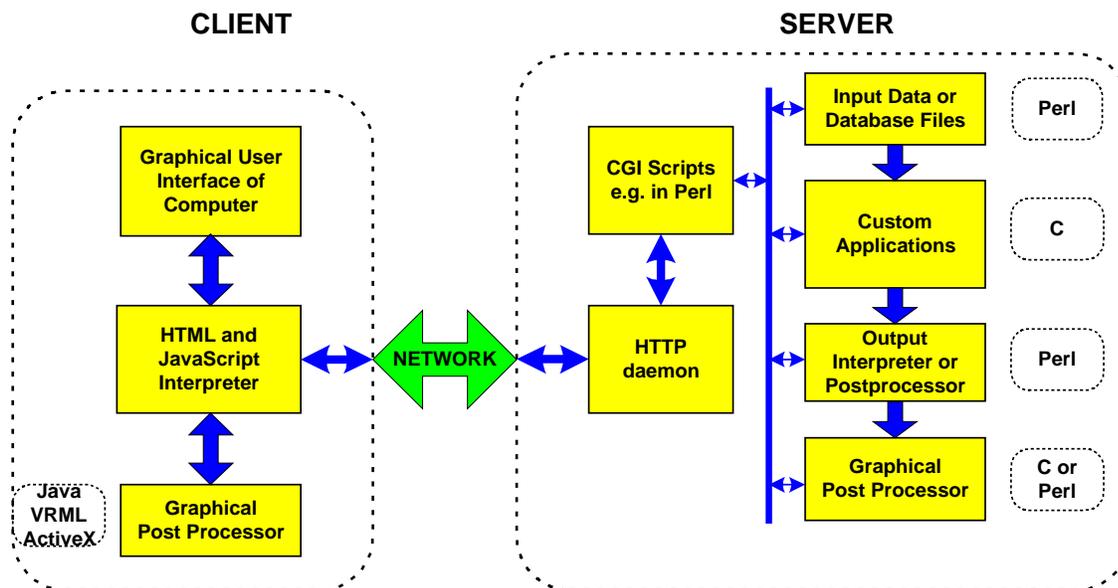

Fig. 1. Data flow in a network-based application.

Network programming uses distributed resources. Part of the computation is done on the server and another part on a client machine. Certain information must be frequently sent both ways between client and server. It would be nice to follow the JAVA applet concept and have most of the computation done on Client machine. This approach, however, is not visible for three major reasons:

- EDA programs are usually very large and thus not practical to be sent entirely via network as applets.

Software developers are giving away their software without the ability of controlling its usage.
 JAVA applets used on-line and on demand are slower than regular software.

The Internet bandwidth is already adequate for many EDA applications if their data flow is carefully designed. Furthermore the bandwidth limitation will significantly improve with time. The key issue is to solve problems associated with a new way of software development so that EDA will be possible through the Internet and intranets. It is therefore important to develop methods, which take an advantage of networks and then platform independent browsers. This would require solving several issues such as:

 Minimization of the amount of data which must be sent by a network
 Task partitioning between the server and client
 Selection of programming tools used for various tasks
 Development of special user interfaces
 Use of multiple servers distributed around world and job sharing among them
 Security and account handling
 Portability of software used on servers and clients
 Distributing and installing network packages on several servers
 Other

For example, should graphics be generated on the server and sent to a client as a GIF or JPG file, or should only text and binary data be sent to the client and a Java applet used to generate graphics there. Both approaches have advantages and disadvantages. The data traffic between the server and client in the first case is bi-directional with little data sent as requests from client to server, and much more sent back to the client as images created on the fly after receiving the request. In the latter case all data is transferred to a client machine together with a Java applet. The job of this applet is to process user's requests of data visualization without further transmissions from the server. Should the software developer choose the latter case, there would usually be large overhead in data transfer before data could be displayed.

To be successful two conditions must be met:

 Internet should have a proper bandwidth
 EDA (Electronic Design Automation) tools must be developed differently than software operating on a single computer or a uniplatform network.

The authors have already developed several network programs. One of them is the Spice Internet Package (SIP). Another example is the Internet Compilers. Both software packages will be described in the following sections.

III. Spice Internet Package

The Spice implementation used in this presentation is just one example of a networked application. Several methods of computer network programming are available including Java and CGI. An application called the Spice Internet Package has been developed for use through Internet and intranet networks. The SIP provides an operating system independent interface, which allows Spice simulation and analysis to be performed from any computer that has a web browser on the Internet or an intranet. The SIP has a user-friendly GUI (Graphical User Interface) and features include password protection and user accounts, local or remote files for simulation, editing of circuit files while viewing simulation results, and analysis of simulated data in the form of images or formatted text.

In the case of the Spice Internet Package it only makes sense to use CGI for the Spice simulation because it would be impossible to send the Spice engine through the network every time it was requested and this would be extremely slow. Java technology could also be used for functions like generating and manipulating graphs and implementing the graphical user interface on the client side. The SIP program currently incorporates CGI, PERL, HTML, and JavaScript. A unique feature of the SIP versus other Spice simulators is that it is operating system independent. Anyone that has access to the Internet and a web browser, such as Netscape Navigator or MS Internet Explorer, can run a Spice simulation and view the results graphically from anywhere in the world using any operating system.

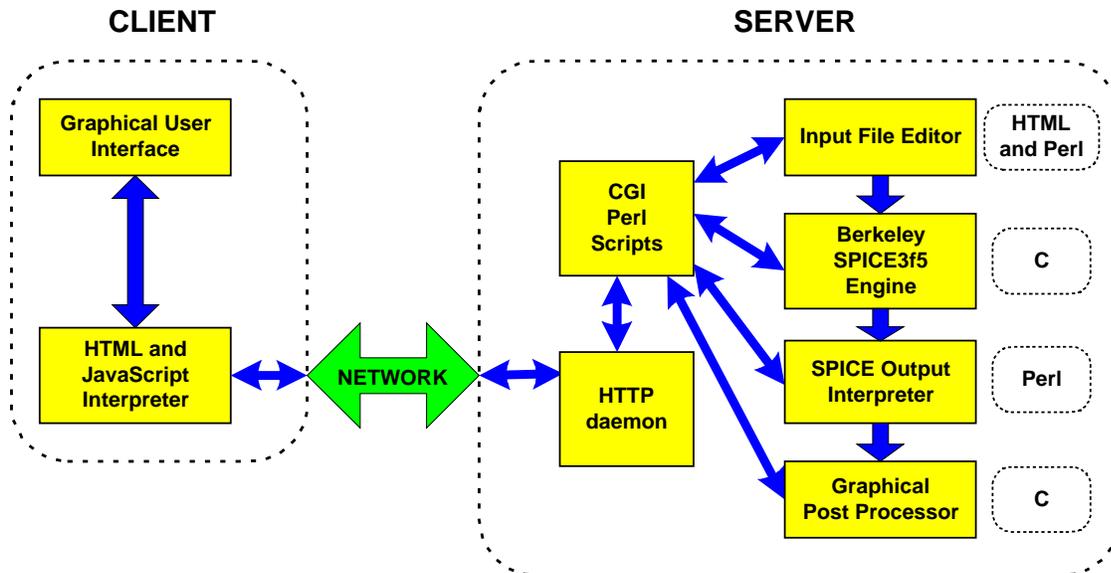**CLIENT**   **SERVER**

Fig. 2. Spice intranet package network model.

The server is configured to accept requests from web browsers through network connections. The server processes the request for Spice simulation or analysis and returns the results to the requesting web browser as an HTML document. The graphical analysis is embedded in the HTML as an image or returned as formatted text. Fig. 2 shows the flow of information and the distinction from the client and the server. The heart of the software is a server located PERL script. This script is executed first when the user logs in to SIP. Then each time the user selects any activity a new dialog box in the form of an on-the-fly generated JavaScript enhanced HTML web page is sent back. Such pages may contain a text editor, a simulation report, a graphic postprocessor menu, or graphic image of plotted results. To complete some tasks the PERL script may run additional programs installed only for the server such as: the main CAD program - Berkeley Spice, GnuPlot generates plots, and some utility programs (netpbmp) to convert plotter files into standard images recognized by all graphical web browsers.

SIP is a very good example for network traffic considerations. The amount of data produced by a single simulation may differ significantly from a few hundred bytes to a few hundred KB depending on the number of simulation steps requested. In case of large data files it is better to generate graphical images of plots and send them to the user. Users frequently inspect the obtained results a few times for example by changing the range or variables to display. In case when there may be many requests for different plots of the same data, it could be better to send

the data once together with a custom Java applet which could display the same information in many different forms without further communicating with the server.

The current versions of SIP are written both for the Unix and the Windows NT operating systems. The two versions differ only slightly because of the operating system independence of the programming languages being used. Some of the SIP features include:
- Simulation and analysis of Spice files stored on the server or on your local machine.
- Graphing of DC/AC/Transient analysis data from a remote or local file. The data used for graphing is generated with the Spice ".PRINT <AC|DC|TRAN> ..." command.
- Customizing the graphical analysis including zoom and scale.
- One copy of the Spice engine runs on a server and many users can access the SIP program simultaneously.
- Password protection and separate file areas for each user password.
- Editing of personal input and output files stored on the server.
- Multiple windows open at a time to allow viewing or editing of circuit files while viewing simulation results.
- Analysis output can be specified as a GIF image, or raw text containing the data points.

The SIP graphical user interface for the version implemented on a UNIX server is shown in Fig. 3. At the beginning it is necessary to select whether the server or local location should be used to store files. This could be done by clicking either at the **REMOTE** or **LOCAL** radio button. When remote (server) storage is used files can be selected from the list of files existing in the user's directory or a new file can be created. Then the file may be edited and simulation performed. In case of using local files a user has to save the simulation output file on the local machine and then enter that name in the **Local OUT Filename** edit box in order to plot the simulation results. As an example of the use of the SIP program, consider the following bipolar transistor amplifier example.

The Spice3 input file in Fig. 4 describes the amplifier circuit in terms of the Spice3 language. The Spice2 and Spice3 input code significantly differs from other popular Spice versions such as PSPICE for example. All differences between various Spice programs are well described in [9]. Notice in the Spice code there are commands for the Spice simulator to analyze the dc sweep, the ac analysis, and the transient response of the circuit and to print the output of V(1) and V(2).
To enter a Spice input file select the filename from the CIR drop down box or type the name into the **CIR Filename** edit box and then press the **VIEW/EDIT** button. After entering or modifying the input file press **SAVE CHANGES** in the edit window to save the changes to the file. After entering the input file and saving it, the simulation is then run by selecting the **RUN SIMULATION** button. The output from the simulation is displayed in the output window. If remote files were used the output is also saved to the file in the **Remote OUT Filename** edit box automatically. Next generate a graph of the output data by pressing the **PLOT DATA** button. A window will open allowing you to select what variables you want to plot and other customization variables as shown in Fig. 5. Select **PLOT DATA** again and the analysis is returned to the output window as shown in Fig. 6. To close all the windows and return to the original web browser window press the **EXIT SIP** button. There is also a **HELP** button describing how to use the SIP program. It is also possible to save the output data in a space delimited ASCII file so a high quality graph can be generated by MS Excel or other plotting packages.
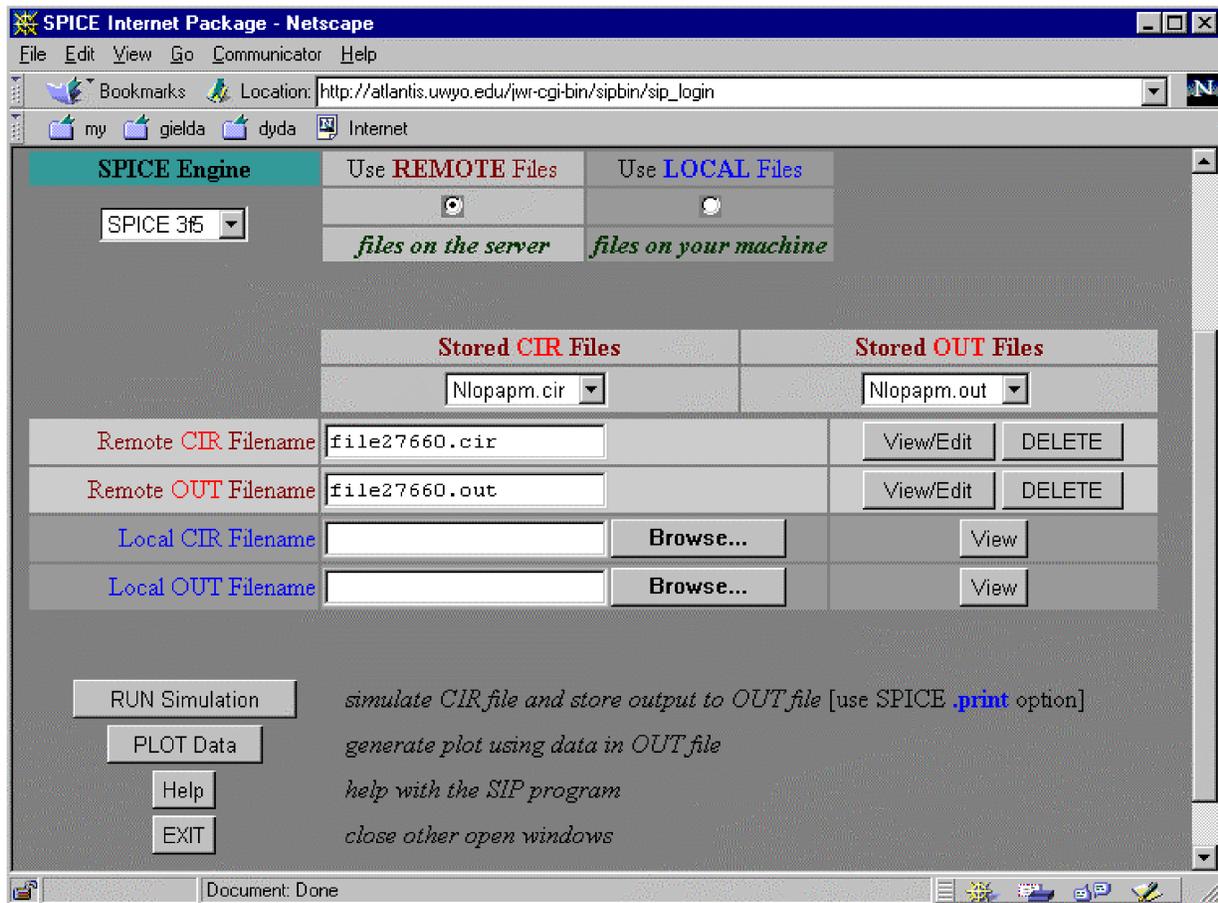
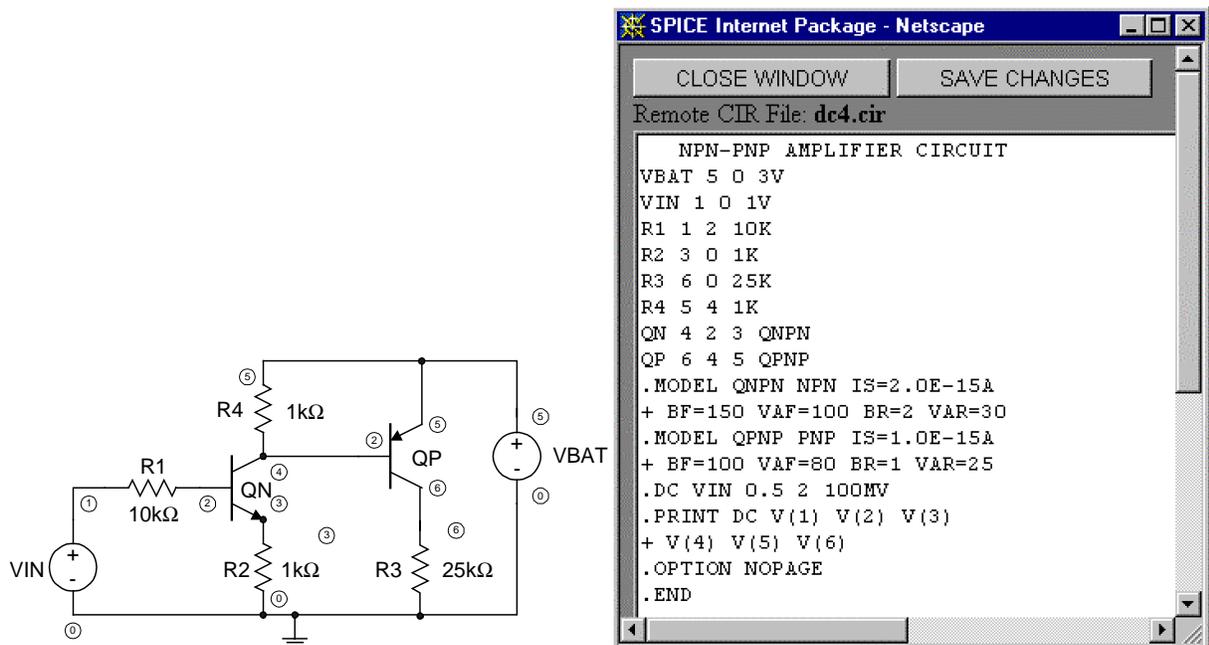Fig. 3. Graphical user interface for SIP package.

Fig. 4. Simple bipolar transistor amplifier: circuit diagram and editing window with Spice input file.

Several features make the Spice Internet Package a desirable program for computer-aided engineering and design. Only one copy of the Spice engine needs to be installed and configured. One machine acts as the server and other machines can simultaneously access the Spice engine through network connections. Remote access to SIP allows users to run Spice simulations from any computer on the network, and that might be from home or another office in another building or town. Also the current Spice engine being used is Spice3f5 from Berkeley [10, 11], which allows an unlimited number of transistors, unlike various "student versions" of Spice programs that are available.
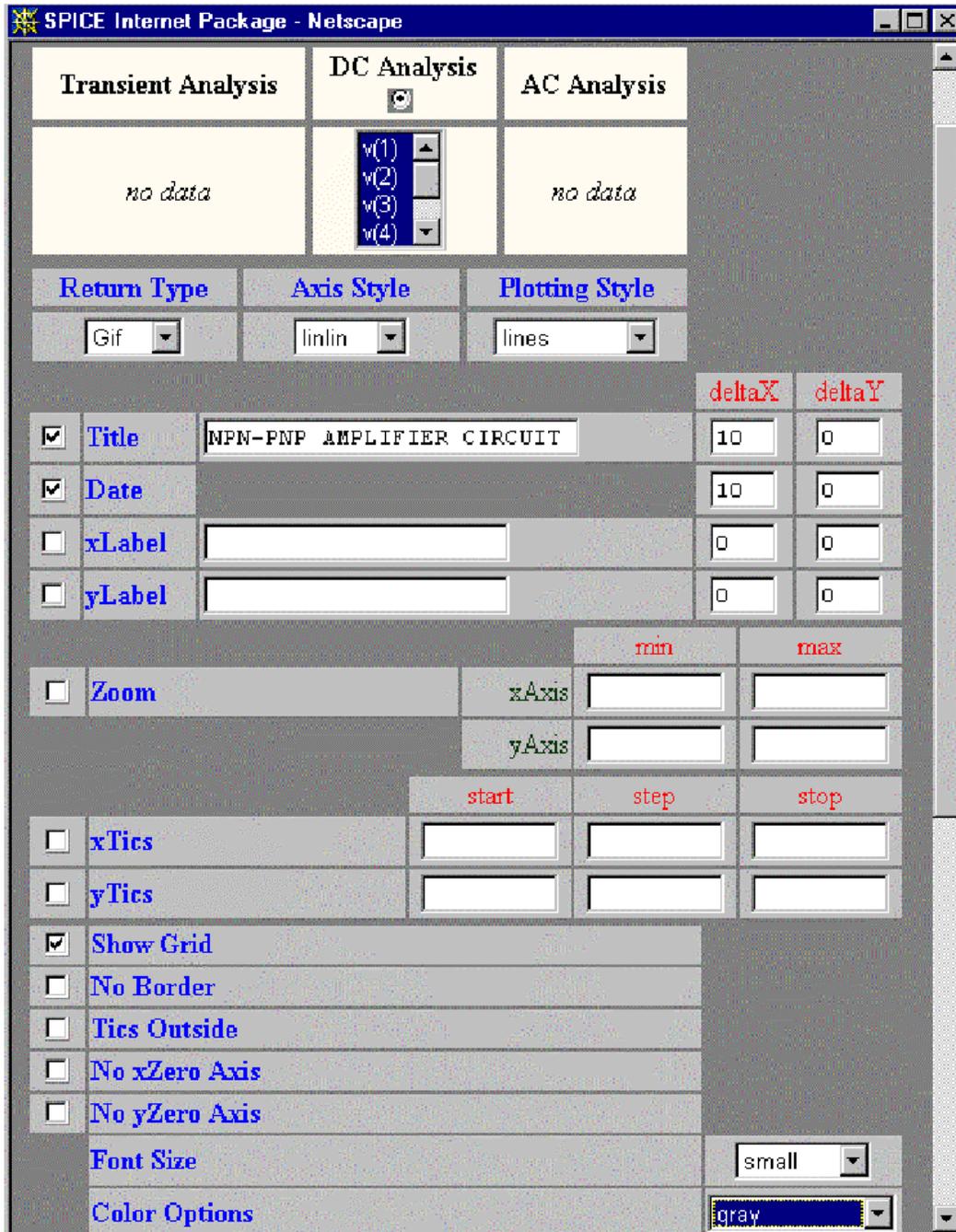

Fig. 5. Upper part of the plotting configuration screen.

User passwords are required because the SIP program is accessible from around the world and a heavy number of users would slow the server down noticeably. For undetermined time you can access the Spice Internet Package and try some examples with the password "**sip**" at the following URL:

- http://atlantis.uwyo.edu/~regnier/sip - UNIX server
- http://nn.uwyo.edu/sip/ or http://sant.bradley.edu/sip/ - Windows NT server

The Spice Internet Package has been successfully used in several classes at the University of Wyoming [12]. Students appreciate a convenience of using it on any computer platform at the University or at home. It was designed originally for the intranet networks, however it works also very well on the Internet.
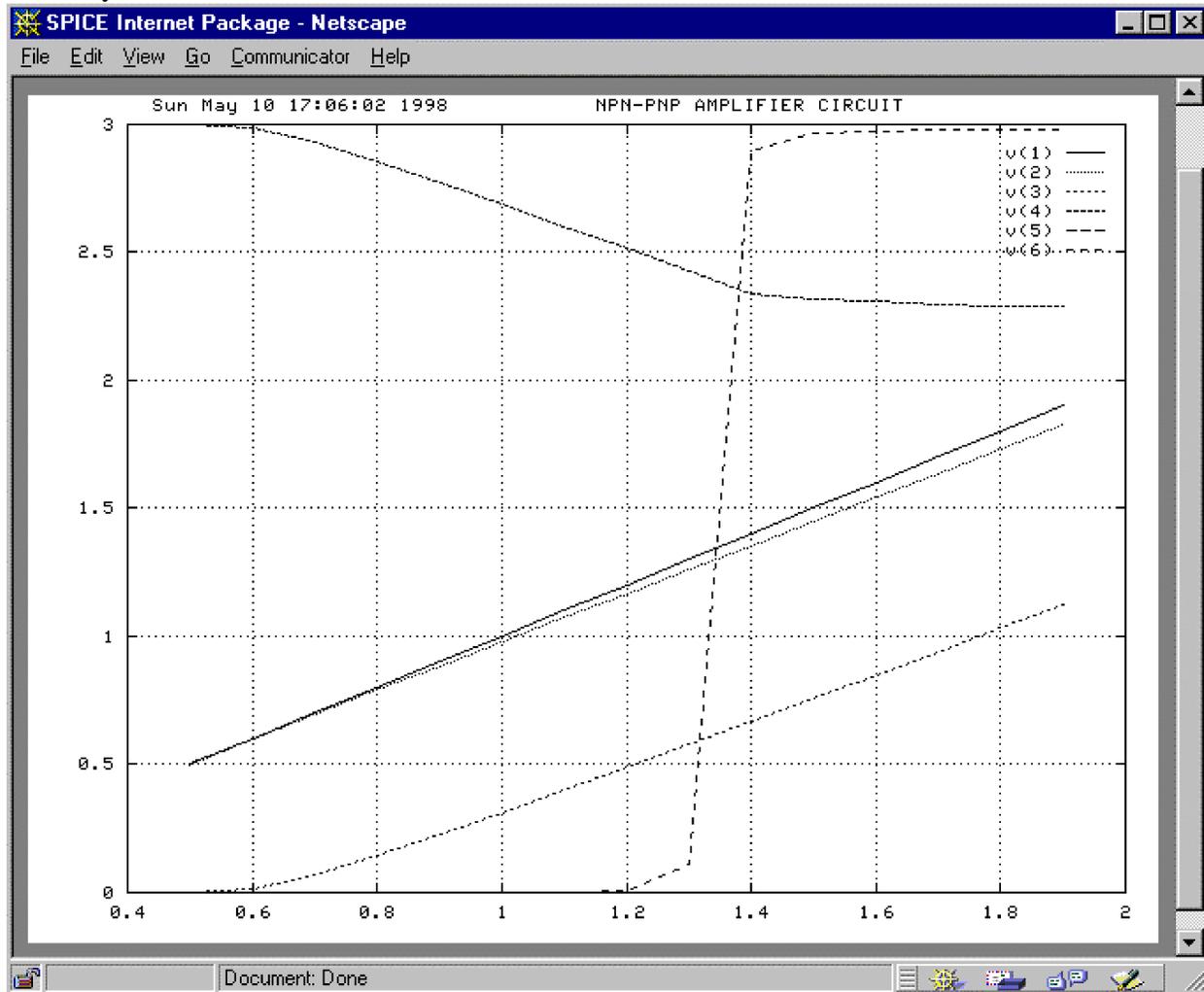

Fig. 6. Results of a transient analysis of the amplifier.

IV. Conclusion

The paper shows how to use the new opportunity created by Internet technologies for efficient and platform independent usage of CAD tools. The presented SIP is just examples but it show a way in which the technology can be implemented. They also illustrate how significant is the impact of the particular data flow on the programming tools and approaches taken to implement the Internet access to the server installed software. The authors are convinced that such approach

will spread and it will revolutionize the general approach toward CAD tool development.

This approach will also have a synergetic effect on the development of better design tools since the market will increase significantly. Availability of design tools via the Internet will boost design process in many new communities and improve our education processes at universities by allowing students to use the same sophisticated software as is used by leading industries. For example, designing of customized VLSI chips will be possible for small business and this will have a synergetic effect on the entire industry.

Several features make the Spice Internet Package a desirable program for computer-aided engineering and design. Only one copy of the Spice engine needs to be installed and configured. One machine acts as the server and other machines can simultaneously access the Spice engine through network connections. Remote access to SIP allows users to run Spice simulations from any computer on the network. This may be from home, or another office in another building, or town. Also, the current Spice engine in use is Spice3f5 from Berkeley, which allows an unlimited number of transistors, unlike various "student versions" of Spice programs that are available.

The Spice Internet Package has been successfully used in several classes at the University of Wyoming. Students appreciate the convenience of using it on any computer platform at the University or at home. It was designed originally for Intranet networks, yet it works very well on the Internet.

Bibliography

1. Sweet W., Geppert L., "http:// It has changed everything, especially our engineering thinking", IEEE Spectrum, January 1997, pp. 23-37.

2. Jamsa K., Lalani S., Weakley S., Web Programming, Jamsa Press, Las Vegas, NV, 1996.

3. Flanagan D., JavaScript, The Definitive Guide, O'Reilly & Associates, Sebastopol, CA, 1997.

4. Hank Shiffman, Making Sense of Java, http://www.disordered.org/Java-QA.html

5. Hank Shiffman, Boosting Java Performance: Native Code and JIT Compilers, http://www.disordered.org/Java-JIT.html

6. Mall L., T. Christiansen, and R. L. Schwartz, Programming PERL, O'Reilly & Associates, 2nd edition 1996.

7. Object management Group, The Common Object Request Broker: Architecture and Specification, v. 2.2, published by Object Management Group, February 1998. see also http://www.cobra.org/

8. CookieCentral.com, Cookie Central, URL: http://www.cookiecentral.com/

9. Wilamowski B. M. and Richard C. Jaeger, Computerized Circuit Analysis Using Spice Programs, McGraw-Hill, 1997.

10. Vladimirescu A., K. Zhang, A. R. Newton, D. O. Pederson, and A. SangiovanniVincentelli, SPICE Version 2G6 User Guide, Department of Electrical and Computer Science, University of California, Berkeley, CA, 1981.

11. Quarles T., A. R. Newton, D. O. Pederson, and A. SangiovanniVincentelli, SPICE3 Version 3F5 User's Manual, Department of Electrical and Computer Science, University of California, Berkeley, CA, 1994.

BOGDAN M. WILAMOWSKI
Bogdan M. Wilamowski (IEEE Fellow) is a Professor of Electrical Engineering at the University of Wyoming. He is the treasurer of IEEE Industrial Electronics Society, a member of the IEEE Neural Network Council, an associate editor of IEEE Trans. on Neural Networks and IEEE Trans. on Education. Dr. Wilamowski is the author of 4 textbooks, more than 200 refereed publications, and 27 patents. He received his MS in computer engineering in

1966, Ph.D. in neural computing in 1970, and D.Sc. in integrated circuit design in 1977, all from the Technical University of Gdansk, Poland.

ALEKSANDER MALINOWSKI
Aleksander Malinowski is an Assistant Professor at the Electrical and Computer Engineering Department of the Bradley University. He is also an associate editor of the IEEE Industrial Electronics Society Newsletter. He received his MS in electronics in 1989 from the Technical University of Gdansk, Poland and Ph.D. in computer science and engineering in 1996 from the University of Louisville, KY. Dr. Malinowski is the author of 4 journal papers and 31 other refereed publications. The areas of his main interests are Web programming, network computing, computer simulation and modeling, computational intelligence and computer architecture.

JOHN W. REGNIER
John W. Regnier received his BS at the University of Wyoming with Major in Electrical Engineering and with a Minor in Computer Science in 1997. He received MS in 1998 from Department of Electrical Engineering of the same University. He has been with Micron Technology, Inc. in Boise Idaho, as a CAD Engineer in Research and Development, since May of 1998