A New Concept for Enhanced Simulation of Power Systems

T. S. Sidhu, *Fellow, IEEE*, M. Tamije Selvy, *Student Member*

Abstract-- Power system simulation software has become an essential tool to today's power engineers. Software models of various components of a power system in the form of equations representing their operating characteristics have been used by academics, consultants and manufacturers. Very accurate models of many power system components, such as transmission lines, generators, motors etc., are available today. However, software models of some components such as protective relays, meters, specialized transformers etc., are not available to power engineers. The main reason for this being that modeling of these components require very accurate design details and these details are not available. Manufactures are not willing to divulge the design details of their products for proprietary reasons.

A new paradigm based on a client-server approach is proposed where the manufacturer need not provide the software model of their product to the customers and the customers can still use these models for their simulations.

Keywords: Power system simulation, Software modeling, Client-servers, Relay Modeling

I. Introduction

THERE are many power system simulation packages available in the market today and they have become an indispensable tool for power engineers. Some of the power system simulation packages are EMTDC/PSCAD, EMTP, EDSA etc. These simulation packages can be used for the accurate simulation of various components of the power system. A detailed study of this kind helps power engineers comprehend and design complex power systems. In the present deregulated energy industry scenario in North America, where power systems are operated dangerously close to their operating limits, understanding the complexities involved has become more crucial.

In these simulation packages, various components of the power system like transmission lines, motors, generators, etc are modeled in the form of equations representing their operating characteristics. Very accurate models of many power system components are available today. Simple to complex networks can be built from these basic components and then the network can be simulated to study the performance of a specific component or the whole system for various operating conditions. Most of these commercial software programs come with a user-friendly graphical interface and are easy to use.

T. S. Sidhu is with Department of Electrical and Computer Engineering, University of Western Ontario, London, ON, Canada (e-mail: sidhu@eng.uwo.ca).

M. Tamije Selvy is with the Department of Electrical and Computer Engineering, University of Western Ontario, London, ON, Canada (e-mail: tmunian@uwo.ca).

Presented at the International Conference on Power Systems Transients (IPST'05) in Montreal, Canada on June 19-23, 2005
Paper No. IPST05 - 223

II. Software Models of Specialized Components

Generally, generic models of power system components like transmission lines and generators are readily available, and also are sufficient for most simulations. But for components such as protective relays, meters, specialized transformers etc, generic models do not represent all the details of the component. This means that the accurate performance of these components, their response for marginal cases and precise response times cannot be studied.

With accurate and detailed models of these components, their internal operation can be examined [1-4]. The interaction of the component with the rest of the network can be studied precisely. For example, a generic model of a mho, phase comparator type distance relay can only give the output of the comparator for the given operating conditions. But a commercial relay has more complex logics built on top of the comparator to form a rugged protection system. Also, the comparator logic itself could be specific to the relay. The model of a relay without all the logics built in cannot be used to investigate unexpected operation of the relay.

Unfortunately, the detailed models of these specialized power system components are not available to power engineers. To develop the model of a commercial product, the manufacturer should provide information concerning the basic operating principle of the product and all additional features. But understandably, manufactures are not willing to divulge the design details of their products for proprietary reasons. The manufacturers may already have developed the software model of their product, before the hardware design and the manufacturer is in a position to develop very accurate models. But as mentioned above, the manufacturers cannot provide the models to the customers as it may include their trade secrets.

III. The Client-Server Paradigm

A new paradigm based on a client-server approach is proposed where the manufacturers need not provide the software model of their products to the customers and the customers can still use these models for their simulations [7],[8].

The setup for the new paradigm consists of two parts: the client side on the customer end, which has the entire simulation package except the model from the manufacturer

and the server side on the manufacturer's end, which has the software model of their product. The two ends can communicate through the internet using an appropriate client-server interface. Thus, both the manufacturers' apprehension to divulge their design details can be alleviated and the models can be made available to the users at the same time.

A. *The Client-Server Setup*

The client-server setup proposed is shown in Fig. 1. The figure shows three servers and two clients. Let Server 1 be the server that has the model of the specialized current transformer. Let Server 2 be the server that has the model of the protective relay R1 and let Server 3 be the server that has the model of the protective relay R2.

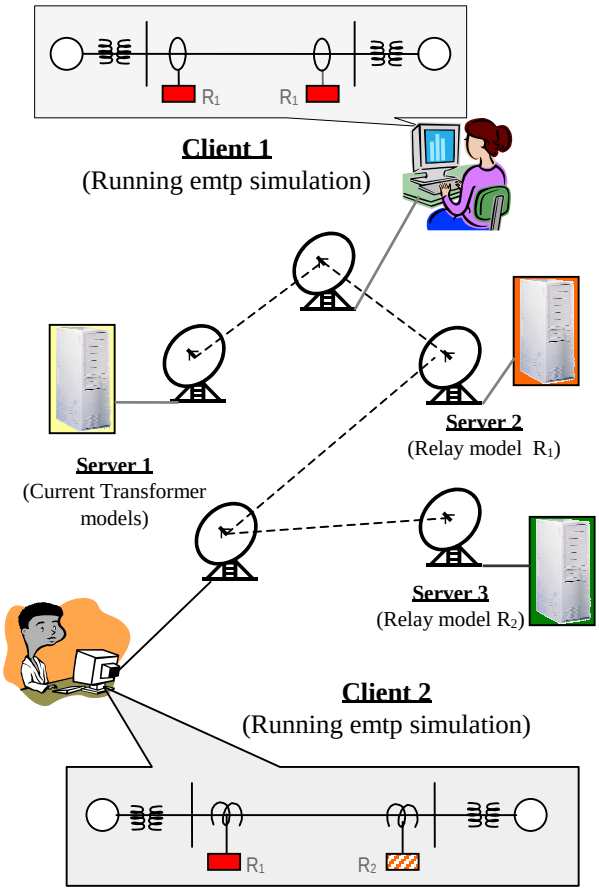

Fig. 1. The client-server setup.

Clients 1 and 2 can be utilities or industrial customers. Let the Client 1 be running an emtp simulation of a system consisting of the specialized current transformers (CT) and two protective relays from the same manufacturer. The simulation package on the client side will have models of the generators, transmission lines, loads, buses etc. But the client would have to access the CT model and the relay model from Server 1 and Server 2 respectively through the internet. Similarly let Client 2 be running an emtp simulation of a system consisting of two protective relays from different manufacturers. The Client 2 has to access the relay models from Server 2 and Server 3.

The emtp running on the client-side treats the software model as though it exists in the same system. The models on the servers behave as if it is getting the data from the server itself. This is same as normal simulation except that some of the specific models reside in another computer (the server). This is achieved using the proper interfaces on the client and the server sides.

B. *The Client-Server Interface*

The client-server interface should have the following features.

- There can be more than one instance of the same external model in a test system. For example, the test system running in Client 1 of Fig. 1 has two relays of the same type R_1.
- There can be more than one external model in the test system. For example, the test system running in Client 1 of Fig. 1, has relay models and transformer model. The client in this case has to send appropriate data to each of the corresponding models in the server.
- The test system might be using more than one instance of the same component but from different manufacturers as in the case of Client 2, in which case the client has to send data to different manufacturers' servers located at different places. The client has to wait till all the servers respond before proceeding with the next simulation step.

The objective of the interfaces is to achieve this. One interface is required on the client side between the emtp and the internet layers and also an interface is required on the server side between the internet layers and the software models.

The client side interface performs the following functions.

- Obtain data from an emtp application (PSCAD/EMTDC).
- Change the data into a pre-specified format.
- Send files for each model to the corresponding server.
- Obtain reply from the server and convert it into the format required for the emtp application.
- Send the reply to the emtp application.

The server side interface performs the following functions.

- Obtain data from the client side
- Change the data into the format required by the software model
- Send the data to the appropriate software model.
- Get the processed data from the software model and convert it into a pre-specified format and send it to the corresponding client.

C. *Testing using the Client-Server Setup*

Three types of testing can be done using the client-server setup, namely open-loop, closed-loop and pseudo-closed-

loop testing. For all the three testing types using the client-server interface, there are two main steps involved. The first is the setting up of the connection and customizing the software model to the current simulation specifications. The second step is the actual simulation itself.

During the first step, the client has to send authentication data to the server(s) and establish connection with the servers. Then the software models present in the servers must be customized to the client's specification. For example, if the external model is a current transformer, the software model in the server will be a general one which can take CT ratios within a range as its input. In this case a custom model should be created in the server for this specific simulation with a specific CT ratio. If the required model is protective relay, then the relay model has to be customized with a specific set of relay settings. The second step, which is the actual simulation, varies for each type of testing.

D. Open-loop Testing

In this type of testing, the entire simulation of the test system is run and the voltage, current and the other control data required for the model is stored into a file in a standard data format. This file is then given to the model, which processes the input data and gives the appropriate response. The client-server setup for the open-loop testing is shown in Fig. 2. In this case there is no interface required between the power system simulation package and the client interface.
The steps involved in this type of testing are as follows:

- The test system is simulated using a power system simulation package and the current and voltage information at specific locations are recorded in a COMTRADE file [9].
- The server is started and a daemon program running in the server listens for input from clients.
- The startup program on the client side is initiated.
- This startup program, first contacts the server with authentication information (username and password)
- The server validates the username and password and if valid starts a new thread for this client and notifies the client.
- The client now sends the customization data to the server.
- The thread that was started accepts this data and creates a custom model and notifies the client.
- The client then sends the recorded COMTRADE file to the server for the custom model to process it.
- The custom model setup on the server processes the input data, and gives the appropriate output to the client.
- This information can be stored or displayed on the client machine.

Open-loop testing is faster and utilizes lesser amount of resources. But the effect of the response of the model on the rest of the system cannot be studied. This type of testing is more useful when the response of the component is discrete and time delayed, such as a protective relay.

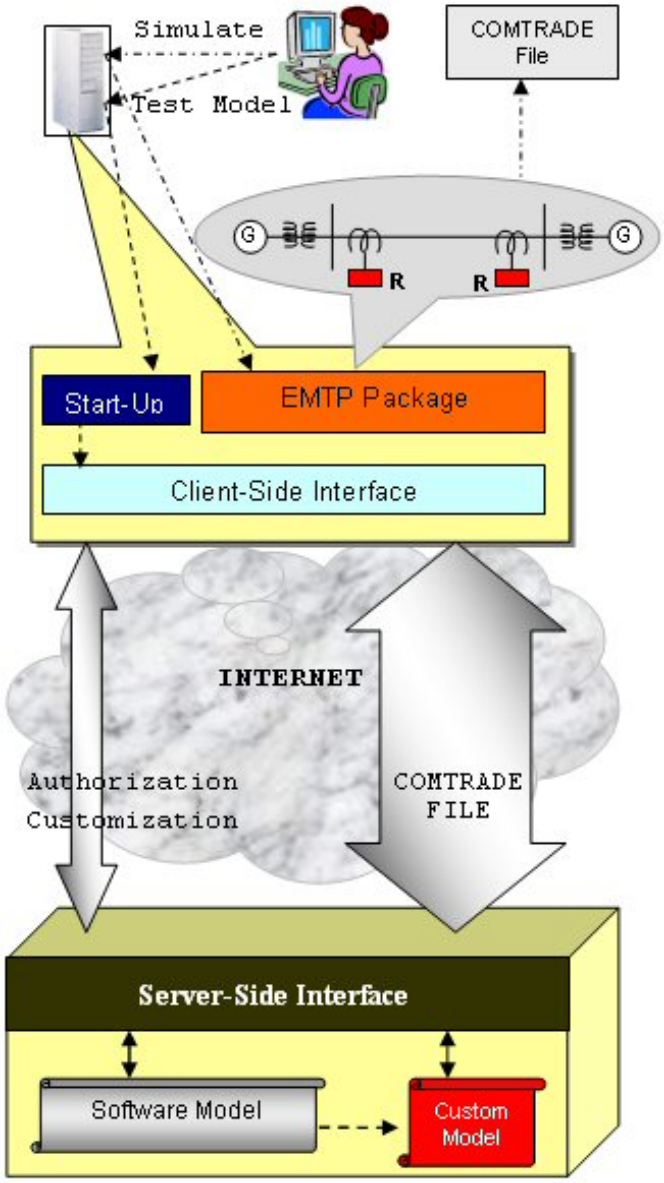

Fig. 2. The client-server setup for open-loop testing.

E. Closed-loop Testing

In closed-loop testing, after every simulation step the voltage, current and control information for that sampling interval is sent to the server, which will process it and will send an appropriate response. The next simulation step on the client side is executed only after the server responds. The client-server setup for the open-loop testing is shown in Fig. 3. Here an interface is required between the power system simulation package and the client interface. A subroutine must be written as a part of the emtp program which can contact the client interface, which in turn can communicate with the server-side interface.

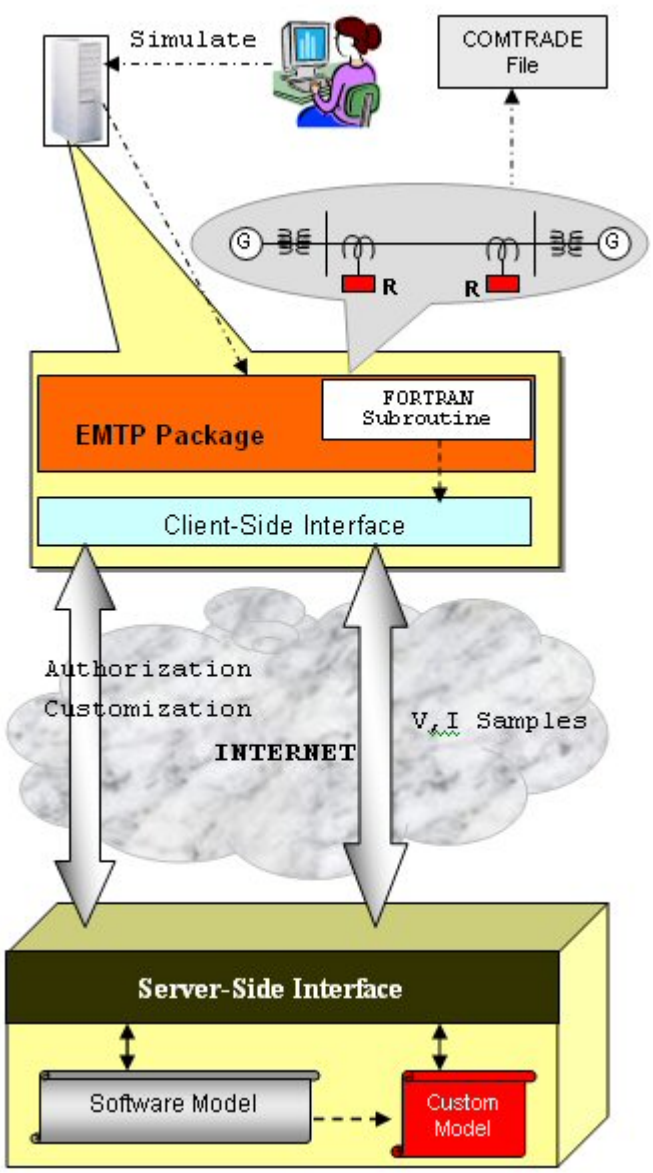

Fig. 3. The client-server setup for closed-loop testing.

The steps involved in this type of testing are as follows:

- The server is started and a daemon program running in the server listens for input from clients.
- The emtp simulation is initiated.
- The subroutine in the emtp simulation initiates the client-side interface before starting the simulation.
- The client-side interface first contacts the server with authentication information (username and password) The server validates the username and password and if valid starts a new thread for this client and notifies the client.
- The client now sends the customization data to the server.
- The thread that was started accepts this data and creates a custom model and notifies the client.
- The client notifies the subroutine that the connection has been set up and the simulation can begin.
- The emtp simulation now begins, and at the end of each sampling interval, the required information is passed to the client-interface by the emtp subroutine.
- The client-interface then sends this data to the server for the custom model to process it.
- The custom model setup on the server processes the input data, and gives the appropriate output to the client.
- The output is passed back to the emtp subroutine by the client-interface.
- This output is used in the next simulation step.
- This continues till the complete simulation is over.

In this type of testing the exact performance of the device and its effect on the test system can be studied online. This type of testing is essential to study the effect of components, such as controllers whose response changes continuously. But the simulation would take a much longer time, due to the delay introduced by the emtp and the device model. Also the server resources would be tied up for a longer time.

F. Pseudo Closed-loop Testing

In the pseudo-closed loop testing, the entire emtp simulation is run without including the external software model, the voltage and current information is stored in a file and sent to the server. The server processes this data and sends the result back to emtp on the client side. The response from the device model is included and the simulation is run again. In this way the effect of this component on the test system can be studied. The client-server setup for the pseudo closed-loop testing is the same as that for open-loop testing shown in Fig. 2

The steps involved in this type of testing are as follows:

- The test system is simulated using a power system simulation package and the current and voltage information at specific locations are recorded in a COMTRADE file.
- The server is started and a daemon program running in the server listens for input from clients.
- The startup program on the client side is initiated.
- This startup program, first contacts the server with authentication information (username and password)
- The server validates the username and password and if valid starts a new thread for this client and notifies the client.
- The client now sends the customization data to the server.
- The thread that was started accepts this data and creates a custom model and notifies the client.
- The client then sends the recorded COMTRADE file to the server for the custom model to process it.
- The custom model setup on the server processes the input data, and gives the appropriate output to the client in a file.
- The information returned from the server by the model is now included as a part of the simulation and the system is simulated again.

Pseudo-closed loop testing does not introduce as much delay as the closed-loop testing and still the effect of the device on the rest of the system can be studied. This type of testing may be the most efficient for the client-server

approach in terms of simulation speed and the utilization of the server resources. This type of testing is also useful when the response of the component is discrete and time delayed but in addition the effect of this response on the rest of the system can be studied.

IV. IMPLEMENTATION

The client-server setup described was setup to verify the proposed client-server approach. A commercial relay was modeled using the specifications provided by the manufacturer. The model was written in Visual C++. The client-server interface described above was coded in Java [5].

A. Commercial Relay Model

A commercial distance relay was modeled. The software model of the commercial relay has the following features.

- It accepts a standard setting file from the manufacturer as the settings input and creates a specific model for the given settings.
- It accepts a standard COMTRADE data file as the data input
- It processes the data as the relay would process it using proprietary algorithms, applies trip logics and gives an output indicating if the relay operated or not.
- The model can also plot the trajectory of the apparent impedance seen by the six impedance loops of the relay superimposed on the impedance characteristics of the relay.

A plot of the apparent impedance seen by the six impedance loops for a phase A to ground fault is shown in Fig.4. This feature is available only when the model is tested locally. When the model is tested through the client-server interface, only the trip/no-trip decision made by the relay model is sent back to the client.

B. Client-Server Interface

The client-server interface was setup to perform open-loop testing as shown in Fig. 2 and described in Section III.D. The setup consisted of a server and a client [6]. The model was put in the server. The client-side interface was loaded in the client machine and the server-side interface was loaded in the machine acting as the server.

C. Testing

Simulations of various test system was run using the power system simulation package EMTDC/PSCAD. The voltage and current outputs at various locations were recorded in a standard COMTRADE file. The recorded data file was given to the model in the server through the interface and the output from the model was observed from the client machine.

The output from the model in a local machine and the output from the model on a server were identical. The results from some of the test cases are shown in Table 1. It should be noted that the total time taken for simulation is not constant. It depends on the status of the network, such as bandwidth, propagation delay, current internet traffic, reliability of the network.

TABLE 1
COMPARISON OF OUTPUT FROM MODEL IN LOCAL MACHINE AND OUTPUT FROM MODEL IN A SERVER

Case	Local Model		Server Model	
	Output	*Delay*	*Output*	*Delay*
1	No Trip	Instantaneous	No Trip	Instantaneous
2	Trip	Instantaneous	Trip	Instantaneous
3	No Trip	Instantaneous	No Trip	Instantaneous
4	No Trip	Instantaneous	No Trip	Instantaneous
5	Trip	Instantaneous	Trip	Instantaneous

D. Standardization

The software model of a device can return different kinds of data. For example, the software model of a distance relay can just return a trip/no trip signal or it can return the apparent impedance as seen by the relay. The trajectory of the apparent impedance can be plotted on the client side. The type of data that is returned from the models and the format in which it is returned must be standardized and need input from standard organizations such as IEEE.

V. FUTURE WORK

The next step is to setup the client-server interface to perform closed-loop testing as shown in Fig. 3. In this case, all processing on the client side will take place from the emtp program. A component will be created in EMTDC as a FORTRAN subroutine which will contact the client-side interface directly. At the end of each simulation step, the voltage and current information will be sent to the server. The model on the server-side will be modified to accept voltage and current information sample by sample instead of in a COMTRADE file. Accuracy tests and timing tests will be done using the closed-loop setup. Then the simulation times taken for open-loop and closed-loop testing will be compared.

VI. CONCLUSIONS

Even though the power system simulation packages available today are powerful tools, the unavailability of models of commercial products is a significant shortcoming. The main reason being the absence of design details for these products, for the manufacturers do not want to divulge their proprietary information.

The proposed client-server paradigm solves this problem elegantly. Using this concept the manufacturers can make the models of their products available to their customers without divulging their design details. The power engineers can perform more meaningful simulations, which helps them to comprehend and design more complex power systems.

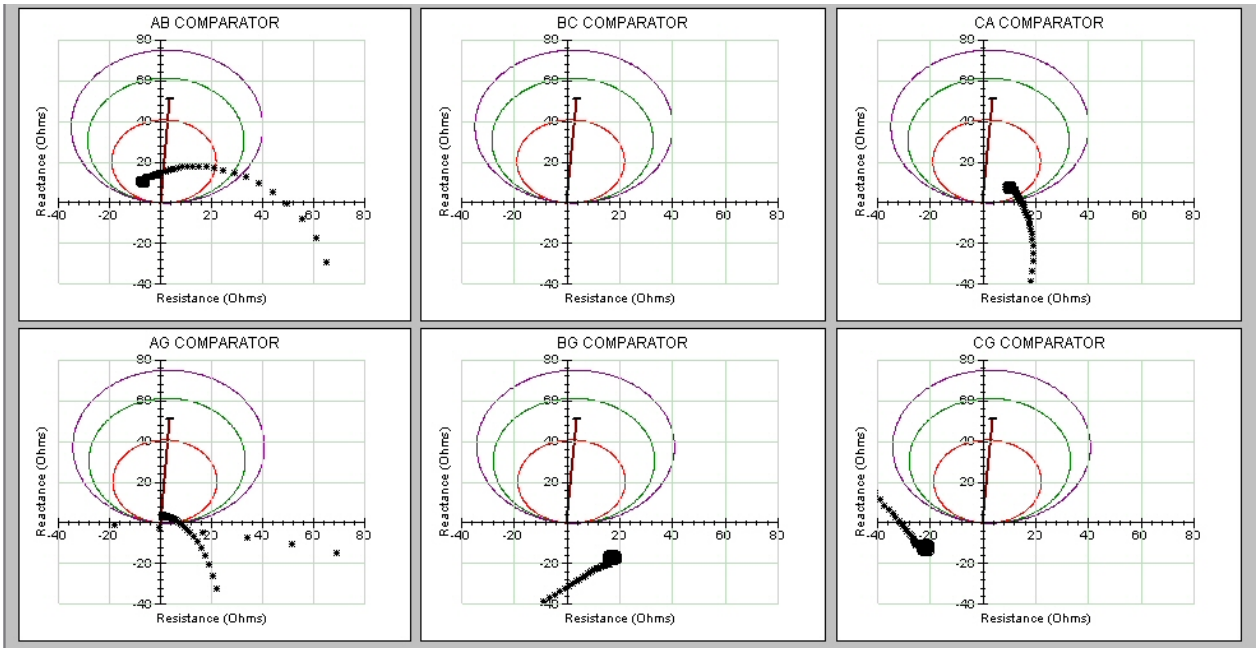

Fig. 4. Plot of the trajectory of the apparent impedance seen by the six impedance loops of the relay for a phase A to ground fault.

VII. References

Periodicals:

[1] T.S. Sidhu, M.S. Sachdev and H.C.Wood, "A Computer Aided Design Tool for Developing Digital Controllers and Relays," *IEEE Trans. On Industry Applications*. Vol. 28, No. 6, pp. 1376-1383, 1992.

[2] P.G.McLaren, G.Benmouyal, S.Chano, A.Girgis, C.Henville, M.Kezunovic, L.Kojovic, R.Marttila, M.Meisinger, G.Michel, M.S.Sachdev, V.Skendzic, T.S.Sidhu, D.Tziouvaras, " Software models for relays," *IEEE Transactions on Power Delivery*, Vol. 16, No. 2, April 2001, pp. 238-245.

[3] T.S.Sidhu, M.S.Sachdev, R.Das, "Modern Relays: Research and Teaching Using PCs," *IEEE Computer Applications in Power*, April 1997, pp. 50-55.

[4] T.S.Sidhu, M.A.Hfuda, and M.S.Sachdev, "Generating Relay Models for Protection Studies," *IEEE Computer Applications in Power*. Vol 4. pp. 33-38. October 1998

Books:

[5] Herbert Schildt, *Java 2, The Complete Reference*, 5th Edition, McGraw Hill Inc.

[6] James F. Kurose and Keith W. Ross, *Computer Networking: A top-down Approach Featuring the Internet*, 2nd Edition, Addison Wesley, 2002

Papers from Conference Proceedings (Published):

[7] T.S.Sidhu and M. Tamije Selvy, "Advancements in Relay Modeling", *6th International Power Engineering Conference*, November 2003.

[8] T.S.Sidhu, M. Tamije Selvy and A.Das "A Client-Server Approach for Advanced Relay Software Modeling and Testing", *2003 IEEE/PES Transmission and Distribution Conference, Septemeber2003.*

Standards:

[9] *IEEE standard common format for transient data exchange (COMTRADE) for power systems*, IEEE Std C37.111-1999, Oct. 1999.

VIII. Biographies

Tarlochan S. Sidhu (M' 90 –SM'94 -F'04) received the B.E (Hons.) degree from the Punjabi University, Patiala, India, in 1979 and the M.Sc. and Ph.D degrees from the University of Saskatchewan, Saskatoon, and SK, Canada in 1985 and 1989, respectively. He worked for the Regional Computer Center, Chandigarh, India, Punjab State Electricity Board, India, and Bell-Northern Research Ltd., Ottawa, ON, Canada. From July 1990 to June 2002, he was with the Department of Electrical Engineering, University of Saskatchewan, where he served as Professor and Graduate Chairman of the Department. He is currently the chair of the Department of Electrical and Computer Engineering, Professor and the Hydro one Chair in Power Systems Engineering at the University of Western Ontario, London, ON. His areas of research interest are power system protection, monitoring and control. Dr. Sidhu is a Fellow of the Institution of Electrical and Electronics Engineers, a Fellow of the Institution of Engineers (India) and a Fellow of the Institution of Electrical Engineer (U.K). He is also a Registered Professional Engineer in the Province of Ontario and a Chartered Engineer in the U.K.

Tamije Selvy Munian was born in Pondicherry, India, in 1976. She received the B.Tech (Hons.) from the Pondicherry University, India in 1998 and the M.Sc degree in electrical engineering in 2002 from the University of Saskatchewan, Saskatoon, SK, Canada. She is currently pursuing her PhD degree from the University of Western Ontario, London, ON in the Department of Electrical & Computer Engineering.

She worked for DSQ Software Limited, India as a Software Engineer from 1998-2000.