

GRAPHICAL USER INTERFACE FOR A DIGITAL REAL TIME SIMULATOR

M. Kezunovic, Senior Member F. Ji
Texas A&M University

S. M. McKenna, D. Hamai
Western Area Power Administration

Abstract This paper presents a design of a graphical user interface(GUI) for a real time simulator(RTS) for relay testing. The GUI software is developed in the X-Window/Motif environment under the AIX/IBM RISC6000 operating system using C language. The editing portion of the software allows the construction of one-line diagrams of power system networks in a paint-brush-like environment. Real time simulation preparation and initialization is also performed by the GUI. Both off-line and real-time simulation can be invoked from the GUI. Its efficiency is demonstrated by utilizing it in actual applications.

1 INTRODUCTION

The use of digital simulators for relay testing has been known since the early eighties [1]. Digital simulators of an open-loop type are extremely effective in replaying either simulated or recorded waveforms into relays for testing purposes. Nevertheless, the open-loop simulators can not easily be utilized to support transient computations under real-time constraints, since the relay testing requires multiple dynamic changes of the power system configuration based upon the relay's response. Electromagnetic Transients Program(EMTP), commonly known in the power industry, can be used for the open-loop transient simulation but not for the real-time simulator implementation [2]. A new simulation software has been designed to operate under a closed-loop control of relays under test. It is implemented using EMTP computational techniques enhanced with novel numerical solutions for dynamic power system configuration change and nonlinear element modeling [3, 4].

Various graphic interface packages on different platforms using Microsoft Windows, X-Windows, or commercial GUI

builders such as LabWindows have been reported [5]-[8]. However, none of commercially available GUI software today can directly be used as a general solution for applications in power system simulation and control. A dedicated Graphical User Interface(GUI) for the new real-time simulator is thus needed for support of its functions. The GUI for real-time simulator has been developed in X-Window/Motif based environment under the AIX/IBM RISC6000 operating system using C language. A set of standard mouse-based manipulations such as *clicking*, *selecting* and *dragging* is utilized together with the keyboard entry of alphanumeric data.

This paper starts with an introduction of the simulator architecture. Then we address the design and implementation issues of the GUI. In the last section, we present some experience of using the GUI in real-time simulation for relay testing.

2 REAL-TIME SIMULATOR ARCHITECTURE

The most demanding real-time simulator design requirement is computation of fault transients under the condition of real-time change of power system configuration due to relay operation [4]. This problem is solved, using EMTP computational techniques enhanced with novel numerical solutions for dynamic power system configuration change and nonlinear element modeling [3, 4]. An advanced Reduced Instruction Set Computer(RISC) architecture is utilized to achieve further optimization of the execution time for the transients computation code. The main advantage of this design is the use of conventional single processor computer architecture in combination with advanced digital signal processors(DSPs). This makes this simulator an off-the-shelf product with all the benefits of commercially available computers priced at a relatively low cost.

The real-time simulator hardware and software architecture are shown in Figure 1 and Figure 2 respectively.

The overall organization of software reflects the existence of two distinct subsystem as shown in Figure 1: the Graphical User Interface(GUI) and the Real Time Simulation(RTS). The GUI software is performed entirely by the GUI workstation(RS6000/320H), while the RTS software is performed

This paper has been presented at the First International Conference on Digital Power System Simulators - ICDS '95, College Station, Texas, U.S.A., April 5-7, 1995.

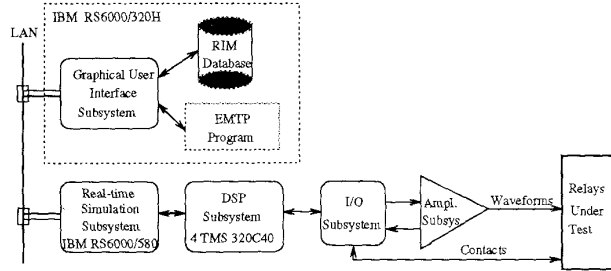


Figure 1: Real-Time Simulator Hardware Architecture

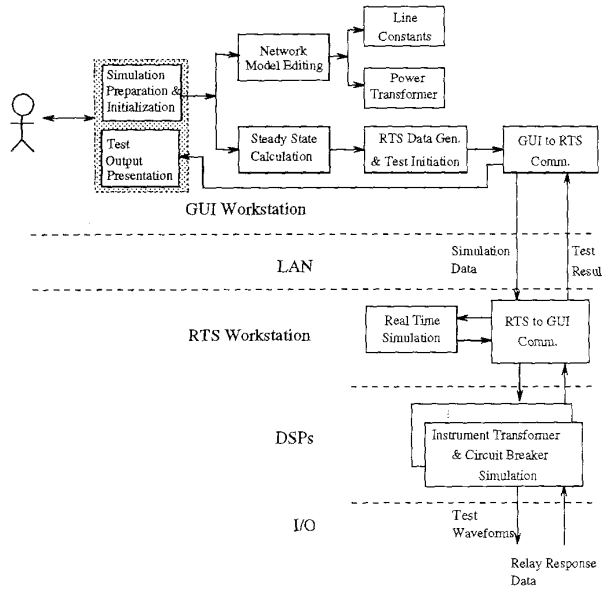


Figure 2: Real-Time Simulator Software Architecture

by the RTS workstation(RS6000/580) and the DSPs. The features of both RISC workstations, as well as the software operating environments, are selected to support in the most appropriate way the specific tasks assigned to each workstation.

The workstations communicate among themselves by using Local Area Network(LAN) for data exchange. The communication is limited to a small set of functions, namely:

- Real-time subsystem initialization
- Simulation initialization
- Result collection

All of the communication between the GUI and RTS is provided by the GUI support modules. The program downloading, simulation initializing, and result collecting are supported by the RTS-to-GUI communication module indicated

in Figure 2. The main task of the GUI is to provide the one-line network model editing, which will be discussed in more detail in the following section. The EMTP program is used for calculation of the steady-state, the line data are computed by the Line Constant auxiliary routine, and the transformer parameters are obtained from the Power Transformer auxiliary routine. A relational information management(RIM) database is utilized for data management for the GUI software.

3 GUI DESIGN AND IMPLEMENTATION

3.1 Functionality

This graphical user interface is designed as a CAD tool, performing the following groups of tasks as shown in Figure 3:

- preparing and initiating input data for the simulator
- collecting and presenting the results of the simulation.

To prepare and initiate a simulation, the user is provided with an interactive graphical editor and a set of auxiliary modules, including the GUI-to-RTS communications module. All the auxiliary modules are accessed through unique user interface, at appropriate points of network model definition. In general course of preparation and initialization of a simulation, the user may perform the following steps:

- Edit a one-line diagram by representing the network topology and modeling the network components.
- Invoke the EMTP auxiliary routines including Line Constants and Power Transformers in order to perform calculations required for completing the network model.
- Perform the steady state calculation for a network model, after a complete network has been defined.
- Generate the input data files in the form appropriate for RTS usage, and initiate simulation by calling a sequencer located in the real-time simulation machine.
- Accept the simulation test results from the RTS subsystem and make them available to the user. The results pertain only to the relay response data and are presented to the user upon request.

3.2 Network Modeling and the GUI Representation

In the real-time software development, well known EMTP modeling techniques have been applied [2]. The GUI provides data for the network components as required by preparing their EMTP cards. These components include uncoupled R, L, C and R-L branches, coupled R-L, C branches, constant parameter(CP) overhead transmission lines, transmission lines with frequency dependent(FD) parameters, voltage sources.

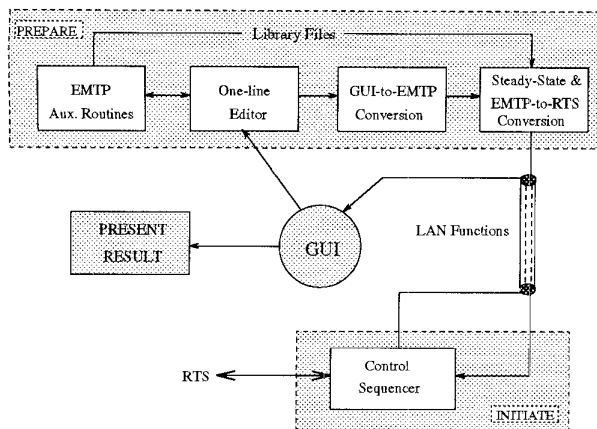


Figure 3: Data Flow of the GUI Software

Other components supported by the real-time simulator are modeled differently from what is used in the EMTP [3, 4]. A brief description of the components treated differently than in the EMTP is given below.

In the GUI, a **fault** is represented as a special component with fault resistance, incidence angle, the initial time at which the fault happens, the length of time the fault lasts, as well as the fault location.

There may be maximum of three **relays** under consideration in any simulation run. Each of them is associated with a set of three switches on which they act. The circuit breakers are modeled as three-phase ideal switches for this purpose. Therefore, the same parameters must be entered for each phase: resistance, initial state, reference node (node for which the voltages and the currents are outputted), etc. Additionally, several parameters describing physical breaker characteristics must be specified. These parameters are used when the circuit breaker model is implemented in the DSP boards. These parameters are saved in the device description files associated with different breaker types, manufactures, or installation sites. These files are treated as RTS system library files and are manipulated through the GUI environment.

Instrument transformers are modeled separately from the remaining parts of the power system. They are available as separate components and they are simulated by the DSP subsystem. Internal network topologies of instrument transformer models are fixed, as described in [9, 10]. Each topology is accompanied by at least one set of parameters representing a commercially available device. The parameters are stored in disk files and manipulated through GUI environment. Connections of the inputs of instrument transformers are implicitly given by the GUI. They are determined by the locations of the relays.

Two different models of series capacitor with MOV(CMOV) protection can be used for representing those components: one-line segment and three-line segment models [3]. The **voltage arresters** are assumed to be of

ZnO kind and gapless. They are permitted to appear only in a shunt configuration. The MOV characteristic for this component is modeled as the one-line segment.

In order for the user to efficiently define the component parameters during a data preparation session, two data libraries are manipulated by the GUI through the database. One of the libraries is designed for the default component parameters, the other for the one-line diagram object attributes. Therefore, all the component parameters can be dynamically updated from graphical user interface functions provided. All the component models are represented in the format defined by the database schemes. For example, the following data attributes are defined for a CMOV and a source component:

modelname CMOVs

type 08

parameters: model type No.,

(type 0) C, bypass L, bypass R, current, energy, V1, R1, V2, R2, V3, R3;

(type 1) C, bypass, clipping voltage, current, energy, clipping R, bypass R;

(type 2) C, bypass L, R in series with C, current, energy, V1, R1, V2, R2, V3, R3

modelname Source

type 01

parameters: frequency, magnitude, phase angle

For each component, either its default parameters or specific values for a given network can be presented to the user through various dialog menus. For example, the dialog menu for component "fault" is shown in Figure 4. The user is able to select, update, or input new values.

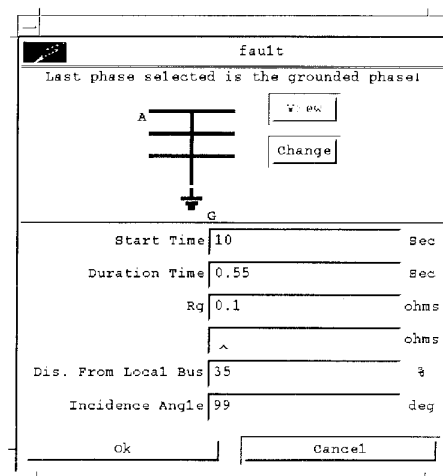


Figure 4: An Example of Dialog Menus in the GUI

3.3 GUI Software Implementation

When the user starts the real-time simulator system, a window shown in Figure 5 is created. The user can work in two sessions:

- simulation
- network editing and simulation

The status of the above sessions is printed in the upper-left message window.

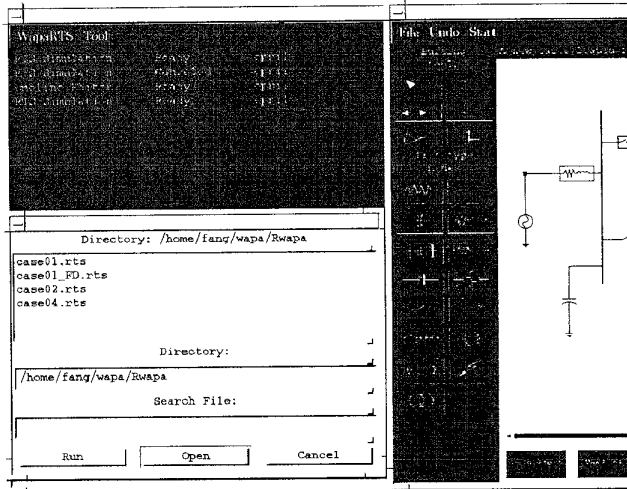


Figure 5: Typical Screens of the GUI System

Users interact with various menus of the GUI to issue requests for actions. These menus are designed in a hierarchical structure. Every menu except the toplevel window has a parent window and can also have submenus. Figure 6 illustrates such a window hierarchy of the GUI and the relationship between several windows.

The most important part of the GUI software system is the one-line network diagram editor shown in the right hand side of Figure 5. The editor enables users to construct one-line diagram representation of a power system through simple point-and-click, paint-brush like, mouse directed movements. These editing functions are initiated from the protocol icons specified in the **Toolbox** of the one-line editor. The display can then be used by putting the GUI in the simulation mode. Additionally, a pseudo-EMTP file can be created for general EMTP applications.

If a case has been edited and its pseudo-EMTP format file generated using the one-line editor, the user can repeat a simulation easily from the simulation window shown in the bottom-left of Figure 5.

The vital task in designing the GUI is to determine the hierarchy of objects needed. The bottom level objects are extracted from the power system itself. These include the physical types of equipment such as sources, breakers and lines. As these objects are studied and decomposed, similarities may be detected between the objects. Using the techniques of object-oriented programming these similarities may be taken advantage of to allow for the sharing of attributes and operations among similar objects. A prototype class hierarchy has been defined in Figure 7. This hierarchy contains

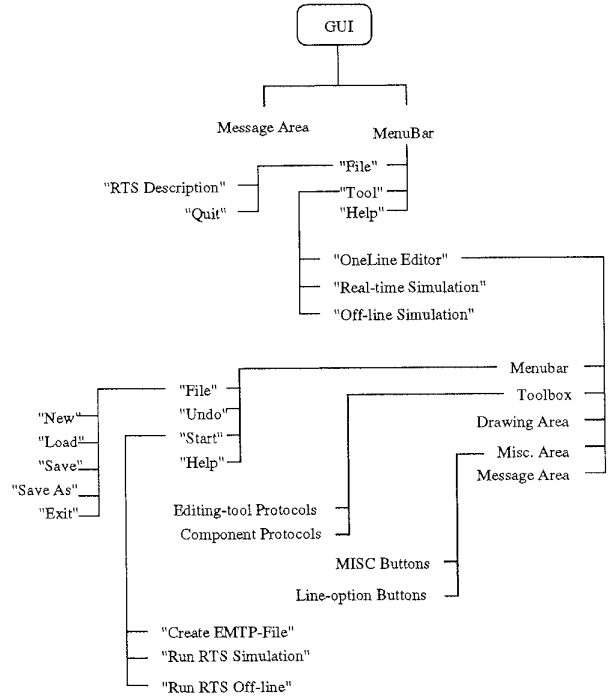


Figure 6: Window Hierarchy of the GUI System

the basic elements required to model a power system for the simulation purpose. It may easily be expanded to incorporate other component objects.

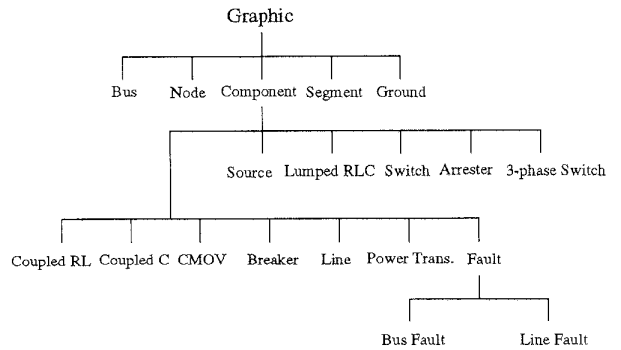


Figure 7: Class Hierarchy of GUI Design for the RTS

4 EXPERIENCES WITH GUI APPLICATIONS

A typical Western Area Power Administration(WAPA) 345K network section has been chosen for new simulator testing as well as the GUI evaluation. The network circuit is shown in Figure 8. This system consists of both nonlinear components including Metal Oxide Varistors(MOVs), Capacitor Coupling

Voltage Transformers(CCVTs), Current Transformers(CTs) and arresters, and linear components including sources, coupled RL/C branches, capacitors, switches and relay-controlled breakers.

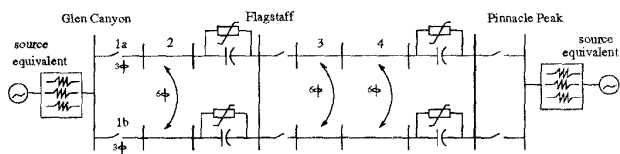


Figure 8: WAPA Reference System

For WAPA reduced cases, the transmission lines between the Flagstaff and Pinnacle Peak substations are made as an equivalent voltage source with an impedance. For both the reduced and full WAPA cases, up to three relays are installed on the lines between the Glen Canyon and Flagstaff substations. Totally 11 WAPA reduced and 6 WAPA full cases are studied for testing the relay performance for various fault and trip-reclose sequences.

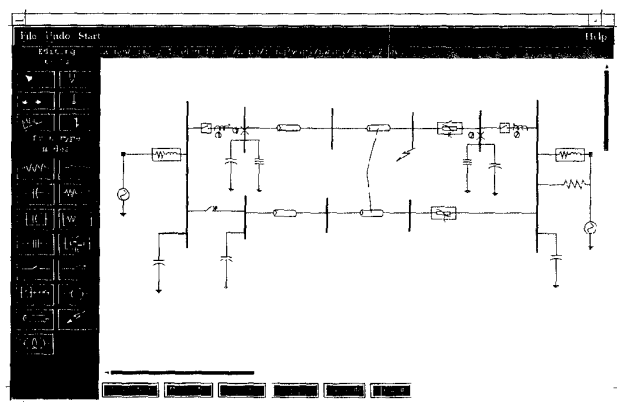


Figure 9: One-line Diagram for a WAPA Reduced Case

For the first case, the user needs to build the topology from beginning and define all the component properties. This process becomes very simple involving only mouse clicking and dragging movements in the GUI environment. To define component property the user is required to input or select all necessary parameters. However, if the user has a well defined component default parameter library, this process is made much easier. The user just confirms the default values and makes some minor updating if needed. The average time to prepare from the scratch a new case with 25 components, 12 buses/nodes (WAPA reduced case) is about 12 minutes and 25 minutes with 42 components, 21 buses/nodes (WAPA full case).

The effort to prepare input data for the real-time simulation using the GUI tools is much less from that using a text editor. Furthermore, one network can be reused for other similar cases. Much efficiency can thus be gained.

Figure 9 is a screen dump of the the one-line representation of the WAPA reduced network. The output of the GUI editor was used for the RTS simulation. Figure 10 and 11 shows the resulting current and voltage waveforms measured at the Flagstaff substation. A three-phase fault(incidence angle is 0°) happens at time 0.05sec after the simulation starts. Three poles of the breaker at Flagstaff are tripped simultaneously when the fault is detected. No reclose is considered for this case. The results are exactly as expected.

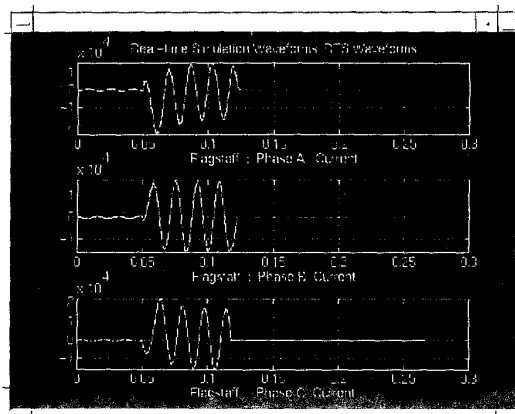


Figure 10: Simulation Results — Currents of Flagstaff

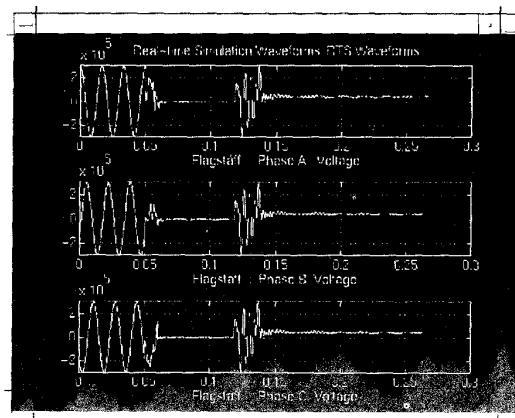


Figure 11: Simulation Results — Voltages of Flagstaff

Besides the WAPA system, Houston Lighting and Power (HL&P) system has also been utilized for testing the RTS system. Very efficient results have been achieved in this case as well. It is expected that the GUI and the RTS software can be used for any other power system network. The critical requirement is that the network should be modeled in the way supported by the RTS software system.

5 CONCLUSION

A graphical user interface for real-time simulation for relay testing has been designed and implemented incorporating the entire simulator system requirements. X-Window system is utilized for graphics and protocols. Complex technical notations are presented in a friendly graphic environment. The user is released from dealing with complicated and laborious data preparation and result presentation tasks.

6 ACKNOWLEDGMENTS

The study reported in this paper has been funded by Western Area Power Administration under contract FC65-90WA 07990.

References

- [1] A. Williams and R. H. J. Warren, "Method of Using Data From Computer Simulation to Test Protection", *IEE Proc.*, vol. 131, Pt. C, No. 7, pp. 349-356, November 1984.
- [2] *Electromagnetic Transient Program (EMTP) Rule Book*, EPRI EL 6421-1, vol. 1, 2, June 1989.
- [3] M. Kezunovic, M. Aganagic, S. M. McKenna, D. M. Hamai, "Computing Response of Series Compensation Capacitors with MOV Protection in Real-time", *IEEE/PES 1994 Summer Meeting*, paper No. 94SM 400-2 PWRD, San Francisco, July 1994.
- [4] M. Kezunovic, M. Aganagic, V. Skendzic, J. Domaszewicz, J. K. Bladow, D. M. Hamai and S. M. McKenna, "Transients Computation for Relay Testing in Real-time", *IEEE/PES 1993 Summer Meeting*, paper No. 93SM 383-0 PWRD, Vancouver, July 1993.
- [5] M. Foley, A. Bose, W. Mitchell and A. Faustini, "An Object Based Graphical User Interface for Power System", *IEEE Trans. on Power System*, vol. 8, pp. 97-104, February 1993.
- [6] R. R. Shoults and E. Barrera-Cardiel, "Use of a Graphical User Interface Approach for Digital and Physical Simulation in Power System Control Education: Application to an HVDC Transmission System Model", *IEEE Trans. on Power System*, vol. 7, pp. 1598-1603, November 1992.
- [7] P. G. McLaren, R. Kuffel, R. Wierckx, J. Giesbrecht and L. Arendt, "A Real Time Digital Simulator for Testing Relays", *IEEE Trans. on Power Delivery*, vol. 1, pp. 207-213, January 1992.
- [8] S. Li and S. M. Shahidehpour, "An Object Oriented Power System Graphics Package for Personal Computer Environment", *IEEE Trans. in Power System*, vol. 8, No. 3, pp. 1054-1060, August 1993.
- [9] M. Kezunovic, Lj. Kojovic, A. Abur, C. W. Fromen, F. Phillips and D. R. Sevcik, "Experimental Evaluation of EMTP-based Current Transformer Models for Protection Relay Transient Study", *IEEE Trans. on Power Delivery*, vol. 9, No. 1, pp. 405-413, January 1993.
- [10] Lj. Kojovic, M. Kezunovic V. Skendzic, C. W. Fromen, D. R. Sevcik and S. L. Nilsson, "Digital Models of Coupling Capacitor Voltage Transformers for Protective Relay Transient Studies", *IEEE Trans. on Power Delivery*, vol. 7, No. 4, pp. 1927-1935, October 1992.

M. Kezunovic (S'77, M'80, SM'85) received his Dipl. Ing. degree from the University of Sarajevo, the M.S. and Ph.D degrees from the University of Kansas, all in electrical engineering in 1974, 1977, and 1980, respectively. Dr. Kezunovic's industrial experience is with Westinghouse Electric Corporation in the U.S.A., and the Energoinvest Company in Sarajevo. He also worked at the University of Sarajevo. He was a Visiting Associate Professor at Washington State University in 1986-1987. He has been with Texas A&M University since 1987 where he is an Associate Professor. His main research interests are digital simulators and simulation methods for relay testing as well as application of intelligent methods to power system monitoring, control and protection. Dr. Kezunovic is a registered professional engineer in Texas.

F. Ji was born in Henan, China on February 11, 1965. She received her B.S. and M.S. degrees in electrical engineering from Tianjin University, Tianjin, China, in 1985 and 1988 respectively. Ms Ji had been employed with Tianjin University from April 1988 to March 1991 as a teaching and research assistant. She is pursuing another M.S. degree in electrical engineering from Texas A&M University and is expecting to graduate in December 1994. Ms Ji's main research interests have included power system protection applications and computer software.

S. M. McKenna was born in Spokane, Washington in 1951. He earned his B.S. degree from the Colorado School of Mines in 1978. Mr. McKenna joined the Western Area Power Administration (Western) in September 1980. Mr. McKenna is currently in the Substation Control Branch where he is involved in control system design for advanced technology devices, protective relaying application and testing, and power system testing. Mr. McKenna is a registered professional engineer in Colorado.

D. M. Hamai was born in San Rafael, California on October 14, 1963. He received his B.S. degree in electrical engineering from the University of Colorado in 1986. Mr. Hamai has been employed with the Western Area Power Administration since 1986 and is currently in the Technical Branch where he is involved with various Electromagnetic Transient Program (EMTP) studies, power system protection applications, and major power equipment requirements.