

Computer Networks as Human System Interface

Nam Pham¹, Bogdan M. Wilamowski², A. Malinowski³

^{1,2}Electrical and Computer Engineering, Auburn University, Alabama, USA

³Electrical and Computer Engineering, Bradley University, Peoria Illinois, USA

nguyehu@auburn.edu

wilam@ieee.org

olekmali@ieee.org

Abstract – With the dramatic increase of network bandwidth and decrease of network latency and because of development of new network programming technologies, the dynamic websites provide dynamic interaction to the end user and at the same time implement asynchronous client-server communication in the background. Many applications are being deployed through the computer and are becoming more popular and are effective means in computing and simulation. Running software over the Internet introduced in this paper is one of the applications of computer networks which enhance interaction between human and CAD tools while computer networks are used as a human system interface. Instead of the required design, software must be installed and license purchased for each computer where software is used, now only one user interface handled by a network browser will be required. Furthermore, instead of purchasing a software license for each computer, the software can be used on a pay per use basis. The application of running software over the Internet has some advantages: all users can access and use software remotely from all over the world by Web browsers; because of only one software on the central machine, so it is working independently with the operating systems which is the limitation of most software; and the most important thing is software developers can protect their intellectual property when the Web browsers are used as the user-friendly GUIs (Graphical User Interface). However, running software over the Internet also has some issues that need to be addressed, especially in case of applications requiring a long simulation process. One of the big issues is how users can control the simulation process and how users can use it at the same time without overloading the system. This requires the GUI to have the ability to create some controlling tools to manipulate it with full functions. In order to do that, the simulation speed is traded off so that the software has the capability to understand and process requests from users.

Keywords — network programming, remote operation

I. INTRODUCTION

Internet becomes more and more an effective means of the communication in the world as nowadays [1][2]. Many Internet applications have been done in the past and more will be done in the future to optimize the work and will continue to expand to companies and research institutions to some extent on a regular basis [3][4][5]. Explosion of the network and network computing, so called network computing or Web-based architecture, changes the design of applications. As networking technology has made great progress in its speed and connectivity in the past twenty

years, network software has also evolved and many technologies have been tried and applied to network software development [6]. Computer networks have been changing the philosophy of not only the engineering view but also the business view as well. Nowadays, many companies rely on computer systems and networks for functions such as order entry, order processing, customer support, supply chain management, and employee administration, etc. Therefore, reliability and availability of such systems and networks are becoming critical factors of the present-day enterprises or institutions [7]. The improvement of network security, network speed, etc opens the possibility for new kinds of applications. Because of their accessibility and uniqueness, computer networks provide the ability to access all kinds of information even for smaller, more isolated domains like a company or a school. The recent technological advances in the World Wide Web with hypertext and JavaScript, capable browsers along with the widely available client server technology have created new software development in the telecommunication network management industry. Traditionally, network capacity planning, delivery and network information technology software have been custom developed due to the nature of the business. With ubiquitous Web use at work and home, access to application software via the media is a reasonable expectation. Software availability of telecommunication network on an intranet basis is cost effective, secure, efficient and widely accessible [8]. Running software through computer networks is a typical example. Users can run software through computer networks by interacting with a user interface. Simulation through computer networks has several benefits.

- Universal user interface on every system: every system can run software simulation with a web browser through a user interface.
- Portability: software located only on the central machine can be accessed anytime and anywhere.
- Software protection: users can run software through computer networks but no version of software will be released.

- Legacy software: old software can be run in a dedicated environment on the server while its new user interface runs through a Web browser on new systems for which the particular application is not available.

- Remote control: computer networks are used to control objects remotely.

- Limitation: software can interact with any platform which is independent of the operation systems, users do not have to set up or configure software unless it is implemented on the server in the form of stored user profile.

The Internet bandwidth is already adequate for many software 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 application of software will be possible through the Internet and Intranet. It is, therefore important to develop methods, which operate on different platform and Web browsers. This would require solving several issues such as:

- Minimization of the amount of data which must be sent through the 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
- Other

II. OVERVIEW OF NETWORK PROGRAMMING

HTML alone does not provide the functionality needed for a dynamic, interactive environment. Additional technologies are used to implement the dynamic behavior both on the client side (inside a Web browser) and on the server side to render a Web page dynamically. Most common network programming tools used for developing dynamic websites on the client side are JavaScript, Ajax extension to JavaScript, and Java or ActiveX applets. Most common tools on the server side are PHP, Sun Microsystems' Java Server Pages, Java Servlets, Microsoft Active Server Pages (APS) technology, and Common

Gateway Interface (CGI) scripts using scripting languages such as PERL, server-side JavaScript, ActiveX and Python, or precompiled binary programs [9].

JavaScript is an object-oriented scripting language and is primarily used in the form of client-side JavaScript embedded in HTML- or XML-coded Web pages. It is implemented as an integrated component of the web browser, allowing the development of enhanced user interface and dynamic website. However JavaScript is only implemented on the client side, therefore, clients cannot retrieve data from server which limits interaction between the server and client. This was changed by adding a new library that is typically referred to by the name of Ajax Technology. Ajax (asynchronous JavaScript and XML) is a group of interrelated web development techniques used on the client side to create interactive web applications. With Ajax, web applications can retrieve data from the server synchronously in the background without interfering with the display and behavior of the existing page. The use of Ajax techniques has led to an increase in interactive or dynamic interfaces on web pages.

Active Server Pages (ASP) was Microsoft's first server side script engine for dynamically-generated web pages. Developing functionality in ASP websites is enabled by the active scripting engine's support of the Component Object Model (COM) with each object providing a related group of frequently-used functions and data attributes. Pages with the .aspx extension are ASP.NET (based on Microsoft's .NET Framework) and compiled which makes them faster and more robust than server side scripting in ASP which is interpreted at run time. Most ASP pages are written in VBScript. ASP, however, can be executed presently only on a PC with a Windows operating system, thus making the application platform dependent. Although this technology is very popular already, it does not allow for the development of applications running on multiple platforms.

Common Gateway Interface (CGI) is the standard way for a Web server to pass a Web user's request to an application program and to receive data back to forward to the user. When the user requests a Web page, the server sends back the requested page. However, when a user fills out a form on a Web page and sends it in, it usually needs to be processed by an application program. The Web server typically passes the form information to a small application program that processes the data and may send back a confirmation message. This convention for passing data back and forth between the server and the application is called the Common Gateway Interface. Such applications are known as CGI scripts and they can be written in any programming language such as PERL, PHP, Python, etc.

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 machine. CGI is quite different from writing Java applets. 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.

The next section will discuss two examples which use computer networks as an interface to enhance interaction between the client and the server.

III. EXAMPLES OF DISTANCE OPERATION

1. NEURAL NETWORK TRAINER

Neural network trainer NBN 2.0 is developed based on Visual Studio 6.0 using C++ language hosting on server and communicating with clients through PHP scripts [10]. Its main interface is shown in Fig. 1.

Training Information	Control Area	Parameter Settings
Cur Iteration: 0	Training Algorithm: EBP for ACN	Training times: 40
Cur SSE: 0	Upload the topology: <input type="button" value="Browse..."/>	Max error: 0.002
Avg Iteration: 0	Upload the data: <input type="button" value="Browse..."/>	Max iteration: 40
Avg Time(ms): 0	<input type="button" value="Set parameters"/>	<input type="button" value="Clear all data"/>
Total Times: 0	<input type="button" value="Start Training"/>	<input type="button" value="Exit"/>
Success Rate: 0	<input type="button" value="View Result"/>	<input type="button" value="Stop"/>

Fig. 1. Neural Network Trainer Interface

NBN 2.0 is developed with four different types of training algorithms:

- Error Back Propagation for arbitrarily connected neuron (EBP for ACN)
- Levenberg Marquardt for multilayer perceptron (LM for MLP)
- Neuron By Neuron (NBN)
- Neuron By Neuron, forward-only (NBN- forward only)

The neural network interface will receive the request from users with uploading files and input parameters to generate data files and then transfer the command to the training software located on server. There are two files required for the training process: the topology file and the training data file. These two files have to follow certain syntax so that the training tool can make sense in the correct way.

- Topology file

The topology files are named “*.in”, and they are mainly used to construct neural network topologies for training. The topology files consist of four parts: topology design,

weight initialization (optional), neuron type instruction and training data specification.

The topology design is aimed to create the neural structures. The general command is “n [b] [type] [a1 a2 ... an]”, which means inputs/neurons indexed with a1, a2..., an are connected to neuron b with a specified neural type (bipolar, unipolar or linear). Fig. 2 presents the topology file for the neural networks parity-3.

- Training pattern file

The training pattern files include input patterns and related desired outputs. In a training pattern file, the number of rows is equal to the number of patterns, while the number of columns is equal to the sum of the number of inputs and the number of outputs. However, only with the data in the training pattern file, one can't tell the number of inputs and the number of outputs, so the neural topology should be considered together in order to decide those two parameters (Fig. 3). The training pattern files are specified in the topology files as mentioned above, and they should have the same route as the related topology files.

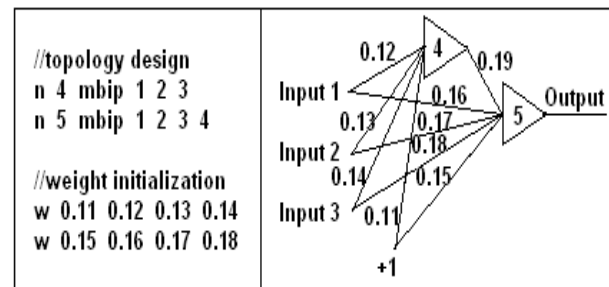


Fig. 2. Weight initialization for parity-3 problem with 2 neurons in FCN network.

training data	topology	explanation
-1 -1 -1 -1	//2 inputs and 2 outputs n3 mbip 12 n4 mbip 12	The first command line of topology shows that there are 2 inputs, since there are 4 columns in training data, so the number of output is 2.
-1 -1 1 1		
-1 1 -1 1		
-1 1 1 -1		
1 -1 -1 1	//3 inputs and 1 output n4 mbip 123 n5 mbip 123 4	The first command line of topology shows that there are 3 inputs, since there are 4 columns in training data, so the number of output is 1.
1 -1 1 -1		
1 1 -1 -1		
1 1 1 1		

Fig. 3. Get the number of inputs and the number of outputs from the data file and topology

All parameters can be edited in the user interface Fig. 1 with the blanks written in HTML form and Java-Script. There are three main parameters: training times, max-error and max-iteration have to be filled in. These values must be the numbers. If users do not fill in these parameters in the proper forms, the warning message will pop up. If these parameters are left blanks, the trainer will automatically assign their values zeros. Besides these parameters, there are other parameters with the default values which will dynamically pop up for inputting data of “combination coefficient”, “scale constant”, “momentum constant”, “learning constant”, “alpha constant”, “beta constant”, “gamma constant” which depend on the selected algorithm when the users click the “Set parameters” button. All these

parameters will be checked on the client side automatically. Using JavaScript to check input parameters will reduce unnecessary data exchange between the server and client.

When all the requirements are fulfilled and set up properly, users can start the training process. The training result is updated every second to let users know that the training process is still running. When the training process is finished, the training result will be generated with data and figures dynamically. The training result file is used to store information such as training algorithm, training pattern file, topology, parameters, initial weights, resultant weights. All results for each training time will be saved automatically in the database, therefore, users can retrieve their training results anytime.

The issue of this trainer is how users can have total control of the training process over the Internet as they can with any installed software which is supported by the operating systems. JavaScript has certain limitations due to the security model of its implementation by a Web browser. One of those limitations is the inability to retrieve data on demand dynamically from the server. This was changed by Ajax technology. Ajax technology allows a JavaScript program embedded inside a Web page to retrieve additional documents or Web pages from the server, store them as local variables, and parse them in order to retrieve data and use it for dynamic alteration of the Web page where the JavaScript is embedded. The additional data is typically generated on the server by means of ASP or CGI used to interface the Ajax query to the database on the server. Data is then sent back typically in the format of an XML formatted Web page. Typically every day application of this technology is an auto-complete suggestion list in a Web page form, for example auto-complete suggestions in a search engine Web page before a user clicks the search button. In this particular application, when the users click “Stop” or “Exit” button, Web browsers will send the message to the trainer located on the central machine. When the trainer recognizes this authoritative message, it will stop training and send the result up to this point back to clients.

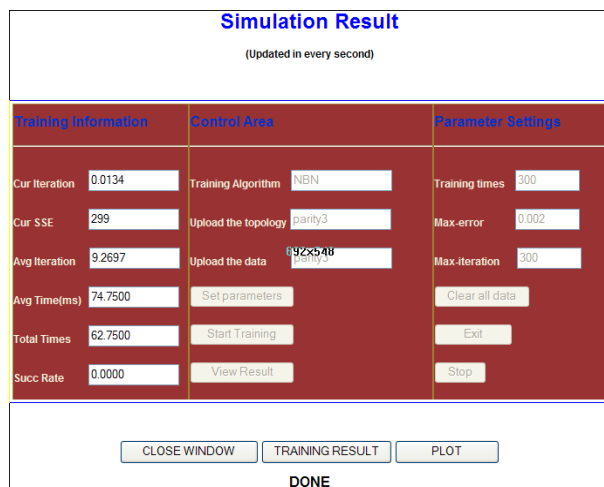


Fig. 4. Training Result

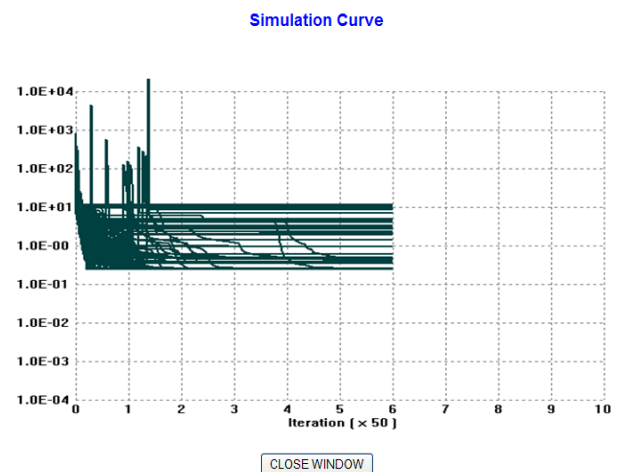


Fig. 5. Output Figure

Parameters
 NBN mu = 0.01000000 scale = 10.00000000
 Data File: parity3.in
Topology
 3 1 2
 4 1 2 3
 5 1 2 3 4
Neurons
 biplor gain=1.00, der=0.01
 biplor gain=1.00, der=0.01
 linear gain=1.00, der=0.05
Initial Weights
 -0.28000000 0.58000000 0.16000000
 0.92000000 0.46000000 -0.08000000 0.28000000
 -0.82000000 -0.46000000 0.94000000 -0.62000000 -0.34000000
Results Weights
 -0.21352129 0.08990222 0.26889254
 -8.09987819 3.40040394 11.61576715 -85.00550976
 3.57991511 -1.46565558 -8.29415476 58.78724502 5.95297590
Training Results
 Total iteration: 300 Total error: 0.42256835 Training Time: 766

Fig. 6. Training Result

2. SIP PROGRAM

The Spice program implemented through computer networks is just one of the examples of networked applications. The SIP allows Spice simulation and analysis to be performed from any computer which has a Web browser on the Internet or an Intranet. The SIP has a user-friendly interface and features include password protection and user account, 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 [3][5]. The current Spice engine being used is Spice3f5 from Berkeley, which allows an unlimited number of transistors, unlike various student versions of Spice programs that are available (Fig. 7).

Authorized User: Anonymous User
 Accesses: 53
 Last Login: 15:36:04 03/08/2010
 User Level: 1

SPICE Engine	Stored CIR Files	Stored OUT Files
SPICE3F5	choose:	choose:
Remote CIR Filename: a_dc4.cir		View/Edit DELETE
Remote OUT Filename: a_dc4.out		View/Edit DELETE

Note that everybody can modify and store here files with any name and those could be incorrect. Therefore several exemplary files are write protected. These examples are: A_BP.CIR, A_DC4.CIR, A_DC5.CIR, A_DC6.CIR, A_DIODE.CIR, A_MEM3.CIR, A_SOL3.CIR, and A_TR2.CIR. If you want to modify then you may copy/paste them to a window which is open using a different name.

RUN Simulation simulate CIR file and store output to OUT file [use SPICE .print option]
 PLOT Data generate plot using data in OUT file
 Help help with the SIP program
 EXIT close other open windows

Fig. 7. SIP Main Interface.

To simulate the Integrated Circuits with the SIP program, users have to select the file name from the CIR drop down box or type the name into the CIR File name and then select the “View/Edit” button to enter the input file. The commands of the input file have to follow the same syntax as the capture syntax in the PSPICE version by MicroSim. After entering and modifying the input file, the users have to save it by pressing the “SAVE CHANGES” button. The simulation process will start by selecting the RUN Simulation button and the output window will popup as Fig. 8.

NPN-PNP AMPLIFIER CIRCUIT
 DC transfer characteristic Mon Mar 22 13:03:18 2010

Index	sweep	v(4)	v(5)	v(6)
0	5.000000e-001	2.999509e+000	3.000000e+000	1.528497e-007
1	6.000000e-001	2.986426e+000	3.000000e+000	1.524884e-007
2	7.000000e-001	2.932099e+000	3.000000e+000	1.512295e-007
3	8.000000e-001	2.856962e+000	3.000000e+000	1.551910e-007
4	9.000000e-001	2.774824e+000	3.000000e+000	3.011039e-007
5	1.000000e+000	2.689688e+000	3.000000e+000	4.263887e-006
6	1.100000e+000	2.602510e+000	3.000000e+000	1.198754e-004
7	1.200000e+000	2.514241e+000	3.000000e+000	3.627945e-003
8	1.300000e+000	2.425360e+000	3.000000e+000	1.120403e-001
9	1.400000e+000	2.339270e+000	3.000000e+000	2.896294e+000
10	1.500000e+000	2.318968e+000	3.000000e+000	2.965063e+000
11	1.600000e+000	2.307525e+000	3.000000e+000	2.973980e+000
12	1.700000e+000	2.298912e+000	3.000000e+000	2.977156e+000
13	1.800000e+000	2.292466e+000	3.000000e+000	2.977441e+000
14	1.900000e+000	2.287252e+000	3.000000e+000	2.977870e+000

elapsed time since last call: 0.000 seconds.
 Total elapsed time: 0.000 seconds.
 Current dynamic memory usage = 0,
 Dynamic memory limit = 0.

Fig. 8. Simulation Output

With the actual network programming technology, any computation process can be done both on the server side and client side and certain information must be transferred back and forth frequently between both sides. We can follow the concept of JAVA Applet, and all the computation can be done only on the client machine.

However this topology has several major issues that need to be considered.

- The programs are often very large, so it is inefficient and impractical to send the whole package via the network. It takes a lot of bandwidth.
- 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.

In case of the Spice Internet it would be impossible to send the Spice engine through the network every time a client requests, this makes the traffic to the server extremely slow. A unique feature of the SIP versus other Spice simulators is that it is operating system independent. Anyone can access and run a simulation and view the results graphically from anywhere with a Web browser via Internet or Intranet. The heart of this program is PERL script located on the server. Each time a user selects any activity a new dialog box will be generated. This dialog box may contain information such as: a text editor which allows the user to edit input of a circuit, a simulation result which edits an output file of a circuit, an output analysis would be set according to purpose of the user or graphic image of plotted results, etc (Fig. 9). In order to generate images recognized by all Web browsers, A PERL script has to run additional programs to create plots from data files (Fig. 10).

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 best to generate graphical images of plots and send them to users. Users frequently inspect the obtained results a few times for example by changing the range or variables to display. In the 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.

CLOSE WINDOW		PLOT Data		Defaults	
Transient Analysis		DC Analysis		AC Analysis	
no data		v(1) v(2) v(3) v(4)		no data	
Return Type		Axis Style		Plotting Style	
Gif		linlin		lines	
				deltaX	deltaY
<input checked="" type="checkbox"/>	Title	NPN-PNP AMPLIFIER CIRCUIT		10	0
<input checked="" type="checkbox"/>	Date			10	0
<input type="checkbox"/>	xLabel			0	0
<input type="checkbox"/>	yLabel			0	0
				min	max
<input type="checkbox"/>	Zoom	xAxis			
		yAxis			
		start	step	stop	
<input type="checkbox"/>	xTics				
<input type="checkbox"/>	yTics				
<input checked="" type="checkbox"/>	Show Grid				

Fig. 9. Output Analysis

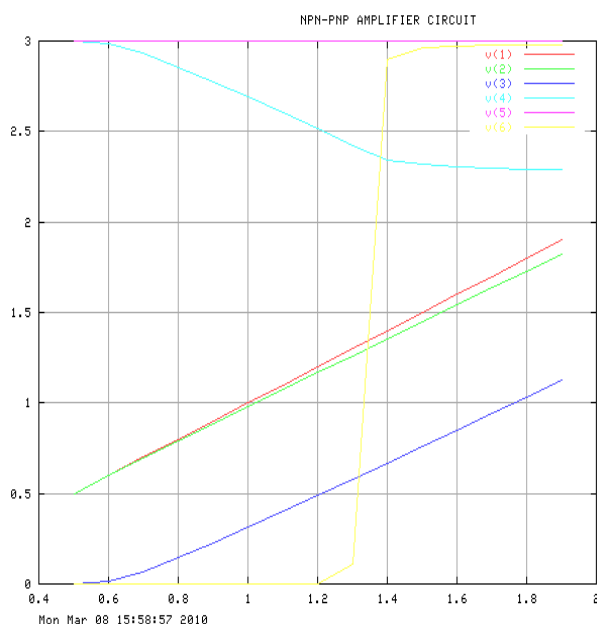


Fig. 10. Output Plot

IV. CONCLUSION

This paper shows how to use the new opportunity created by Internet technologies for efficient and platform independent usage of software applications. The presented examples are just examples but they 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.

This approach will also have a synergetic effect on the development of better software applications since the market will increase significantly. Availability of software application 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.

Several features make the software run over the Internet a desirable program. Only one copy of the software engine needs to be installed and configured. One machine acts as the server and other machines can simultaneously access the server engine through network connections. Remote access to server allows users to run simulations from any computer on the network. This may be from home, or another office in another building, or town.

Computer networks are also used in the systems for database management, econ Websites, controlling objects as robots, etc. The programming network technologies as Java, Visual Basic, PHP, etc are combined with MySQL that is based on the structure query language (SQL) to process data in the database and manage multi-user access. These technologies have made the task of building and accessing the relational databases much easier and more effective. A web server is used to provide the client application to the operator. Client uses custom TCP/IP protocol to connect to the server, which provides the interface to the specific robotic manipulators. Data monitoring and control through a computer network or some other proprietary network is no longer prohibitively expensive and thus restricted only to industrial or luxurious products. Continuously decreasing cost of microprocessors and network interfaces opened additional possibilities in the home automation applications.

The NBN 2.0 is available at:

<http://131.204.128.91/NNTrainer/index.php>

The SIP is available at:

<http://gdansk.bradley.edu/sip/>

REFERENCES

- [1] Wilamowski BM, Malinowski A, " 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.
- [2] Manic M, Wilamowski BM, Malinowski A, "Internet based Neural Network Online Simulation Tool", Proc. of the 28th Annual Conference of the IEEE Industrial Electronics Society, pp. 2870-2874, Sevilla, Spain, Nov 5-8, 2002.
- [3] Wilamowski BM, John Regnier, Aleksander Malimowski, "SIP- Spice Intranet Package", IEEE International Conference on Industrial Electronics, pp.192-195, June 7-10 1998.
- [4] Malinowski A, Wilamowski BM, "Web-based C++ compilers", ASEE 2000 Annual Conference, St. Louis, MO, CD-ROM session 2532, June 18 to 2, 2000.
- [5] Wilamowski BM, Malinowski A, Regnier J, "SPICE based Circuit Analysis using Web Pages", ASEE 2000 Annual Conference, St. Louis, MO, CD-ROM session 2520, June 18 to Aug 2, 2000.

- [6] Arano T, Aoyama M, "Emerging technologies for network software development: past , present, future", Computer Software and Applications Conferences, pp 428, August 21-23, 1996.
- [7] Juan AA, Faulin J, Marques JM, Sorroche M, "J-SAEDES: A java-based simulation software to improve reliability and availability of computer systems and networks", Simulation Conference, 2007 Winter, pp. 2285-2292, Washington, DC, Dec 9-12, 2007.
- [8] Logethran A, Pratiwadi R, Logenthiran D, Porebski A, Thomas DW, "Software migration of telecommunication network management systems to the Web using CORBA and Java", System Sciences, Proceedings of the Thirty-First Hawaii International Conference on, pp. 637-644, Kohala Coast, HI, Jan 6-9 1998.
- [9] Malinowski A, Wilamowski BM, "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.
- [10] Hao Yu, Wilamowski BM, "Efficient and Reliable Training of Neural Networks" IEEE Human System Interaction Conference, HSI 2009, Catania. Italy, pp. 109 – 115, May 21-23, 2009.