

Web-Based SCADA Display Systems (WSDS) for Access via Internet

B. Qiu H. B. Gooi

School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798

Abstract: The World Wide Web (WWW) has become a convenient way to access information on the net because the WWW browser integrates different network services into a common easily accessible user interface. These features coupled with low investment cost are especially suited for accessing information of the SCADA system. This paper describes a unique Web-based application which is implemented based on the client/server architecture. The user can view the real-time data superimposed on one-line diagrams generated automatically using the VLSI's placement and routing techniques. In addition, the user can also control the operation of the substation at the server site. The choice of the Java language offers unique and powerful features such as zero client installation, on-demand access and platform independence to the design of the SCADA display system.

Keywords: Java, Internet, WWW, SCADA, Client/Server, GUI, Auto-Generation, One-Line Diagrams.

1. INTRODUCTION

SCADA systems are essential parts of the DMS and EMS that employ a wide range of computer and communication technologies. A SCADA system gathers incoming power system data for further processing by a number of distributed processes. The control commands for operating a power system can also be issued through the SCADA system. The SCADA system is characterized by geographical spread, as well as the need to match the control hierarchy with the organization hierarchy.

The privatization of electric utility companies all over the world has brought along not only competition in the electricity supply industry but also numerous changes in the SCADA/DMS/EMS. One of the changes requires that the EMS exchange information with external organizations such as generators, distributors and brokers. This data exchange is neither time critical nor needed on a dedicated basis, but it requires quick access and data security. Technological advances in Internet and WWW have made it possible to develop a low cost WWW display system for accessing information via tabular displays and one-line diagrams on the net.

Currently, the generation of one-line diagrams for SCADA/EMS/DMS applications is done manually. The process is tedious, time-consuming and error-prone. This paper presents a method to auto-generate the one-line diagrams based on the placement and routing algorithms [1,2,3,4] proposed for the VLSI design.

Graphical User Interfaces (GUI) [5,6,7] of various complexities have been developed to simulate and display the dynamics of power systems. They are mostly stand-alone

programs. SCADA systems of various designs [8,9,10] have also been implemented for monitoring and controlling the power system. However, they are normally not designed for Internet access.

Web-based conception has been elaborated and implemented [11,12] by researchers all over the world since it was first introduced in the early 90s. Their experience has indicated that Java can be successfully employed in the real-time system. In this paper, a new Web-based SCADA display system is described.

2. SOFTWARE TECHNOLOGIES

2.1 Client/Server Architecture

Traditional enterprise applications are, for the most part, self-contained monolithic programs that have limited access to one another's procedures and data. They are usually cumbersome to build and expensive to maintain because even simple functional changes require the entire program to be rewritten, recompiled, and re-tested. By contrast, the client/server structure provides the scalability and robustness required to support mission-critical applications throughout the enterprise comprising thousands of users. This clean, modern architecture which is object-oriented, server-based and database driven is ideally suited for web-based workflow solutions. In this paper, a typical three-tiered client/server applications architecture which contains a service tier, a data store tier and a MMI tier is shown in Fig. 1.

2.2 Network Programming

Networking enables programs to retrieve information stored in computers located anywhere in the world. People can communicate with each other on the net. Many computers can be harnessed to work on one problem. Network programming is ideal for the SCADA display project for two reasons:

- Easy to access - The SCADA real-time data can be accessed by an authorized user anywhere. For example, the current operating condition of the Singapore power system can be accessed and visually viewed by an authorized technical consultant and equipment manufacturer in America on the net.
- Low investment - No special hardware investment is needed for Internet access. All the user needs is a web browser for linking the WSDS Server machine. The data transfer is completely transparent to the user.

2.3 The Java Language

Java is the first programming language designed from the ground up with networking in mind. As the global Internet continues to grow, Java is uniquely suited to build the next generation of network applications. It provides solutions to a number of problems which are difficult to address in other

programming languages. The level of safety with Java applets (programs which adhere to a set of conventions that allow them to run within a Java-compatible browser) is far greater than what can be obtained from other software. Java makes writing networking programs easy [13]. It is relatively straightforward for Java applications and applet to send and receive data and to communicate across the Internet, limited only by security restrains.

Java is portable and platform-independent. In the past, software developers had to work with a specific native instruction set which locked them into a specific hardware and operating environment. Unlike other programming languages, Java executes in a run time environment called a virtual machine. The Java virtual machine executes bytecode (platform independent code) that a Java compiler generates and it can be incorporated or embedded in Web browsers, or the kernel of the operating system. Java virtual machines and Java APIs (Application Program Interfaces) insulate the Java program from hardware dependencies. Using them, Java's bytecode can run on a wide range of platforms, much larger than what C and C++ would allow. The Java bytecode helps make the "write once, run anywhere" possible.

Currently, a program written in the Java language runs a bit slower than the native code. However, with the wider bandwidth, smart bytecode compilers, well-tuned interpreters, Java can perform close to that of native code without threatening portability.

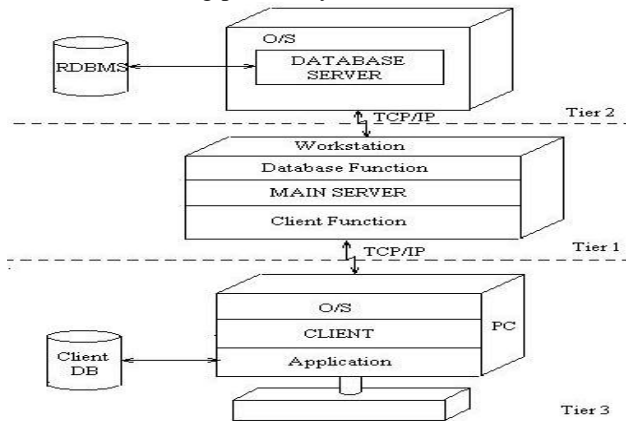


Fig. 1. Three-Tier Client/Server Applications Architecture

2.4 Database Programming

SQLJ [14, 15] is a way to embed the Structured Query Language (SQL) in Java programs and to reduce the development and maintenance costs of Java programs that require Database connectivity. SQLJ provides a simple model for Java code containing SQL statements. SQLJ offers a much simpler and more productive programming APIs than JDBC (Java Database Connectivity) to develop applications that access relational data and it can communicate with multi-vendor databases using standard JDBC drivers.

3. SYSTEM DESIGN

Based on the above three-tier architecture, a software prototype named PowerNet is developed. PowerNet is a workflow software that models, coordinates, and integrates

some of the SCADA functions such as the one-line diagram generation and real-time data display. Fig. 2 describes the main configuration of PowerNet. JavaCON, which stands for Java connectivity to SCADA, is used as a data communication bridge between the SCADA system and the WSDS Server. The WSDS Server receives messages from the SCADA system through JavaCON and provides the necessary support for MMI.

MMI accessed through a network browser is used for displaying the SCADA system data. MMI provides a means for utility engineers, planners and higher management to access the SCADA system data. The functionality of the MMI, for most of the time, is only limited to viewing the overall picture of the system being monitored. People accessing MMI do not take active part in controlling the system. The MMI at the client site has the same graphical user interface as that of the server site. This design enables the user to view the details of the substation conveniently. When the user accesses the WSDS Server, a security mechanism is triggered to allow only the authorized user to view the SCADA information.

The WSDS database stores the static and dynamic data of the SCADA system. The Oracle RDBMS is used for storing historical data. It provides a convenient data accessing mechanism. The historical data consists of logged data, events and dynamic data of one-line diagrams. The system status and the real-time data are stored in the database every 5 seconds or at a periodicity specified by the user. Access to the database is provided through a series of interface modules that read and write data to the database tables.

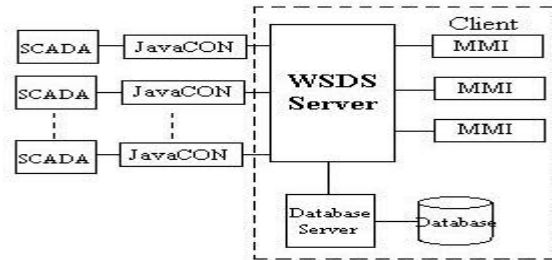


Fig. 2. Main Configuration of PowerNet

The WSDS Server is written entirely in Java and does not require any native code. A Java interpreter is required to parse the bytecode each time the WSDS Server program runs. As shown in Fig. 3, the server supports four roles: Client function, JavaCON function, MMI function and Database operation.

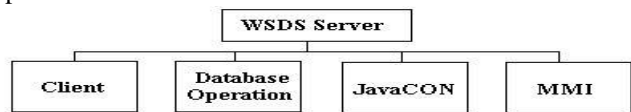


Fig. 3. WSDS Server Function

3.1 JavaCON

JavaCon communicates with the remote SCADA system based on TCP/IP and sockets (end points of a two-way communication link between the two programs running on the network). The WSDS Server gets the real-time analog data, breaker status information, transformer taps from the

SCADA system, and displays them on the screen. Multithreading is used to process multiple JavaCONs. When a JavaCON communicates with the WSDS server, a thread will be set up to process the data transmission from the JavaCON.

3.2 Database Operation

The database operation on the server site is through JDBC and SQLJ. JDBC is a Java API for executing SQL statements. It lets the Java program send SQL statements to the appropriate database. Two sets of JDBC drivers are used to operate two different databases. One is JDBC-Net pure Java driver which translates JDBC calls into a DBMS-independent net protocol which is then translated into a DBMS protocol by the server. This driver is mainly used for the network Oracle database manipulation. Another driver is JDBC-ODBC (Open Database Connectivity) bridge plus ODBC driver. This bridge provides JDBC access via ODBC drivers which is used for the host database. Three steps are executed before using the JDBC API to access the database. They are 1) Import the JDBC classes; 2) Register JDBC driver and 3) Open a connectivity to the database. After these steps, the Java program can talk to the database.

The substation database contains data which include static data such as component names and positions and dynamic real-time data. The linkage between static and dynamic data tables is the component name.

3.3 Client

Java makes programming for communication between the client and server easy [11,16]. The communication between server and client contains three parts: the server site program, the client site program and the protocol between them. The multithreading structure enables the server handle multiple client requirements concurrently. When the client site program is started, it connects to the WSDS server and requests for services. The server site will start a thread to handle this client. Every client must have a user name and identification number so that only authorized users can visit the SCADA system information. The rendezvous between the client and the server is realized by using the specified protocol. After these steps, the client can visit the real time SCADA system.

3.4 Main Machine Interface (MMI)

The MMI is supported by the auto-generation of one-line diagrams and GUI displays. As shown in Fig. 4, the auto-generation of one-line diagrams contains stream input data, placement, routing, dynamic data linkage and graphic editor. The stream input data supply substation component information for creating the topology of the substation. The vertices of the topology without the components will be placed using the VLSI's placement algorithm. Routing will route connections of all the components according to some connectivity priorities. The component symbols are then added to the topology. Dynamic information such as voltages and MW/MVar flows will be automatically linked to each component to display the updated SCADA information on the screen. The final layout will be stored in the ASCII file.

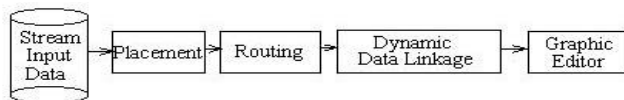


Fig. 4. Process Involved in One-Line Diagram Generation

3.4.1 Stream Input Data

The stream input data contain the connectivity information of the network. They include the component's names, their start and end nodes, component ratings and modeling parameters. When the data are read in by the Auto-Generation module, a substation topology is created according to the connectivity of the components. The topology is accessed by the placement and routing modules.

3.4.2 Placement

Component placement determines the location of the component within the one-line diagram. It determines the minimum length of interconnection and minimizes line crossovers. It distributes the nodes of components evenly in the frame. Placement is crucial in the overall design cycle as its output must be routable, The overall quality of the layout is determined in this stage.

In this project, the Force-Directed method is used for the placement of power components. The method models the graph as a physical system of rings and springs. In this context, rings represent the vertices in a graph and springs represent the edges of a graph. Repulsive forces are calculated between every pair of vertices, but attractive forces are calculated only between neighbors. The method treats the placement problem as a process of reducing the total energy for a system of steel rings interconnected by springs. By minimizing the sum of compression or tension on all the springs, the rings would be most nearly at their ideal distances from one another. The energy of the system state is:

$$\sum_{1 \leq i < j \leq n} k_{ij} (|\mathbf{V}_{pos,i} - \mathbf{V}_{pos,j}| - l_{ij})^2 \quad (1)$$

where i and j are vertex numbers; n is the total number of vertices; $\mathbf{V}_{pos,i}$ or $\mathbf{V}_{pos,j}$ is the position of the ring corresponding to vertex v_i or v_j ; k_{ij} is the spring constant for the spring between $\mathbf{V}_{pos,i}$ and $\mathbf{V}_{pos,j}$; l_{ij} is the optimum distance between vertices v_i and v_j . In a fixed frame of width W and length L , the area of the frame is

$$area = W * L \quad (2)$$

and the optimal distance between the two vertices is:

$$l_{ij} = \sqrt{\frac{area}{n}} \quad (3)$$

This provides the ideal distance between vertices because the two forces would exactly cancel each other. If f_a and f_r are attractive and repulsive forces respectively, and d is the distance between the two vertices, then

$$\begin{aligned} f_a(d) &= d^2/k \\ f_r(d) &= -k^2/d \end{aligned} \quad (4)$$

In a graph, each vertex v_i has two vectors: $\mathbf{V}_{pos,i}$ and $\mathbf{V}_{disp,i}$, where $\mathbf{V}_{pos,i}$ is v_i 's position and $\mathbf{V}_{disp,i}$ is v_i 's displacement departing from its initial position. The vertices are assigned random initial positions when the algorithm starts. The

locations of vertices are constantly updated according to the attractive and repulsive forces among the vertices. In each iteration, there are four steps:

1. Calculate the effect of attractive forces on each vertex. Here, the attractive force will be calculated between two vertices v_i and v_j on a connectivity line.

$$\mathbf{V}_{disp,i} = \mathbf{V}_{disp,i} - (\Delta/|\Delta|) * f_a(|\Delta|) \quad (6)$$

$$\mathbf{V}_{disp,j} = \mathbf{V}_{disp,j} + (\Delta/|\Delta|) * f_a(|\Delta|) \quad (7)$$

where $|\Delta|$ is the distance between two vertices in a connectivity line.

$$|\Delta| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (8)$$

$$\Delta = \mathbf{V}_{pos,i} - \mathbf{V}_{pos,j} \quad (9)$$

2. Calculate the effect of repulsive forces. The repulsive force will be calculated between any two vertices v_i and v_j .

$$\mathbf{V}_{disp,i} = \mathbf{V}_{disp,i} + (\Delta/|\Delta|) * f_r(|\Delta|) \quad (10)$$

3. Modify the position of a vertex.

$$\mathbf{V}_{pos,i} = \mathbf{V}_{pos,i} + \mathbf{V}_{disp,i} / |\mathbf{V}_{disp,i}| \quad (11)$$

4. The energy of the system state will then be calculated using (1). The iteration will continue by returning to step 1 and until the energy falls below a preset threshold. At that time, the position of each vertex will be determined.

The optimal positions of vertices determine the vertex placement of the one-line diagram. Fig. 5 shows the diagram of how the vertices of a sample system are placed using the Force-Directed method.

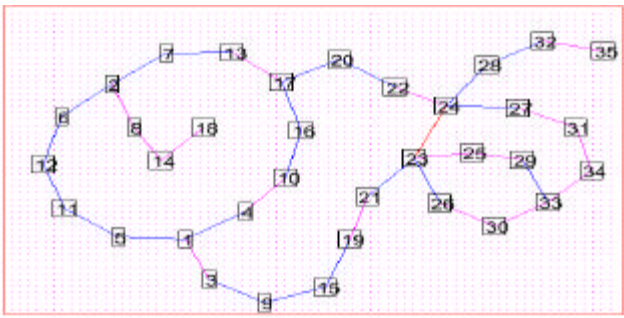


Fig. 5. Diagram of Force-Directed Placement of a Sample system

After placement, the layout must satisfy some general accepted aesthetic criteria, but some adjustments are still needed to make it more fitful for the requirements of the one-line diagram. For example, the transformer and its related breakers should be at the same x- or y-coordinate. Different sections of a main busbar in a substation should be at the same x-or y-coordinate. Simple AI algorithms are employed. If needed, the operator can also use the mouse to manually modify the final layout according to each individual taste.

3.4.3 Routing

After placement, the diagram is ready for line routing. In this paper, the Hightower algorithm is chosen. This method does not store the entire grid in its memory. Instead it only stores lines. So it executes faster and has a lower memory

requirement than that of Lee and Maze algorithms [4]. In this paper, some modifications were added to the Hightower algorithm to make it more efficient in route searching.

Fig. 6 shows a diagram routed using the modified algorithm. The dashed lines represent projected horizontal and vertical lines while the solid lines represent existing blockades in the routing area. A blockade is defined as the cover of a point on the routing surface if any horizontal or vertical lines projected from the point intersect the cover. An escape line is a horizontal or vertical line passing through the point. An escape point is a point on the escape line which has at least one side not obstructed by any cover. Otherwise the new escape line from this escape point cannot be projected. Points B and C are the source and target respectively. Lines V, W, X, Y and Z are blockades. Points A, D and E are escape points. Lines e, f, g, h, i, j and m are escape lines. Point F is the intersection where two escape lines meet.

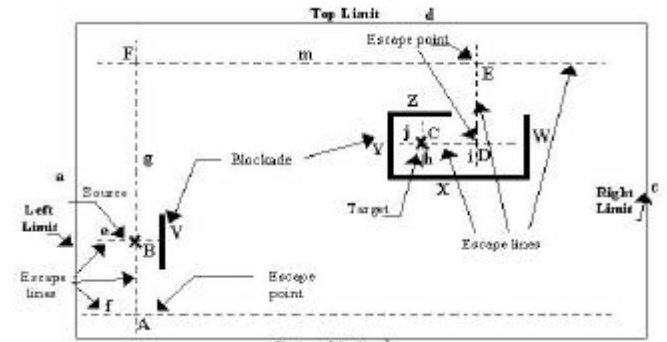


Fig. 6. Hightower Routing Method

In Fig. 6, escape lines e and g are projected from source point B. Escape lines h and j are also projected from target point C. Escape lines e, h, and j encounter boundaries or obstacles. Only escape line g has escape point F which can connect to the target point. D is an escape point along escape line j from target point C. Escape point D connects to E, an escape point along escape line i. From escape point E, escape line m meets escape point F. Thus the source and target connect with each other.

The basis of line routers is to project horizontal and vertical lines from both the source and target points on the routing surface. These lines are projected as far as possible. If during the line projection process, two lines from the two different points intersect, then a connection is found. On the other hand, if an obstacle is encountered or the edge of the routing area is reached, a new escape point is chosen along the current line. New horizontal and vertical lines are projected from the escape point. This process continues until a connection is made.

The original Hightower algorithm stores every occupied point in the cell array. The cells with fixed vertices are called occupied cells. When the search line expands from the source or target point, it must compare to each occupied cell stored in the array. If the grid is large, this comparison is very time-consuming. In the modified algorithm, a block of occupied cells are stored as a line. When the search line expand itself, it only compares with the start and end points of a line to decide whether it hits the obstacle or not. The main algorithm

expands the horizontal and vertical lines from both the source and target points to find the connectivity. When the connection is found, the Retrace module is used to find the lines composing horizon and vertical connectivities. It then stores the lines into the occupied cell array and draws the lines on the screen.

One of the disadvantages mentioned using line routing is that an earlier connection tends to block those that are to be routed later. This problem is mitigated by routing the paths according to some priority order. In this paper, the algorithm modifies the connections according to the area of rectangle formed by the line. In Fig. 7, S1 to S4 are four source points, and T1 to T4 are four target points. A pair of source and target points forms a rectangle area. The priority of the connectivity is weighted according to the number of points included in the rectangle area. The more the points included in the rectangle area, the lower the priority is. In Fig. 7, the S1-T1, S2-T2, S3-T3, S4-T4 pairs contain 5,3,1,0 point(s) respectively and thus their weights are 5,3,1,0 respectively. So the S4-T4 pair will be routed first, followed by S3-T3 and S2-T2 pairs and finally the S1-T1 pair.

The sample system in Fig. 5 is processed using the Hightower routing method discussed above. After routing, the layout shown in Fig. 8 is obtained.

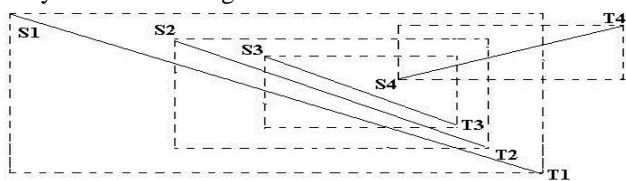


Fig. 7. How Priority of Connectivity is determined

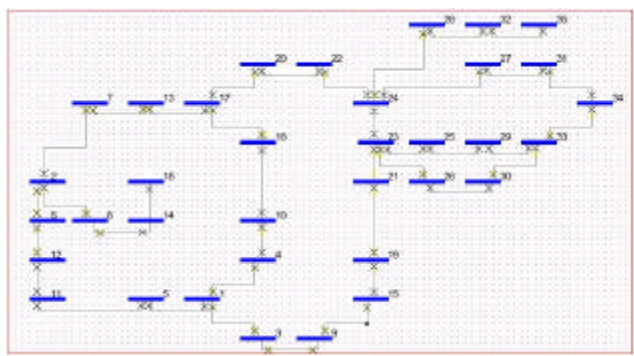


Fig. 8. Final Layout of One-Line Diagram after Routing.

3.4.4 Dynamic Data Linkage

After routing, the SCADA system data will be dynamically linked to the static one-line diagram. The real and reactive power (MW/MVar) and/or voltage of the power system components will be linked to the database and displayed at the position close to the respective component. When the display is refreshed each time, their values will be retrieved automatically from the database and updated on the one-line diagram.

3.4.5 Graphic Editor

Full graphical editor functions are supported by the GUI at the WSDS Server site. Within a substation, components

can be manipulated and edited via the mouse and keyboard. Component data may be edited by double clicking the mouse on the respective component which in turn calls up a dialog box. Panning and zooming are also supported when viewing the details of a network or a substation. Uniformity throughout the GUI is achieved via the use of a small number of mouse manipulations that trigger the same class of response for each object type. The applicable manipulations are select, drag and double-click. Multiple objects may be selected and the object's attributes may be altered via a pop-up dialog.

At the server site, control may be issued to open or close breakers through the one-line display. After the SCADA receives the command, the display will reflect the new status of the breaker.

4. LABORATORY IMPLEMENTATION

The laboratory setup consists of three major modules running on three computers as show in Fig. 2. The first module is the SCADA simulator that simulates the real-time SCADA and runs on a PC. The SCADA simulator runs a Power Flow program which provides all the analog and digital data that are normally supplied by the real-time SCADA system. The second module is the WSDS Server program that runs on a SUN workstation equipped with the Solaris 2.5 operating system. Netra-i version 3.2 is employed to support Web publishing and gateway access server function. Slstice FireWall-1 3.9 is installed to provide a data security for the enterprise network in a distributed environment. The third module is an Internet browser such as Netscape or Explorer, which runs on a PC. Communication between the WSDS Server and Java Client is linked by the high speed Ethernet within the campus. The system database for PowerNet resides in the Delta server of the University Computer Center.

4.1 Real-Time Power System Simulator

The real-time power system simulator is developed using Visual C++ 5.0. The power system model is initially loaded into the simulator from the Access 97 database. Due to change of load, the Power Flow calculates a new solution every 5 seconds and the results are stored in the Access database. Through the simulator, the user can open or close the breaker on the one-line diagram. This triggers a new Power Flow solution in the next update.

JavaCON retrieves the data from the Access database and sends the data to the WSDS Server. The communication is based on the TCP/IP protocol and socket. The WSDS Server receives the SCADA data through JavaCON. It stores the data in the Oracle database, and displays the data on the one-line diagram. Subsequently it waits for further processing requests from the client.

4.2 Client Displays

The Java MMI program establishes its connection to the server by typing in the server's IP address. The MMI consists of multiple displays, each giving a unique window into the overall system condition. The highest level is the map of

Singapore that presents an overview distribution of the substations. The second and third levels represent a smaller area of the entire Singapore map but with more details in the area displayed. The lowest level of the display is the substation's one-line diagram. Through the zoom in and zoom out facility, the user can view the details of every substation component.

Consider a hypothetical substation in the Buona Vista area of Singapore. Figs. 9-12 display the steps to view the one-line diagram of the hypothetical substation. The hypothetical substation is accessed by selecting the rectangular area to be enlarged at each level. The dynamic data of the one-line diagram are being updated. Data such as voltages, MW/MVar flows and breaker statuses of the simulated power system are automatically transferred from the server's Oracle database.

5. CONCLUSIONS

This paper has described a Web-based SCADA display system designed for the WWW. The object-oriented design approach and the client/server module allow the user great flexibility to dynamically interact with the SCADA system. The one-line diagram can be generated automatically using the VLSI's placement and routing algorithms. A laboratory implementation of such a system is developed for testing the



Fig. 9. Map of Singapore

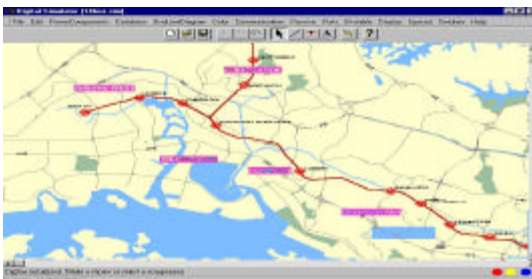


Fig. 10. Map of Buona Vista

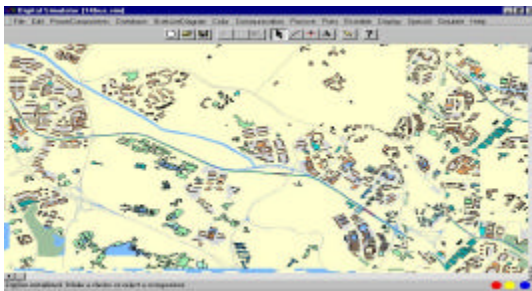


Fig. 11. Detailed Map of Buona Vista

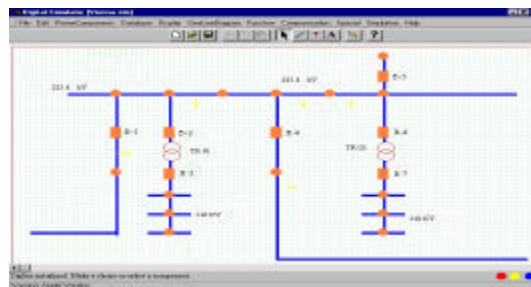


Fig. 12. One-Line Diagram of A Hypothetical Substation

capability of the system. It implements some features that take advantage of Java and Internet's capabilities [16]. The SCADA system can be accessed from anywhere in the world via the Internet. No special hardware and software and application are required at the remote location.

6. ACKNOWLEDGEMENT

The authors wish to thank Mr Richard Koh of InfoGrid Pte Ltd for the use of static pictures from Bizmap Navigator.

7. REFERENCES

- [1] H B Gooi, A N Arunasalam, "Automatic Generation of One-line Diagrams", *International Power Engineering Conference*, Singapore, 1993.
- [2] M J Thomas, "Graph Drawing by Force-directed Placement", *Software Practice and Experience*, pp 1129-1139, Nov, 1991.
- [3] J Lee, J H Chou, "Hierarchical Placement for Power Hybrid Circuits under Reliability & Wireability Constraints", *IEEE Transactions On Reliability*, Vol 45, No 2, pp 200-207, Jun, 1996.
- [4] S M Sait, H Youssef, "VLSI Physical Design Automation: Theory and Practice", McGRAW-Hill Book Company, 1995.
- [5] T J Overbye, P W Sauer, "A User-friendly Simulation Program for Teaching Power System Operations", *IEEE Transaction on Power System*, pp 1725-1732, Nov, 1995.
- [6] M Foley, Y Chen, A Bose, "A Modern Digital Simulation Laboratory for Power Systems", *IEEE Computer Application in Power*, pp 16-19, Apr, 1990.
- [7] D C Yu, N Jamal, "A graphical User Interface for Design, Simulation and Analysis of Power Plant Electrical Auxiliary Systems", *IEEE Transactions on Energy Conversion*, pp. 263-269, Jun, 1994.
- [8] D Shirmohammadi, E Liu, "Distribution Automation System with Real-time Analysis Tools", *IEEE Computer Application In Power*, pp. 31-35, Apr, 1996.
- [9] M J Menz, B Payne, J.R., "Servers in SCADA Applications", *IEEE Transactions on Industry Applications*, pp.1295-9, Sep, 1997.
- [10] D Guilfoyle, E Connolly, "Distributed SCADA Systems for electricity Distribution Control", *Power Technology International*, pp. 169-172, 1994.
- [11] Ulrich Gall, Franz J. Hauck, "Promondia: A Web-Based Framework for Real-Time Group Communication in the Web", *Computer Networks and ISDN Systems*, pp. 917-926, Sep, 1996.
- [12] K Nilsen, "Java for Real-time", *Real-Time Systems*, pp. 197-205, Sep, 1996.
- [13] E R Harold, "Java Network Programming", O'Reilly & Assoc., 1997.
- [14] Oracle Web Page, [http://www.oracle.com/products/free_software/index.htm].
- [15] K North, "Java, JDBC, Stored Procedures and Servermania", *WEB Techniques*, pp. 22-26, May, 1998.
- [16] E Yourdon, "Viewpoint: Java, the Internet Will Reshape Software Engineering", *IEEE Spectrum*, pp. 66-67, Jan. 1997.

Qiu Bin is a Ph.D. candidate of Nanyang Technological University. His main research interests are computer and network application in the Power System. His email address is: p143278020@ntu.edu.sg.

H. B. Gooi is a Senior Lecturer at Nanyang Technological University. His main research interests are power system operation, economics and computation. His email address is: ehbgooi@ntu.edu.sg.