

A Simulation Tool for Analysis of Alternative Paradigms for the New Electricity Business

Thomas J. Overbye Peter W. Sauer George Gross Mark J. Laufenberg Jamie D. Weber
overbye@uiuc.edu p-sauer@uiuc.edu gross@uiuc.edu lauf@uiuc.edu jdweber@uiuc.edu

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Abstract

Various proposals have been made for new paradigms for operating the restructuring electrical system. One of the key challenges associated with this restructuring is to effectively analyze and evaluate the effects these alternative paradigms may have on major aspects of power system operations, including the so called ancillary services. This paper provides preliminary results on the development of a simulation tool to perform this analysis. Since power systems are ultimately governed by the laws of physics, the heart of the tool is a power system simulation engine, valid over a simulated time frame of minutes to days. The modular structure of the tool then allows other packages to be superimposed on this structure to represent different proposed paradigms. Integration of effective visualization into an interactive simulation environment is then used to gain understanding and insights concerning the attributes of each paradigm.

1. Introduction

The electricity industry is undergoing its most turbulent period in history. Throughout the world there are myriad rapid and deeply impacting changes which are resulting in a restructuring of the industry. Various proposals have been made for new paradigms for operating this restructuring industry. One of the key challenges associated with this restructuring is to effectively analyze and evaluate the effects these alternative paradigms may have on major aspects of power system operations, including the so called ancillary services.

In this paper we will discuss POWERWORLD™, a power system simulator that we have developed to help meet this need. Since power systems are ultimately governed by the laws of physics, the heart of this tool is a power system simulation engine, valid over a simulated time frame of minutes to days. The modular structure of the tool then

allows other packages to be superimposed on this structure to represent different proposed paradigms. Integration of effective visualization into an interactive simulation environment is then used to gain understanding and insights concerning the attributes of each paradigm.

2. Package Overview

The POWERWORLD package was developed to clearly communicate the intricacy and complexity of many of the issues involved in power system operation. The design goals were to develop a simulation package which could accurately model power system operation in the longer time frame associated with industry restructuring, yet also be modular to allow easy interaction with external packages. This allows researchers to use the package both to study the impacts of different restructuring scenarios, and to clearly communicate their findings to non-technical audiences. The package runs in the Microsoft Windows® environment.

In general, the package simulates the operation of a multi-area power system over a specified period of time, typically from several hours to several days. The simulation runs at a rate proportional to actual time, but because of its emphasis on longer time frame concepts, this rate is usually many times actual time. The simulation reproduces the various continuous and/or discrete events that affect a power system. Examples would include continuous load variation or sudden loss of a transmission line.

The overall block-diagram representation of the package is shown in Figure 1, with directions of data flow as indicated. The blocks shown within the dashed rectangles are modules within the program itself; data flow is internal. The other blocks in Figure 1 are external files and programs; all data flow between these blocks is done using external files. In the following sections we provide a more detailed description of these blocks.

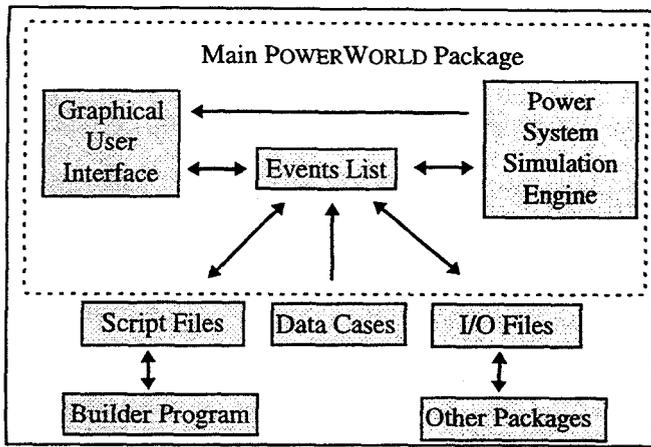


Figure 1: Package Block Diagram

3. Graphical User Interface

The main interaction between the user and the simulation is through a user-friendly, graphical user interface (GUI). In order for POWERWORLD to be an effective platform for studying industry restructuring, it is crucial that the GUI provide the user with easy access to just about all of the information associated with a power system model, even for large systems. To accomplish this, the GUI was designed using object-oriented programming (OOP) techniques. The advantages of this include flexibility for the user to interact with virtually all objects on the screen, small source and executable files, extensibility and reusability of existing code, and a consistent GUI.

The most important part on the GUI is the one-line diagram window. POWERWORLD allows the display of multiple one-line diagrams of virtually unlimited size. Panning and zooming are supported, along with conditional display of one-line objects. For example, Figure 2 shows a case modeling the 6400 bus North American Western Systems Coordinating Council (WSCC) system with just the high voltage transmission initially visible. Figure 3 shows this same system with the display zoomed into the San Diego area. Note that the lower voltage 230 kV transmission lines are now visible. Finally, Figure 4 shows a bus level view. When the package is running all windows are continuously updated with bitmap copies used to provide a "smooth" animation. The display update rate depends upon the computer's speed, size of the simulation case, and the number of windows shown. As an example, when run on a 166 MHz Pentium machine with 1024 by 768 display resolution, the WSCC case one-lines shown in Figures 2 through 4 update at about several times per second.

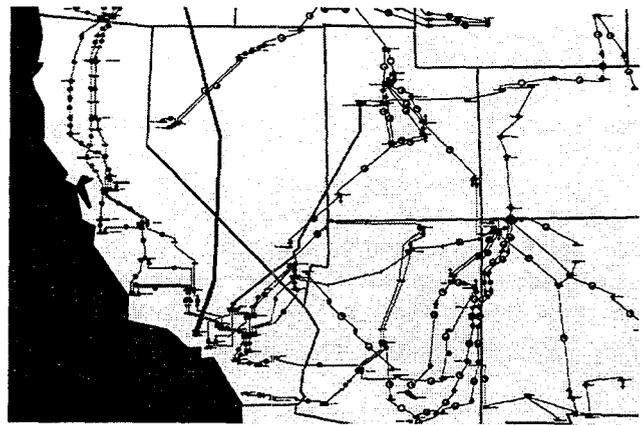


Figure 2: Southern Portion of WSCC System

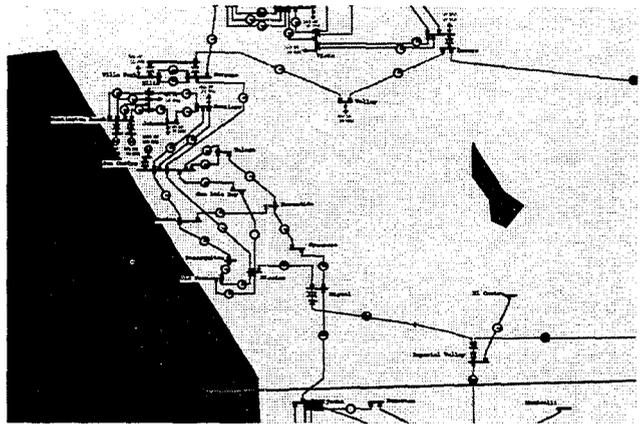


Figure 3: Zoomed View of San Diego Area

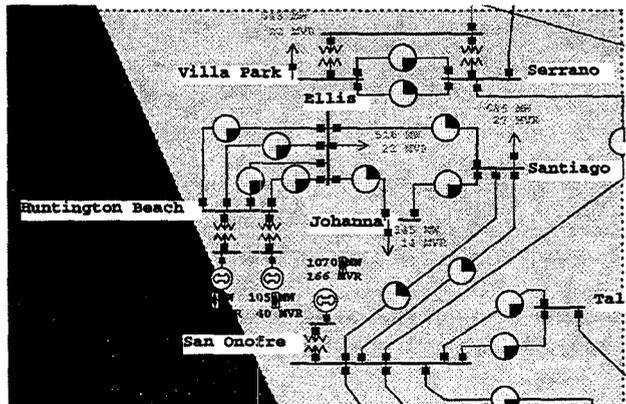


Figure 4: Bus Level View

The one-line uses graphical symbols to represent system equipment such as buses, generators, loads, transmission lines and transformers. A combination of numeric values and graphical symbols are used to show system values. For example numeric values are used to show bus voltage magnitudes, and load/generation values,

while graphical pie charts are used to show percentage line loading. In addition, because of the display's smooth refresh, animation can be used to convey additional information. For example, the user can optionally have colored circles moving along the transmission lines to indicate the direction and magnitude of real power flow within the system. While it is obviously difficult to convey the impact of this animation in a paper, Figure 5 shows a portion of a one-line for a 3900 bus model of the Mid-American Interpool Network (MAIN) system. For this example the circle sizes are also proportional to the real power flow on the lines.

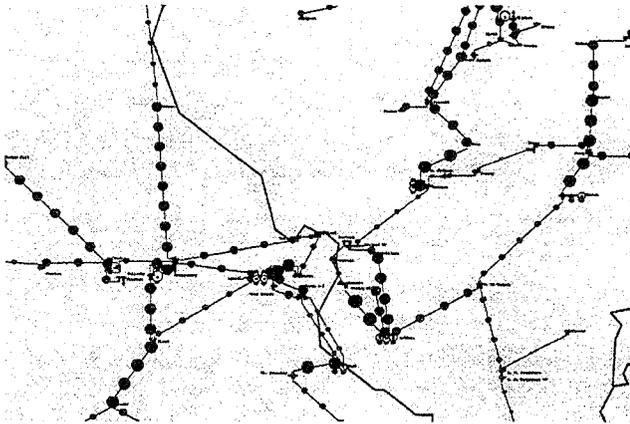


Figure 5: 345 kV Transmission in St. Louis Area

The object-oriented approach allows the user to interact with essentially all objects on the window, providing a convenient means both to control the system and to gain access to detailed system parameters. For example circuit breakers can be opened or closed by positioning the cursor on the circuit breaker symbol and clicking the left mouse button. Because of the fast display update rate, the new flows appear almost instantaneously. Similarly, the MW output of the generators is increased or decreased by positioning the cursor on the arrows shown by the generator MW value and clicking with the left button. However, right clicking on the actual device displays a submenu, allowing users to view either a dialog window in which they can view and change many of the parameters associated with the device, or graph windows showing generator cost curve information.

Another important display is the Area Transactions/Information Window. This display is used to provide a convenient summary of power transactions, both actual and scheduled, between different operating areas. Areas in POWERWORLD can represent either existing utilities or independent power producers (IPPs). As an example, Figure 6 shows this window for the MAIN case. Left clicking on the area box is used to display additional

information about an area and/or enter into scheduled MW transactions between areas. As is the case with the one-lines, the Area Transactions/Information Window supports zooming and panning.

Additional information about the power system is available through a number of list displays. These displays provide information about the entire case, irregardless of whether a bus or line is shown on the one-line diagram. Therefore, they provide a convenient means to view and modify case parameters and results. Examples include lists of bus records, generator records, transmission line/transformer records, limit violations, and power flow at system buses. The user can use the Area Filter display to limit the information shown on these displays to those records associated with particular areas. Additionally, many of these displays allow the user to sort the information presented in a number of ways, such as alphabetically by bus name or by bus voltage magnitude.

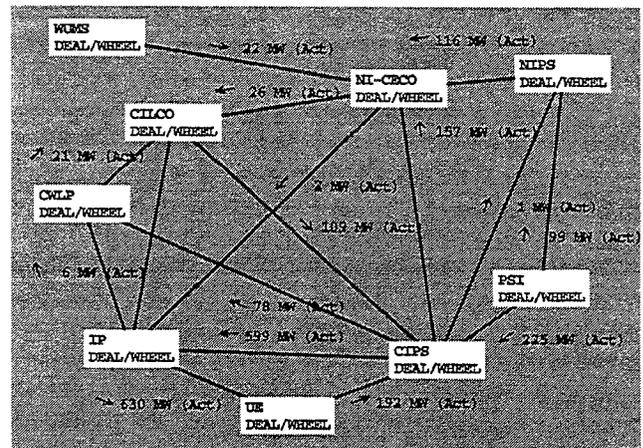


Figure 6: Area Display for MAIN

The GUI functionality discussed above is succinctly provided using the OOP techniques of encapsulation, inheritance and polymorphism. Encapsulation is the process of combining a data record and the procedures that affect that record into a single entity known as an object. The object hierarchy then defines the relationships between various objects, with descendant objects inheriting access to all its ancestor's procedures and data. Polymorphism is giving an action one name that is shared up and down an object hierarchy, with each object implementing the action in a way appropriate to itself [1]. For example, polymorphism permits clicking the left mouse button on the circuit breaker to open the device, while the same action on the generator MW value increases the generator's MW output.

All portions of the POWERWORLD simulation package, including the simulation engine, have been developed

using Borland's Delphi™ [2], an object-based application development environment based upon object-oriented Pascal. Therefore much of the general functionality needed to develop the GUI is already provided by Delphi. To implement the more specific GUI functionality, POWERWORLD actually uses two interdependent hierarchies of objects, one descending from the Delphi TForm object to represent the windows themselves, and one descending from the Delphi Tobject object to represent the devices shown on the windows. These hierarchies are shown in Figures 7 and 8.

The relationship between the two hierarchies is that a PokeObject (or any of its descendants) can be shown on any NumericWindow (or any of its descendants). Specific GUI functionality is then implemented as high up the hierarchy as possible. For example the functionality notifying a PokeObject that the user has clicked (or double clicked) on it with the left or right mouse button (i.e., "poked" the object) is implemented at the NumericWindow level. Likewise the generic functionality defining the "poke region" (the area on the window the object considers its own) is implemented at the PokeObject level, consisting of the rectangular region bounding the object. For most descendent objects this generic functionality is sufficient and thus needs to only be coded once. However some objects, such as multi-segment transmission lines utilize polymorphism to override this default functionality and only respond pokes that are sufficiently close to the line itself.

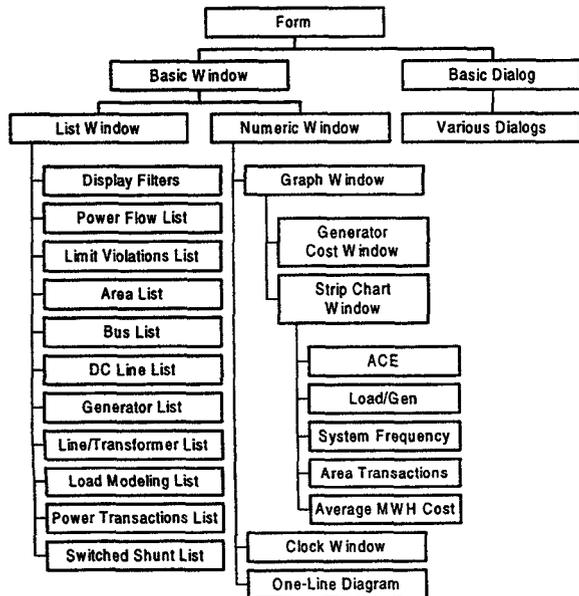


Figure 7: POWERWORLD Window Object Hierarchy

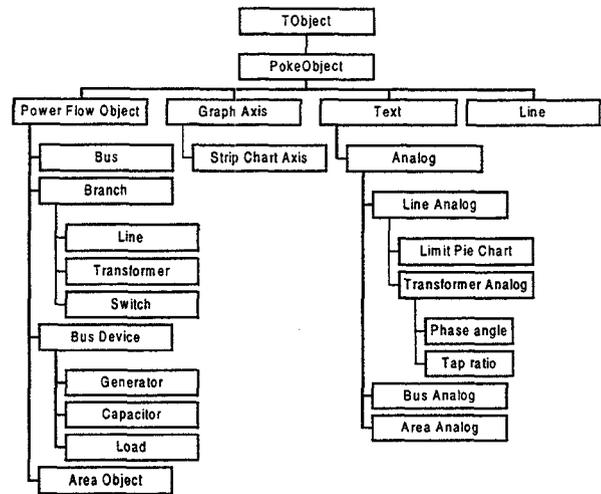


Figure 8: POWERWORLD Device Object Hierarchy

4. Simulation Engine

POWERWORLD was developed with a strong desire to provide a simple construct for explaining power system operation in the time frame of minutes to hours. We believe that most of the key concepts involved in industry restructuring on this time scale can be adequately simulated using a constant frequency model (i.e., the power flow approach). The simulation solution cycle is then as follows:

1. Read user selected simulation case, script file, and main one-line diagram information from disk files; set simulation time to start time, do initial events and perform initial power flow solution.
2. While simulation time \leq end time Do
 - a. Change load, perform any scheduled or user requested events, and, on an optional basis, introduce stochastic events.
 - b. Perform power flow solution, including network topology processing and enforcement of longer timeframe dynamics.
 - c. Update GUI and write any requested text output.
 - d. Update time according to a user selected scaling relative to real-time.
3. Stop the simulation

With the constant frequency model the heart of the simulation engine is a power flow solution. Because of the need to simulate longer time frames, simulation time is usually scaled to be many times real-time (typical values might be 30 or 60). For each time step in the simulation a number of adjustments are first made to the system. These include changing the loads at each bus according to a piecewise linear model, enacting any scheduled or user

requested events, checking for transmission line thermal limit violations, and, optionally, introducing stochastic events. Examples of scheduled or user requested events include opening circuit breakers, setting up MW transactions between areas, or changing the tap position for LTC transformers. All transmission lines are assumed to have thermal loading limits, with the line capable of indefinite operation for any loading below this value. However, for loadings above this limit the line temperature starts to increase. Line heating is approximated by integrating over time the square of the line current. Eventually, if the line flow does not decrease to a value below its limit the line is automatically removed from service. The only stochastic events considered are loss of transmission lines and/or transformers.

Following these adjustments, topology processing may need to be performed to take into account changes in bus connectivity. Topology processing permits the modeling of multiple islands, with an island marked as viable if it has sufficient generation reserves to match the total island load plus losses. Buses that are not connected to a viable island are marked as dead, with their load and/or generation set to zero. Conversely, loads at buses that had been dead but are now reenergized are immediately reset to their correct (time-dependent) values; generation is optionally either immediately set to its pre-outage value, ramped up to this value, or requested to remain at zero (to simulate when a generator has tripped off-line and cannot be immediately put back on-line).

Following topology processing the power flow is solved. Three solution methods are optionally available: the full Newton-Raphson algorithm, a fast decoupled power flow algorithm [3], or a Newton based optimal power flow (OPF) [4]. Sparsity techniques have been used to decrease both execution time and storage requirements. When using either the first or second solution method the package is able to solve systems with up to 20,000 buses. During the power flow solution longer timeframe dynamics are considered as time dependent constraints external to the main iteration, with the time period equal to the simulation time elapsed since the last time step. Examples include limiting the allowable change in generation to enforce ramp rate limits on generators, and limiting transformer tap changes according to transformer tap delays. When using the full Newton Raphson algorithm the optimal multiplier technique of [5] is used to prevent power flow divergence. However if the user stresses the system beyond its point of maximum loadability, no power flow solution exists and hence the power flow will not converge. If such a situation were to occur in actual operation, the system would probably experience a voltage collapse induced blackout. In POWERWORLD this is simulated by dimming all the windows to dark gray and sending a message indicating that a blackout has occurred.

Currently, the OPF solution method is limited to solving systems with up to approximately 300 buses. However, on such systems, by continuously tracking the optimal state, the OPF algorithm provides invaluable costing information. The following example illustrates the information available when using the OPF solution method.

A simple power system operating under the OPF solution method is shown in Figure 9. Information available from the OPF solution method includes the marginal cost of MW and MVAR generation at each bus. These costs could be used in developing pricing algorithms for ancillary services such as VAR production.

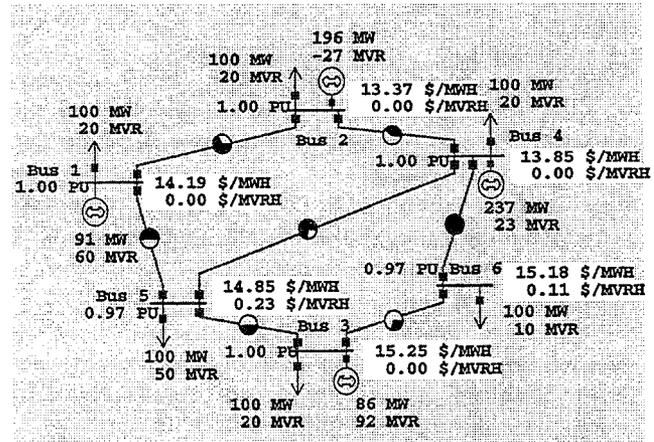


Figure 9: Six Bus, Single Area System on OPF Control

A major advantage of the OPF solution method is the ability to handle limits of power system components such as transmission line MVA limits. In Figure 9, the line between bus 4 and 5 is carrying approximately 70 MVA. When this limit is lowered to 40 MVA, the OPF will re-dispatch the generators in a manner which lowers the MVA flow on the line to 40 MVA. This is shown in Figure 10.

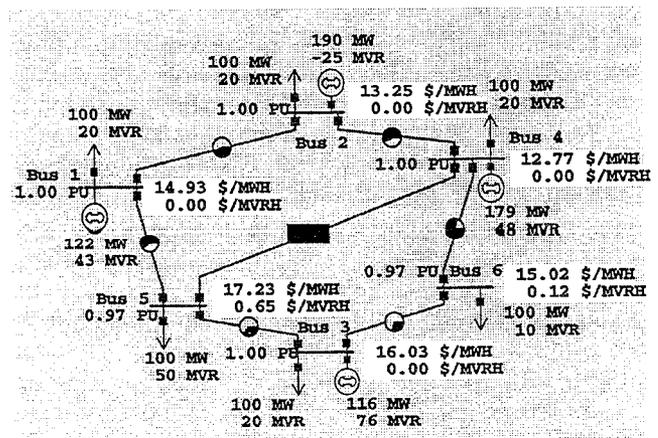


Figure 10: Line Limit Enforcement

The total system cost without the line limited is \$ 7824/hr. With the line limited this costs increases, as would be expected, to \$ 7895/hr. Also note the differences between the bus MW marginal costs in Figures 9 and 10. Because the generators at buses 2 and 4 were forced to decrease their output in order to remove the overload, their bus MW marginal costs also decreased. Conversely, the bus MW marginal costs at busses 1, 3, and 5 increased. As might be expected, the largest changes occurred at the ends of the limited line, buses 4 and 5.

The OPF solution method may also be used with multi-area power systems. The OPF will enforce the scheduled area interchange in these systems. In Figure 11, the simple six bus system is split into two areas as shown. For this case the generation in Area Two is less expensive than in Area One. Using the POWERWORLD Area Transactions/Information display scheduled transactions can be set up between the two areas to optimize their costs. Table 1 below summarizes several possibilities. As can be seen, the least expensive scenario for the sum of the areas is an interchange of 65.5 MW. This is of course the same solution as shown in Figure 9 for the single area case.

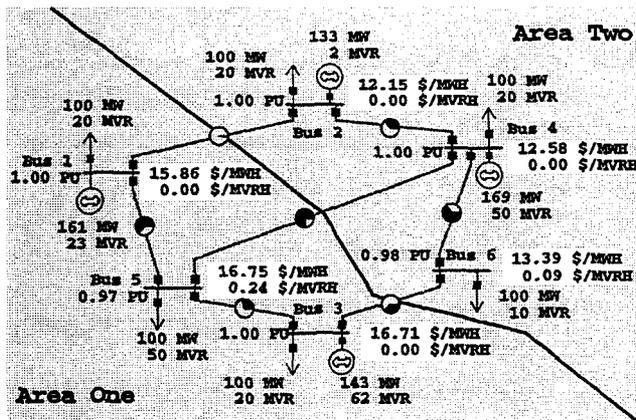


Figure 11: Six Bus, Two Area System on OPF Control

Table 1 : Sample Transaction Scenarios

Transaction [MW]	Area One Cost [\$/hr]	Area Two Cost [\$/hr]	Sum of both Areas [\$/hr]
None	4564	3496	8060
50.0	4489	3423	7912
65.5	4482	3413	7895
70.0	4481	3415	7896
80.0	4481	3428	7909

5. Interaction with Other Packages

Key to the versatility of POWERWORLD are its ability to play script files and its ability to dynamically interact with

other modules. As the name implies a script file contains specific action that take place automatically at specific times during the simulation. Through the use of script files, just about any action a user can do manually can be setup to occur automatically. In addition, it is possible to add written annotations that appear on the screen at various times throughout a simulation. Thus an entire demonstration of a simulation case can be setup beforehand and then simply "played" for an audience.

This functionality is implemented internally in POWERWORLD through the use of the events list. This linked list is used to hold both pending commands to the simulation engine and requests for data from the simulation engine. All entries in the list are time tagged with processing of the entry occurring when the simulation time has reached the tagged value. Referring again to Figure 1, note that all commands from the GUI pass through this list, including commands that only affect the GUI (such as resizing a window). Therefore implementing the scripting capability just required the development of an encoding module that takes a command from the script file (which is simply a text file) and inserts it into the events list. Once in the list the command is processed just as if it had come from the GUI. The easiest way to create a script file is to use POWERWORLD's "recording" capability. While recording, each entry in the events list is written into a script file after it has been processed. Thus recording allows the user to keep a permanent copy of a given POWERWORLD scenario.

The events list also provides the functionality to allow POWERWORLD to dynamically interact with external packages. These packages can be used to superimpose alternative paradigms for system operation. To accomplish this, at each time step in the simulation POWERWORLD first checks to see if an "input" file exists. The input file, which uses the same format as the script file, can be created by any just about any other type of program. If the file exists and can be successfully opened, POWERWORLD reads the input commands and places them in the Events List. The input file is then deleted. Likewise, at the conclusion of the timestep POWERWORLD writes any requested information into an "output" file. This output file can then be used by other modules.

The simulation of the so called minimum ISO, as presented in [6] and [7], can be used to illustrate this approach. The main idea behind the minimum ISO proposal is to give more freedom to electric market participants to engage in multilateral trades. The independent system operator (ISO) plays a more modest role compared with other proposed paradigms. Here the ISO is concerned only with system security, rather than also system economics. In the event of congestion, the ISO maintains system security through the use of cost information

Such a paradigm could be simulated using a multi-area model in POWERWORLD together with external modules (or players) to model the market participants. During the ex-ante period (before the current simulation time), market participants would determine all of the necessary multilateral trades. An external module would be used to setup these transactions. The scheduled transactions would then only need to be provided to POWERWORLD before their simulated start time. POWERWORLD would then simulate the system in the real-time period, providing the necessary feedback to the other modules of the current system state, and any other required information (such as sensitivity data). During the real-time period the researchers would be able to easily interact with the simulation through the GUI to determine how the minimum ISO paradigm responds to various system disturbances (such as contingencies or load not matching the forecasted demand). Finally, other modules could be used in the ex-post period to settle any outstanding charges.

6. Constructing Cases

Each simulation case has at least two associated data files. The POWERWORLD case file contains the information necessary to model the power system, while the POWERWORLD display file(s) contain the one-line information. The actual power system information is either defined in the beginning of the case file or in an external file using the IEEE common data format [8] or the PTI raw data format. The remainder of the file contains various simulation options and more detailed device modeling information. The display file(s) is a non-ASCII format file used to store the information necessary to display a one-line diagram. Each case must have at least one such diagram, but may have multiple one-line diagrams.

Both of these types of data files are created/modified using the separate Builder program. With Builder a user can graphically create a new case by simply selecting various power system devices from a menu and then placing them on a one-line diagram; the program automatically creates all the necessary files. This allows users to rapidly generate a wide variety of different systems. Users can also initialize a new case from an existing IEEE common format or PTI raw data format power flow file and then use Builder to create one or more one-line diagrams for the system.

7. Conclusion

Recent changes in the electric power industry have resulted in the need for a simulation and visualization tool for explaining the impacts of policy decisions on power system operations. This paper has described such a simulation package. The object-oriented design approach allows users great flexibility to dynamically interact with the power system simulation, while retaining a small size for the source and executable files. The use of data files and options windows allow users to simulate a wide variety of different operating conditions.

8. Acknowledgments

The research work underlying the results of this paper has been supported by the Power Affiliates program of the University of Illinois at Urbana-Champaign, the Grainger Foundation and the Edison Electric Institute (EEI). The authors gratefully acknowledge their support. The authors would also like to thank Kollin Patten for setting up most of the one-lines discussed in this paper.

9. References

- [1] *Borland Pascal with Objects User's Guide*, Borland International, Scotts Valley, CA, 1992.
- [2] *Borland Delphi for Windows User's Guide*, Borland International, Scotts Valley, CA, 1995.
- [3] B. Stott and O. Alsac, "Fast decoupled load flow," *IEEE Trans. on Power App and Sys.*, vol. PAS-93, pp. 859-869, May/June 1974.
- [4] D.I. Sun et al., "Optimal power flow by Newton approach," *IEEE Trans. on Power App and Sys.*, vol. PAS-103, pp. 2864-2880, Oct. 1984.
- [5] S. Iwamoto and Y. Tamura, "A load flow calculation method for ill-conditioned power systems," *IEEE Trans. Power App. & Sys.*, vol. PAS-100, pp. 1736-1743, April 1981.
- [6] F.F. Wu and P. Varaiya, "Coordinated multi-lateral trades for electric power networks: theory and implementation," Dept. of Electrical Engineering and Computer Science, Univ. of California at Berkeley, June 1995.
- [7] F.L. Alvarado, R.J. Camfield, and R. Rajaraman, "Open transmission access and congestion: an efficient minimal role for the ISO," L.R. Christensen Associates, April 1996.
- [8] H.E. Pierce Jr. et al., "Common format for exchange of solved load flow data," *IEEE Trans. Power App. & Sys.*, vol. PAS-92, pp. 1916-1925, Nov/Dec. 1973.