2010 24th IEEE International Conference on Advanced Information Networking and Applications

Design of network based software

Bogdan M. Wilamowski, *Fellow, IEEE*
Electrical and Computer Engineering, Auburn University, Alabama, US
wilam@ieee.org

Abstract:
With the increase of Internet bandwidth the World Wide Web is changing the approach for software development. Traditionally, most of the software was developed for one particular platform such as DOS, Windows, Mac, Unix or Linux and it was not portable from one system to another. Usually the user interface is system dependent and it has to be developed individually for every system. It is, however, possible to standardize web browsers to use interface and such interface can be moved easily from one platform to another. At the same time the software can run on the same computer or run remotely on a network computer. Interestingly, such software is very portable and can be easily recompiled on various systems. Several examples of such network based software were developed in order to test the concept such as: SIP - Spice Internet Package which let users to run the unlimited version of SPICE program over internet, ICP – Internet Compilers Package where many compilers were implemented for various computer languages such as C, C++, Fortran, Pascal and JAVA; and Neural network simulation tool running over the network are demonstrated.

I. INTRODUCTION

The increase of Internet bandwidth on the World Wide Web may have a significant impact on the computer technologies. Google is offering a new operation system Chrome dedicated for network operation and Android for iPhones and other network gadgets. The Chrome OS is for portable computers and should be available for individual users in the second part of 2010. In Chrome the information, instead of being stored on local hard drive, it will be preserved on large network servers. Google is already providing Gmail service where each user is receiving over 7GB of free storage space on their servers. In Chrome the software will not be stored on individual computers but it it will be stored on remote servers. It is expected that notebooks with Chrome will be resistant to viruses because software on network servers can be only read by internet users and cannot be modified by internet hackers. Email will be based on Gmail and text edition and spreadsheet calculation will be using Google Docs. Other applications in Chrome will be Google Calendar, Picasa Web for photos, Last.fm for music and Pandora may replace Skype. Chrome will be able to boot up in several seconds, but notebooks can run only with presence of the connection to Internet. The Chrome operation system will be available for free. Many people who are using Gmail need not to store anything on local hard disk and also more and more people are storing their personal files on the network using software like Gmail, Yahoo, YouTube, Facebook, and Twitter. Software on the network can be automatically updated and there is no need for backup or for scanning for viruses. Google is already offering a Chrome web browser, which indeed works with lightning speed.

Computer networks may have a significant impact also on the design process by using 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, who have an access to sophisticated design tools, will increase significantly [1][2]. This rapid increase in the number of people involved in sophisticated design processes may have significantly accelerated technological development. Currently, most of the software is physically delivered to the user. It is password protected but it is seldom effective in protection usage usage by unauthorized users. In case of usage of the software over computer networks the user will never see the original software but will only be able to use it over computer network. Users may only send the specification data over network and would receive results of his design also over the network. With this approach actual software will not be transmitted over the network and in most cases actual amount of data transmitted through network would not be very significant. Another limitation is that EDA software must be licensed for each computer where software is installed. Sometimes, very expensive licenses must be purchased no matter if the software is used very extensively, or very rarely. With the presented approach, only one user interface handled by a network browser would be required. Furthermore, instead of purchasing a software license for each computer, EDA tools can be used on a pay-per-use basis [1]. A similar trend exists in electronic publishing, where readers are asked to pay per access to the material, instead of purchasing journals, books, or abstracts, and IEEE is a driving force there.

By moving the design platform to one common user interface may save a lot of money for software companies as well. The common problem faced by many electronic engineers in industry is that their design tools often work only on one or two operating platforms such as UNIX, DOS, Windows 95, Windows NT, or Macintosh. With the network approach, only one user interface handled by a network browser would be required. Availability of design tools via 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.

1550-445X/10 $26.00 © 2010 IEEE
DOI 10.1109/AINA.2010.181

The internet bandwidth is already adequate for many of such tools and it will significantly improve with time. Several issues must be solved such as: (a) how to minimize the amount of data which must be sent by a network, (b) what task should be performed on the server and what task should be done on the client, (c) what programming platforms should be used for various tasks, (d) the problem of multiple servers distributed around world and issues of job sharing between servers must be solved, (e) several issues of security and account handling must be addressed, (f) portability issues of software on servers and clients is very important, (g) there is a problem of distributing and installing network packages on several servers, and others. Several network-programming tools, which are available today, include Java, CGI, ActiveX, JavaScript, VBScript, HTML, PHP, and PERL. 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. The CGI is quite different from writing Java applets in this aspect. 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.

This presentation demonstrates the capability of the use of Internet and intranet networks for software engineering, computer-aided design, and education. A networked application can be used remotely through the network and any operating system can be used to access a networked application, making it operating system independent. Several network base design tools will be presented in next sections. One of the main tools used by electrical engineers is the SPICE program, which is usually platform dependent. The Spice Internet Package (SIP) was developed using an Internet browser as a platform-independent graphics user interface [3][4][5]. This particular application allows users to run SPICE simulations and view graphical analysis of electronic circuits through a network connection using a Web browser. The SIP application has many options that include simulation of Spice files, graphical postprocessing data, online editing of Spice files, password protection with separate file areas for each user, and a user friendly graphical user interface. Another example developed, is the Internet Compilers Package (ICP) [6][7][8]. Different compilers can be used for various stages of application development providing either maximum error detection, or code optimization. The most recently implemented on the network design tool is Neural Network Trainer [9] which is able to train neural networks with arbitrarily connected neurons and its training speed is often 1000 times faster than when traditional Error Back propagation algorithm is used.

II. IMPLEMENTATION ISSUES

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 network when requested and execution is performed entirely on the client machine that made the request. With CGI much less information has to be passed to the server. The server executes instructions based on the given information and sends the results back to the local machine that made the request. Network programming uses distributed resources. Part of the computation is done on the server and another part on the 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 computations done on the client machine. This approach, however, is not feasible for three major reasons:

- Many programs are very large and thus not practical for sending 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 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 the 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

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 the 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 the client machine together with a Java applet. The job of this applet is to process the user's requests of data visualization without further transmissions from the server. Should the software developer choose the latter case, there would usually be a large overhead in data transfer before data could be displayed.

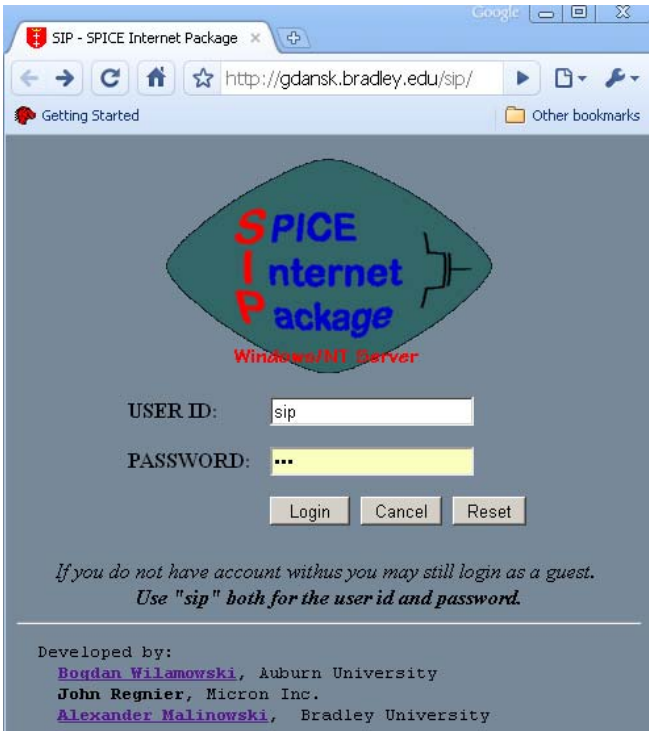


Fig. 1. Entry page to SIP

III SIP - - SPICE INTERNET PACKAGE

An application called the Spice Internet Package (SIP) has been developed for use through Internet and Intranet networks [3][4][5]. SPICE program [10][11][12] is known to most electrical engineers and compilers be of interest of computer scientists and computer engineers. The SIP provides to 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 intranet (see Fig 1).

In the case of the Spice software it only makes sense to use CGI for the Spice simulation, because it would be impossible to use applet technology and send the Spice engine through the network every time it was requested. This would be extremely slow. 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, can run a Spice simulation and view the results graphically from anywhere in the world using any operating system.

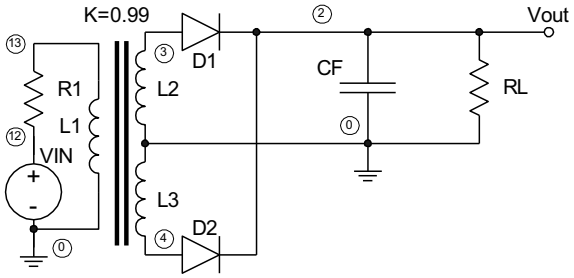


Fig 2. Analyzed circuit

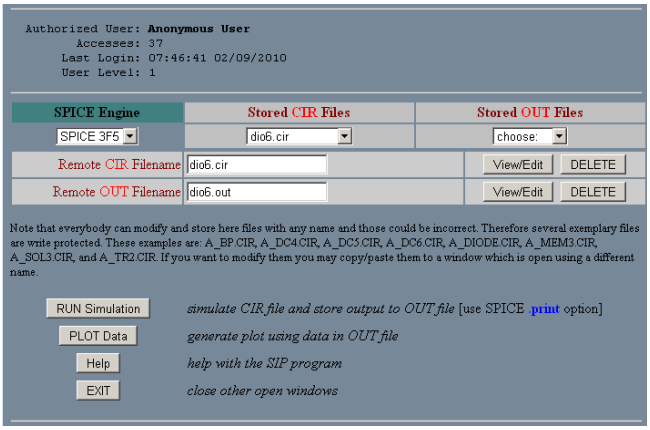


Fig 3. The main screen for SIP

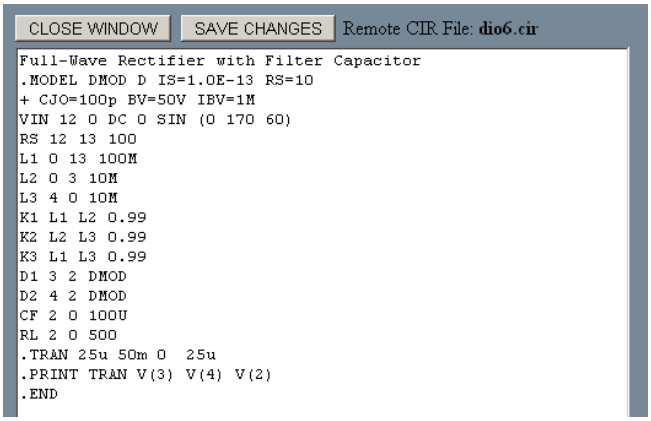


Fig 4. Editing of the netlist

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. The heart of the software is a server located PERL script. This script is executed first when the user logs in to SIP (Fig. 1).

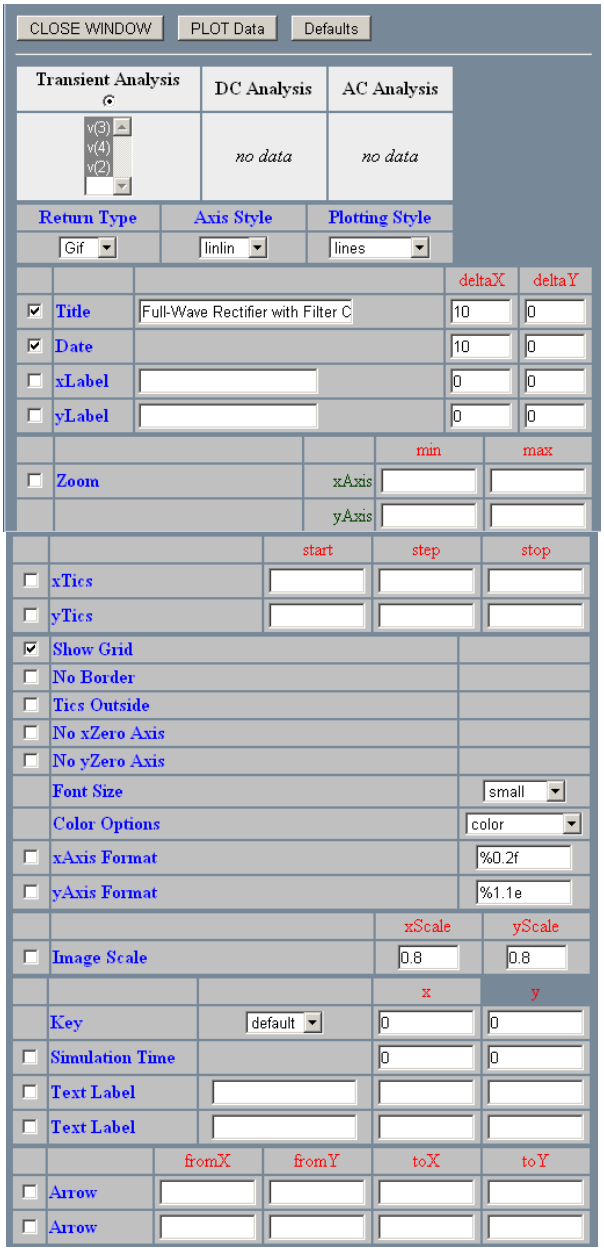


Fig. 5. Configuration of plots

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 (Figures 3 through 6) through may contain a text editor (Fig 4), a simulation report, a graphic postprocessor menu (Fig. 5), or a graphic image of plotted results (Fig 6). To complete some tasks the PERL script may run additional programs installed only for the server such as the main CAD program - the Spice engine does number crunching, Gnuplot generates plots, and a utility program like NETPBMP converts plotter files into standard images recognized by all graphical web browsers.

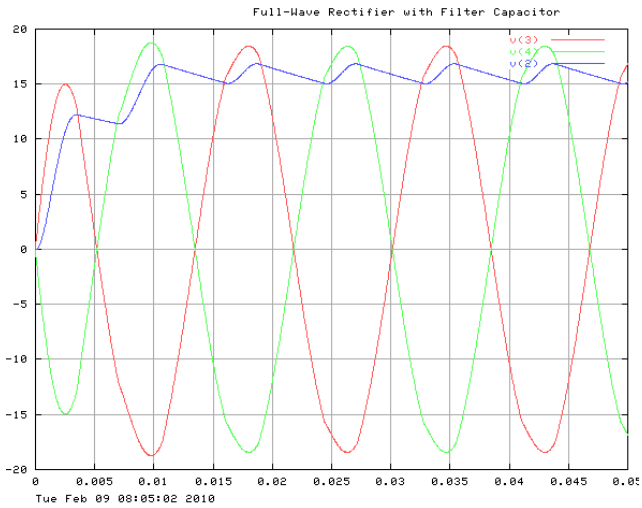


Fig. 6 Plotting results

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 by changing, for example, the range or variables to display. In case when there may be many requests for different plots of the same data, it would 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.

SIP can be used for both Unix and Windows NT operating systems as servers, and could be run on any computer which has access to web pages. 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 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 which 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 Spice2 and Spice3 input code significantly differ from other popular Spice versions such as PSPICE for

example. All differences between various Spice programs are well described in [12].

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.

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.

IV ICP INTRANET COMPILERS PACKAGE

Frequently, during the process of software development, more than one compiler package is required. Some products are known to be very useful for locating errors or debugging, while others perform extremely well when a program or library is in the final stage of development and should be optimized as much as possible. When facing obscure error messages, which may result in a time-consuming search for the error, a different error message from the second compiler frequently cuts that time dramatically. The user interface of ICP [6][7][8] is programmed in HTML enhanced with JavaScript. A PERL script handles the data flow, which is shown in Fig. 7. That script does the file managing, runs compilers and processes the compilation results. The result is both the source code listing and a binary code to download or a list of errors sent back to the user.

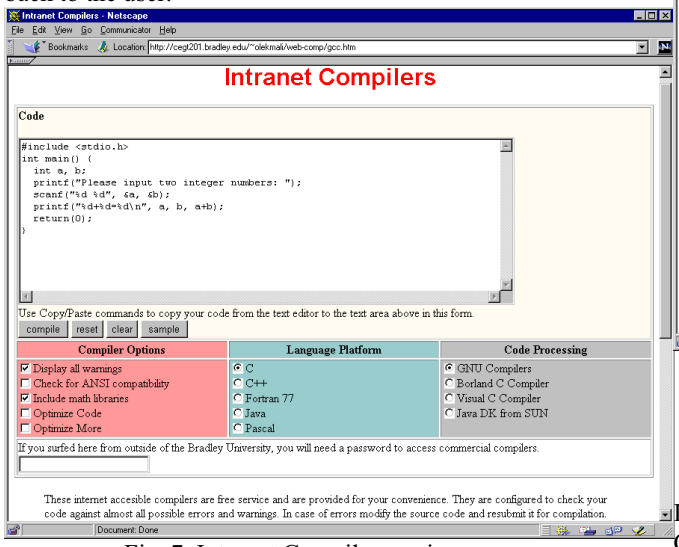
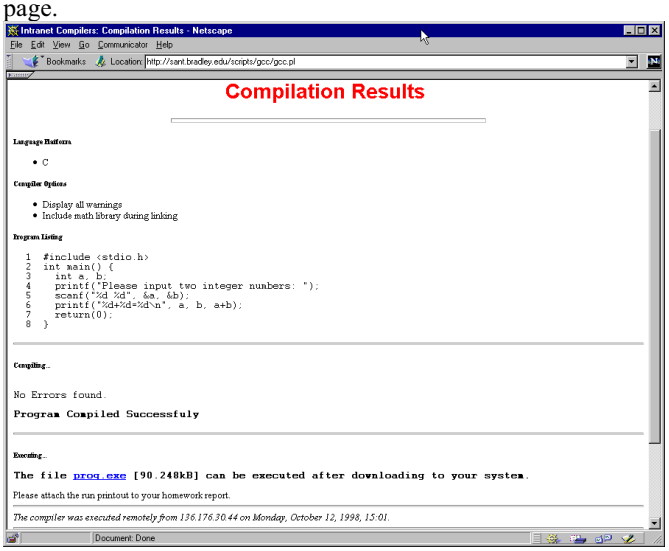
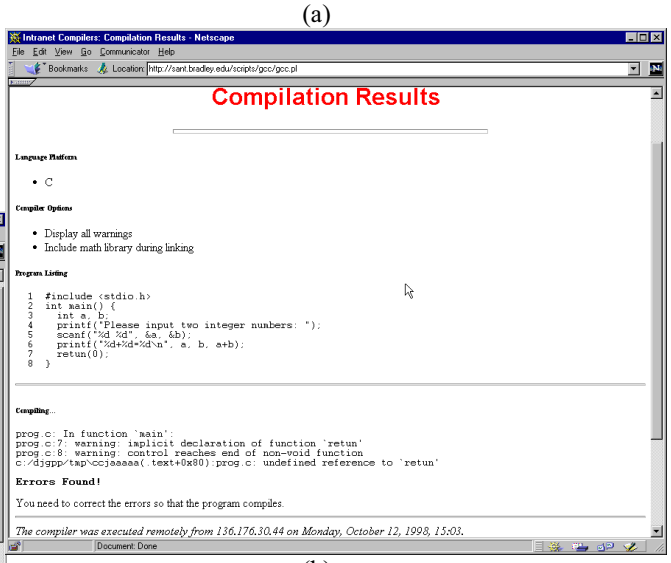


Fig. 7. Intranet Compilers main page.

To use ICP, paste the program code from your compiler text editor, or from any text editor, to the web page form. Then submit the form. The compilation will be performed by PERL script on the server in batch mode. Although the front end is designed to be as simple as possible with only a few commonly used options, it is sufficiently functional and can be used quickly. The PERL script located on the server has to deal with the translation of these common options to the actual options of compilers from different vendors. It also handles the compilation errors and processes the report. The cookie technology, together with JavaScript, is used to store each user's preferences for compilation so that the settings are preserved when the user returns to the ICP web page.



(a)



(b)

Fig. 8. Intranet Compilers - successful compilation (a) and compilation errors (b)

ICP package supports C, C++, Pascal, Fortran, and Java. It utilizes GNU, Borland, and Microsoft compilers for C and C++, GNU compiler for Fortran and Pascal, and Sun's JDK for Java. A common front end is used for all compilers (Fig. 7). This HTML page allows for selecting a vendor and a language, and for setting a few basic compilation options. The process of compilation is performed in batch mode. After setting the desired options and pasting the source code into the appropriate text box, the task can be started by pressing the ***COMPILE*** button. As a result another web

page with HTML wrapped information is sent back to the user as shown in Fig. 8. It is displayed in another browser window so that the user can correct the source code in the original window and resubmit it if necessary.

V NNT - NEURAL NETWORK TRAINER

The trainer NBN 2.0 was written in C++ to assure high speed operation. Then the C++ executable was called from PHP scripts. The communication between PHP script and C++ executable were secured by writing input/output data to temporary files on the server. The entry page written in HTML is shown on Fig. l

Fig. 9 Entry page for the Neural Network Trainer.

Everyone should btheir e able to create their own username and password and have own account on the server. This is important because this way the user preference such as working directories, preferred algorithms, etc are stored on the server and next time user has an easy access.

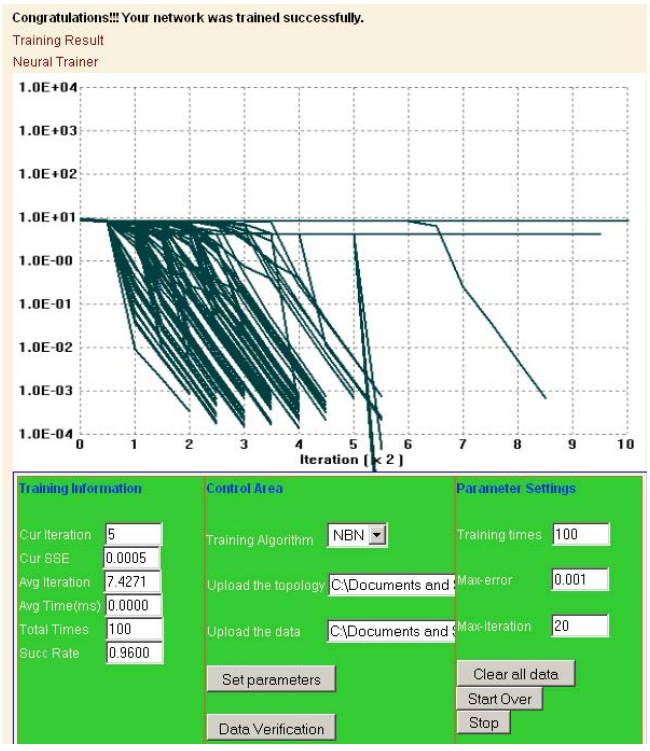


Fig 10. Parity-3 problem trained 100 times with NBN – Neuron by Neuron training algorithm

Parameters
NBN mu = 0.01000000 scale = 10.00000000
Data File: parity3.in
Topology
4 1 2 3
5 1 2 3 4
Neurons
biplor gain=0.50, der=0.01
biplor gain=0.50, der=0.01
Initial Weights
-0.78000000 -0.96000000 0.86000000 0.06000000
-0.70000000 -1.00000000 -0.66000000 -1.00000000 -0.08000000
Results Weights
0.69755431 -9.23717699 9.07814653 -9.60620262
0.09125257 -10.59999962 9.65821652 -9.14054200 -18.83804431
Training Results
Total iteration: 8 Total error: 0.00772545

Fig. 11. Training data of the Parity-3 problem

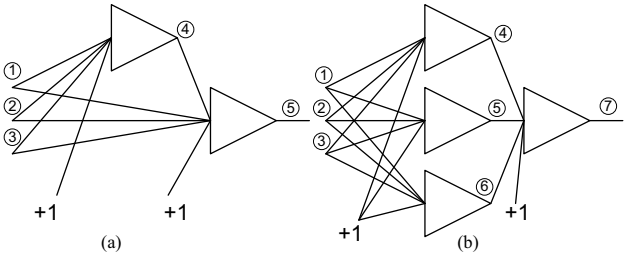


Fig. 12. Neural networks for Parity-3 problem: (a) ACN topology and (b) MLP topology

Fig. 10 shows the training results for the Parity-3 problem which was trained 100 times with randomly selected starting weights. The new NBN algorithm is very fast 96 of 100 training were successful and all job was completed in almost no time. When the same problem was trained with popular EBP – Error Back Propagation algorithm the learning process was 10,000 times longer.

Sample training parameters are shown in Fig 11. In contrast to most of existing neural network training software, which train only layered architectures (such as MLP) the NNT package can train neural networks with arbitrarily connected neurons ACN. These neural networks with connections across layers are much more powerful than popular MLP network [13]. For example the Parity-3 problems with ACN topology (Fig 12 (a)) needs only 2 neurons, while the MLP topology require 4 neurons (Fig 12 (b)).

Any topology can be entered to the network so the software is not limited only to layer-by-layer type of architectures. As one can see from Fig. 11 the topology of the Fig 12(a) is entered with two lines:

4 1 2 3
5 1 2 3 4 5

It means that the first neuron with output node 4 is connected to inputs (nodes) 1, 2, 3 and second neuron with output node 5 is connected to inputs 1, 2, 3, 4. In similar way the topology for the network of the Fig. 12 (b) can be written as:

4 1 2 3
5 1 2 3
6 1 2 3
7 4 5 6

Neural network trainer can be used remotely through any network connection or any operating system can be used to access, making the application operating system independent. Also much less installation time and configuration time is required because the training tool locates only on central machine. Many users can access at the same time. Users can train and see training results directly through networks.

VI CONCLUSION

It is possible to standardize web browsers to use interface and such interface can be moved easily from one platform to another. At the same time the software can run on the same computer or run remotely on a network computer. Interestingly such software is very portable and can be easily recompiled on various systems. Such an approach has several advantages:

(1) Fully portable user interface
(2) Easy recompilation for different platforms
(3) Superb pirate protection (company need not to give user the code)
(4) Possible pay-per-use (small companies need not to pay for giant software if it is not used often)

Software can run over different programming platforms and it is also important that software developers can protect their intellectual property when network is used as user interfaces. Several examples of such network based software were developed in order to test the concept such as: SIP - Spice Internet Package [5] ICP – Intranet Compilers Package [7] Neural network simulation tool running over the network [9] was presented. But also other applications can be found in the literature such as: Manuscript collection and evaluation; GradeWatch which is a system for interactive presentation of students' grades [[14]] Internet controlled robots [15], Paper collection and review [16], Internet robots for the literature search [17][18], etc.

REFERENCES

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

[2] A. Malinowski and B. M. Wilamowski, " Internet Technology as a Tool for Solving Engineering Problems ", *The 27th Annual Conference of the IEEE Industrial Electronics Society* (tutorial) pp. 1622-1630, Denver CO, Nov 29-Dec 2, 2001.

[3] B. M. Wilamowski B., R. Regnier R., A. Malinowski "SIP - Spice Internet Package", *Proc. of the IEEE International Symposium on Industrial Electronics (ISIE'98),* Pretoria, South Africa, July 7-10, 1998, pp. 493-496.

[4] Regnier J. W. and B. M. Wilamowski, " SPICE Simulation and Analysis through Internet and Intranet Networks" *IEEE Circuit and Devices Magazine,* May 1998, pp. 9-12

[5] B.M. Wilamowski, A. Malinowski, and J. Regnier, "Internet as a New Graphical User Interface for the SPICE Circuit Simulator", *IEEE Transactions on Industrial Electronics*, vol. 48. No. 6, pp. 1266 –1268, Dec. 2001

[6] [A. Malinowski and B. M. Wilamowski, "Internet Accessible Compilers in Software and Computer Engineering" *International Conference on Simulation and Multimedia in Engineering Education (ICSEE'99)*, San Francisco, CA, January 14-15, 1999, pp. 221-224

[7] A. Malinowski and B.M. Wilamowski, "Compiling Computer Programs Through Internet", ITHET-2000 - International Conference on Information Technology Based Higher Education and Training, Istanbul, Turkey, July 3-5, 2000, pp. 343-348

[8] A. Malinowski and B. M. Wilamowski, "Web-based C++ compilers", *ASEE 2000 Annual Conference, St. Louis, MO*, CD-ROM session 2532, June 18 to 2, 2000.

[9] Nam Pham, Hao Yu, B. M. Wilamowski, Neural Network Trainer through Computer Networks, Aina,10, Perth, Australia, April 20-23, 2010.

[10] A.Vladimirescu, 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] T. Quarles, 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.

[12] B. M. Wilamowski and R.C. Jaeger, *Computerized Circuit Analysis Using Spice Programs*, McGraw-Hill, 1997.

[13] B. M. Wilamowski, " Neural Network Architectures and Learning algorithms – How Not to Be Frustrated with Neural Networks", *IEEE Industrial Electronics Magazine,* vol 3, no 4, pp.56-63, (2009)

[14] B. M. Wilamowski and A. Malinowski, "GradeWatch - the software package displaying on web pages students grades", *ASEE 2000 Annual Conference, St. Louis, MO*, CD-ROM session 2793, June 18 to 2, 2000.

[15] A. Malinowski and B. M. Wilamowski, "Controlling Robots via Internet" *Proc. of the 1st International Conference on Information Technology in Mechatronics*, pp. 101-107, Istanbul, Turkey, October 1-3, 2001.

[16] A. Malinowski and B. M. Wilamowski, " Paper Collection and Evaluation Through the Internet", *Proc. of the 27th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1868-1873, Denver CO, Nov 29-Dec 2, 2001.

[17] S. Neeli, K. Govindasamy, B. M. Wilamowski, and A. Malinowski,"Automated Data Mining from Web Servers Using Perl Script" *12th INES 2008 -International Conference on Intelligent Engineering Systems,* Miami, Florida, USA, February 25-29, 2008, pp. 191-196

[18] Hung Nguyen and B. M. Wilamowski "IEEE Article Data Extraction from Internet" accepted for *13-th IEEE Intelligent Engineering Systems Conference, INES 2009,* Barbados, April 16-18, 2009.

[19] http://gdansk.bradley.edu/sip/

[20] http://131.204.128.91/NNTrainer/index.php