

FAULT SIGNATURE-BASED CASE LIBRARY SYSTEM AND METHOD FOR POWER SYSTEM TROUBLE-SHOOTING AND RECOVERY

FIELD OF THE INVENTION

This invention relates to fault diagnosis and system recovery. More particularly, this invention relates to a computational method and software system that employs a fault signature-based system for diagnosing and trouble shooting of complex systems.

BACKGROUND OF THE INVENTION

Electric power is the most commonly used form of energy. Unlike other forms of energy, the production and transmission of electricity are relatively efficient and inexpensive. However, electricity is not easily stored and must generally be used as it is being produced. Therefore, the states of electric power systems must constantly be monitored to prevent wastage due to a system disruption. If faults are detected, the faults must be timely diagnosed and corrected quickly to prevent the generated electricity from being wasted.

When a fault occurs in the electric power system, which causes failure of the system or interruption of power supply, system monitors must take immediate recovery actions to restore the system. However, it is a problem to determine the correct recovery actions when using the existing SCADA technology (SCADA stands for Supervisory Control And Data Acquisition). One prohibitive factor in determining the proper recovery action is the lack of fault diagnosis knowledge and efficient systems which could advise the engineers to perform system restorations.

Existing technologies in the field of fault diagnosis can be classified into the following categories: model-based diagnosis (MBD), experience-based diagnosis, Case Based Reasoning (CBR) systems.

A considerable amount of effort has been made in the area of model based diagnosis. For example, U.S Patent No. 7,260,501 B2 issued to Pattipatti et al entitled "Intelligent Model-Based Diagnostics for System Monitoring, Diagnosis and

Maintenance” discloses an intelligent MBD system for system monitoring, diagnosis and maintenance. However, the main disadvantage of such systems is that models for each possible scenario or error pattern must be generated for use in diagnosing and providing solutions to scenarios observed while monitoring the system. Generally, these models are difficult to create and the difficulty increases as the complexity of the monitored system increases. Further, the modelling of the systems becomes nearly impossible when the configuration of the monitored system is constantly changing such as in a power supply system. Given these difficulties, the MBD systems are only suitable for small or medium-sized systems that have a fairly static configuration and relatively simple problems to diagnose.

Experience based diagnosis (or diagnosis by an expert system) is based on experience with the system rather than mathematical models. Traditional expert systems use production rules to store the domain knowledge for a monitored system. A production rule consists of two parts: a sensory precondition (or “IF” statement) and an action (or “THEN”). If a sensory precondition matches the current state of the system, then the action will be triggered. If an action is executed, the action is said to have fired. In a knowledge based diagnosis system, the precondition may be a description of the fault symptoms, and the action may be a system restoration. An expert system includes a database, sometimes called working memory, which maintains current state or domain knowledge, and a rule interpreter. The rule interpreter is software that prioritizes actions when more than one action is triggered. Problems in expert systems with a rule based diagnosis system include knowledge capturing and representation of conditions. It is difficult to describe the preconditions or state of the diagnosis system, in general, in a consistent manner. Often, it is more efficient to automatically record the state of the system through an online monitoring system.

More recently, the CBR systems have become more commonly used, in particular for large and complex systems, such as power supply systems. CBR systems include a set of observations, a set of diagnostic solutions and a map correlating the observations to the diagnostic solutions. These three components are stored in a case memory.

CBR systems heavily depend upon the structure and content of the case memory because an observed problem is diagnosed by recalling a previous observed problem and using the map to provide the diagnosis of the previously observed problem to resolve the new problem. In CBR systems, the matching is performed by case search and matching processes. Both processes need to be both effective and reasonably time efficient. Further, since the diagnosis for a new problem has to be retained for future use, these requirements also apply to the method of integrating a new case into the memory.

The representation problem in CBR systems is primarily a problem of deciding the information to store an observed problem, finding an appropriate structure for describing and storing the information. Furthermore, the structures stored in the case memory must be organized and indexed for effective retrieval and reuse. An additional problem is integration of the case memory structure into a model of general domain knowledge, to the extent that such knowledge is incorporated.

In the prior art, there are two influential case memory models: the dynamic memory model and the category-exemplar model.

The dynamic memory model is described in the book titled "Dynamic Memory: A Theory of Reminding and Learning in Computers and People" by RC Schank - 1983 - Cambridge University Press New York, NY, USA. The described case memory is shown in Figure 1. The case memory in this model is in a hierarchical structure of episodic memory organization packets or Generalized Episodes (GE). In this hierarchy, specific cases sharing similar properties are grouped under a more general structure. As shown in figure 1, GE 50 contains three different types of objects: norms 52, indices 54 and cases 58. Norms 52 are features common to all cases indexed under a GE. Indices 54 are features used to group cases. An index may point to a more specific GE, or directly to a case. An index is composed of two terms: index name 54 and index value 56.

A new case description is given and the hierarchical structure is searched to find the best match. During the search, the input case structure is "pushed down" the

hierarchical structure, starting at the root node. When one or more features of the case matches one or more features of a GE, the case is further discriminated based on its remaining features. Eventually, the case with most features in common with the input case is found.

New cases are stored to the hierarchy in a similar manner. During storing of a new case, when a feature of the new case matches a feature of an existing case, a GE is created. The two cases are then discriminated by indexing each of the cases under a different index below the new GE. If during the storage of a new case, two cases (or two GEs) end up under the same index, a new GE is automatically created. Hence, the hierarchical structure is dynamic in that similar parts of two case descriptions are dynamically generalized into a GE, and the cases are indexed under this GE by different features in the cases.

A matching case is retrieved by finding the GE with the most norms 52 in common with the problem description of the new case. Indices under that GE are then traversed in order to find the stored case that contains most of the additional matching features. Storing a new case is performed in the same way, with the additional process of dynamically creating GE, as described above.

Since the index structure is a discrimination network, a case (or pointer to a case) is stored under each index that discriminates the case from other cases. This leads to exponential growth of indices as the number of cases in the hierarchy increases. Most CBR systems using this type indexing scheme must limit to the choice of indices for the cases to reduce searching time. In some systems, for example, only a small vocabulary of indices is permitted.

Besides limitations on the size of the indexing scheme, it is difficult and costly for the diagnostic solution software developers to define these complex indexing structures when using RC Schank's dynamic memory model. In fact, in order to develop a workable memory structure, the developers must possess a good understanding of the specific application domain.

The category exemplar model is described by B. Porter, R. Bareiss, and Robert H an article entitled "The Category & Exemplar Model to Organize Cases in a Case Memory" (see Concept learning and heuristic classification in weak theory domains. *Artificial Intelligence*, vol. 45, no. 1-2, September 1990. pp 229-263). In the described model, cases are also referred to as exemplars.

The described model is shown in Figure 2. In the model, a case memory is embedded in a network structure of categories, cases, and index pointers. Each case is associated with a category 66. An index may point to a case or a category. The indices are of three kinds: feature links 64 pointing from problem descriptors (features 62) to cases or categories 66, case links 68 pointing from categories to associated cases 70 (called exemplar links), and difference links 72 pointing from cases to the neighbor cases that only differs from the case by one or a small number of features. A feature 62 is described by a name and a value. A category's exemplars 70 are sorted according to their degree of "prototypicality" in the category.

Matches to a new case are determined by searching for a case in the case base performed in the following manner. Input features of a problem case are combined to form a pointer. The pointer is used to search the case base for a case or category that shares most of the features of the pointer. If a remainder of the features in the pointer directly matches a category, the links to the most prototypical cases of the category are traversed, and the cases at the end of the links are returned. As indicated above, general domain knowledge is used to enable matching of features that are semantically similar between the new case and cases already in the network.

A new case is stored by searching for a matching case. After a matching case is found, appropriate feature indices are formed between the new case and the matching case. If a case is found with only minor differences from the new case, the new case may not be retained or the two cases may be merged by following taxonomic links in the semantic network.

As described by B. Porter et al, a case memory is embedded in a complex network structure of categories, cases, and index pointers. IT professionals often have difficulties constructing the knowledge mapping and representation models properly. The generated models often lack robustness. A poorly designed network structure often generates conflicting outputs for the same input. Another disadvantage of the complex network structure model is that both software development cost and maintenance cost are considerably high. For instance, even if a small modification is made on the system, the process of constructing the model system must be repeated. In addition, case retrieval by these complex approaches is slow and not efficient due to the complexity of the network.

Regardless of fault diagnosis model used, the observations, or state of the system, or fault condition, must be obtained and presented to the diagnosis system for analytical or digital analysis.

Fault signature (or fault signature analysis) is a common technology used in diagnosis study and application. Essentially the fault signature is the signal that is monitored electrically and has a characteristic indicative of a corresponding fault condition.

Fault signature analysis is often used to diagnose mechanical equipment, IC and other electronic systems. For example, U.S Patent No. 7,509,551 B2 issued to Koenemann et al and entitled "Direct Logic Diagnostics with Signature-Based Fault Dictionaries" (Koenemann et al) discloses a logic diagnosis method and system to diagnose faults in an integrated circuit using fault dictionary approach. The disclosed method comprises receiving a signature produced by a signature generator where the signature corresponds to the response of the circuit to no more than one test pattern; comparing the signature to entries of a fault dictionary; matching an entry of the fault dictionary to the signature and storing the fault in a list of fault candidates.

Another example is U.S Patent No. 7,246,039 B2 issued to Moorhouse and entitled "Fault Diagnosis System" (Moorhouse), which discloses a fault diagnosis system

and a fault diagnosis system for diagnosing faults in complex equipment. The system includes means for storing a set of diagnostic signatures which relates a set of known faults, and means for processing the diagnostic signatures and a set of fault symptoms identified for a current state of the equipment to calculate diagnostic data for identifying a fault causing the current state of the equipment.

In the system described in Moorhouse, the diagnostic signatures are a set of numerical values in the form of matrix, which is used to calculate probabilities to identify a fault that causes the current state of the equipment. There are numerous fault signature models developed for fault diagnosis. One example is a power signature which refers to "type and shape, curves of the power signal" as disclosed in U.S Patent No. 7,421,378 B2, issued to Pernestal and entitled "Power Signature Diagnosing".

Case based system technology may be used to diagnose complex and large systems. However, case based systems have limitations that include, but are not limited to, the case memory models being very complex, and case descriptions not being consistent. It is clear that performance of the case based diagnosis systems can be significantly improved if simplified case memory models and more consistent fault description method are utilized.

Further, there are many ways for human experts to write descriptions of a fault conditions, as well as related trouble shooting and system recovery knowledge. Therefore, traditional manual approaches lack efficiency and consistency. In fact, the quality of case library construction largely depends on individuals' expertise and training background.

Considering inconsistency and low efficiency of the traditional manual approaches used for constructing the library case record, there is a need to improve and to address these issues in any monitoring and diagnosis system.

SUMMARY OF THE INVENTION

The above and other problems are solved and an advance in the art is made by a fault signature-based case library diagnosis system and method in accordance with this invention. A first advantage of a system in accordance with this invention is that the system may be installed in existing monitoring system. Hence, no additional circuitry is required. A second advantage of a system in accordance with this invention is that, it improves traditional case based diagnosis system performance in terms of case presentation consistency, solution search efficiency and robustness. A third advantage of a system in accordance with this invention is that, there is no additional effort required to redevelop a new software system even for a complete new environment because the new system can be configured for any new environment with minimal effort. Therefore the new method can save up to 95% of the total development time which is normally required to build a new knowledge over the traditional KBS or CBR systems.

In accordance with embodiments of this invention, the system receives the fault report from a monitoring system. The fault report comprises a node value pair for each error detected. When one or more error is detected, a fault signature sequence (FSS) is generated. A solution search program will perform solution searching and display a case record associated with the FSS to provide recommendations for trouble shooting and system restoration. The system may then prompt a user to input additional recommendation for the case record. The case record is then updated to the database.

In accordance with one of the embodiments of this invention, the FSS is generated by serializing a fault signature list (FSL). The FSL contains a list of node value pair for each error detected. The FSS is serialized according to a substation number, panel number, time stamp, or alphabetic order. One advantage is that the system ensures that the fault signature sequence (FSS) is always in a consistent manner so that a FSS is a unique diagnostic representation of the faulty system symptoms, even if the power network has been altered and the network nodes re-organized.

In accordance with one of the embodiments of the embodiments, the FSS is partitioned into sub-groups according to substation structure, panel structure, or circuit line structure based on time stamp. One advantage of partitioning is that the FSS carries rich information of the system state; the partitioned FSS into a number of sub-groups at sub-station or circuit route level allows a user to perform advanced diagnosing and solution searching.

In accordance with one of the embodiments of this invention, a case number is generated for each of the FSS sub-groups. If the sub-group is not available in the database, a case record will be assigned to the case number. The user will be prompted to input handling information to the case record. The case record will then be stored in the database for future references.

In accordance with some embodiments of this invention, the database is a multi-dimensional database for reliable and efficient storage and searching of information.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other advantages and features of a signature-based case library diagnosis system in accordance with this invention are described in the following detailed description and are shown in the following drawings:

Figure 1 illustrating a hierarchical structure of the case memory model known in the art;

Figure 2 illustrating a network structure of categories, cases, and index pointers known in the art;

Figure 3 illustrating a system that includes a diagnosis system in accordance with an embodiment of this invention;

Figure 4 illustrating a processing system that performs process for providing a diagnosis in accordance with an embodiment of this invention;

Figure 5 illustrating a functional block diagram of a software system for providing a diagnosis system in accordance with an embodiment of this invention;

Figure 6 illustrating a schematic representation of an electric power system;

Figure 7 illustrating a typical representation of substation and control panels;

Figure 8 illustrating a node value pair;

Figure 9 illustrating a relationship between the fault signature list and the fault signature sequence;

Figure 10 illustrating partitioning of fault signature sequence into sub-groups according to substation or panels definition;

Figure 11 illustrating partitioning of fault signature sequence into sub-groups according to circuit line.

Figure 12 illustrating the types of libraries;

Figure 13 illustrating the components of each case record;

Figure 14 illustrating a process flow of the diagnosis system in accordance with an embodiment of this invention;

Figure 15 illustrating a process flow of a case construction;

Figure 16 illustrating a process flow of searching case records; and

Figure 17 illustrating a conceptual database image for searching, storing, and retrieving of case record in accordance with an embodiment of this invention.

DETAILED DESCRIPTION

This invention relates to fault diagnosis and recovery system. More particularly, this invention relates to a computational method and system that employs fault signature-based system. Still more particularly, this invention relates to applications stored on the media and loaded into executable memory of a system to cause the system to perform fault signature-based system for diagnosing, trouble shooting for complex systems, and recommending recovery plans for system restoration.

Figure 3 illustrates a processing system 100 that may include a fault diagnosis and recovery system in accordance with this invention. In Figure 3, processing system 100 is shown as a convention desktop personal computer. However, for purposes of this invention processing system 100 may be a desktop personal computer, a laptop personal computer, a computer terminal, server, router, or any system having a processor, memory, and interfaces for communicating with a Monitored

System 150. One example of a Monitored System 150 is shown in Figure 6 and will be described below.

In the shown embodiment, processing system 100 includes display 105, keyboard 130, and mouse 140 that are connected to the system as described in Figure 4 to allow user interaction with processing system 100. Processing system 100 also includes optical disk drive 115 and magnetic disk drive 120. Optical disk is inserted into optical disk drive 115 to allow processing system 100 to read data from and write data to optical disk. Magnetic disk is inserted into magnetic disk drive 120 to allow processing system 100 to read data from and write data to magnetic disk.

Furthermore, processing system 100 includes Universal Serial Bus (USB) ports 125 that allow a USB compliant device to connect to and interface with processing system 100 via an UBS connector.

Although a transceiver is not shown, processing system 100 may include a RF or other type of signal transceiver that allows devices to connect to and interface with processing system 100 via signalling using a known protocol such as Bluetooth. Thus, in accordance with this invention, a portable media device is any device that connects to a processing system to allow data to be read from and written to a data storage media in a connected device. The only requirement being that the device has sufficient memory to store the instructions for a fault diagnosis and recovery system in accordance with this invention.

Figure 4 illustrates a block diagram of the processing components of processing system 200 that executes instructions to provide applications to provide a diagnostic system in accordance with an embodiment of this invention. Processing system 200 includes Central Processing Unit (CPU) 205. CPU 205 is a processor, microprocessor, or any combination of processors and microprocessors that execute instructions to perform the processes in accordance with the present invention. CPU 205 connects to memory bus 210 and Input/Output (I/O) bus 215. Memory bus 210 connects CPU 205 to memories 220 and 225 to transmit data and instructions between the memories and CPU 205. I/O bus 215 connects CPU 205

to peripheral devices to transmit data between CPU 205 and the peripheral devices. One skilled in the art will recognize that I/O bus 215 and memory bus 210 may be combined into one bus or subdivided into many other buses and the exact configuration is left to those skilled in the art.

A non-volatile memory 220, such as a Read Only Memory (ROM), is connected to memory bus 210. Non-volatile memory 220 stores instructions and data needed to operate various sub-systems of processing system 200 and to boot the system at start-up. One skilled in the art will recognize that any number of types of memory may be used to perform this function.

A volatile memory 225, such as Random Access Memory (RAM), is also connected to memory bus 210. Volatile memory 225 stores the instructions and data needed by CPU 205 to perform software instructions for processes such as the processes for providing a system in accordance with this invention. One skilled in the art will recognize that any number of types of memory may be used to provide volatile memory and the exact type used is left as a design choice to those skilled in the art.

I/O device 230, keyboard 235, display 240, memory 245, network device 250 and any number of other peripheral devices connect to I/O bus 215 to exchange data with CPU 205 for use in applications being executed by CPU 205. I/O device 230 is any device that transmits and/or receives data from CPU 205. Keyboard 235 is a specific type of I/O device that receives user input and transmits the input to CPU 205. Other examples of I/O devices include a mouse, Personal Digital Assistant (PDA) and other USB compliant devices. Display 240 receives display data from CPU 205 and display images on a screen for a user to view. Memory 245 is a device that transmits and receives data to and from CPU 205 for storing data to a media. One skilled in the art will recognize that more than one memory 245 may be connected to system 100. Examples of memory devices include optical disk drive 115 and magnetic disk drive 120 shown in Figure 1; and memory sticks and the like that connect to processing system 100 via USB Ports. Network device 250 connects CPU 205 to a network for transmission of data to and from other processing systems.

Figure 5 illustrates block diagram of a power system diagnosis system 300. One skilled in the art will recognize that this diagram includes physical components, process steps, and software components used to provide power system diagnosis system 300 in accordance with this invention as is presented for illustrative purposes for understanding system 300. Diagnosis system 300 comprises a fault message processor module 318, a library case builder module 322, a user interface 324, and a storage media 330 for storing the system diagnosis knowledge in case library 332, and the power system information 334. The storage media 330 has a processing unit for executing embedded programs such as the solution search programs 342.

Diagnosis system 300 may be integrated with a monitoring system 314. One typical monitoring system is a SCADA system. Monitoring system 314 monitors the power system 312. Both monitoring system 314 and power system 312 are drawn in dotted lines to show that these components are external systems that interact with system 300 in accordance with this embodiment of the invention. The monitoring system 314 sends fault report 316 to fault message processor 318. Fault Signature Sequences (FSS) 320 is output from fault message processor 318 and will be forwarded to library case builder 322. In this embodiment, fault message processor 318 and library case builder 322 are software components executed by a processing system to perform the described functions in accordance with this embodiment of the invention. For each FSS, library case builder 322 creates a case number. For each fresh case number, library case builder 322 constructs a case record in the case library 332. Power system information 334, such as substations and panels definition, circuit line definition, is also saved in the storage media 330.

User interface 324 includes diagnostic navigator 326 and knowledge I/O board 328. In accordance with this embodiment, user interface 324, diagnostic navigator 326 and knowledge I/O board are software processes executed by a processing system to provide the functions described. Upon receiving fault messages from the fault message processor 318, diagnostic navigator 326 displays a fault message by

blinking the faulty elements which are located on a simplified power network system diagram which is drawn by the system according to power system information 334, stored in the storage media 330. A user can click on any blinking element to activate the solution search program 342.

User interface 324 may be provided by devices electronically connected to the processing system providing diagnostic system 300 either through wired or wireless means to enable a user to input into the electronic device the information for storing or retrieving knowledge for system diagnostics and restoration purpose.

Once solution search program 342 is activated, solution search program 342 performs a solution searching processes and returns recommendations 340 in the form of a display or a print-out to help the user perform trouble shooting and system restoration. The recommended solutions can be confirmed by a user input using user interface 324 if the recommendations are implemented. The confirmation 344 of system restoration is then saved to the storage media 330 for consideration in subsequent searches.

In electrical engineering, line drawing diagrams are frequently used for power system analysis. However, for brevity and simplicity, a simplified schematic diagram is drawn to represent the network structure of a power system being monitored in accordance with the described embodiment of this invention, as shown in Figure 6.

Figure 6 illustrates an electric power system 400. Power system 400 includes power intake 402 or generator 406; a set of transformers 404 to lower the distribution voltage to the level used by the consumer equipment; the transmission and distribution lines 414; substations 408a, 408b...408n at which the power system 400 is further distributed from substations to the consumers normally at relatively lower voltage through networks of electric circuits.

Substations 408a to 408n generally have switching protection and control equipment as well as one or more transformers. In a large substation, circuits

breakers are used to interrupt any short-circuit or overload currents that may occur on the network. Smaller distribution stations may use re-closer circuit breakers or fuses for protection of distribution circuits. Substations 408a to 408n may, although not necessarily, include power generators. Other devices such as power factor correction capacitors and voltage regulators may also be located at a substation. Each of the circuit breakers and other devices located at a substation can be hosted in a panel. Figure 6 uses rectangles to represent the substations 408a to 408n and solid black squares to represent panels or breakers 412a, 412b and 412c in each substation.

Figure 7 illustrates an enlarged view of substation-j 408 as a more detailed illustration. The substation-j has two rows of panels 412; the first row includes two panels, 412d and 412e; the second row includes six panels, 412f to 412k.

In one embodiment of the invention, the power grid elements, such as breakers or panels, are also named nodes. A generic form of a substation, denoted as Substation^s (s = 1 to N, N is the number of substations in the power system), consisting of a plurality of nodes, can be represented as follows.

$$\text{Substation}^s = \{\text{Node}_1, \text{Node}_2, \text{Node}_3, \dots, \text{Node}_k, \dots, \text{Node}_{N_{ds}-1}, \text{Node}_{N_{ds}}\} \quad (1)$$

It is noted that in the above expression, N_{ds} is the number of nodes located in the sth substation. The definition of a plurality of substations and panels, as part of the power system information 334, is stored in the system 300 as shown in Figure 5.

Referring to Figure 6, any electric-distribution system, such as the power system 400, involves a large amount of supplementary equipment to protect generators 406, transformers 404 and transmission lines 414. The system 400 often includes devices designed to regulate the voltage or other characteristics of power delivered to consumers. For example, to protect all elements of a power system from short circuits and overloads, and for normal switching operations, circuit breakers 412a, 412b and 412c are employed. These breakers can be large switches that are activated automatically in the event of a short circuit or other condition that

produces a sudden rise of current. There is also various electrical current control devices used in the power system. A typical example is network switch 418, which provides an alternative component for controlling electricity current.

As shown in Figure 6, some panels are outgoing panels which provide the electric power connections to other substations, for example, the panel 412a provides electric power to the substation 408b by an incoming panel 412b. In this case, a circuit line or route is formed. A generic form of a circuit line, denoted as Line^c ($c = 1$ to N_L , where N_L is the number of circuit lines defined in the power system), including a plurality of the power grid nodes, can be represented as follows.

$$\text{Line}^c = \{\text{Node}_1, \text{Node}_2, \text{Node}_3, \dots, \text{Node}_k, \dots, \text{Node}_{N_{dc}-1}, \text{Node}_{N_{dc}}\} \quad (2)$$

In Expression (2), N_{dc} is the number of nodes which construct the circuit line, Line^c , each node in the circuit line is denoted as Node_k , where $k = 1$ to N_{dc} . Those skilled in the art will recognize that a node can be any device used in the power system for power distribution and control purpose within the circuit line construction.

Figure 6 also shows a typical circuit line, which is marked using a dotted line 410a. Circuit line 410a starts at main intake 402, and continues through the transformer 404; substation 408a (panel 412a), distribution line 414, substation 408b (panel 412b and 412c) and finally to the transformer 416a. Similarly, other circuit lines can be defined accordingly, such as circuit line 410b. Those skilled in the art will recognize that the definition of a plurality of circuit lines, as part of the power system information 334, is stored in the system 300 as shown in Figure 5.

Each of nodes in the power grid has an identification number designated as node ID. A node ID may contain substation information as well, for example, a node ID can be S02-03, where S02 is substation ID while 03 is panel ID. Following is generic formula for constructing a node ID.

$$\text{Node ID} = [\text{Substation ID}]-[\text{Node ID}] \quad (3)$$

When a power system is monitored by a monitoring system, the state of the power system can be described using a list of the monitored nodes with their status values. The monitored nodes can be the panels or any other devices shown in Figure 6. The node and status value of the node construct a node value pair, denoted by NVP_k , meaning that the node-value pair for node_k. In general, NVP_k can be expressed as follows.

$$NVP_k = [Node_k \text{ ID}]:[Value^k_s] \quad (4)$$

Now referring to Figure 8, node value pair 600 includes of a Node_k 682 and a Value_s^k 684. The Value_s^k 684 represents the status value of Node_k 682. Those skilled in the art will recognize that the Value_s^k 884 may further include two elements: 1) fault type 686 and 2) time t_k 688, where, t_k is the time at which the specific fault, denoted pF_k (k can be any index value from 1 to N_f), was triggered. In general, the node Value^k can be expressed as follows.

$$Value^k = (pF_k @ t_k) \quad (5)$$

In the above expression, the symbol @ may read "at" which can also be used as a delimiter to separate data elements pF_k and t_k .

The node status value, pF_k , namely Fault Type_k 686, can be any value selected from a range of possible fault types: pF_1 690 pF_2 692, pF_{N_f-1} 694 and pF_{N_f} 696; where N_f is the number of possible fault types which may be returned by the monitoring system.

Following is a typical example of the possible fault types defined in a SCADA system.

Type of Alarm	Description
OCTrip	Over Current Trip Status
EFTrip	Earth Fault Trip Status

LowOil	Low Oil Status
HiPress	High Pressure Status
HiTemp	High Temperature Status
LowOilAlm	Low Oil Level Alarm
HiTempAlm	High Temperature Alarm
BatAlarm	Battery Alarm
BatChrTrip	Battery Charger Trip
PLTrip	Pilot Trip Status
DiffTrip	Diff Trip Status
SBEFTrip	SBEF Trip Status
TFFault	Transformer Fault Alarm
TFDoorLimit	Transformer Door Limit Switch
SEFTrip	SEF
InterTrip	Inter Trip

According to Expressions (4) and (5), if Over Current Trip Status –“OCTrip” is triggered to a node with ID “S02-03” at time 6/4/2008 11:37:45 AM, a sample node-value pair, denoted as NVP, is expressed as follows.

$$\text{NVP} = \text{S02-03:OCTrip@6/4/2008 11:37:45 AM} \quad (6)$$

A fault report generated by the SCADA system can be presented in a text file. The report file often includes of a plurality of fault description lines. Each of the lines in the report describes a fault or alarm tag, alarm description, and date/time at which the alarm tag is triggered. In general, the alarm tag includes the following information.

[Substaion ID]-[Panel ID]_(Type of Alarm)@(Record Time)

A typical example of the alarm tag is shown as follows.

S02-03_OCTrip@6/4/2008 11:37:45 AM

The above alarm tag states that the Panel with Id “S02-03”, that is, the panel numbered 03 located at Substation with substation ID S02, has a reported alarm “OCTrip”, the alarm tag was triggered at 6/4/2008 11:37:45 AM.

An exemplary complete fault description line can be as follows.

S02-03_OCTrip, Substation S02 OC Trip at P-03, @6/4/2008 11:37:45 AM

In the above alarm description line, coma is delimiter; “S02-03_OCTrip” is alarm tag; “Substation S02 OC Trip at P-03” is alarm description, @ introduces the record time “6/4/2008 11:37:45 AM”. It should be noted that the alarm tag, alarm description and alarm triggering date/time formulation may vary with different SCADA systems which may have different reporting format and fault definitions.

Those skilled in the art will recognize that in a report, one fault may be reported more than one times although the same fault may be reported at different times. These duplicated fault description lines must be filtered out by the fault message processor 318 of the system 300 shown in Figure 5. In some embodiments fault message processor 318 may ignore these alarm description lines which only contain alarm information and do not generate any power system failure. The fault message processor 318 may ignore these alarm messages according to the alarm tag selection policy which is predefined and stored in the system 300 as system parameters 336.

The sample node value pair NVP illustrated in expression (6) represents the node faulty symptom for the node with ID “S02-03”. The sample node value pair NVP is also considered as fault signature of the node. One skilled in the art will note that more than one faulty node may be reported in a particular instance when the monitored power system is large and complex. In this case, these faulty nodes form a fault signature list 302 shown in Figure 9.

Figure 9 illustrates fault signature list 302 consists of a list of node value pairs: NVP₁ 302a, NVP₂ 302b, NVP₃ 302c. . . NVP_k 302d. . . NVP_{Nr-1} 302e, NVP_{Nr} 302f,

where the index $k = 1$ to N_r , N_r is the number of faulty nodes reported by monitoring system in a single report. The fault signature list 302, denoted as FSL, can be expressed as follows:

$$FSL = \{NVP_1, NVP_2, NVP_3, \dots NVP_k, \dots NVP_{N_r-1}, NVP_{N_r}\} \quad (7)$$

The presentation order or sequence of the node value pairs in the fault signature list 302 can vary significantly because the faulty nodes are triggered at different time and the faulty nodes in a report are arranged by the monitoring system in a random manner. In accordance with the present invention, the inconsistency in sequence of the node value pairs in the fault signature list 302 is eliminated in order to provide a consistent and robust fault diagnosis system. As shown in Figure 9, the consistency of the node value pair's sequence is achieved by a serialization process 306 in which the sequence of the node value pairs in a fault signature list is reorganized or re-ordered using a predefined serialization method.

One of the serialization methods is a time based serialization method. For the faulty nodes which are connected serially, their node value pairs are serialized or re-ordered according to the time stamp t_k 688 as shown in Figure 8. In this method, the faulty nodes are sorted according to their time stamp, date/time t_k 688 in ascending order. The serially connected nodes are the nodes which are in a circuit line 410a as shown in Figure 6. For a serial structure, the time stamp t_k 688 is important because failure of any node may have significant impact on other nodes in the circuit line.

Another serialization method is based on a predefined power system network structure. This method arranges the position of the node value pairs in the fault signature sequence list according to their position in the power system network 400 shown in Figure 6. This method is applicable to power grid nodes that are interconnected in a parallel manner. The method may also be applicable to all the nodes in the power system. For example, the nodes located in Substation_j 408a shown in Figure 6 are connected to the same busbar as the Substation_j 408b. In this case, the time stamps t_k 688 that is associated with the faulty nodes in the

report can be ignored as they do not have significant influences. Therefore, for nodes that are interconnected between Substations, their node value pairs are arranged according to their physical arrangement in the substation. For example, 412d, 412e, and 412f, 412g, 412h, 412i, 412j, 412k shown in Figure 7.

In yet another serialization method, an alphabetical order based serialization is provided. In alphabetical order based serialization, the other node value pairs in the fault signature list are ordered according to their alphabet sequence.

Figure 9 shows an example of the result of the serialization process, the fault signature list FSL 302 is changed into the Fault Signature Sequence 308, denoted as FSS_{Σ} , which can be expressed as follows.

$$FSS_{\Sigma} = \{NVP_3, NVP_1, NVP_2, \dots NVP_k, \dots NVP_{Nr}, NVP_{Nr-1}\} \quad (8)$$

Note that only the node value pairs sequence has been changed according to Expression (8). The subscribed indexes of all the node value pairs remain the same as in Expression (7).

Figure 10 shows fault signature sequence FSS_{Σ} 800 in a unique description of a state of a fault system. Typically, FSS_{Σ} 800 contains a large amount of information about the power system. Therefore, it is necessary to partition the fault signature sequence FSS_{Σ} 800 into useful sub-groups 802 through a partitioning process 806.

One partitioning process is to group the node value pairs according to the configuration of the substation as previously defined above with Figure 6. For example, three node value pairs NVP^{S1}_1 808, NVP^{S1}_2 810, and NVP^{S1}_3 812 are selected from FSS_{Σ} 800 to form a sub-group FSS_{S1} 814. These three nodes are grouped because the three nodes are located in the substation indexed S1. A similar partitioning method may be used to select respective elements from FSS_{Σ} 800 to construct other sub-groups as shown in Figure 10, i.e. FSS_{S2} 816, FSS_{S3} 818, and FSS_{Sn} 820. The partitioning process according to the configuration of the

substation is depicted as FSS_{Sk} ($k = 1$ to n), where n is the number of faulty substations reported in a single report generated by monitoring system and is shown as follows:

$$FSS_{\Sigma} = \{FSS_{S1}, FSS_{S2}, FSS_{S3}, \dots, FSS_{Sn}\} \quad (9)$$

Each of the sub-groups may include one or more node value pairs. This sub-group partitioning may be used for construction and retrieval of trouble shooting and recovery knowledge at substation level, which will be explained in detail later.

As previously described, a list of nodes or panels may often be interconnected to each other to form a network or to construct a circuit line as shown in Figure 6. Thus, it is useful to identify these faulty nodes that are in a circuit line structure.

With reference to Figure 11, the same fault signature sequence FSS_{Σ} 800 is partitioned into sub-groups FSS_{C1} 916, ..., FSS_{Cn} 918 according to a circuit line structure. The node value pairs NVP^{C1}_1 906, NVP^{C1}_2 908, NVP^{C1}_3 910, NVP^{C1}_4 912 and NVP^{C1}_5 914 are selected from FSS_{Σ} 800 to form a sub-group FSS_{C1} 916. These nodes and their faulty type values are grouped because they are located within the same circuit line ($C1$). Similar partitioning method is applied to select respective elements from FSS_{Σ} 800 to construct other sub-groups shown in Figure 11, i.e. FSS_{Cn} 918. After partitioning, relationship between FSS_{Σ} and circuit line based fault signature sequence sub-groups is depicted by FSS_{Ck} ($k = 1$ to n) and is shown as follow:

$$FSS_{\Sigma} = \{FSS_{C1}, FSS_{C2}, \dots, FSS_{Cn}\} \quad (10)$$

The fault signature sequence sub-groups FSS_{Ck} ($k = 1$ to n), where n is the number of circuit lines reported in the fault message report. Each of the sub-groups may consist of one or more node value pairs. This sub-group will be used for construction and retrieval of trouble shooting and recovery knowledge at substation level, which will be explained in detail later.

It is important to note that the serialization process and partitioning process may be performed at different stages. For example, the initial fault signature list FSL may be serialized into a fault signature sequence FSS_{Σ} according to a structure based or alphabet order based serialization. Upon which, the fault signature sequence FSS_{Σ} is partitioned according to a circuit line structure. Subsequently, a time based serialization method is applied to the sub-groups obtained after partitioning according to circuit line structure. Further, one skilled will understand that different arrangement in the serialization and partitioning is possible. However, once the serialization and partitioning policy is determined; the policy should not be changed during the whole life time of the application.

As described previously, after partitioning according to circuit line structure, the time based serialization may be applied to the fault signature sequence sub-groups FSS_{Ck} ($k = 1$ to n). This serialization process will sort or re-arrange the node value pairs in each of the sub-groups FSS_{Ck} ($k = 1$ to n) according to their time stamp, date/time t_k 688 in ascending order.

Those skilled in the art will recognize that after serialization and partitioning, the time stamp t_k can be removed from all the fault signature sequence sub-groups, FSS_{Ck} and FSS_{Sk} for brevity purpose. Therefore, the node value pair "S02-03:OCTrip@6/4/2008 11:37:45 AM" shown in Expression (6) may be changed to "S02-03:OCTrip".

Three types of fault signature sequences are shown in Table 1. The first type is the original fault signature sequence without partition, FSS_{Σ} . The second type is the fault signature sequence sub-groups FSS_{Sk} ($k=1$ to n) which are obtained by partitioning FSS_{Σ} according to substations. The third type is the fault signature sequence sub-groups FSS_{Ck} ($k=1$ to n) which are obtained by partitioning FSS_{Σ} according to circuit line definition.

Table 1 Type of Fault Signature Sequence

S/N	Notation	Description
1	FSS_{Σ}	system fault signature sequence
2	FSS_{Sk} (k=1 to n)	substation node fault signature sequence
3	FSS_{Ck} (k=1 to n)	circuit line fault signature sequence

Each type of the fault signature sequences is associated with a specific type of diagnostics knowledge. For instance, FSS_{Σ} is associated with the knowledge at system level, such as generic system diagnostics and recovery guide lines; FSS_{Sk} is associated to the trouble shooting and restoration procedures at substation level; and FSS_{Ck} is associated with the diagnostics knowledge at circuit line level, and the steps and sequence of re-setting breakers and relays in order to restore the circuit line functions.

Referring to Figure 12, the case library system 1002 accordingly contains three case libraries: case library (1) 1004 stores the generic system diagnostics knowledge and system recovery guidelines, case library (2) 1006 stores the substation trouble shooting and restoration procedures; and case library (3) 1008 contains the circuit line trouble shooting and breaker resetting sequences.

The system utilizes objects to model and store the relevant entities, data elements and knowledge in the case library system. Now there is provided detailed description of the fault report, library case ID and library case record objects.

With reference to Figure 13, the fault report object 1112, represents a typical memory model of the fault report 316 generated by monitoring system 314 as shown in Figure 5. The fault report object 1112 may include information such as fault report ID, fault signature list (FSL), as well as recording time, type and staff-in-charge. The recording time is the time when the report is generated. After serialization, the resultant fault signature sequence FSS_{Σ} , which normally contains more than one fault signature elements, will be forwarded to Case Library (1) 1004 for case ID construction and case knowledge recording operations. The signature sequence sub-groups FSS_{Sk} (k=1 to n) obtained by the substation based partition

will be forwarded to Case Library (2) 1006 and the fault signature sequence sub-groups FSS_{ck} ($k=1$ to n) obtained by circuit line based partition will be forwarded to Library (3) 1008.

The case ID object 1114 contains a case ID and the fault signature sequence (FSS). A case ID is created automatically by the system according to the fault signature sequence (FSS). It is noted that for each FSS, there is only one unique case ID created and stored in the system. In fact, the fault signature sequence can be directly treated as case ID although a case ID can be generated according to a predefined protocol.

Case record object 1116 contains basically three types of information: (1) identification information, (2) trouble shooting and restoration knowledge and (3) case recording information. One skilled in the art will recognize that for one library case ID, there can be one or more case records in the case library.

For brevity, the knowledge contents in the case record object 1116 in Figure 13 only describe the attributes for Case Library (2) 1006, i.e. the substation trouble shooting and restoration procedures. One skilled in the art will recognize that other attributes may also be stored in each case record. Further, one skilled in the art will recognize that the above is only provided as an illustration and the exact configuration and information required is left to the person skilled in the art.

The case identification information contains a composite key including a record number and a case ID. It is noted that one case record has only one record number, but different records may share a same case ID.

The knowledge contents may vary with the case library types. The case knowledge content shown in Figure 13 may include the fault description, isolation and recovery procedures at substation level. Those skilled in the art will recognize that the knowledge contents are not generated by the system, but captured from human experts and stored in the system for future use. Following is an example of isolation and recovery procedures.

*Isolation Steps:**Isolation of Breaker S02-01 to 10**Isolation of S02-02 & S07-01**Isolation of S02-03 & S11-01**Isolation of S02-05 & S12-01**Isolation of S02-06 & incoming Circuit**Recovery Procedure:**Ensure Generator switch to "OFF"**Verify Insulation Resistance of Switchboard Busbar**Recovery substation S07**Recovery substation S11**Recovery substation S12*

In the above isolation and recovery procedures, the breaker nodes with ID S02-01 to S02-10 are located at substation S02. S07-01, S11-01, S12-01 are breakers nodes which connected to S02-02, S02-03, S02-05 respectively. The substations with ID S07, S11 and S12 receive electric power supply from the substation S02.

The knowledge contents for circuit line recovery include the detailed step-by-step procedures to reset breakers and O/C relays for recovery.

The case recording information may also include other miscellaneous information such as, but not limited to, date and time of records; and an identification of the staff that saved the case record and the number of implementations of the case solution.

Figure 14 provides a flowchart of the method 1200 of fault message processing used by the Fault Message Processor 318 in accordance with one embodiment of the present invention. In particular, method 1200 is described using fault signature sequences as single-indexing key elements to store diagnostic knowledge into and retrieve diagnostic knowledge from the case library system. Method 1200 begins in

step 1202. The system receives fault report 1204 that is generated by the monitoring system 314. The system then proceeds to step 1206 and generates a fault message included in the fault report, into fault signature list (FSL). Subsequently, the fault signature list (FSL) is serialized to form fault signature sequence (FSS_{Σ}) in step 1208.

The fault signature sequence (FSS_{Σ}) is partitioned into sub-groups FSS_{sk} ($k = 1$ to n) in step 1210 according to the substation structure, panel structure or another parameter. FSS_{Σ} may then be further partitioned into subgroup FSS_{ck} ($k = 1$ to n) in step 1212 according to circuit line definition 334 stored in the memory storage 330 of the system 300.

There are two outputs from the Fault Message Processor 318 as shown in Figure 5. Firstly, fault signature sequence sub-groups are sent to visual diagnostic navigator 326 for information display. In step 1214, the fault message is displayed on the visual diagnostic navigator 326. In one embodiment, the fault message may be represented on a display by blinking the objects representing the faulty elements, such as faulty substations, panels, or faulty circuit lines. At the same time, these faulty elements or objects may be used by the program to allow the user to select fault message handlers in order for the solution search program 342 to perform solution search and recommend trouble shooting steps and recovery procedures to the user.

The fault signature sub-groups are also forwarded to the case Library Case Builder 322 for case ID creation and case recording case construction in Step 1216. For each of the sub-groups, a new case ID is created if the sub-group does not exist in the system. The new case ID is saved to case library. At the same time, a template case record may be created which is associated with the case ID. Those skilled in the art will recognize that the corresponding case solution can be captured and saved into the case library at a later time. Detailed method for case construction is illustrated in Figure 15. Method 1200 then ends with 1218.

Figure 15 is a flowchart illustrating the details of the case construction 1300 performed in step 1216 of method 1200. Case construction 1300 begins with 1302. Case construction 1300 receiving fault signature sequences FSS_{Sk} ($k = 1$ to n) 1304. Those skilled in the art will recognize that the case construction 1300 illustrates the details for constructing a library case for substation and panel trouble shooting and recovery. However, similar methods can also be used to build library cases for a generic diagnostic library and circuit line trouble shooting and recovery libraries.

In step 1306, the case construction 1300 initializes a counter k to 1, and then proceeds to step 1308 to create a case ID for every fault signature sequence. The newly created case ID (ID_{Sk}) 1310 for the fault signature sequence FSS_{Sk} is then forwarded for further processing.

In step 1312, case construction 1300 searches the case library to determine whether the case ID (ID_{Sk}) exists. If case ID (ID_{Sk}) does not exist in the case library, the method proceeds to step 1314 and creates a case record for the new case with a new record number, CsN_{Sk} . The new case number is associated with said case ID (ID_{Sk}). Those skilled in the art will recognize that the case record number CsN_{Sk} and case ID (ID_{Sk}) formulate a composite key of the now case record in the case library. After step 1314, the case construction cycle counter k is incremented by one at step 1316.

If case ID (ID_{Sk}) exists in the case library, case construction 1300 proceeds to step 1316.

In checking step 1318, the case construction cycle repeats if the case construction counter k is less than or equal to m , where m is the number of fault signature sub-groups. If the cycle is to be repeated, case construction 1300 returns to step 1308. Otherwise, case construction 1300 ends at 1320.

With reference to Figure 16, a general flowchart illustrates a method 1400 for providing solution search program 342 by using the fault diagnosis system 300 in

accordance with one embodiment of the present invention. Method 1400 begins in step 1402. The search program captures the click event, if any, by the user, which occurs on one of the blinking faulty objects in the visual diagnostic navigator 326 at step 1404.

When capturing a click event, the program will read all the information embedded in the object, such as object identification information and case ID (denoted as ID_{blinking}) 1406.

A case record may not contain a solution. For instance, only a template case record is stored in the library if no solution has been captured from the electrical engineers.

In step 1408, the search program 342 retrieves a case record available in the case library. If the case record is available, the search program will retrieve all the solutions that match the case ID (ID_{blinking}) and display retrieved solutions in a predefined order in step 1410. Those skilled in the art will recognize that the display order is determined by pre-defined system initialization parameters 336. When the user implements the recommended recovery procedure, the user may or may not confirm the implementation. If implemented, the program updates the case implementation information at step 1418 for future use. Otherwise, the search program prompts the user whether to continue or quit at step 1420. The search program goes back to step 1404 if user inputs a request to continue. Otherwise, the search program ends at step 1422.

When there is no available case record in the case library, the search program will search a relevant case record for the user in step 1412. The user may then edit the relevant case record and the updated case record will be saved to the case library for future use in step 1414.

As illustrated in Figure 17, all the case records stored in the case library are constructed into a solution space, A , 622 in an n -dimensional space, Ω , 620. Each of the case is denoted as case_i and marked using letter x 624.

Unlike complex network structured memory models in the prior art, the present invention utilizes a simple memory structure, namely single-indexing structure to store diagnostic knowledge in the case library. The main advantage of the single-indexing structure is that the case construction is that the structure is simple, and a template case can be automatically constructed and added into the case library. The single-indexing structure system is robust and does not generate conflict outcome for a given case ID. The reliability of the system is high because its brevity and simplicity in memory structure. In addition, the software development cost and system maintenance cost is significantly reduced.

The strategy used in the present invention reduces the time needed for the solution search as follows. The solution space set A 622 includes N_{SS} sub sets (ψ_k , $k=1$ to N_{SS}), where N_{SS} is the number of subsets in solution space set A 622, which is shown as follows:

$$A = \{\psi_1, \psi_2, \psi_3, \dots, \psi_{N_{SS}}\} \quad (11)$$

The case library disclosed in the present invention includes three subsets. In Expression (11), ψ_1 is the subset that includes the cases associated with generic diagnostic knowledge and system recovery guidelines (i.e. Case Library (1) 1004 in Figure 12); ψ_2 is the subset that includes the cases associated with substation trouble shooting and restoration procedures (i.e. Case Library (2) 1006 in Figure 12); ψ_3 is the subset that includes the cases associated with circuit line trouble shooting and breaker/relay resetting sequences (i.e. Case Library (3) 1008 in Figure 12).

When searching for a solution case, for a given fault signature sequence, FSS_{Σ} , sub-groups FSS_{Sk} or sub-groups FSS_{Ck} sequence, the search program 342 only checks the subset which the fault signature sequence points. For example, the subset 626 is such a subset, which contains the solution case 630. As the subset

ψ 626 is much smaller than the solution space set A 622, the search time required to reach the solution set 630 is significantly reduced.

As discussed previously, there may be more than one record available in the solution set 628. In this case the search program 342 will list all the solution records in order. The solution display order is determined according to the system setting parameters 336 pre-defined by the user. For example, the solution records may be displayed according to their implementation records, or their record date/time, or combination of both. On the solution list, the first case record is considered the best solution 630.

It will be appreciated that one skilled in the art will understand that numerous variations and/or modifications may be made to the invention as shown in the above embodiments without departing from the spirit or scope of the invention as described. The present embodiments as described above are therefore to be considered in all respects as illustrative and not restrictive.

Claims

1. A method of fault diagnosis for a power grid system comprising:
 - receiving a fault report from a monitoring system comprising a plurality of errors wherein each of said plurality of errors comprises a node value pair;
 - generating a target fault signature sequence that is an ordered listing of said plurality of errors;
 - comparing said target fault signature sequence with a plurality of signature fault sequences stored in a database;
 - determining one of said plurality of fault signature sequences that matches said target signature fault sequence;
 - retrieving a case record associated to said one of said plurality of fault signature sequences that matches said target signature sequence; and
 - displaying said case record.
2. The method of claim 1 wherein said step of generating said fault signature sequence comprises:
 - generating a fault signature list comprising said plurality of errors from said fault report; and
 - serialising said fault signature list to form a target fault signature sequence wherein said plurality of errors are in a predefined sequence based upon said node value pair of each of said plurality of errors in said target fault signature sequence.
3. The method according to claim 1 wherein said node value pair comprises a substation number and said step of generating said target fault signature sequence comprises:
 - ordering said plurality of errors in accordance with said substation number in said node value pair of each of said plurality of errors.

4. The method of claim 1 wherein said node value pair comprises a panel number and said step of generating said fault signature sequence comprises:
ordering said plurality of errors in accordance with said panel number in said node value pair of each of said plurality of errors.
5. The method according to claim 4 wherein said ordering of said plurality of errors is according to a predefined circuit line comprising a list of panel numbers.
6. The method of claim 1 wherein said node value pair comprises a status value and said step of generating said target fault signature sequence comprises:
ordering said plurality of errors in accordance with said status value in said node value pair of each of said plurality of errors.
7. The method of claim 1 wherein said node value pair comprises a time stamp and said step of generating said target fault signature sequence comprises:
ordering said plurality of errors in accordance with said time stamp in said node value pair of each of said plurality of errors.
8. The method of claim 1 wherein said node value pair comprises an alphabetic code and said step of generating said target fault signature sequence comprises:
ordering said plurality of errors in accordance with said alphabetic code in said node value pair in each of said plurality of errors.
9. The method according to claim 1 further comprising:
partitioning said target fault signature sequence into a plurality of sub-groups of said plurality of errors.
10. The method according to claim 9 wherein said step of partitioning comprises:
forming each of said plurality of sub-groups according to a substation number in said node value pair of each of said plurality of errors.

11. The method according to claim 9 wherein said step of partitioning comprises:
forming each of said plurality of sub-groups according to a panel number in said node value pair of each of said plurality of errors.
12. The method according to claim 9 wherein said step of partitioning comprises:
forming each of said plurality of sub-groups according to a time stamp in said node value pair of each of said plurality of errors.
13. The method according to claim 9 further comprising:
generating a case number for each of said plurality of sub-groups;
determining one of said plurality of sub-groups is not recorded in said database based on said case number;
assigning a case record to said one of said plurality of sub-groups;
receiving inputs from a user regarding handling information for said case record; and
storing said inputs to said case record in said database wherein said case record includes said one of said plurality of sub-groups and said handling information.
14. The method according to claim 1 further comprises:
receiving inputs from a user including additional handling information for said case record; and
updating said case record to include said additional handling information in said database.
15. The method according to claim 1, wherein said database is a multi-dimensional database.
16. The method according to claim 9 wherein said node value pair comprises a substation number and said step of generating said target fault signature sequence comprises:

serializing said plurality of errors in accordance with said substation number in said node value pair of each of said plurality of errors.

17. The method of claim 9 wherein said node value pair comprises a panel number and said step of generating said fault signature sequence comprises:
serializing said plurality of errors in accordance with said panel number in said node value pair of each of said plurality of errors.

18. The method of claim 9 wherein said node value pair comprises a status value and said step of generating said target fault signature sequence comprises:
serializing said plurality of errors in accordance with said status value in said node value pair of each of said plurality of errors.

19. The method of claim 9 wherein said node value pair comprises a time stamp and said step of generating said target fault signature sequence comprises:
serializing said plurality of errors in accordance with said time stamp in said node value pair of each of said plurality of errors.

20. The method of claim 9 wherein said node value pair comprises an alphabetic code and said step of generating said target fault signature sequence comprises:
serializing said plurality of errors in accordance with said alphabetic code in said node value pair in each of said plurality of errors.

21. A system for fault diagnosis in a power grid system being executed in a software system comprising:
circuitry configured to receive a fault report from a monitoring system comprising a plurality of errors wherein each of said plurality of errors comprises a node value pair;
circuitry configured to generate a target fault signature sequence that is an ordered listing of said plurality of errors;

circuitry configured to compare said target fault signature sequence with a plurality of signature fault sequences stored in a database;

circuitry configured to determine one of said plurality of fault signature sequences that matches said target signature fault sequence;

circuitry configured to retrieve a case record associated to said one of said plurality of fault signature sequences that matches said target signature sequence; and

circuitry configured to display said case record.

22. The system according to claim 21 wherein said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to generate a fault signature list comprising said plurality of errors from said fault report; and

circuitry configured to serialise said fault signature list to form a target fault signature sequence wherein said plurality of errors are in a predefined sequence based upon said node value pair of each of said plurality of errors in said target fault signature sequence.

23. The system according to claim 21 wherein said node value pair comprises a substation number and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to order said plurality of errors in accordance with said substation number in said node value pair of each of said plurality of errors.

24. The system according to claim 21 wherein said node value pair comprises a panel number and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to order said plurality of errors in accordance with said panel number in said node value pair of each of said plurality of errors.

25. The system according to claim 21 wherein said node value pair comprises a status value and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to order said plurality of errors in accordance with said status value in said node value pair of each of said plurality of errors.

26. The system according to claim 21 wherein said node value pair comprises a time stamp and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to order said plurality of errors in accordance with said time stamp in said node value pair of each of said plurality of errors.

27. The system according to claim 21 wherein said node value pair comprises an alphabetic code and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to order said plurality of errors in accordance with said alphabetic code in said node value pair in each of said plurality of errors.

28. The system according to claim 21 further comprising:

circuitry configured to partition said target fault signature sequence into a plurality of sub-groups of said plurality of errors.

29. The system according to claim 28 wherein said circuitry configured to partition comprises:

circuitry configured to form each of said plurality of sub-groups according to a substation number in said node value pair of each of said plurality of errors.

30. The system according to claim 28 wherein said circuitry configured to partition comprises:

circuitry configured to form each of said plurality of sub-groups according to a panel number in said node value pair of each of said plurality of errors.

31. A system according to claim 28 where said circuitry configured to partition comprises:

circuitry configured to form each of said plurality of sub-groups according to a time stamp in said node value pair of each of said plurality of errors.

32. A system according to claim 28 further comprising:

circuitry configured to generate a case number for each of said plurality of sub-groups;

circuitry configured to determine one of said plurality of sub-groups is not recorded in said database based on said case number;

circuitry configured to assign a case record to said one of said plurality of sub-groups;

circuitry configured to receive inputs from a user regarding handling information for said case record; and

circuitry configured to store said inputs to said case record in said database wherein said case record includes said one of said plurality of sub-groups and said handling information.

33. A system according to claim 21 further comprises:

circuitry configured to receive inputs from a user including additional handling information for said case record; and

circuitry configured to update said case record to include said additional handling information in said database.

34. A system according to claim 21, wherein said database is a multi-dimensional database.

35. The system according to claim 28 wherein said node value pair comprises a substation number and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to serialize said plurality of errors in accordance with said substation number in said node value pair of each of said plurality of errors.

36. The system according to claim 28 wherein said node value pair comprises a panel number and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to serialize said plurality of errors in accordance with said panel number in said node value pair of each of said plurality of errors.

37. The system according to claim 28 wherein said node value pair comprises a status value and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to serialize said plurality of errors in accordance with said status value in said node value pair of each of said plurality of errors.

38. The system according to claim 28 wherein said node value pair comprises a time stamp and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to serialize said plurality of errors in accordance with said time stamp in said node value pair of each of said plurality of errors.

39. The system according to claim 28 wherein said node value pair comprises an alphabetic code and said circuitry configured to generate said target fault signature sequence comprises:

circuitry configured to serialize said plurality of errors in accordance with said alphabetic code in said node value pair in each of said plurality of errors.

40. A fault diagnosis system in a power grid system connectable to a processing system comprising:

a memory is said fault diagnosis system; and
instructions stored in said memory for directing a processing unit to:
 receive a fault report from a monitoring system comprising a plurality of errors wherein each of said plurality of errors comprises a node value pair;
 generate a target fault signature sequence that is an ordered listing of said plurality of errors;
 compare said target fault signature sequence with a plurality of signature fault sequences stored in a database;
 determine one of said plurality of fault signature sequences that matches said target signature fault sequence;
 retrieve a case record associated to said one of said plurality of fault signature sequences that matches said target signature sequence; and
 display said case record.

41. The system of claim 40 wherein said instructions to generate said target fault signature sequence further comprises:

 instructions stored in said memory for directing said processing unit to:
 generate a fault signature list comprising said plurality of errors from said fault report; and
 serialise said fault signature list to form a target fault signature sequence wherein said plurality of errors are in a predefined sequence based upon said node value pair of each of said plurality of errors in said target fault signature sequence.

42. The system according to claim 40 wherein said node value pair comprises a substation number and said instructions to generate said target fault signature sequence further comprises:

 instructions stored in said memory for directing said processing unit to:
 order said plurality of errors in accordance with said substation number in said node value pair of each of said plurality of errors.

43. The system according to claim 40 wherein said node value pair comprises a panel number and said instructions to generate said target fault signature sequence further comprises:

instructions stored in said memory for directing said processing unit to:
order said plurality of errors in accordance with said panel number in said node value pair of each of said plurality of errors.

44. The system according to claim 40 wherein said node value pair comprises a status value and said instructions to generate said target fault signature sequence further comprises:

instructions stored in said memory for directing said processing unit to:
order said plurality of errors in accordance with said status value in said node value pair of each of said plurality of errors.

45. The system according to claim 40 wherein said node value pair comprises a time stamp and said instructions to generate said target fault signature sequence further comprises:

instructions stored in said memory for directing said processing unit to:
order said plurality of errors in accordance with said time stamp in said node value pair of each of said plurality of errors.

46. The system according to claim 40 wherein said node value pair comprises an alphabetic code and said instructions to generate said target fault signature sequence further comprises:

instructions stored in said memory for directing said processing unit to:
order said plurality of errors in accordance with said alphabetic code in said node value pair in each of said plurality of errors.

47. The system according to claim 40 further comprising:

instructions stored in said memory for directing said processing unit to:
partition said target fault signature sequence into a plurality of sub-groups of said plurality of errors.

48. The system according to claim 47 wherein said instructions to partition comprises:

instructions stored in said memory for directing said processing unit to:
form each of said plurality of sub-groups according to a
substation number in said node value pair of each of said plurality of
errors.

49. The system according to claim 47 wherein said instructions to partition comprises:

instructions stored in said memory for directing said processing unit to:
form each of said plurality of sub-groups according to a panel
number in said node value pair of each of said plurality of errors.

50. The system according to claim 47 wherein said instructions to partition comprises:

instructions stored in said memory for directing said processing unit to:
form each of said plurality of sub-groups according to a time
stamp in said node value pair of each of said plurality of errors

51. A system according to claim 47 further comprising:

instructions stored in said memory for directing said processing unit to:
generate a case number for each of said plurality of sub-
groups;

determine one of said plurality of sub-groups is not recorded in
said database based on said case number;

assign a case record to said one of said plurality of sub-groups;

receive inputs from a user regarding handling information for
said case record; and

store said inputs to said case record in said database wherein
said case record includes said one of said plurality of sub-groups and
said handling information.

52. A system according to claim 40 further comprises:

instructions stored in said memory for directing said processing unit to:
receive inputs from a user including additional handling
information for said case record; and
update said case record to include said additional handling
information in said database.

53. A system according to claim 40, wherein said database is a multi-dimensional database.

54. The system according to claim 47 wherein said node value pair comprises a substation number and said instructions to generate said target fault signature sequence further comprises:

instructions stored in said memory for directing said processing unit to:
serialize said plurality of errors in accordance with said
substation number in said node value pair of each of said plurality of
errors.

55. The system according to claim 47 wherein said node value pair comprises a panel number and said instructions to generate said target fault signature sequence further comprises:

instructions stored in said memory for directing said processing unit to:
serialize said plurality of errors in accordance with said panel
number in said node value pair of each of said plurality of errors.

56. The system according to claim 47 wherein said node value pair comprises a status value and said instructions to generate said target fault signature sequence further comprises:

instructions stored in said memory for directing said processing unit to:
serialize said plurality of errors in accordance with said status
value in said node value pair of each of said plurality of errors.

57. The system according to claim 47 wherein said node value pair comprises a time stamp and said instructions to generate said target fault signature sequence further comprises:

instructions stored in said memory for directing said processing unit to:
serialize said plurality of errors in accordance with said time stamp in said node value pair of each of said plurality of errors.

58. The system according to claim 47 wherein said node value pair comprises an alphabetic code and said instructions to generate said target fault signature sequence further comprises:

instructions stored in said memory for directing said processing unit to:
serialize said plurality of errors in accordance with said alphabetic code in said node value pair in each of said plurality of errors.