Synchronized Sampling Uses for Real-Time Monitoring and Control/RTGRM

Implementation Specification for Visualization Tools

Project Progress Report

Mladen Kezunovic

Ce Zheng

Texas Engineering Experiment Station

December 31, 2008

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

Table of Contents

Executive Summary

This report is the third deliverable from several consecutive research and development activities. Prior work on the project explored requirement specifications and functional design for the control center visualization tools.

During this phase of the project, the implementation specification of our proposed graphical user interface (GUI) software has been developed and explained in detail. The main research achievements are:

- The overall implementation flow chart of the software has been developed and explained;
- Both external and internal logic of the software represented in the implementation flow chart have been specified;
- The Object-oriented Programming (OOP) technique has been selected as the primary programming technique;
- The software programming flow chart has been defined and general programming sequence has been established;
- Two types of data interpreters have been explained; detailed inputs and outputs of the interpreters have been specified;
- The programming specification of all GUI modules within the graphical software has been outlined and block diagrams for each module has been specified;
- Options for receiving satellite images from commercial providers have been investigated;
- The hierarchical view, which is for normal state monitoring purpose, is set to be a combination of four other views: model, equipment, aerial and topological.
- Implementation plans for software maintenance and security considerations during development have been specified;
- A general schedule for developing the graphical software is established, which has specified three phases throughout the programming process;

Since the implementation specification for the graphical tools has been outlined, our investigation will now move on to the development of the control center visualization software. As soon as the whole development of GUI is completed, on-site demonstration using field cases will be carried out.

Part 1: Introduction

Currently large amounts of data are collected by Phasor Measurement Units (PMUs), and other Intelligent Electronic Devices (IEDs) at the substation level. Beyond the traditional Supervisory Control and Data Acquisition (SCADA) data collected by Remote Terminal Units (RTUs), very little PMU or other IED data are integrated into current Energy Management System (EMS) solutions [1-2].

The goal of this project is to investigate the potential benefits of integrating information obtained by PMUs and other IEDs into the control center real-time monitoring and control solution. This report describes development of new software aimed at automated fault location and visualization. A complete implementation specification of the control center visualization tools is investigated in detail and reported earlier [3-4].

The integration of data sources and control center GUI is shown in Figure 1.

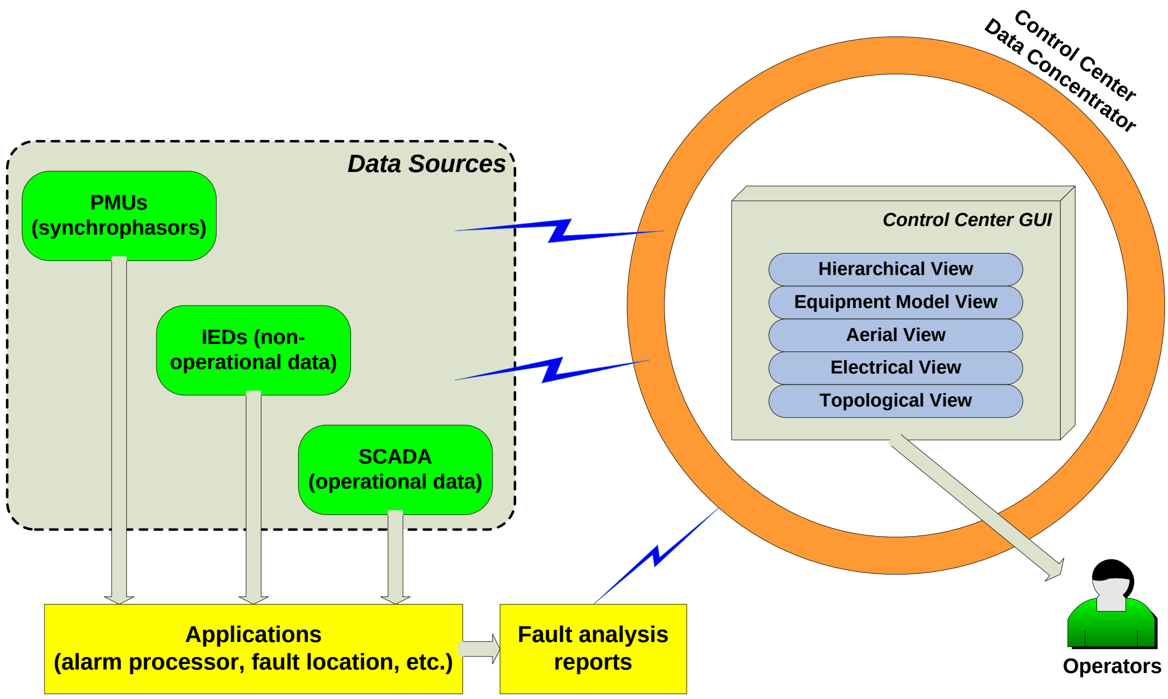

Figure 1: Data sources and control center GUI

Input information such as data from PMUs, IEDs and SCADA, together with application analysis reports are collected by the control center data concentrator (usually a server database). All input information will then be converted and sent to the control center GUI. After internal information processing, the graphical software will display several kinds of graphical view, as outputs, through the five visualization interfaces.

Part 2 will investigate software implementation specification. Other considerations are discussed in Part 3. The schedule for software development has been provided in Part 4.

Part 2: Implementation Specification

2.1 Implementation flow chart

The overall implementation flow chart of the GUI software is shown in Figure 2. Embedded in the flow chart are two types of logic: external (yellow) and internal logic (blue).

2.1.1 External Logic

The external logic explains relationship between applications and GUI software, as well as their implementation sequence.

As demonstrated in Figure 1, after being installed in control center client computers, the software will first be set to its starting status. All inactive software programs will be started up and all running threads will be restarted. Considering the number of applications and user interfaces incorporated in the software, this process may take up to one minute.

As soon as the startup has been completed, a message will be sent to the control center server database automatically, which tells the server computer that real-time data and reports are expected.

On the other hand, the control center server computer will retrieve data and reports from various sources. The data sources include the PMUs, IEDs, and RTUs of SCADA. A control center data concentrator will be responsible for collecting and converting the format of these data and transferring them to the server database. The reports include all application analysis results, e.g. fault reports obtained from Intelligent Alarm Processor, Optimal Fault Location, Cascading Analysis and Optimized Maintenance applications.

After receiving the message from GUI software, the server database will immediately send the requested data files and reports to the client computers through a Local Area Network (LAN). This is a continual process which means the updated files will be sent from server to clients uninterruptedly according to a beforehand configured time interval (usually every several milliseconds).

Coming back to the GUI software, data and reports transmitted from the server will be first interpreted to applicable data files. Then, a hierarchical view will be displayed in the normal state monitoring interface. A time interval for scanning will be configured the same way as it is done for updating frequency of the data collection (every several milliseconds). Thus being configured, the software could scan every data package sent from server.

Next is the logic of decision making: is there a fault or not? If not, the software will maintain current normal state monitoring interface and no alarms will be presented. The scanning thread will wait for the next data package. If yes, the software will enter the fault event mode, most of which relates to the internal logic.

Last but not least, within the fault event mode, the software will need to retrieve fault reports from all applications. Operators will decide which actions to take according to the

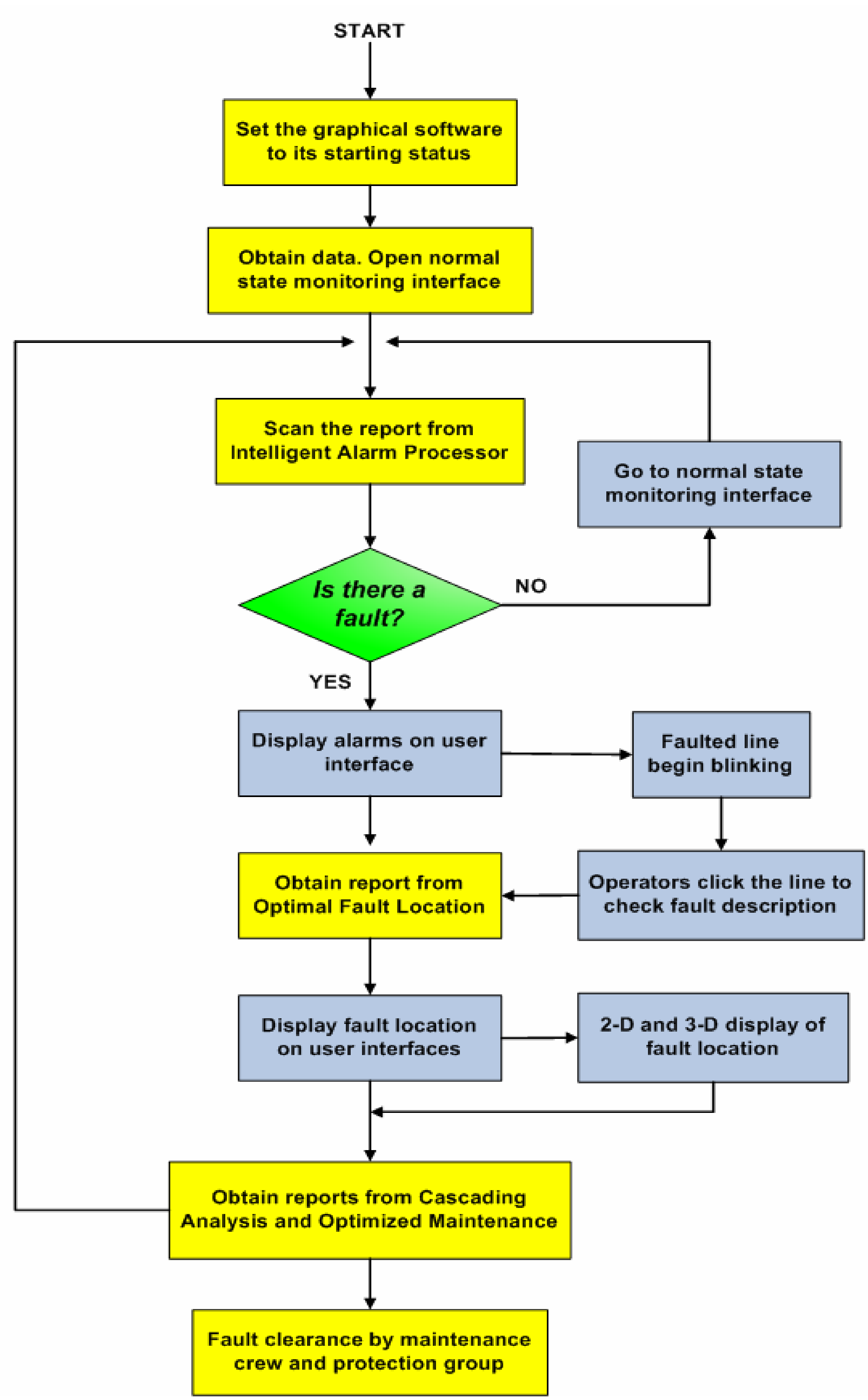

Figure 2: Implementation flow chart of the software

reports. Then they will inform related utility groups, e.g. maintenance crew and protection staff, to repair the fault and restore the normal state of the power system.

2.1.2 Internal Logic

The internal logic explains relationship and implementation sequence of various functional modules and user interfaces within the GUI software.

As is specified in our prior work, there are five modules incorporated in the proposed visualization tools: Equipment Model View, Aerial View, Electrical View, Topological View and Hierarchical View.

Although these five modules are shown as running in parallel in our block diagrams, when it comes to actual implementation, they will execute different tasks and have an internal sequence of occurrence. From the flow chart shown in Figure 1, we can observe that the hierarchical view will be shown on the screen as long as the system is in its normal state.

Once a fault occurs, the software could detect the fault by scanning fault report from the Intelligent Alarm Processor (IAP). Meanwhile alarms will be displayed by blinking the faulted zone/line(s). Operators could then click the blinking area to check the detailed fault description obtained from the fault report.

Equipment Model View is also available at this moment. By directly clicking certain equipment, operators are able to observe its 2-D and 3-D representation of equipment construction and operation model.

As soon as the fault report from Optimized Fault Location (OFL) application is available, calculated fault location, instead of estimated fault section, is presented by the graphical software. Aerial view, electrical view, and topological view of the fault are ready for operators to check. System one-line diagram, system topology, 2-D, 3-D and satellite images of equipment and area around the fault location will be displayed on the user interfaces.

It should be noted: the reason there is an internal sequence for these five modules is because there is a sequence for the external applications. Once an event occurs, IAP will be carried out first to check whether this event is a fault. Once it confirms that a fault is happening, OFL will be activated to calculate fault location. And only then can other applications such as Cascading Analysis (CA) and Optimized Maintenance (OM) be activated.

2.2 Programming consideration

In this section, we will discuss the considerations for software programming, including the programming language to be used, commercial software to be used, object-oriented programming method, and flow chart for programming.

2.2.1 Programming language and commercial software

Java will be used as the primary programming language to write the code for our GUI software. Java derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities [5-6]. Java applications are typically compiled to bytecode that can run on any Java Virtual Machine (JVM) regardless of computer architecture. As a mature programming language, Java has the following advantages [7]:

- It is simple, object-oriented and familiar;
- It is robust and secure;
- It is architecture neutral and portable;
- It could execute with high performance;
- It is interpreted, threaded and dynamic.

Besides Java, some other commercial software may also be deployed in our development process. Matlab, with its powerful numerical computing capability and huge amount of toolboxes, becomes a very good choice and programming language for our software prototyping. For creating graphics of the GUI, some software such as Blender will be deployed. Blender is a free 3-D graphics application. It can be used for modeling, simulating, and creating interactive 3-D applications, and is available for several operating systems [8]. Some other graphical software, e.g. 3ds Max, Lightwave and Maya, may also be used to create our graphic user interfaces.

2.2.2 Object-oriented programming

Object-oriented Programming (OOP) method will be the major programming approach used in our software development process. The OOP is a programming paradigm that uses “objects” and their interactions to design applications and computer programs. OOP method includes features such as encapsulation, modularity, polymorphism, and inheritance.

Object-oriented programming may be seen as a collection of cooperating objects, as opposed to a traditional view in which a program may be seen as a group of tasks to compute. In OOP, each object is capable of receiving messages, processing data, and sending messages to other objects [9].

In our GUI programming, various classes will be created to encapsulate different functionalities incorporated in the user interface. Class is an essential concept in OOP which defines the abstract characteristics of an object, including the object’s characteristics and behavior [10]. Classes provide modularity and structure in our GUI programming. Another point is that the code for a class of our GUI should be relatively self-contained.

2.2.3 Flow chart for programming

Similar to the implementation logic, when it comes to programming, a flow chart should be built to reflect the actual sequence of GUI development.

First, since the GUI will be developed as a computer project, a Java project should be established. Also, the architecture of the project needs to be designated according to our prior specification work.

Soon after the project architecture has been established, the data interpreter should be programmed to provide necessary data interface between control center server and our GUI software.

Then, several generic classes should be developed to provide several common interactive functions, e.g. object zoom in and zoom out, flip and rotate.

The five user interfaces will be developed next. According to their internal logic, the order of programming the modules should be: Equipment Model View, Aerial View, Electrical View, Topological View, and Hierarchical View.

The flow chart for GUI programming is shown in Figure 3.

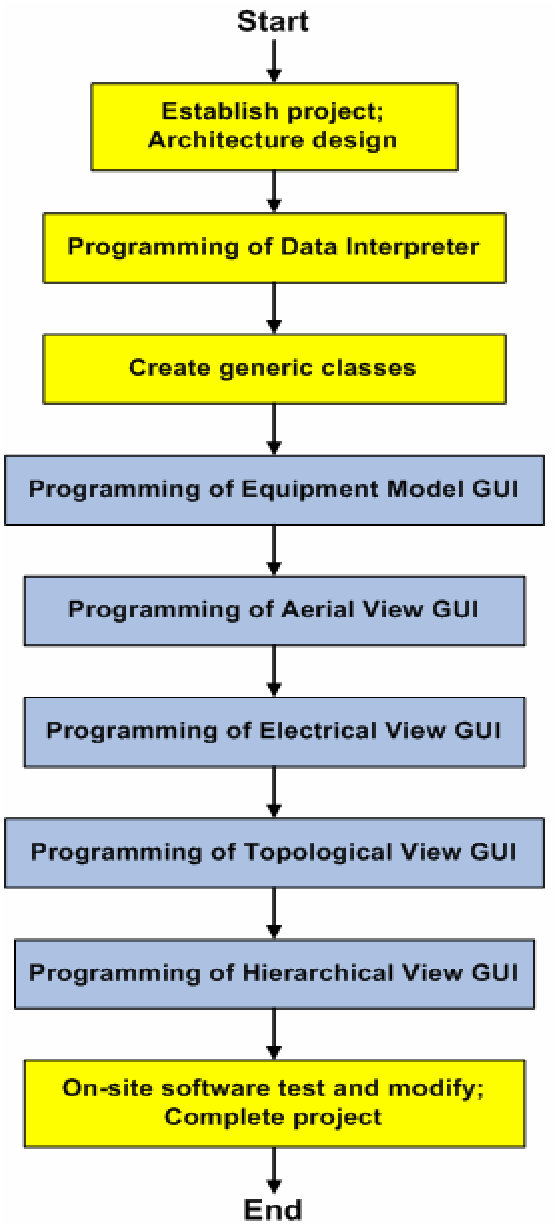

Figure 3: Programming flow chart

2.3 GUI implementation

In this section, we will investigate the GUI implementation process step by step.

Programming for the data interpreter and five visualization modules will be discussed in detail.

2.3.1 Data Interpreter

Now that PMU data, IED data and SCADA measurement data are collected by control center data concentrator, we need to decide how to use these data. Since different data types have different formats and contents, data interpreter is required to convert data types into applicable data files. There are two types of data interpreters: one is for applications, and one for GUI software.

The data interpreter for applications is shown in Figure 4.

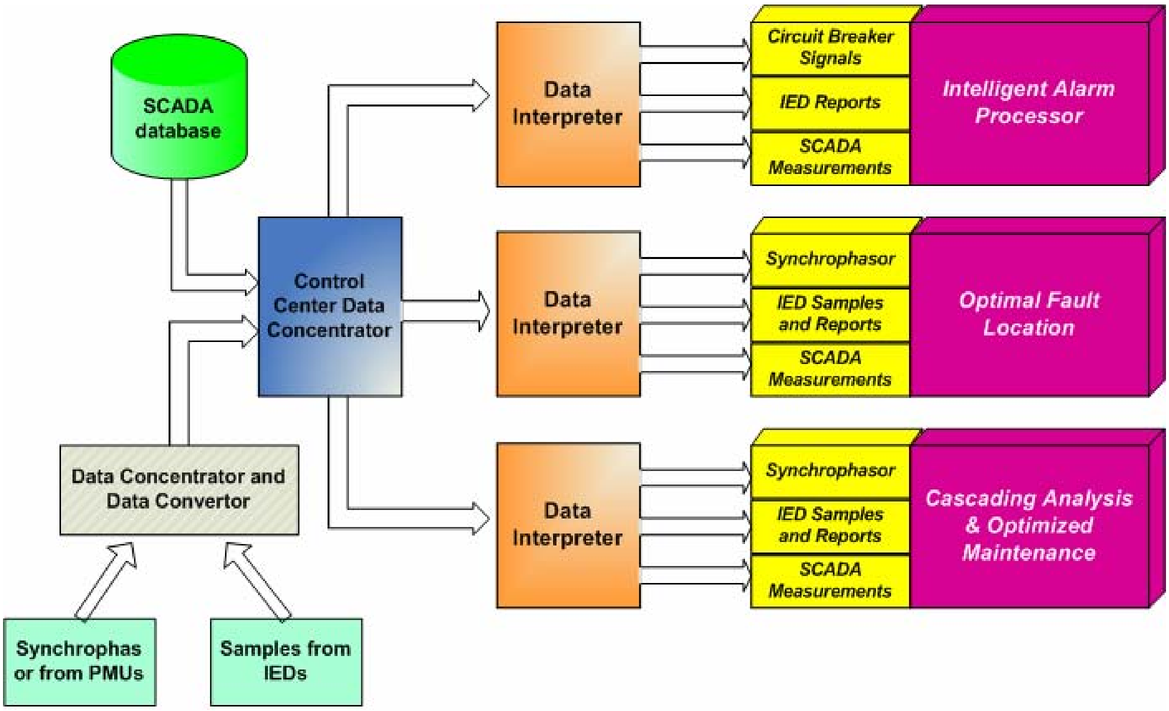

Figure 4: Data Interpreter for applications

As is shown in Figure 4, the data sent from control center concentrator has been converted by data interpreter before they are sent to different applications. For Intelligent Alarm Processor, data has been interpreted into applicable circuit breaker signals, IED reports and SCADA measurements. For Optimized Fault Location and other applications, data has been interpreted into applicable synchrophasors, IED samples and reports, as well as SCADA measurements.

For the control center GUI software, a data interpreter is also necessary for converting data and information obtained from data sources and applications. In our project, different GUI modules requires different data files. This means that for each module an independent data interpreter needs to be programmed. For example, for the Aerial View module, the interpreter outputs should include applicable fault location results, satellite image information, and SCADA measurements.

The data interpreter for GUI software is shown in Figure 5.

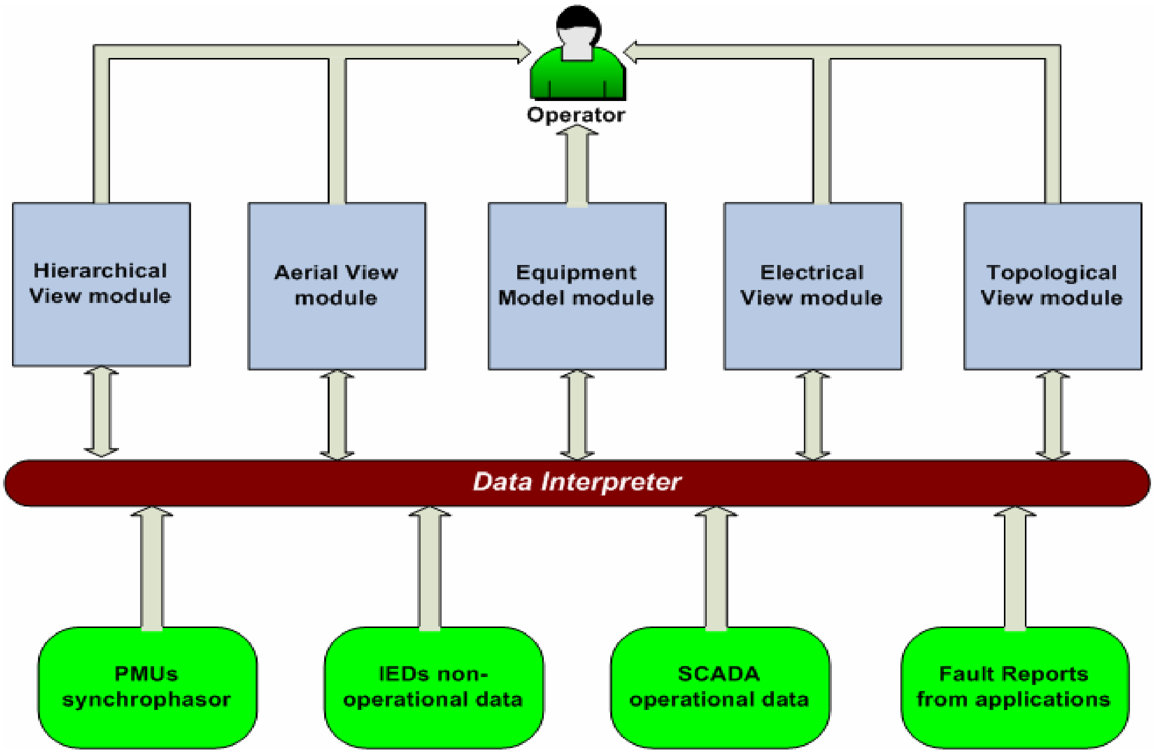

Figure 5: Data Interpreter for GUI software

2.3.2 Equipment Model View Module

The Equipment Model View module is designed to provide constructional and operational view of equipment of interest. Creating a generic model for each kind of equipment is necessary. Both 2-D and 3-D models are needed for a comprehensive view. Also, animated device operation process will be developed to provide a better understanding.

Before showing the equipment models, the GUI needs to know where the faulted area is and which devices are included in that area. This means the faulted reports are required. Thus a data interpreter needs to be programmed first to convert the report to an applicable version.

As soon as the faulted zone has been located and user clicks on an equipment image, corresponding equipment model should pop up immediately. The design of equipment models for constructional view should then be developed and incorporated into the interface. Also, animated equipment operation demo in both 2-D and 3-D modes should be created to provide the operational view.

To make the above two types of equipment model interactive, further programming is needed to enable different user functions. By deploying the OOP method, we will create different classes to accomplish this task. For example, the function of zoom in and zoom out will be encapsulated into one class. The reason for encapsulation is to prevent clients of an interface from depending on those parts of the implementation that are likely to change in future, thereby allowing those changes to be transparent to the clients. Other interactive functions include flip and rotate, as well as operation curve plotting.

The block diagram of equipment model programming sequence is shown in Figure 6.

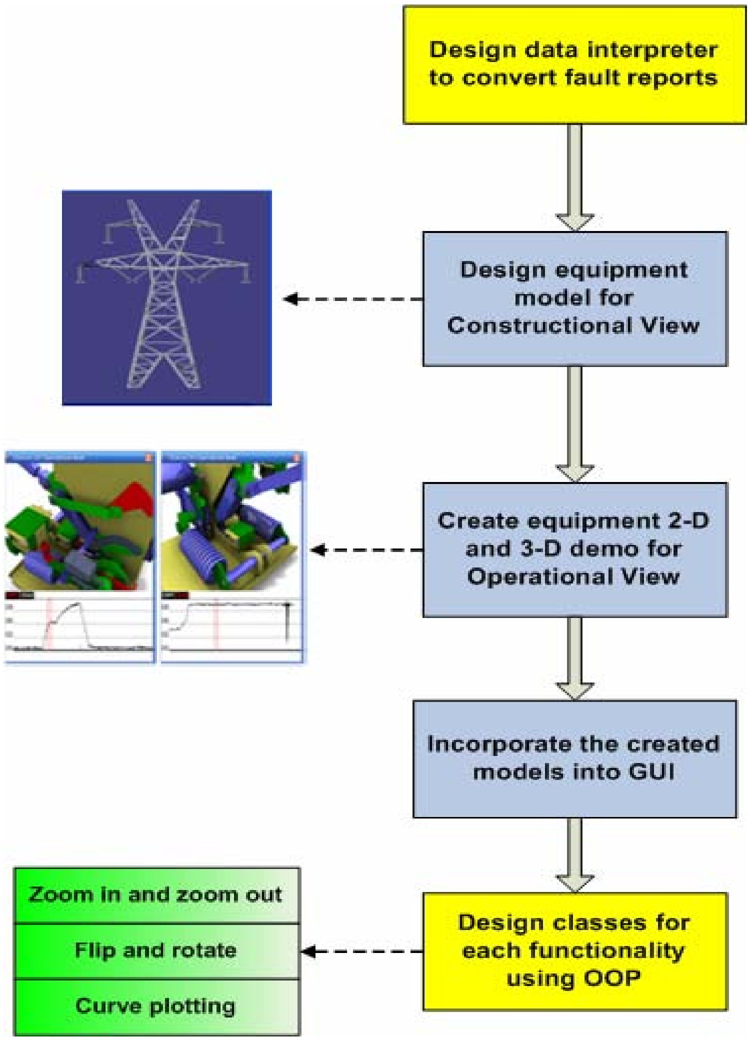

Figure 6: Programming of Equipment Model GUI

2.3.3 Aerial View module

To get a comprehensive view of the terrain around a fault location, satellite image is currently the best tool. The goal of Aerial View module is to convert satellite image into useful information for the purpose of fault location visualization.

It will take several steps to complete this GUI module. First, a data interpreter is needed to convert data and reports from external sources. Next, high spatial resolution images needs to be obtained by accessing the satellite imaging services.

Currently, Ikonos, QuickBird and OrbView services are capable of providing very high-resolution satellite images and are available on a commercial basis. The Ikonos satellite is the world's first commercial satellite to collect panchromatic images with 1 meter ground sample distance (GSD) and multi-spectral imagery with 4 meter GSD. QuickBird is a high resolution satellite owned and operated by DigitalGlobe. Using a state-of-the-art Ball’s Global Imaging System 2000 sensor, QuickBird uses remote sensing to a 0.61 meter GSD. OrbView also offers one of the high-resolution satellite images. OrbView-5 will offer the highest resolution available to date by simultaneously acquiring 0.41 meter panchromatic and 1.64 meter multi-spectral imagery [11-14].

The block diagram of aerial view module programming sequence is shown in Figure 7.

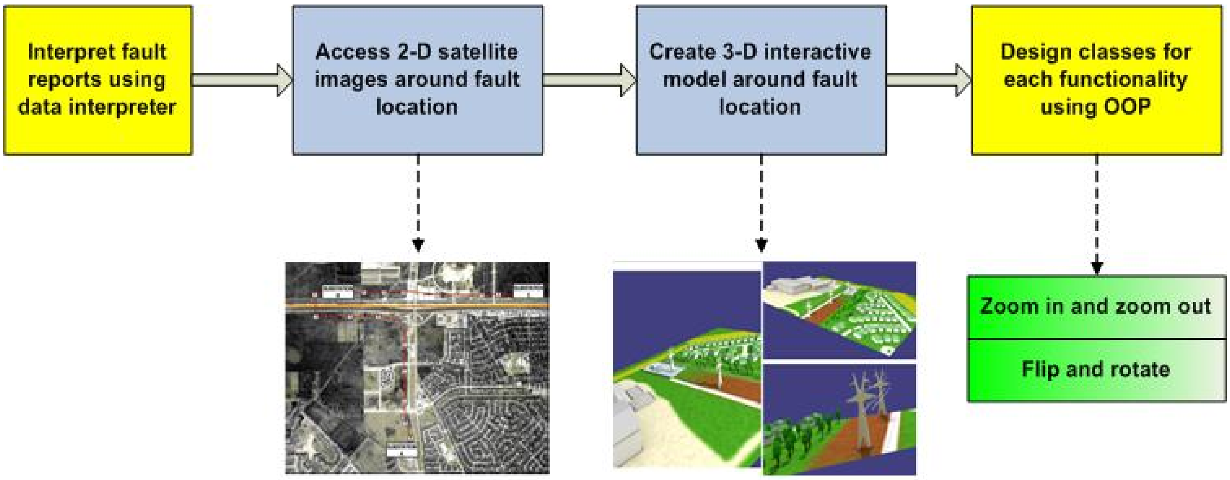

Figure 7: Programming of Aerial View GUI

2-D satellite images could display the real-world terrain around a position. However, sometimes in system monitoring and control, information such as street and building are not as important as transmission lines and tress. By processing the satellite images, we could also abstract some useful information which could be utilized to form the 3-D interactive model. These models include the critical information which operators are eager to know. Having the 3-D models available, our next step is programming for interactive functions such as model zooming, flip and rotation.

2.3.4 Electrical View module

The block diagram of programming sequence for this part is shown in Figure 8.

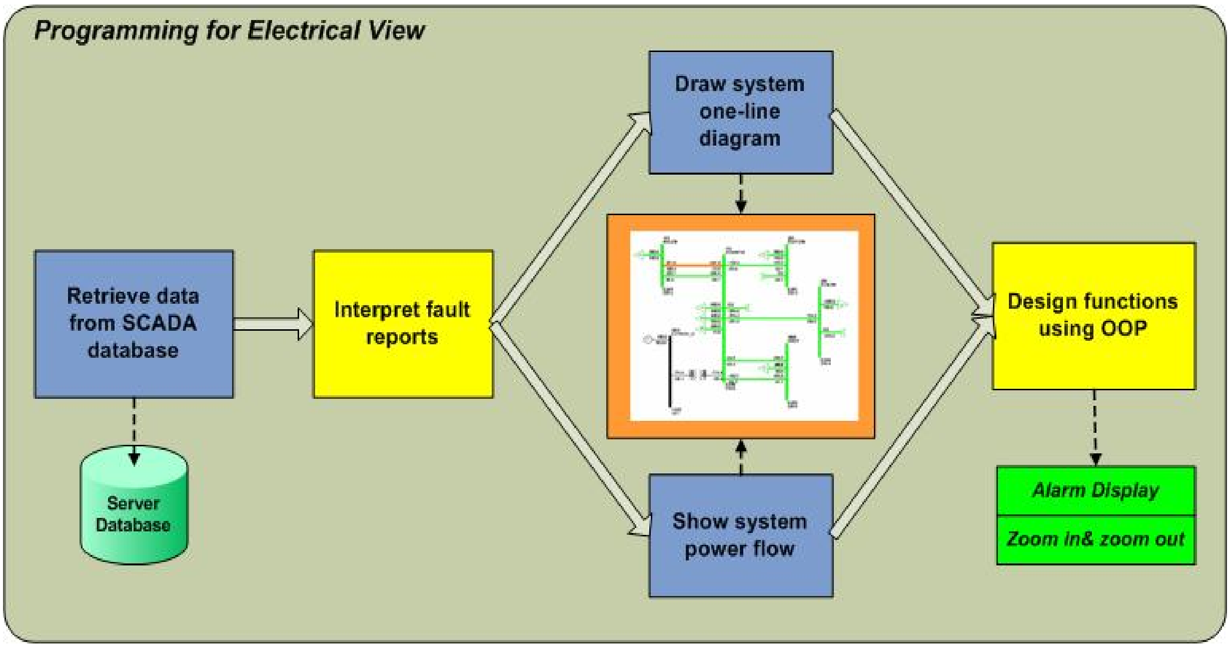

Figure 8: Programming of Electrical View GUI

The Electrical View GUI is going to display the electrical measurements in the system one-line diagram. As shown in Figure 8, for this part we will have the following programming tasks:

- Retrieve data and fault reports from server database;
- Interpret those data and reports into applicable data files. This requires the programming of a data interpreter;
- Use graphical techniques integrated in Java to draw system one-line diagram according to the interpreted data files;
- Display real-time power flow on the one-line diagram;
- Design interactive functions. This includes writing codes for object zooming, flip and rotation, using OOP technique.

2.3.5 Topological View module

Next we will move to the programming of Topological View module. This module is designed to provide topology visualization of the system.

The block diagram of topological view programming sequence is shown in Figure 9.

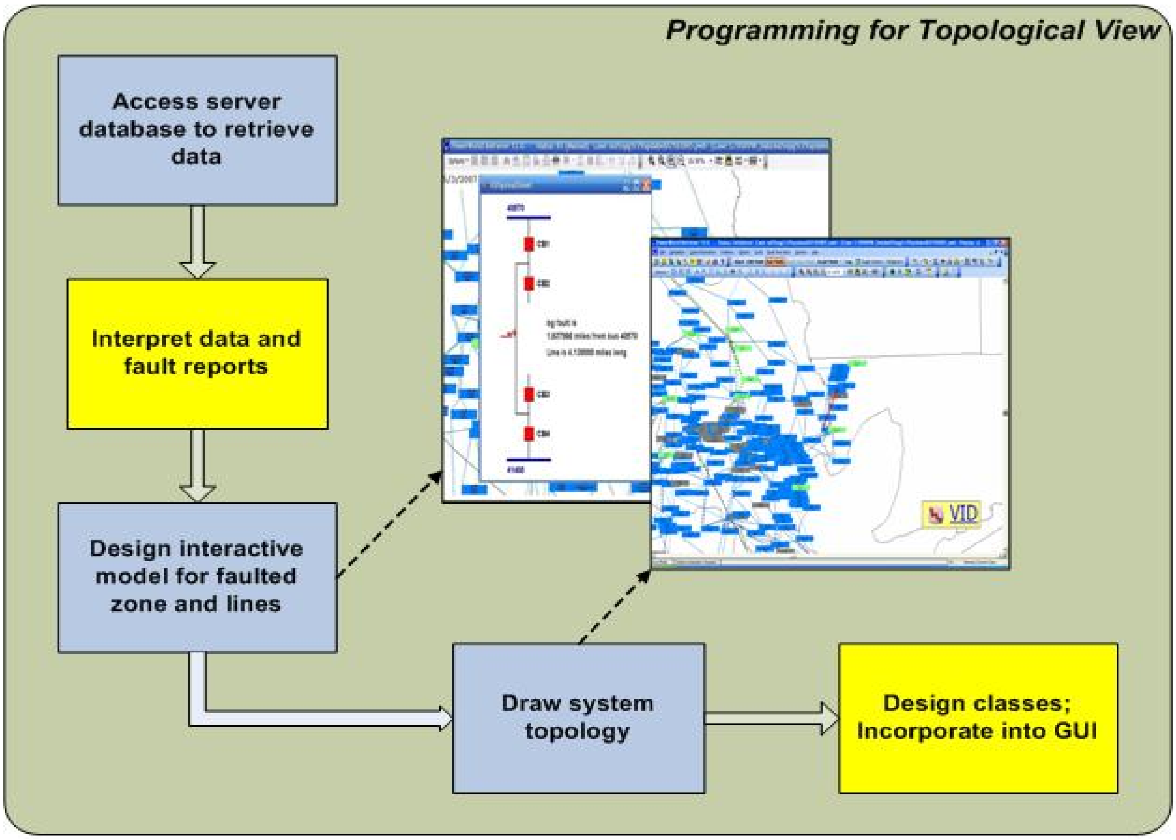

Figure 9: Programming of Topological View GUI

As is shown in Figure 9, we will take the following steps to program this module:

- Retrieve data and fault reports from server database;

- Program a data interpreter to convert data and reports;
- Program interactive model and functions;
- Draw system topology using interpreted data;

2.3.6 Hierarchical View module

The last GUI module to be programmed is the hierarchical view. Basically this module is a normal state monitoring tool, from which operators can use to visualize the system real-time topology, one-line diagram and power flow.

When a fault occurs, this module is also responsible for displaying the alarms. This can be accomplished by blinking the faulted lines/zone. Operators could then click that blinking area to check the fault details.

To achieve the above functions, several programming tasks need to be done in a specific order. First, similar to other modules, a data interpreter will be coded to convert data into applicable form. Secondly, to show the system topology, one-line diagram, and real-time power flow, we will interface to the existing modules. The equipment model view will be obtained by accessing Model View module. The Electrical View module will be accessed to show the system one-diagram and power flow. Topological View module will be interfaced to display system topology. Further development may also be required so that the satellite images included in Aerial View module can be incorporated in the hierarchical view. From above description one can see that the hierarchical view is a combination of all the other GUI modules.

Our next step is to add several interactive functions to the user interface. Here an event-response mechanism will be established. This means the program will be set to continually scan the interpreted data files. As soon as a fault has been detected, the area around the fault will begin blinking as a response to the fault alarm. Another task is to write code to respond to the click actions of users. Once the users click on a certain area on the screen, correlated information such as enlarged pictures or equipment models will show up.

Part 3: Other Considerations

3.1 Software maintenance plan

As described in the international standard ISO/IEC 14764, the following software maintenance processes will be considered throughout our development [15]:

1) Creation of the maintenance plan. Before starting the graphical software programming, we will make preparation for handling problems identified during development, and the follow-up related to product configuration management;
2) Organization of a maintenance group. Once the software has been applied, the maintenance programmer must analyze each request, confirm it, and check its validity, investigate it and propose a solution, document the request and the solution proposal, and, finally, obtain all the required authorizations to apply the modifications;
3) Acceptance of the modifications. This will be done by checking it with the individual who submitted the request in order to make sure the modification provided a solution;
4) Software migration. The migration process (platform migration, for example) is exceptional, and is not part of daily maintenance tasks. If the graphical software must be ported to another platform without any change in functionality, this process will require a maintenance project team will be assigned to this task;
5) Retirement of a part of the GUI software. This is the last maintenance process, also an event which does not occur on a daily basis.

3.2 Security plan

Once the graphical software has been installed in at the control center client computers, the following security plan will be implemented [16]:

1) Assign accounts and passwords for different utility groups. User identification is strictly verified using user authorization and authentication techniques;
2) Software authorization and authentication. Each module and interface should only be accessed by particular utility group who is responsible for that part. Any other groups or individuals will not be allowed to access this module and related graphical user interfaces;
3) Training of utility staff. The security training is scheduled to be carried out before utility employees are permitted to use the visualization software.
4) Regular security inspection and assessment. The assessment of software security is scheduled to be executed regularly, for example, every week.

Part 4: Schedule for Software Development

In this section, a general schedule for developing the control center GUI software is discussed. The schedule is established according to the programming flow chart shown in Figure 3.

According to the project plans, the software development process will start in January 2009, and end in September 2009. After comprehensive consideration, we have divided the 10 months' duration into three phases. During each phase a major milestone is expected to be accomplished:

- The first phase is software preparation, including Java project establishment and generic classes development. This will last for about 2 months;
- Our second phase is the core part, which includes GUI programming, equipment modeling, 2-D and 3-D images creation. This phase may take about 7 months;
- The final phase is for software testing and on-site demonstration using field cases, which may take about 1 month.

The schedule for software development is shown in Table 1.

Table 1: Schedule for software development

No.	Task	Estimated duration
1	Establish software project; Develop generic classes.	2 months
2	User interface programming; Create equipment models and 2-D 3-D images.	7 months
3	On-site demonstration using field cases.	1 month

It should be noted that this is a general programming schedule. As the software development proceeds, detailed task schedule will be established and new time table will be made as needed for each phase.

Part 5: Conclusions

5.1 Accomplishments

As the third and last step in the design of the GUI software, the implementation specification has been studied in detail.

The following is a summary of the progress achieved so far:

- The overall implementation flow chart of the software has been developed and explained;
- Both external and internal logic of the software represented in the implementation flow chart have been specified;
- The Object-oriented Programming (OOP) technique has been selected as the primary programming technique;
- The software programming flow chart has been defined and general programming sequence has been established;
- Two types of data interpreters have been explained; detailed inputs and outputs of the interpreters have been specified;
- The programming specification of all GUI modules within the graphical software has been outlined and block diagrams for each module has been specified;
- Options for receiving satellite images from commercial providers have been investigated;
- The hierarchical view, which is for normal state monitoring purpose, is set to be a combination of four other views: model, equipment, aerial and topological.
- Implementation plans for software maintenance and security considerations during development have been specified;
- A general schedule for developing the graphical software is established, which has specified three phases throughout the programming process;

With the above accomplishments, the implementation specification of the visualization tools has been completed.

5.2 Future work

Since the specification and design work for the software has been completed, our activity will focus now on the development of the control center visualization tools as the next step. Field installation and in-service demonstration of visualization tools will be carried out soon after the software is developed.

References

[1] M. Kezunovic, "Optimized Fault Location," Power Systems Engineering Research Center, Project Final Report, PSERC Publication 08-07, April 2008.
[2] M. Kezunovic, A. Abur, "Merging the Temporal and Spatial Aspects of Data and Information for Improved Power System Monitoring Applications," IEEE Proceedings, Vol. 9, Issue 11, pp 1909-1919, 2005.
[3] T. J. Overbye, et al, "Effective Power System Control Center Visualization,", Power Systems Engineering Research Center, Project Final Report, PSERC Document 08-12, May 2008.
[4] M. Kezunovic, Zheng Ce, "Monitoring Power System Dynamic Performance Using Synchronized Sampling," CIGRE Conference on Monitoring of Power System Dynamics Performance, St. Petersburg, Russia, April 2008.
[5] Bruce Eckel, “Thinking in Java,” 4th Edition, Prentice Hall, 2006.
[6] Brian D. Hahn, Katherine M. Malan, Hahn, “Essential Java for Scientists and Engineers,” Butterworth-Heinemann, 2003.
[7] K. Barclay, J. Savage, “Object-oriented Design with UML and Java,” Elsevier Butterworth-Heinemann, 2004.
[8] http://en.wikipedia.org/wiki/Blender_(software).
[9] Brad J. Cox, “Object-oriented Programming – an Evolutionary Approach,” Addison-Wesley, 1986.
[10] http://en.wikipedia.org/wiki/Object-oriented_programming.
[11] SPACE IMAGING, http://www.spaceimaging.com/products/ikonos/index.htm, Aug. 10, 2006.
[12] S. Kocaman, L. Zhang, A. Gruen, D. Poli, “3D City Modeling from High-resolution Satellite Images,” ISPRS Ankara Workshop 2006, Ankara, Turkey, Feb. 2006.
[13] ORBIMAGE, http://www.orbimage.com/corp/orbimage_system/satellite.html, Aug 10, 2006.
[14] Yoshihiro Kobayashi, George Karady, et al, “Satellite Imagery for the Identification of Interference with Overhead Power Lines,” Project Final Report, PSERC, Jan. 2008.
[15] http://en.wikipedia.org/wiki/Software_maintenance.
[16] H. Mouratidis, P. Giorgini, “Integrating Security and Software Engineering: Advances and Future Visions,” Idea Group Inc, 2006

Appendix A

Synchronized Sampling Uses for Real-Time Monitoring and Control/RTGRM

Requirements Specification for Visualization Tools for

Optimized Fault Location and Intelligent Alarm Processor

Project Progress Report

Mladen Kezunovic

Ce Zheng

Texas Engineering Experiment Station

September 30, 2008

Executive Summary

Real-time monitoring of power system conditions can help control center operators maintain adequate situational awareness. This project addresses a set of visualization tools for future generation of switchable networks, with a particular emphasis on visualization used in the control center.

While synchrophasor measurements are proven to enhance awareness of system operators, it remains unresolved how user interfaces that will aid operators in making decisions should be incorporated in existing interfaces for SCADA functions. This project is aimed at developing an integrated visualization tool that will seamlessly incorporate time correlated information from synchrophasor measurements, SCADA and non-operational data. This will result in intelligent operator tools for alarm processing, fault analysis and breaker switching management, which will increase the effectiveness of power system visualizations and reduce the time needed to make decisions.

The first phase of the project addresses requirements specifications for the visualization tools. A set of achievements obtained and reported in this document are:

- The applications of alarm processor and fault location have been specified;
- The data sources and data preprocessing have also been specified. CIM format and XML file format have been used for data storage and exchange;
- The inputs, outputs, and information processing structure of each application have been summarized and described in detail;
- Control center server is introduced for on-line collection of information from substation applications and SCADA database. A server/client network is established;
- The information exchanging mechanism between control center server and client computers has been demonstrated;
- Different modules within the software have been designed to execute different functionalities. 2-D, 3-D and satellite views are designed to be incorporated in the software to increase the effectiveness of power system visualization;
- By assembling different modules according to their inherent relations, the architecture of the whole data/application/visualization system has been presented;
- Responsibilities of operators and each utility group have been specified. Several other considerations have been discussed.

This report is the first step out of several consecutive research and development activities. Next the functional design and implementation specifications for the visualization tools will be explored. The new tools will be demonstrated using filed cases from ERCOT and CenterPoint Energy. In addition to the participation from organizations acting as the field demonstration hosts, this project will be supported by industry advisors from AEP, Oncor, FirstEnergy, and HydroOne.

Part 1: Introduction

This section summarizes the background of different applications and main motivations for exploring the opportunities of developing new integrated interface which will aid operators in their decision making process.

1.1 Intelligent alarm processor

Alarms are typically generated in power system control centers any time one of two categories of following events happens [1]:

- An analog value measured by a transducer passes an operation constraint, e.g., an overload in a transmission line, an under-voltage or an over-voltage at a bus;
- A digital status value changes state, e.g., the opening or closing of a circuit breaker, or the detection of an excessive temperature alarm point set in a transducer.

With the growth of power system complexity, a major disturbance could trigger hundreds or even thousands of individual alarms and events, clearly beyond the ability of any control center operator to handle. To adapt to the new situation, intelligent alarm processor has been developed to aid operators recognize the nature of the disturbances. An efficient alarm processor has the ability to analyze complex events efficiently within a time constraint, and provide prioritization of alarms. More specifically, the intelligent alarm processors (IAP) are developed to meet the following needs [2]:

- Reduce the number of alarms presented to the operator;
- Convey a clearer idea of the power system condition causing the alarms;
- Recommend corrective action to the operator if such action is needed.

Traditionally, the research focus is on the improvement of hardware and software of alarm processing to guarantee that no alarms are lost when an event triggers a series of alarms within a short period of time. Now that the synchronized sampling technique is gradually being deployed, it is possible to have high precision assessment of the alarm occurrence and sequences leading to a possibility for automated cause-effect analysis at the substation or SCADA level.

Recently an advanced alarm processor has been proposed, which combines alarm processing techniques at both the Substation Automation System (SAS) level and the Energy Management System (EMS) level. Accurate alarm interpretations could be made by computers in a very short time after an event happens, without the intervention of control center operators [3].

Once the alarms are prioritized and processed, the analysis results can be conveyed to the control center where operators are then able to handle the system conditions according to the recommended actions.

This requirements specification relates to the Intelligent Alarm solution reported in [2]. This application, while less effective than solution reported in [3], is easier to implement since it only requires SCADA data and selected data from protective relays located in substations.

1.2 Optimized fault location

A continuous and reliable electrical energy supply is the objective of any power system operation. However, faults inevitably occur in power system due to bad weather, equipment damage, equipment failure, environment changes, human or animal interference and many other reasons. Once fault event in power system occurs, different Intelligent Electronic Devices (IEDs) automatically recognize the fault as abnormality.
Since it is essential that accurate information about fault location (FL) and its nature is provided as fast as possible, various fault location algorithms have been presented in the literature in the past. These algorithms can be categorized into three major types with respect to the placement of measurements: single-end, multi-end and system-wide sparse measurements algorithms. The spatial and temporal considerations indicate that there is no universal FL algorithm suitable for all situations. In order to be able to evaluate which algorithms are applicable for a given fault event, different data sources (measurements) have been utilized and the idea of optimized fault location which takes into account both temporal and spatial considerations has been proposed [4, 5] In the case data from two ends of faulted transmission line are available, the two-end FL algorithm is the most accurate approach and should have priority. In the case of the two-end algorithm, if input samples are synchronized, synchronized sampling two-end algorithm is the most appropriate [6, 7]. Otherwise it is checked whether data from only one end of the faulted line is available.

Once the fault location is calculated, it is very important that the fault reports are effectively presented to operator. In addition, communication between control center and utility groups such as maintenance crew and protection engineers is essential. Knowing the real-world environment around fault location and construction of involved equipment enables utility staff to clear fault quickly and efficiently. Designing user interfaces that can effectively convey the results of fault analysis still remains a challenge in the utility industry [8].

1.3 Cascading outage analysis and optimized maintenance

The cascading outage is a more severe kind of system disturbances, which refers to the uncontrolled successive loss of system elements triggered by a disturbance. Large area blackouts are always the final results of the cascading outages [9]. How to detect, prevent and mitigate cascading outages becomes more and more focus in recent research. In our project, the proposed visualization tool we will incorporate an option for inclusion of an application module that contains state-of-the-art cascading analysis approach, which can provide operators with a clear idea of the cause and possible impacts of complex events.

Besides cascading outage analysis, some other applications such as condition-based maintenance of circuit breakers will also be considered in the specification of the new visualization software. By presenting the analysis results through GUI, operators, as well as maintenance crews and protection engineers will be able to monitor and evaluate real-time conditions of both power system and its components.

Figure 1 summarizes the objectives and expected contribution of our developments.

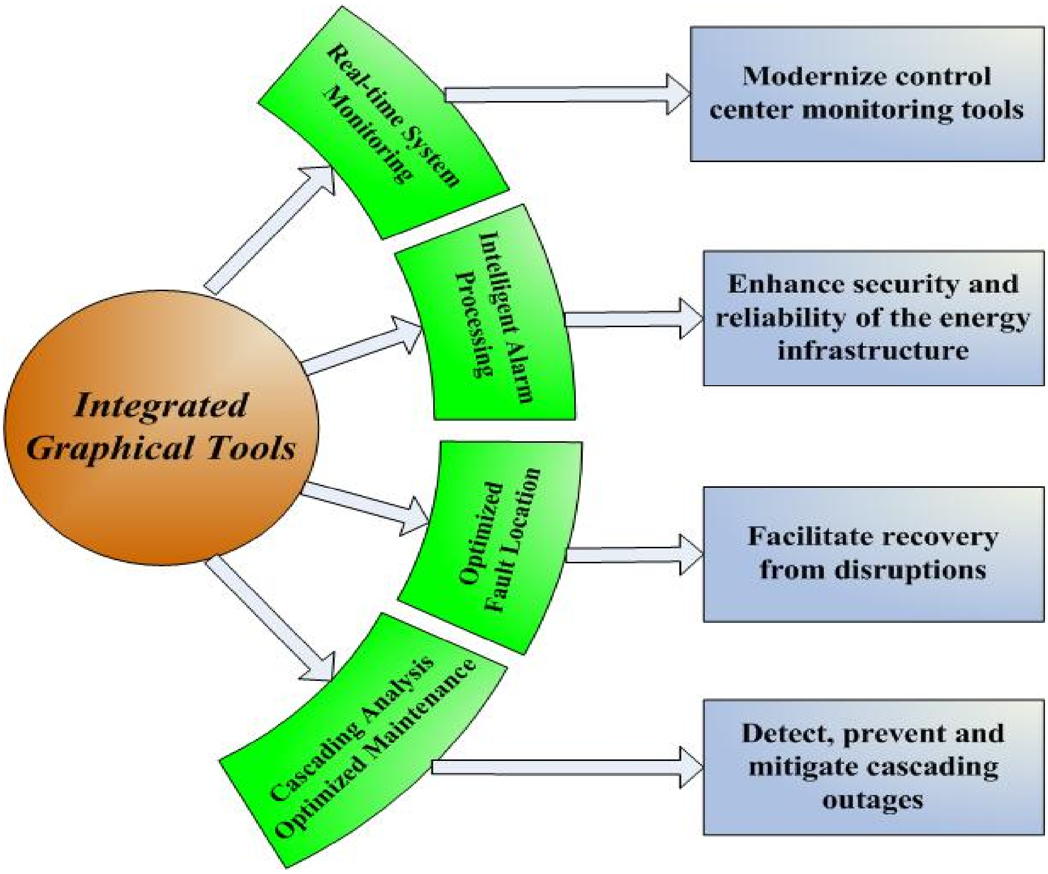

Figure 1: Objectives of the integrated tools for Graphical User Interface (GUI)

Part 2: Requirements Specification

This section discusses the proposed requirements specifications for the integrated graphical tools. The information exchange between GUI and new applications such as intelligent alarm processor and optimized fault location has been specified. Synchronized samples, Synchronized phasor measurements, SCADA database and other information obtained from IEDs have been utilized as the input data sources. The data processing structure as well as specific outputs of each application have been demonstrated and compared.

2.1 Data sources and applications

Currently, the time correlated information in power systems is obtained mainly through three types of equipments: Remote Terminal Units (RTUs) from SCADA, GPS-based equipments (e.g. PMUs), and some other IEDs (e.g. DPR, DFR) installed in substations. Each type of measurement methods has its own particular application areas [9]. In order to fully utilize these recorded data, it is very important that information from various data sources, i.e. synchrophasors, SCADA PI historian database and IED non-operational data can be retrieved and integrated. Since different measurement equipment produces data in quite different formats, the conversion of data into certain standard formats is also necessary [10, 11]. Figure 2 gives the relationship between data sources and their applications. It also describes the general relationship between applications and our proposed control center visualization tools.

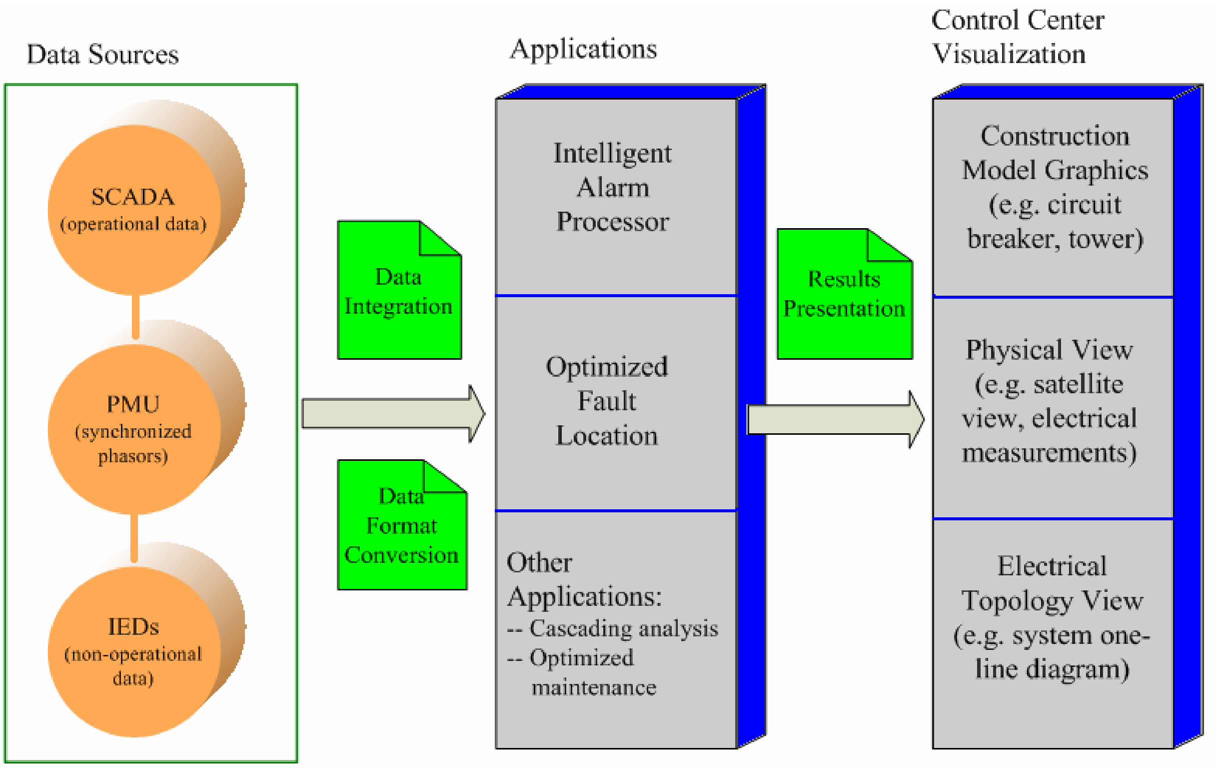

Figure 2: Data sources, applications and control center visualization options

As shown in Figure 2, as soon as the recorded data has been integrated and converted to applicable data files, they are processed in different application modules, including optimized fault location, intelligent alarm processing, and several other applications (e.g. cascading analysis and optimized maintenance). The processing results are then carried to the control center and presented to operators through the proposed visualization tools. According to our design, three categories of views will be provided by the visualization software. They are: Construction Model Graphics, Satellite View and System Electrical Topology View. They will be further explained later on in this section.

2.2 Specification of applications

Figure 3 illustrates the hierarchy of the visualization system. From the bottom layer, data collected from GPS-based devices, SCADA and other IEDs are transmitted to the applications located at a higher layer. Commercial software such as PowerWorld, ATP-EMTP and PSS/E are located at the bottom layer and are utilized in short circuit calculation and fault analysis needed within application modules. The control center visualization tools are customized developments that are obtaining information directly from the applications, located at the second layer.

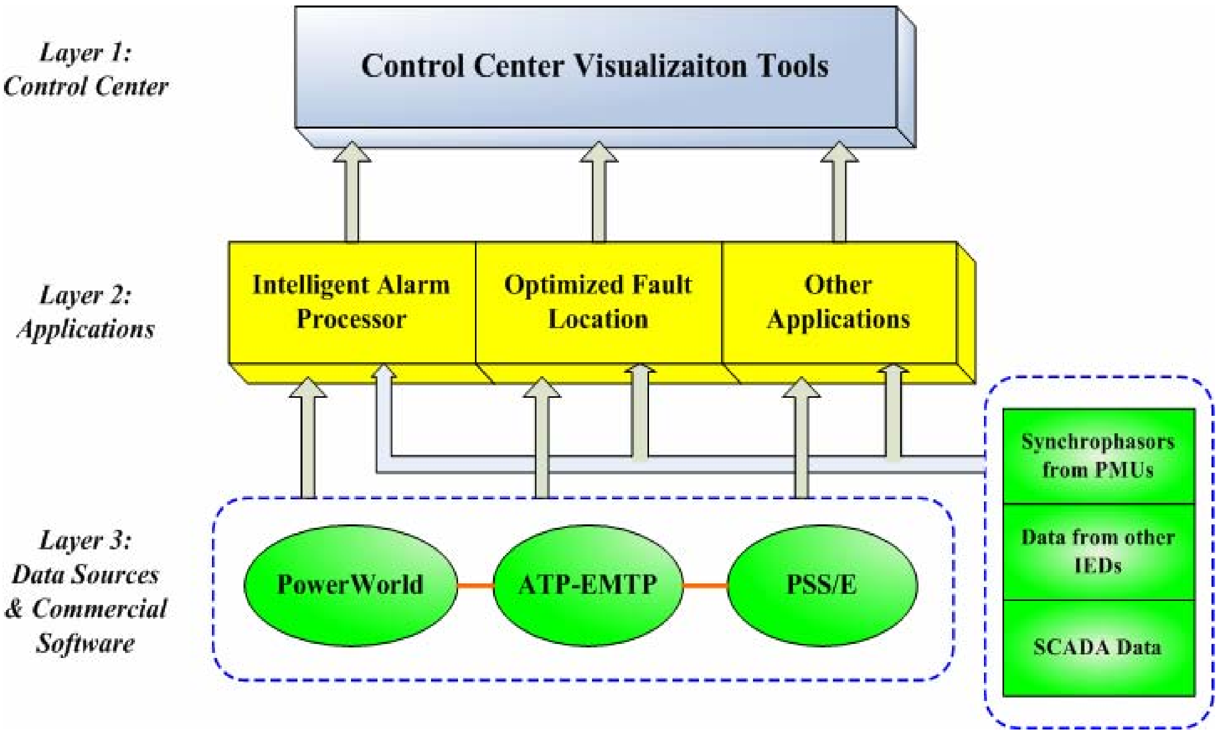

Figure 3: Hierarchy of the visualization system

To define the exchange of information between application modules and control center, we need to know first what the inputs to the applications are, the data requirements of each functional module, their specific data processing structures, and the outputs. All of these have been specified in detail in this section.

2.2.1 Intelligent alarm processor

As is stated in Part 1, the intelligent alarm processor solution proposed in our research has been reported in [2]. The information processing structure of this alarm processor is demonstrated in Figure 4.

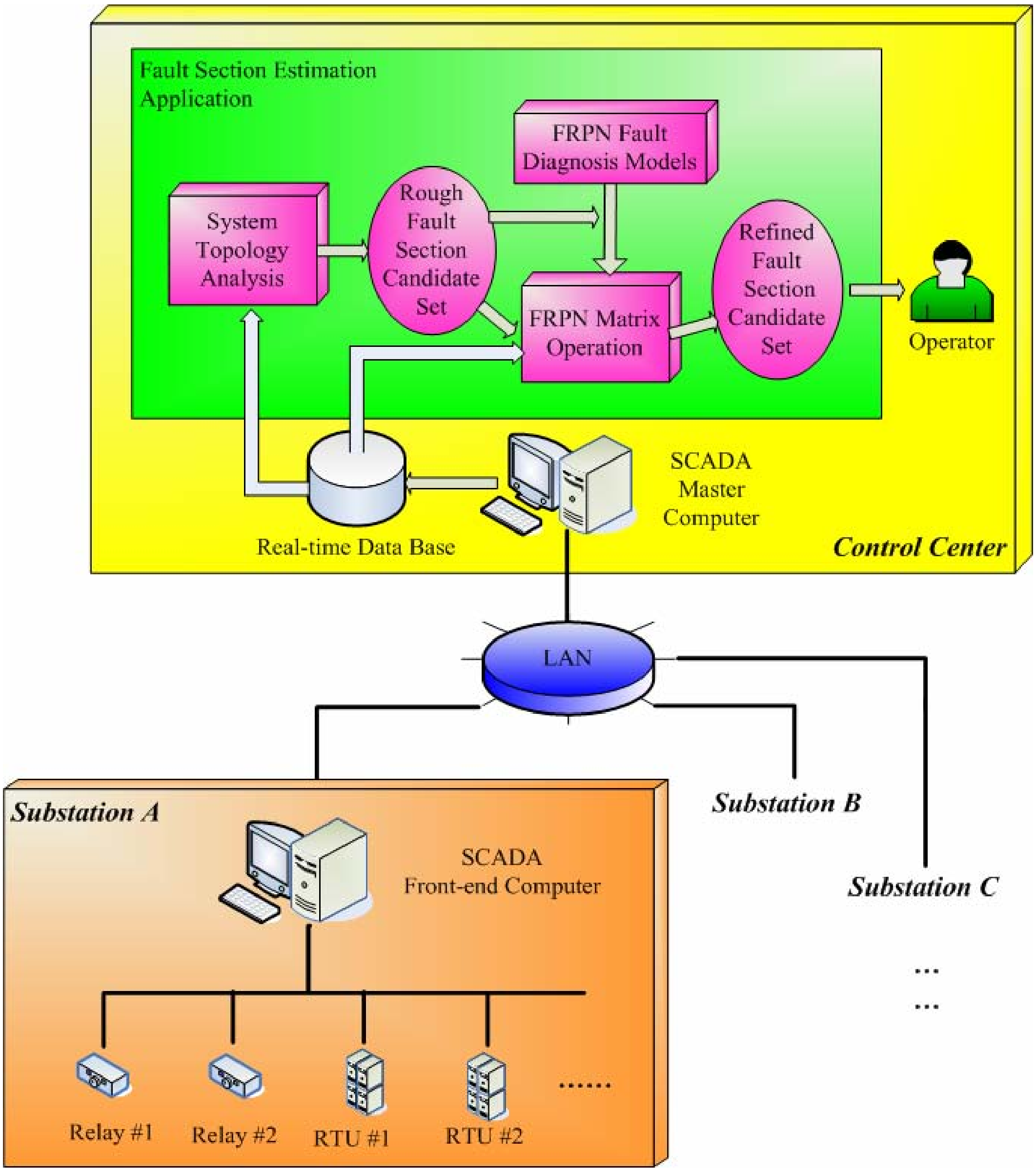

Figure 4: Information processing structure of intelligent alarm processor

Usually SCADA PI historian database and measurements from Remote Terminal Units (RTUs) are the data sources for alarm processor. In our solution illustrated in Figure 4, only SCADA data and selected data from protective relays located in substations are required. Since the synchronized sampling technique is being gradually deployed, it is possible to have high precision assessment of the alarm occurrence and sequences leading to a possibility for automated cause-effect analysis at the substation level and/or control center level. The data requirements for intelligent alarm processor are listed in Table 1.

Table 1: Data requirements for intelligent alarm processor

Type of Data	Function Module	Inputs	Description
System Level	Control Center Alarm Processor	CB status change alarms	This input file includes the opening and closing information of circuit breakers.
		Over-current alarms	The data and alarms are obtained and provided by SCADA.
		Relay operation data around the time of occurrence of the alarm	This file is always used to verify the correctness of analysis results of IAP. It's not necessarily used in alarm processing.
Local Level	Substation	RTU data and selected data from protective relays	

2.2.2 Optimized fault location

The optimal fault location method selects a suitable fault location algorithm from three types of algorithms: two-end synchronized sampling FL algorithm, single-end phasor-based FL algorithm and system-wide sparse measurement based FL algorithm. The measurement equipments used are sparsely located Digital Fault Recorders (DFRs) or other GPS-based IEDs (e.g. PMUs) in the future. Commercial programs like PSS/E and PowerWorld have been utilized to run power flow analysis and visualize the fault location [12, 13]. The detailed descriptions of the field data needed for this application are listed in Table 2.

Table 2: Data requirements for optimized fault location

Type of Data	Function Module	Input Data	Description
System Level	Power Flow Analysis	Power flow raw data from: Input data files for PSS/E (*.raw)	This file contains power flow system specification data for the establishment of a static power system model. This data is used by PSS/E to run the power flow analysis.
	Tuning of Static System Model	SCADA data from: SCADA PI historian database	This file contains the latest load, branch and generator data to tune the static system data with the actual pre and post fault conditions
	Short Circuit Study	Sequence impedance data from: Input data files for PSS/E (*.seq)	This file contains the negative and zero sequence impedance data needed for short circuit study. It is used by PSS/E to add the impedance data to

			the case of interest.
	Visualization of Fault Location	System binary data and graphical data from: Input data files for Power World Retriever (*.pwb and *.pwd)	These files are used by the PowerWorld Retriever to visualize the fault location. It contains power system binary and graphical data.
Local Level	Fault Location	Recorded data during faults from: DFR configuration (*.cfg) and data (*.dat)	These files are the fault events data to be used by Optimal Fault Location algorithm.
		Substation interpretation data from: Interpretation Data (*.int) for each Substation of interest	The "Interpretation File" contains information that relates the channel numbers to the monitored signals. It also represents the correspondence between the nomenclature used in the DFR files and those used in the PSS/E file.

The information processing structure of optimized FL is shown in Figure 5.

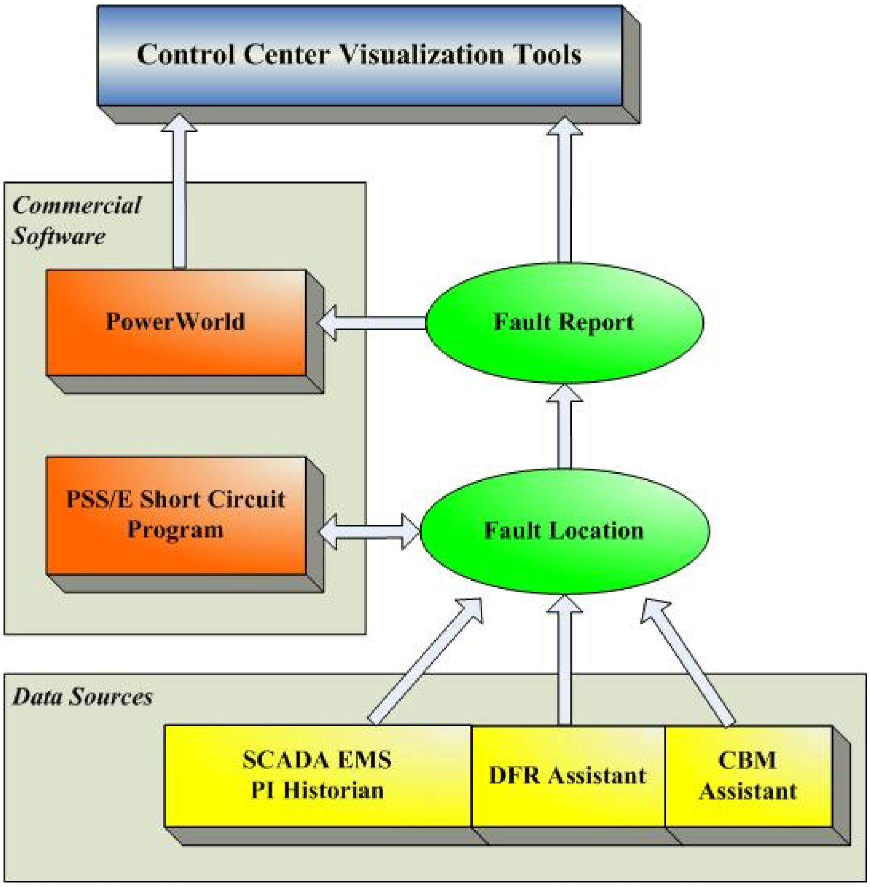

Figure 5: Information processing structure of optimized fault location

2.3 Information exchange specification

Once the data processing has been accomplished within application modules, analysis results as well as detailed reports should be exported to the control center visualization tools and presented to operators as fast as possible. The proposed visualization tool is responsible to collect application outputs, integrate them and display them to the operators.

Before exporting data and reports from application modules to the operators, it is very important to specify what the outputs of different applications are and in what format the output information is. The outputs of both intelligent alarm processor and optimized FL are shown in Table 3.

Table 3: Description of application outputs

Application	Outputs	Original format
Intelligent alarm processor (IAP)	Timestamp	Saved in Windows EXCEL; Analysis result and suggested actions, as well as additional information
	Analysis result	
	Suggested actions	
	Additional information	
Optimized Fault Location (OFL)	Estimated fault section (Bus No. at faulted line at two ends)	PowerWorld Retriever's format; Saved in XML file;
	Fault location within the estimated section	
	Fault type	

It can be observed that different applications have different formats for output files. In order to efficiently exchange information between applications and the proposed visualization tools, the following requirements must be satisfied:

- A web-based file sharing system between control center computers should be established, which could provide a method for sending and receiving digital data files over the network.
- Since the application output reports are produced and updated from time to time, an on-line data transmission and storage method is required.
- Since the outputs of applications serve as inputs to the visualization tool, a data adapter (or data converter) should be utilized to convert different data files into the same file with a uniform format before transmitting them to the visualization tool.

To meet the needs of data exchange between different applications and control center visualization tools, an information exchange structure is proposed and demonstrated in Figure 6. It possesses the following features:

(1) The Common Information Model (CIM) [14], which is defined in IEC-61970, is utilized as standard data modeling format. Synchronized samples from IEDs are preprocessed and converted into CIM format in substation;

(2) After converted to CIM format, metadata is stored to an XML file. Application modules will use this file to carry out analysis. Outputs from all applications will

also be converted to CIM format through data adapters.

(3) XML file which contain outputs of applications will be sent to Control Center Server. The server, which is located in equipment room, is responsible for collecting and saving data files from applications and SCADA PI historian database. It is also connected to client computers through a Local Area Network (LAN). Information exchanging between server and clients is completed within the network;

(4) The server and client computers are connected via Java Remote Method Invocation (Java RMI). The Java RMI system allows an object running in one Java virtual machine to invoke methods on an object running in another Java virtual machine. RMI provides for remote communication between programs written in the Java programming language.

(5) The proposed visualization tools will be installed in all client computers. Once an event occurs in power system, clients will receive fault reports from the server. Analysis results of different applications will be properly presented to operator through Graphical User Interfaces which are incorporated in the visualization tools.

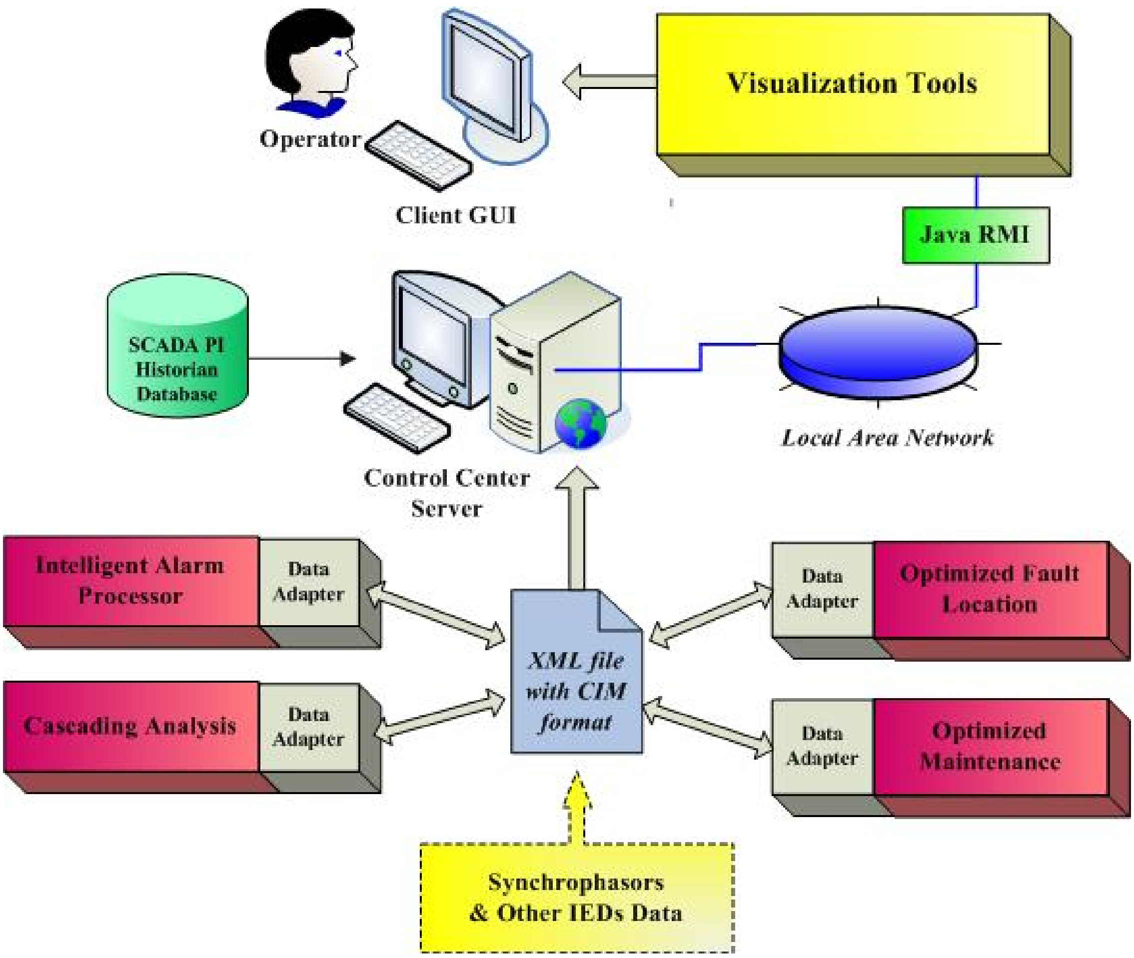

Figure 6: Information exchange structure

2.4 Functionalities of the visualization tool

The main objective of developing the control center graphical tools is to aid operators in monitoring and handling events in the entire system more efficiently. Thus the proposed visualization tool should have the following functionalities:

- Cause-effect report generation, which provides detailed fault report for both simple and complicated (multiple) system disturbances;
- Recommended solution suggestion which specifies recommended switching actions and other operating schemes for operators to adopt;
- Graphical User Interfaces which provide user-friendly display of results from different applications.

2.4.1 Cause-effect Report Function

The cause-effect report function is incorporated in the visualization tools. This function is used for interpreting fault reports conveyed from alarm processor and cascading analysis applications. A full description list including definition of an event, its cause and possible impacts to the system will be presented to operators.

After the information processing has been accomplished, the description list will be stored in different JAVA objects for further uses.

2.4.2 Recommended Solution Function

Similarly, the Recommended Solution function is also incorporated in the integrated graphical tools. It is responsible for extracting and forming recommended actions from the fault reports generated by application modules such as alarm processor and cascading analysis.

After the information processing has been accomplished, the recommended solutions will be stored in different JAVA objects for further uses.

2.4.3 Graphical User Interfaces

The GUI is the core part of our proposed visualization software. The following visualization functions will be incorporated in the graphical tools:

- Equipment Models (circuit breaker, tower, etc.)
- Aerial View (satellite view)
- Electrical View (alarms, power flow, etc.)
- Topological View (system one-line diagram)
- Hierarchical View (normal state monitoring)

The design of graphical user interfaces has been outlined in Figure 7.

Once an event or several complex events occur in power system, field measurement data collected from substation will be immediately sent to the alarm processor module. The alarms will be analyzed efficiently within a very short time period. A judgment whether a fault happened in the system will be made quickly. If a fault is confirmed, a comprehensive description of the power system condition causing the alarms will be generated by intelligent alarm processor, and then be conveyed to the control center through cause-effect function. Meanwhile the GUI will pop up a notification (with alarm sound) to inform operators that an event of interest has occurred.

In addition, the fault reports produced by alarm processor will also be conveyed to optimized fault location and cascading analysis modules. Further calculation and analysis will be carried out there. Reports from fault location module will be sent to visualization tool and processed in order to show the 2-D and 3-D pictures. Operators are able to take a comprehensive view of the geographical environment around fault location via 2-D and 3-D (and satellite, if available) pictures.

Report from cascading analysis module will be conveyed to cause-effect report as well as recommended solution functions, which will in turn produce a full fault description list and recommended actions. By uploading this information into the client GUI, the software will enable operators to get much detailed understanding of the fault and its possible cascading impacts on the entire system. It will also provide recommended actions for operators to prevent or mitigate cascading outages.

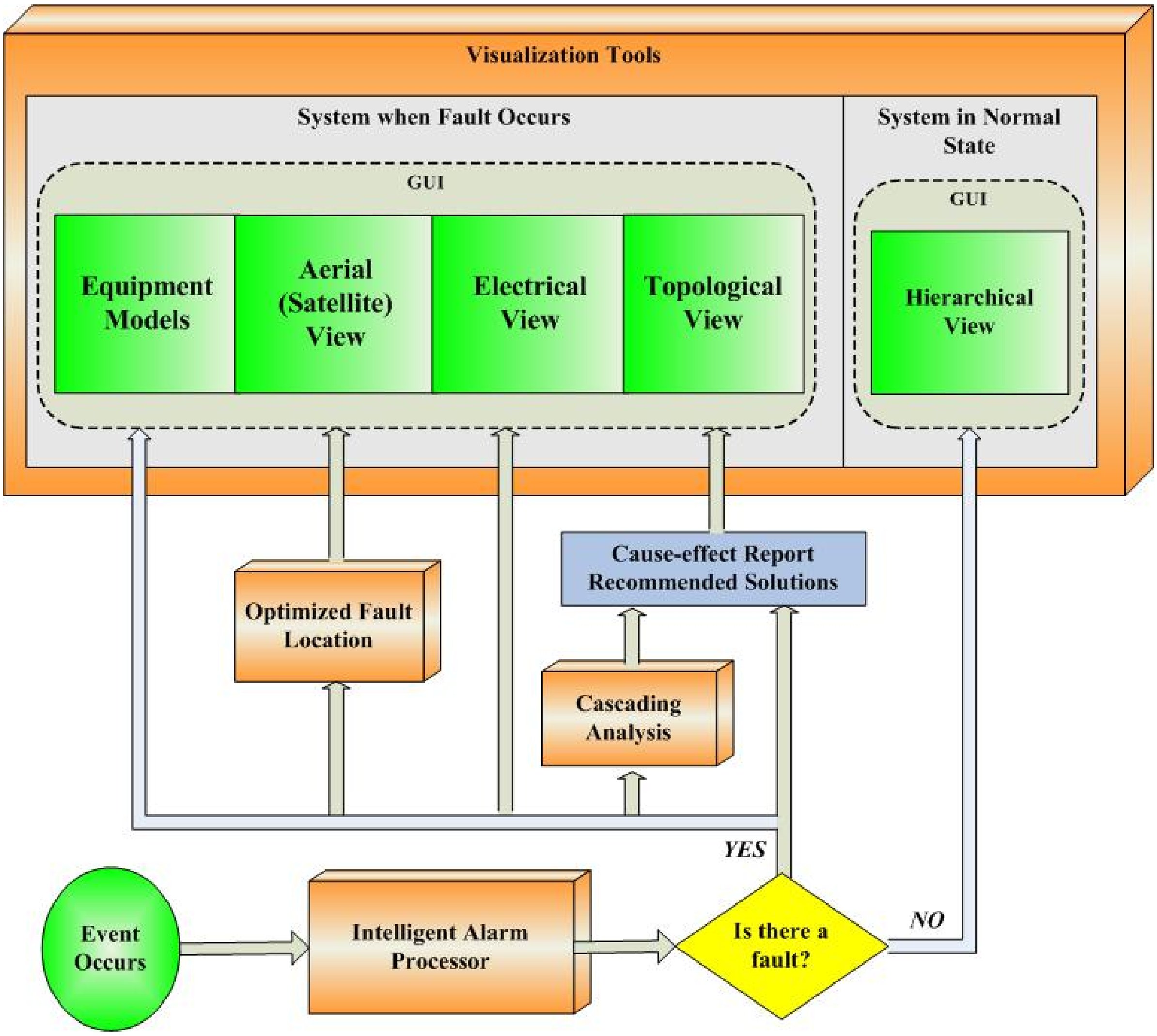

Figure 7: Functionalities of Visualization Tool

It should be noticed that 2-D, 3-D and satellite visualization are also designed to be available under system normal state. Hierarchical view is provided in this case. It will be used for operators to monitor the entire system conditions such as real-time power flow and component conditions on-line.

In summary, when fault occurs, the proposed graphical tools not only possess construction model view such as circuit breaker model or transmission tower model, but also provide aerial view, electrical view and topological view to aid operators collecting as much information about the geographical environment of the fault area as possible. Examples of the proposed 2-D and 3-D (aerial) graphics of one fault location case are demonstrated in Figure 8.

Figure 8: 2-D and 3-D view of the fault location

Part 3: Proposed System Architecture

In Part 2, we have specified the data inputs and outputs, information processing structure and requirements specification for each application module as well as the proposed visualization tool. In this section, an integrated control center GUI system will be presented by assembling all those existing modules together. Both the functional architecture and physical connection of the system will be proposed.

3.1 Roles of operator and utility groups

After the preprocessing of sampled data in substations and applications modules, the control center equipped with visualization tools will now have two distinct features comparing with those of traditional EMS system:

- The substation data and extracted information are shared with different utility groups, including protection engineers, dispatchers, maintenance technicians, etc., making sure the data/information are presented in the form most suitable for a given group;
- Each group receives the best information since the origin of substation data becomes transparent to the users and what they receive is the best information obtained using all available data.

There are several advantages of doing this:

- The information, not data, is sent from substations to the upper levels for the operators to be able to use it in real-time. The information is extracted from the data in the time frame allowing real-time use. This prevents the communication bottleneck. The raw data, if needed, is sent at a later time when the communication traffic is not so intensive;
- The local information is extracted close to the source using abundance of IED data synchronized using GPS receivers. If a coordination of local conclusions is needed for system-wide analysis, this can be accomplished at a centralized location through further exchange of information.

In the proposed system architecture, each utility group will be equipped with a computer with GUI client installed. The clients together with the server are interconnected through a Local Area Network. Operator is at the top level and responsible for monitoring real-time system conditions. Other utility groups also receive information from client computers. The information they receive is not necessarily the same as the one received by the operator, but it is the most relevant information that pertains to their particular responsibilities.

Once an event occurs, the visualization tools will inform operator immediately. Operator could then assign tasks to different groups according to the fault reports and recommended solutions. The maintenance crew will be requested to repair system components identified with accurate fault location while protection engineers will be asked to analyze the fault clearance sequence and dispatchers will be required to re-dispatch the power generation and load flow to balance the whole system.

3.2 Functional architecture of GUI system

The proposed architecture of the GUI system is shown in Figure 9.

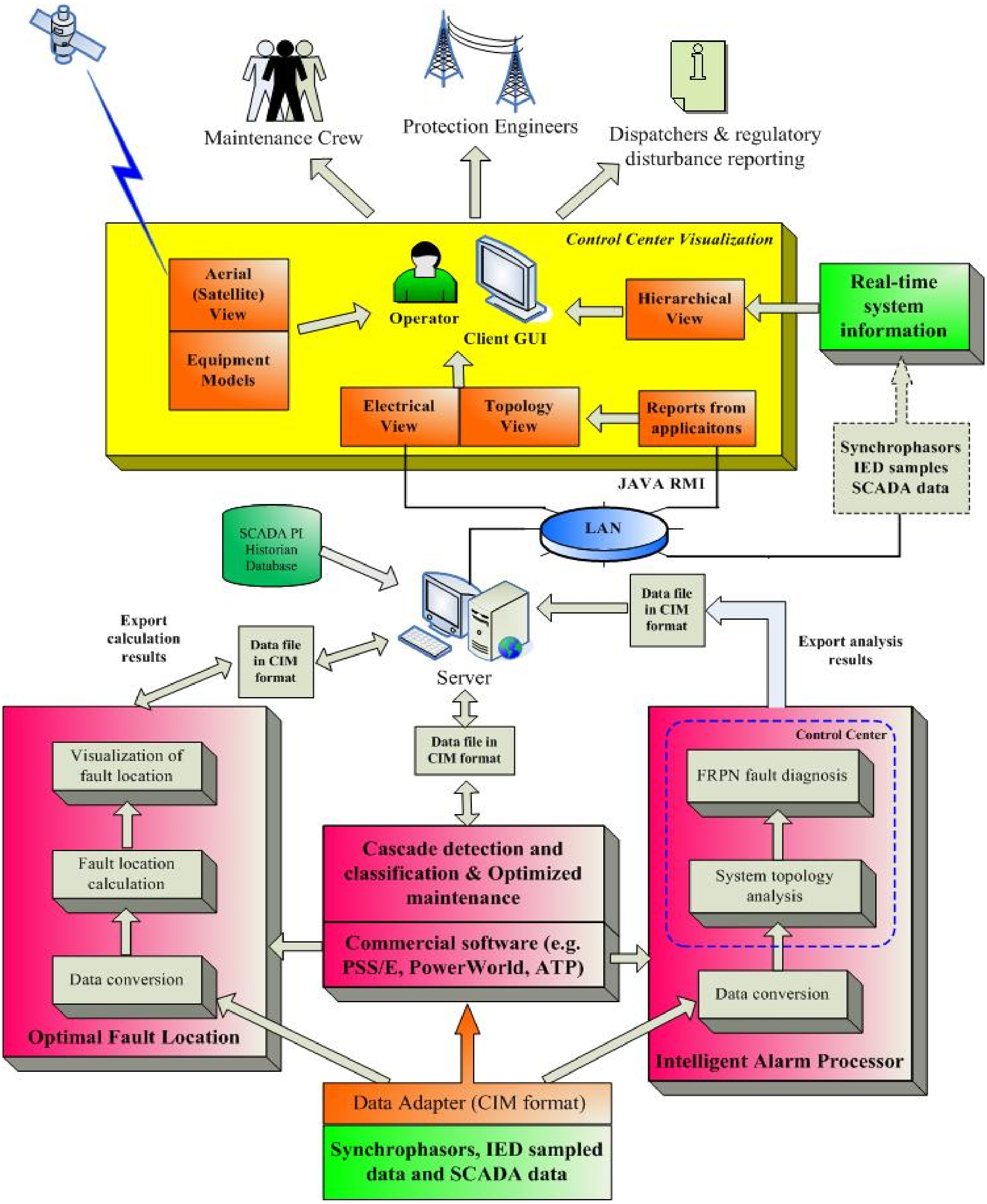

Figure 9: Architecture of the GUI system

Part 4: Other Considerations

Security and maintenance are two fundamental considerations in software development. They have been broadly discussed in this section.

4.1 Security consideration

For the system architecture shown in Figure 8, once the server/client connection has been established, network administrator needs to assign particular accounts and passwords for different utility groups. User identification is strictly verified using user authorization and authentication techniques.

Different utilities will have different procedures for authorization and authentication for the use of the visualization software. Each module and interface should only be accessed by particular utility group who is responsible for that part. Any other groups or individuals will not be allowed to access this module and related graphical user interfaces.

Training of utility staff is necessary before they are permitted to use the visualization software. This is not only for the reason of the security of the local area network, but also for the safety of the entire power system, because any unauthorized or improper operation may cause very severe consequences.

4.2 Software maintenance consideration

In software engineering [15], software maintenance is the modification of a software product after delivery to correct design faults, to improve performance or other attributes, or to adapt the product to a modified environment (ISO/IEC 14764).

Our proposed visualization software requires a group of computer engineers for continuous maintenance after it has been installed in control center. The responsibilities of the maintenance group include:

- Monitoring software configuration and on-line application performance;
- Handling problems identified during software utilization;
- Proposing solutions to any new requests from users
- Making software modification;
- Updating or replacing outdated software.

Part 5: Conclusions

5.1 Accomplishments

The requirements specification for linking the proposed GUI with intelligent alarm processor and optimized fault location has been researched in detail.

The following is a summary of the research progress achieved so far:

- The applications of intelligent alarm processor and optimized fault location have been outlined.
- The data sources and their preprocessing before utilized in application modules have been specified. CIM format and XML file have been introduced for data storage and exchange respectively.
- The inputs, outputs, and information processing structure of each application have been summarized and described in detail. Control center server is introduced for on-line collecting of information from new applications and SCADA database. A server/client network is established.
- The information exchanging mechanism between control center server and client computers has been described. A general structure of the visualization tools has been presented.
- Different modules have been designed to execute different functionalities. 2-D, 3-D and satellite views are proposed to aid operators better monitoring the system.
- The architecture of the integrated data/application/visualization system has been presented. Responsibilities of operators and each utility group have been specified. Several other considerations have been discussed.

With the above accomplishments, the research on requirements specification of the visualization tools for intelligent alarm processor and optimized fault location has been completed.

5.2 Future work

Since the specification work for the software has been completed, our research will focus on the development of the control center visualization tools as the next step. Functional design for visualization tools will be investigated first. Then the implementation specification for visualization tools will be explored.

Field installation and in-service demonstration of visualization tools will be carried out soon after the software is developed.

References

[1] W. Prince, B. Wollenberg and D. Bertagnolli, "Survey on Excessive Alarms," IEEE Transactions on Power Systems, pp. 188-194, August 1989.
[2] X. Luo, M. Kezunovic, "Implementing Fuzzy Reasoning Petri-nets for Fault Section Estimation," IEEE Transactions on Power Delivery, Vol. 23, No. 2, pp. 676-685, April 2008.
[3] Y. Wu, M. Kezunovic, T. Kostic, "An Advance Alarm Processor Using Two-level Processing Structure," Power Tech 2007, Lausanne, Switzerland, July 2007.
[4] M. Kezunovic, "Optimized Fault Location," Power Systems Engineering Research Center, Project Final Report, PSERC Publication 08-07, April 2008.
[5] M. Kezunovic, M. Knezev, "Selection of Optimal Fault Location Algorithm," IEEE PES General Meeting, Pittsburgh, July, 2008.
[6] M. Kezunovic, B. Perunicic, "Automated Transmission Line Fault Analysis Using Synchronized Sampling at Two Ends," IEEE Transactions on Power Systems, Vol. 11, No. 1, February 1996.
[7] M. Kezunovic, Zheng Ce, "Monitoring Power System Dynamic Performance Using Synchronized Sampling," CIGRE Conference on Monitoring of Power System Dynamics Performance, St. Petersburg, Russia, April 2008.
[8] U.S.-Canada Power System Outage Task Force, "Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations," Tech. Rep., April 2004, Available: http://www.nerc.com.
[9] M. Kezunovic, A. Abur, "Merging the Temporal and Spatial Aspects of Data and Information for Improved Power System Monitoring Applications," IEEE Proceedings, Vol. 9, Issue 11, pp 1909-1919, 2005.
[10] M. Kezunovic, B. Matic-Cuka, O. Gonen, N. Badayos, "An Advanced Data/Information Presentation in Power Systems," ISCA 23rd International Conference on Computers and Their Applications-CATA 08, Cancun, Mexico, April, 2008.
[11] T. J. Overbye, et al, "Effective Power System Control Center Visualization,", Power Systems Engineering Research Center, Project Final Report, PSERC Document 08-12, May 2008.
[12] PSS™E Load Flow ©Siemens Power Transmission & Distribution, Inc., PTI, Schenectady, NY, USA.
[13] PowerWorld Retriver, User’s Guide, PowerWorld Corporation, 2006.
[14] Common Information Model (CIM), IEC 61970, Part 301~303, 2001
[15] http://en.wikipedia.org/wiki/Software_maintenance

Appendix B

Synchronized Sampling Uses for Real-Time Monitoring and Control/RTGRM

Functional Design for Visualization Tools

Project Progress Report

Mladen Kezunovic

Ce Zheng

Texas Engineering Experiment Station

October 31, 2008

Executive Summary

Prior work on the project explored requirement specifications for the control center visualization tools. This report is the second deliverable from several consecutive research and development activities.

During this second phase of the project, we focused our efforts on designing functionalities of the proposed graphical user interface (GUI) software. A set of specific solutions for the GUI applications, as well as the user interfaces have been discussed in detail in this document. The main research achievements are:

- The design for the overall control center visualization GUI tool has been specified for future applications
- The GUI has two parts: one for fault event view (fault location) and the other for the power system monitoring view in the normal state (alarm processor).
- The functions of GUI for each application module (fault location and alarm processor) have been specified in detail;
- For faulted systems, four different views are proposed: equipment model view, aerial view, electrical view and topological view;
- Hierarchical view is proposed for the purpose of supervision of normal state of the power system;
- Construction view and operational view using equipment models are designed to better demonstrate behavior of different devices;
- Both 2-D satellite image and 3-D models are integrated to represent an aerial view module;
- Real-time power flow and transmission line alarms are displayed in the electrical view. Component connection and equipment status are shown in topological view;
- Other features such as detection of cascading events and risk based failure analysis for power apparatus could be added to the GUI in the future.

Now when the functional design for GUI has been specified, our investigation will move on to the implementation specifications and development of the visualization tools. As soon as the whole development of GUI is completed, on-site demonstration using field cases will be carried out.

Part 1: Introduction

Utilities are striving to reduce outage time and improve quality of power delivery. An effective way to accomplish this is to speed up fault location procedure through automation of both data retrieval and fault analysis [1-3].

This project aims at addressing a set of visualization tools for future generation of switchable networks, with a particular emphasis on visualization used in the control center. The first phase of the project has explored the specific requirements for developing the graphical software (see Appendix). This is the second report which provides functional design specification for the visualization tools.

While designing the specific interfaces integrated in the GUI software, our main objectives are the following:

- Obtained data and calculated results should be available to different user groups involved in decision making and subsequent actions simultaneously ;
- The interactive view of the events should show both the physical environment that surrounds fault location, as well as detailed views of the equipment involved in fault clearing
- Create user friendly interface that helps the decision making process and does not contain unnecessary details

Our goal is to make the graphical software transparent to users, meaning that the user will not have to worry how the data is obtained as long as the information presented to the user is highly descriptive. The best way to achieve this transparency is to understand the user's required tasks and incorporate that feature into the user's interface [4, 5].

In the design process, we are trying to provide users with the availability of direct manipulation of objects and actions of interest [6, 7]. For this purpose, component models as well as satellite images are available in the GUI.

Specific designs have been described in Part 2. Several fundamental ideas such as model manipulation, fault area view, power flow and alarm display have already been incorporated in the user interface. Since our programming considers maintenance issues as reported in the earlier document (see Appendix), future expansion of functionalities may be added to our GUI as software development proceeds.

Part 2: Functional Design of GUI

Our proposed GUI software is composed of several functional modules: Equipment Models View, Aerial View, Electrical View, Topological View and Hierarchical View. Each of these application modules is implemented through a set of graphical user interface (GUI) tools.

In this part, all the GUI application designs will be described in detail, including various interfaces to the applications and users.

2.1 Structure of the GUI Tools

The general structure of the GUI software is shown in Figure 1. Visualization tools are connected with different applications through local area network (LAN).

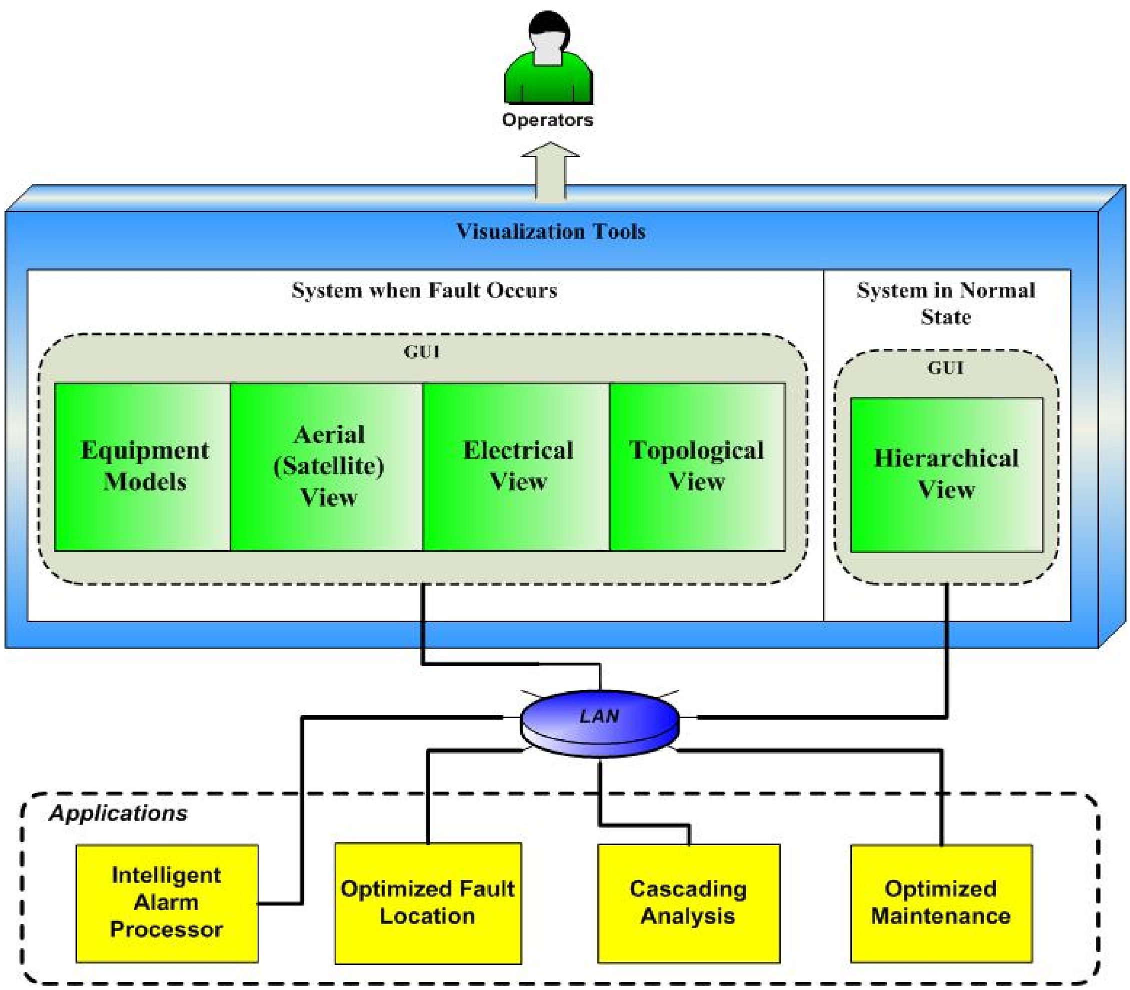

Figure 1: Software Architecture

Information exchange is also achieved using local area network. Analysis results, after being generated, will be transmitted from the application modules to the GUI software and later presented to the operators through different graphical user interface tools.

Our visualization tools are designed to provide user with an easy way to access relevant information when:

- Power system is in normal state;
- Fault is present and system changes its state;
- Power system changes its state from normal to faulted.

Four types of user interfaces will be developed in the case of faults occurring in a power system. They are equipment models view, aerial view, electrical view and topological view. When power system is running in a normal state, hierarchical view interfaces will be deployed to monitor the whole power system. The following sections are focused on describing the detailed designs of all of the four graphical user interface modules.

2.2 Equipment Models View

In our proposed GUI system, various power system equipments will be modeled and presented to operators through user interfaces. Currently the modeling of two types of devices is being carried out: Transmission Tower and Circuit Breaker. The modeling of several other equipments will proceed in the future. The design for Equipment Model module is shown in Figure 2.

2.2.1 Transmission Line Tower Model

Transmission line tower is a complex structure that holds the transmission lines through insulators. If a fault occurs on particular segment of transmission line, it is likely that the fault will occur on the tower insulators. The tower structure varies with voltage level and also depends on type of circuit it carries (single circuit, double circuit, etc). Based on the conductor arrangements, towers can be classified into single-level, two-level, three-level, etc. A construction view of the tower in 3-D is necessary to help the maintenance crew visualize the type of the tower, insulators etc. before heading for the actual fault site.

In our proposed user interface, the tower model view is designed so that the model can be interactively rotated, enlarged, and reduced as needed.

2.2.2 Circuit Breaker Model

The second type of equipment to be modeled in our graphical software is circuit breaker. Circuit breaker is an electro mechanical device which has several mechanical components such as trip and close coils, trip and close latch mechanisms, connecting rod, rollers, cams, etc. Visualization of these parts could help maintenance crew to make better decisions for both diagnostic and maintenance purposes. Visualization of circuit breaker mechanism is divided into two separate sections: constructional and operational view.

Constructional view is basically a 3-D representation of the circuit breaker operating mechanism. In our former work, a Westinghouse make vacuum circuit breaker of type 3 is used in developing the 3-D representation. This particular breaker is equipped with spring operating mechanism (similar treatment can be applied to other types of operating mechanisms such as pneumatic or hydraulic).

Similar to the tower model, it is possible to rotate and change size of the object in all directions to gain better understanding of its constructional details.

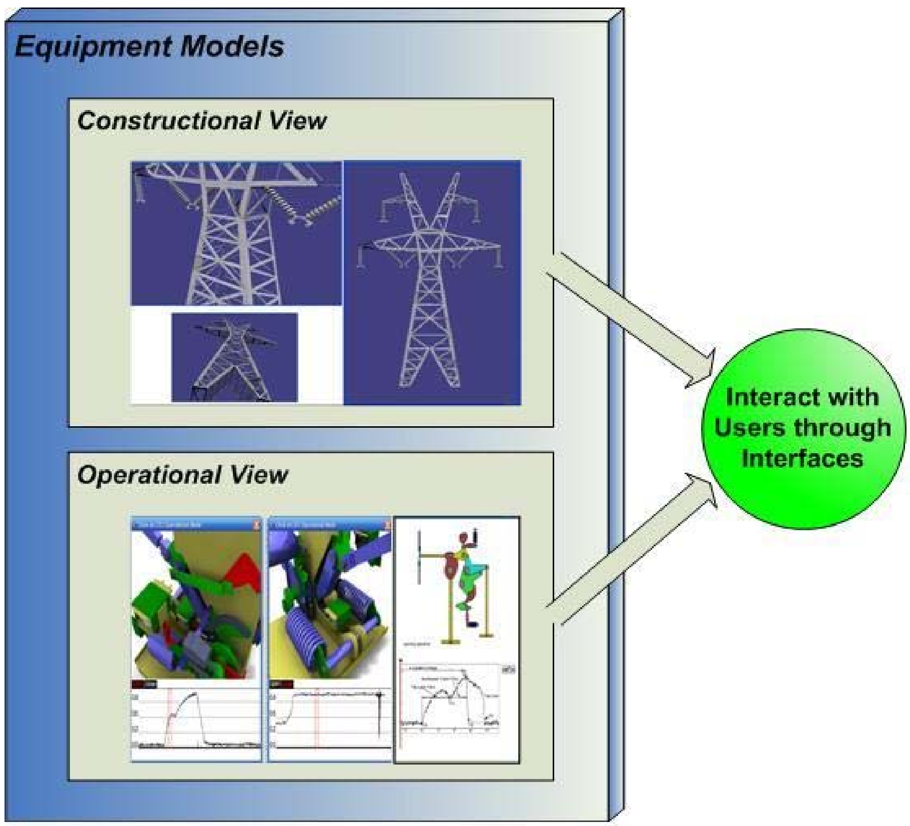

Figure 2: Design of Equipment Models View

The operational view shows the animation of the operating mechanism while circuit breaker operates. With availability of recordings from a circuit breaker monitor (CBM) device, which records circuit breaker control circuit signals, each operation of circuit breaker can be evaluated through an expert system analysis. The idea of operational view is to show how the operating mechanism behaves during the operation of the circuit breaker. As the operating mechanism is often enclosed in a box, it is not easy to visually observe the movement of various parts. For selected CBM recording, corresponding signals are correlated with mechanical movements of the circuit breaker.

2.2.3 Models of Other Equipment

Besides the tower and circuit breaker, there are several other equipment types which need to be considered, for example, insulators, generators, transformers, etc. These devices need to be modeled since faults are likely to occur on them. For example, overstress in insulators result in small cracks, reducing their insulation strength, hence possibly causing faults. By establishing their construction and operation models, operators as well as other utility groups are able to better interpret causes and consequences of a fault or better maintain the integrity of the system after a fault occurs.

2.2.4 GUI Design for Equipment Models View

The functional design of the user interfaces for Equipment Models View is shown in Figure 3.

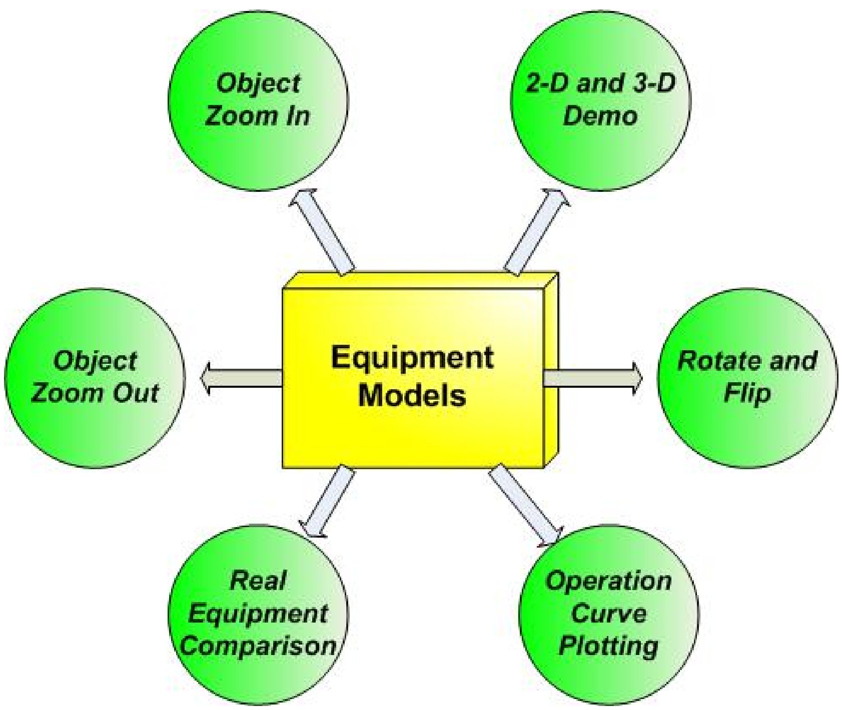

Figure 3: Functional Design of Equipment Model Interfaces

For the purpose of better interaction between users and GUI software, the following basic functionalities are designed and will be integrated in the user interfaces:

- Model zoom in and zoom out;

- Rotation and flip of the equipment model;
- 2-D and 3-D demonstration of the model operation viewed from all directions;
- Plotting the operation performance indicator within a time period specified by users;
- Compare the structure of a model with that of real equipment;

As the software development proceeds, new functionalities and features may be designed and added to the user interface.

2.3 Aerial View

Once the fault location is calculated it is very important that is the details are effectively presented to maintenance crew. The Aerial (satellite) View module translates results from fault location report files into a view of the corresponding faulted zone. Through this module it is possible to see physical environment of the faulted area, as well as the behavior and status of equipment involved in the fault event.

The design of the Aerial View module is shown in Figure 4.

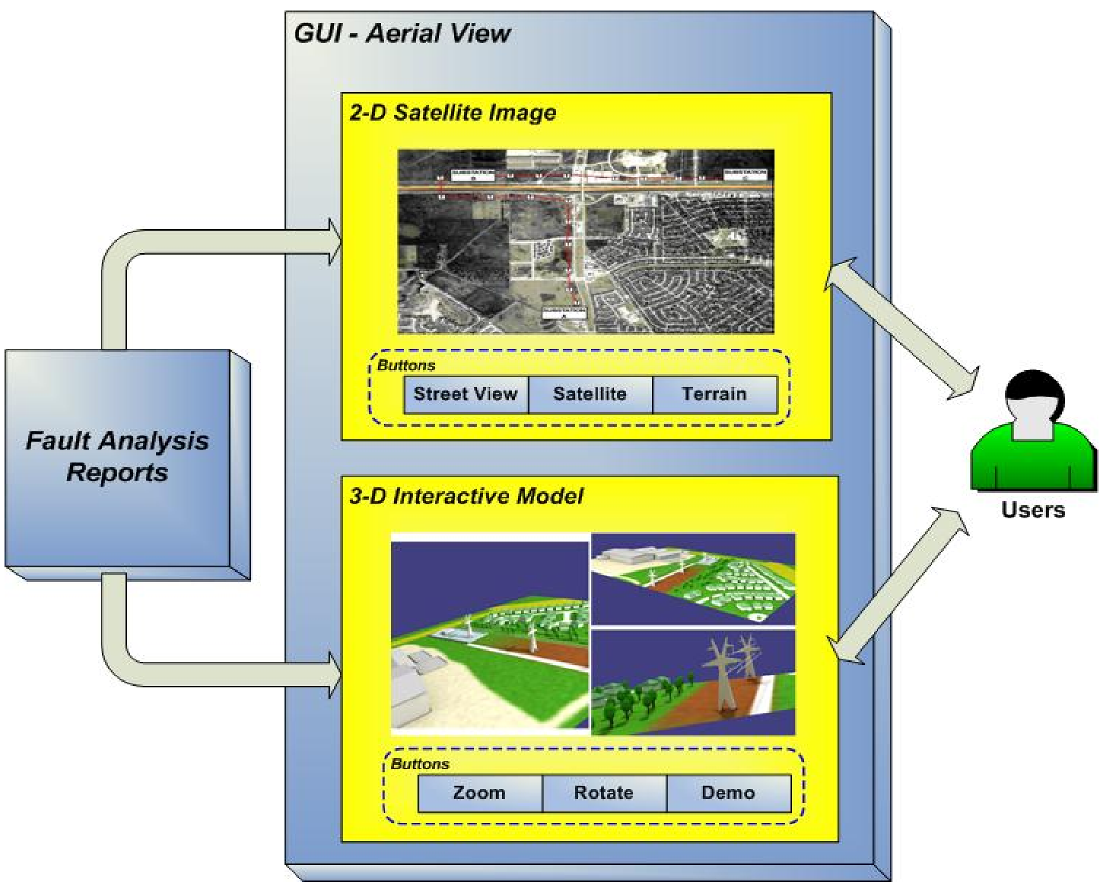

Figure 4: Design of Aerial (Satellite) View

For the graphical user interfaces integrated in this module, two kinds of views are designed to present data about the fault location:

- 2-D View: the fault line and fault location are marked on the 2-D satellite image, while natural environment around fault location is kept;
- 3-D View: the 3-D model is exported for an interactive demonstration of the constructional view of the transmission lines.

2.3.1 2-D Satellite Image

By clicking the button marked ‘2-D Satellite Image’, operators and related utility staff can open the interface for the satellite view of the faulted area. In our design, several features will be integrated in this interface:

- Satellite image of streets, traffic routes and terrain view;
- Zoom in and zoom out of specific area with the image;
- Automatic overlay of transmission lines on the satellite image;
- Automatic indication of the fault location on the satellite image;

2.3.2 3-D Model View

The 2-D satellite image doesn’t provide clear information about the type of the transmission towers that experience the fault nor does it give information about the height of other objects that surround the fault. To overcome these deficiencies, the 3-D model is designed for an interactive demonstration of the constructional view of the transmission line. For the demonstration purpose, the area is modeled manually as shown in Figure 4. Several buttons are designed in the interface for the interactive changing of model size, rotating of the model, and viewing the nearby environment.

For the proposed integrated software, the 3-D aerial view model should be generated in run time by combining pre-modeled elements (e.g. houses, tress, etc.) according to a topology scene file. This would speed up the process of modeling the environment and decrease the required memory amount to store the data.

In the future, several new functionalities may be added to the user interface as the development of software continues.

2.4 Electrical View

This is another independent module integrated in the visualization tools. Electrical View includes the visualization of alarms, power flows [8], system one-line diagram etc. As for the GUI, several basic functionalities are designed:

- One-line diagram of power system is established and updated using the real-time data provided by applications;

- Once new fault event is processed the corresponding faulted line of the diagram starts blinking. Description of the alarm is also presented to users;
- Power flow before and during the fault event is calculated and updated every several seconds, and presented to operators through the interface;

The design of the Electrical View module is shown in Figure 5.

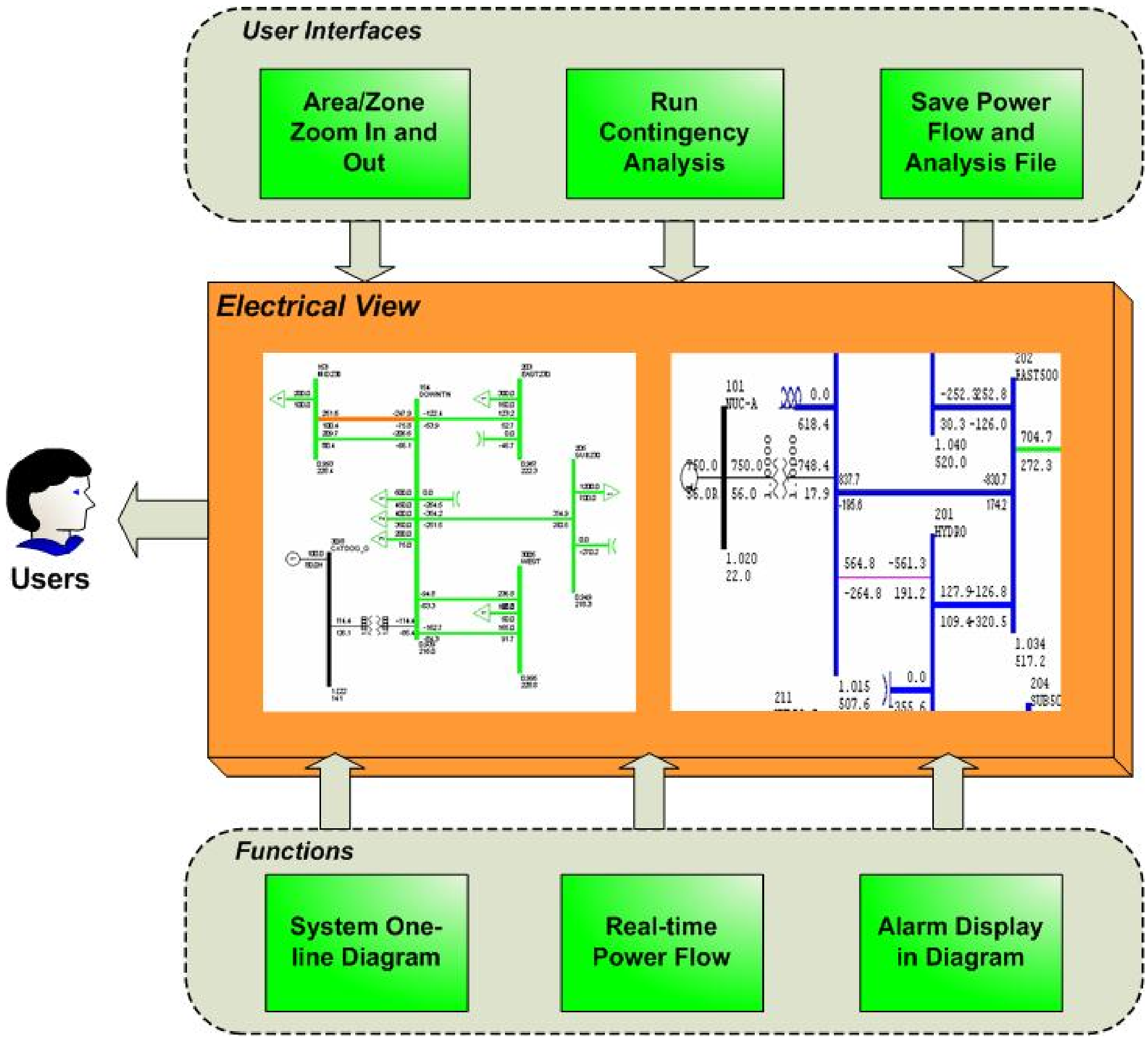

Figure 5: Design of Electrical View

As is shown in Figure 5, the GUI has integrated functions which can facilitate interaction with users. Except those basic operations such as zoom in and out, users can run contingency analysis such as N-1 analysis to analyze system stability and security. Related analysis reports as well as system power flow information can be stored in files for future use. Similarly, as the software development continues and on-site implementation begins, new functions may be added according to needs.

2.5 Topological View

The power grid topology describes connectivity of the various components in the power system. In order to process retrieved fault event recordings, they must be related to a specific position from which they are measured and the information how the measurement positions were interconnected at the time of the fault occurrence. Therefore, the system topology must be known. Beside the connectivity, it is necessary to obtain information about component characteristics at a specific moment.

The design of the Topological View module is shown in Figure 6.

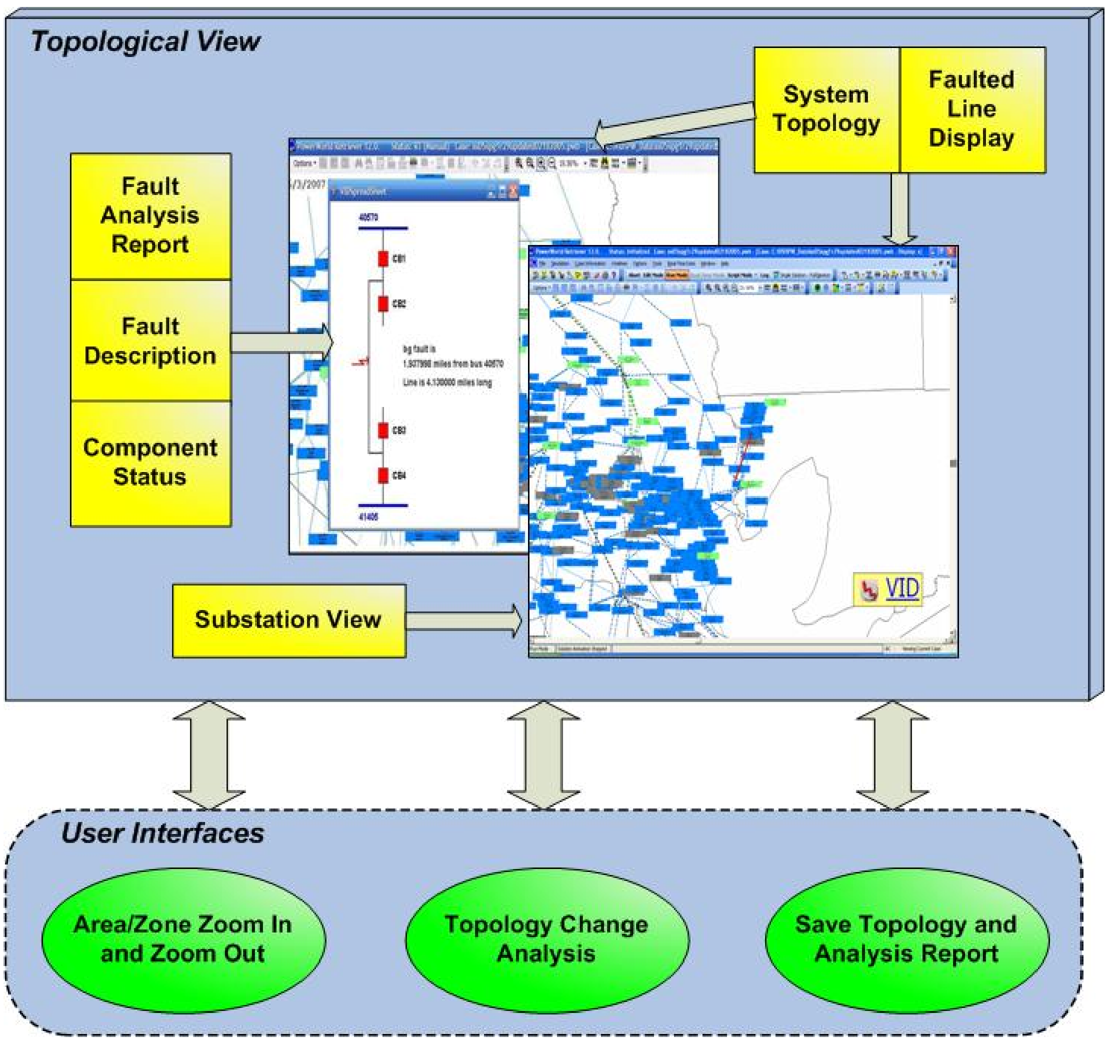

Figure 6: Design of Topological View

The features incorporated in this module include:

- Updated system topology overview;
- Substation number and connectivity view;

- Fault description including accurate fault location and faulted line display;
- Component characteristic and real-time status display;

Users not only can zoom into specific area to check fault effect, component status and so on, but they can also run analysis by changing system topology. Similarly, analysis report and updated system topology can be saved for future use.

2.6 Hierarchical View

When system is in a normal state, real-time visualization and monitoring of the power flow and related operation are necessary. The graphical software can import real-time data by connecting to data sources that enable users to perform supervision and visual analysis of power system operations. The Hierarchical View module is used to track system behavior in normal state. Real time data are obtained from SCADA PI Historian.

The design for Hierarchical View module is shown in Figure 7.

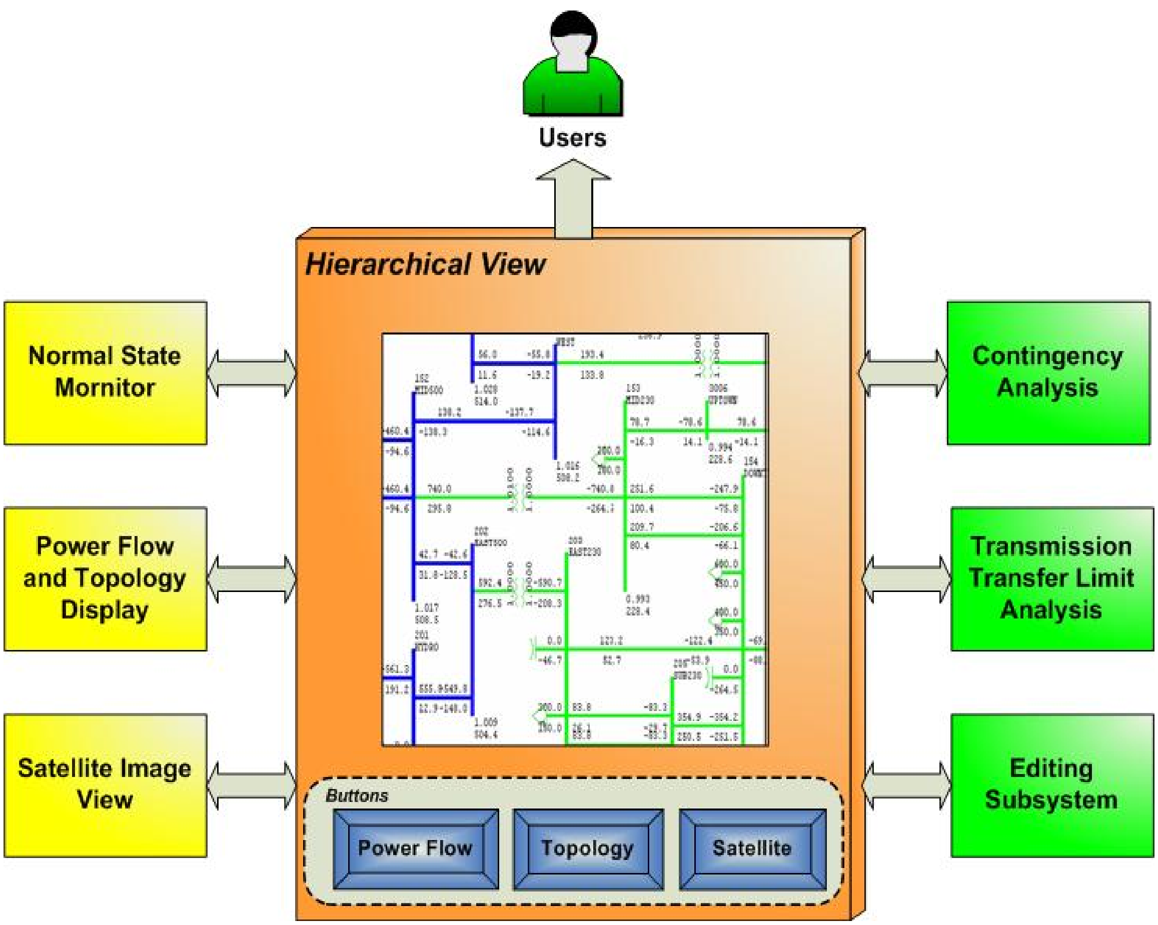

Figure 7: Design of Hierarchical View

As is shown above, the functions incorporated in this module include:

- Real-time monitoring of power system in normal state;
- Updating power flow and system topology display;
- Viewing both 2-D satellite image and 3-D model view are available;
- Selecting and editing subsystems easily;
- Calculating transmission transfer limits;
- Performing contingency analysis and saving analysis reports for future use;

As the software development proceeds and on-site implementation takes place, new functions may be designed and added to the Hierarchical View GUI.

Part 3: Conclusions

3.1 Accomplishments

In this document, the functional design for the GUI tools has been researched in detail. The following is a summary of the design progress so far:

- The basic design for the overall control center visualization tools has been provided;
- The GUI has been divided into two parts. One is for fault event view. The other is for monitoring normal power system state.
- The functions of GUI for each application module the two applications have been designed in detail;
- For fault event in a power system, four different modules have been proposed: equipment models view, aerial view, electrical view and topological view;
- Hierarchical view is proposed for the purpose of normal state supervision of a power system;
- Construction view and operational view of Equipment Model are designed to better demonstrate operation of the devices;
- Both 2-D aerial (satellite) image and 3-D model are integrated in the Aerial View module;
- Real-time power flow and line alarms are to be displayed in Electrical View. Component connection and equipment status will be shown in Topological View;
- Several other interactive features such as contingency analysis and transfer limit calculation are also designed to be integrated in GUI.

With the above achievements, the functional design for the control center visualization tools has been completed.

3.2 Future work

So far, the GUI requirements specification work and functional design for visualization tools have been completed. Next the implementation specification for the proposed visualization tools will be investigated. After that, our research team will focus on the deployment of the control center graphical software.

Field installation and in-service demonstration of visualization tools will be carried out soon after the software is developed and deployed.

References

[1] M. Kezunovic, "Optimized Fault Location," Power Systems Engineering Research Center, Project Final Report, PSERC Publication 08-07, April 2008.
[2] M. Kezunovic, M. Knezev, "Selection of Optimal Fault Location Algorithm," IEEE PES General Meeting, Pittsburgh, July, 2008.
[3] U.S.-Canada Power System Outage Task Force, "Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations," Tech. Rep., April 2004, Available: http://www.nerc.com.
[4] T. J. Overbye, et al, "Effective Power System Control Center Visualization,", Power Systems Engineering Research Center, Project Final Report, PSERC Document 08-12, May 2008.
[5] Wilbert O. Galitz, "The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques," 3rd Edition, Indianapolis, IN: Wiley Pub., 2007
[6] http://en.wikipedia.org/wiki/Graphical_user_interface
[7] Tim Hartley, "GUI Design Fundamentals," Phoenix, AZ. Computer PREP. Inc., 1998
[8] PSS™E Load Flow ©Siemens Power Transmission & Distribution, Inc., PTI, Schenectady, NY, USA.