

Quality Assurance Solutions for Monitoring and Control Software Systems of Electrical Substations

MIHAIELA ILIESCU¹, VICTOR URSIANU², FLORICA MOLDOVEANU²,
MARIUS AVRAMESCU², RADU URSIANU²

¹Faculty of Engineering and Management of the Technological Systems

²Faculty of Automatic Control and Computers

University "POLITEHNICA" of Bucharest

Splaiul Independentei nr. 313, Bucharest

ROMANIA

iomi@clicknet.ro; victor.ursianu@gmail.com

Abstract:- General aspects regarding monitoring and control systems of electrical substations are discussed. Solutions are proposed to ensure software quality including methods for preventing fault injection and techniques meant to eliminate and isolate faults. Using a monitoring system developed in Romania as case study, we proposed modeling this system using SCL language and Bayesian networks in order to estimate its reliability. A software simulator for IEDs was created for this system and was used during specific tests.

Key-Words: - software quality assurance, software metrics, fault tolerance techniques, IEC61850, Smart Grid, energy field, Bayesian networks, Markov chains, fault prediction, recovery blocks, N-version programming.

1. Introduction

The quality problem of monitoring and control software systems from the energy field is of great issue.

The quality of the system depends on the quality of all its hardware and software components.

First of all, we must be aware of the power field importance.

The national and international economic development depends on the power field. Huge financial supplies are available for the development of the Smart Grid concept in the European Union and not only.

There is a common assent of the specialists that the power field and the future Smart Grid cannot work without monitoring the parameters of the power equipment within the electrical substations. The quality of a software product is given by "the ability to be used efficiently, effectively and comfortably by certain users for a set of goals in the specified conditions". Ensuring the software quality of the system is very important in order to have a reliable product with as few failures and errors as possible. In the worst case, the situation may generate a power unbalance in a certain geographical area and cause important material damage for consumers.

2. Analysis of Quality Aspects for Monitoring and Control System of an Electrical Substation

2.1. General architecture

A monitoring and control system of an electrical substation requires the existence of electronic devices that monitor the electrical equipment parameters (IED - Intelligent Electronic Device), one or more computers (servers) that centralize the monitored data and save it into a database and one or more computers (clients) that run a client software which has the purpose to present the monitored parameters to the users.

Generally, such a system has a client-server infrastructure (Fig. 1).

The server software applications receive in real-time the values of the electric equipment monitored parameters from the IEDs, process these values and then save it into a local or central database. The client software application receives data from the database and presents them to the user (the staff of the electrical substation) using a friendly user interface.

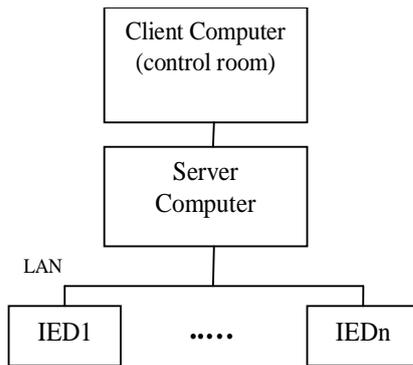
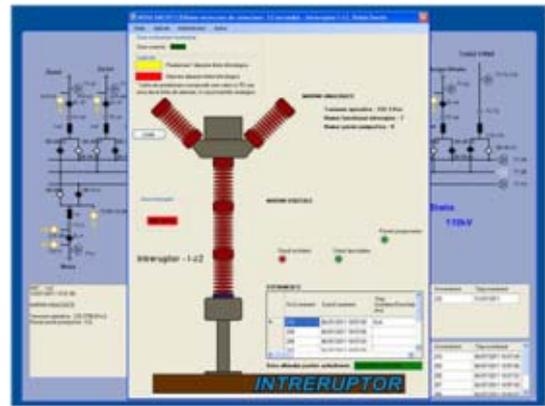


Fig. 1 The general architecture

Fig. 2. The *EMCSIT Station* software

2.2. Problems that may cause system failures

The problems that may cause a malfunction of a monitoring and control system of an electrical substation may vary, from the delay of data transmission, wrong data display, to the failure of one component or of the whole system.

The information obtained with the help of the monitoring and control system is useful to local and national energy dispatcher, each delay in pointing out a fault of the monitored electrical equipment may affect the energy balance in a certain area.

3. Case Study: Solutions for Ensuring the Quality of EMCSIT Monitoring and Control system

3.1. The particularities of the system

EMCSIT is a complex online system that monitors an electrical substation [1].

The EMCSIT system has client-server architecture and consists of IEDs (Intelligent Electronic Device) placed in every relay cabin inside the electrical substation. They are connected to sensors and transducers, mounted on the electrical equipment of the substation and transmit the received data to local servers.

The monitoring system includes computers (referred as servers), which are connected to the monitoring equipment (IEDs), receive the values of the monitored parameters and save these values in a local Oracle database. In addition to these servers, there is a central server that performs the synchronization with the other servers and acquires all the data in order to make it accessible to the EMCSIT client-software (Fig. 2) in the local network.

Because a local server connects to more IEDs, it will have to run one instance of server-type software for every connected IED. In order to prevent running more windows for the server software, each one connected to one IED, *EMCSIT SuperServer* software was developed. It runs one instance of server-type software for every IED, not visible to the user, by using more execution threads. Every minute, the *EMCSIT SuperServer* software interrogates all the IEDs that are connected to the local server and receives the data regarding the monitored parameters of electrical equipment.

After the data package from the IED has been received, it is processed and saved in the database installed onto the central server of every substation inside the component of the electrical substation.

During the system development, the number of faults detected in every software module inside the system was gathered and the failure rate of every system component was determined.

Among the noticed software errors we can mention: ignoring all possibilities that may occur in exploitation, calculations that may lead to wrong results; ignoring division by zero in calculations; wrong timed auto-scale of the monitored parameters that leads to appearance of infinite loops; wrong software design and implementation, that leads to exceeding of the allocated memory or stack and causes the interruption of normal operation, etc.

3.2. Solutions implemented for ensuring the quality of EMCSIT system

3.2.1. The development of an IED simulator for testing the system

The development of a monitoring and control system of an electrical substation involves software components as well as hardware components and demands a huge financial and human effort.

It is necessary that such a system should be as reliable as possible. Therefore the software components are subject to independent tests and then acceptance tests in the substation, with all the components connected and functional. We considered the development of a software simulator for IEDs to be necessary due to the many advantages it offers: much more thorough testing, obtaining a much more reliable product, shorter testing period in the electrical substation, etc.

3.2.2. Generating test cases using Pairwise testing method

Generating test data by analyzing the input data leads to many test cases: the test vectors represent combinations of selected values for input data.

For testing the server software, the simulator generates test data using technical specifications of the monitored parameters for the electrical equipment: minimum value, maximum value and the pre-alarm value.

The Pairwise testing [2] ensures that every possible combination among the selected values for every pair of input variables is tested at least once. Empirical studies have shown that using this method may help detecting approximately 70% of all the faults in the software.

The simulator generates test cases via the Pairwise testing method, and sends these data to the server software. The time interval when the test data packages are sent and the number of analogue inputs can both be modified; the simulator can generate data packages for 16 analogue inputs maximum. By using the log files generated by both the simulator and the server software, correlations can be made in order to observe whether all the test data has been received and to observe the state of equipment for these values. This way, the accuracy of calculations made in the server software can be verified.

3.2.3. Implementing fault tolerance techniques

3.2.3.1. Using recovery blocks

The recovery blocks method is used in the EMCSIT project for event acquisition (operation of different electrical equipment: open/close operation for circuit breaker).

The server software interrogates every connected IED in order to acquire the event list generated in the substation and registered by the IEDs in the internal memory. This list is compared with the already existing data from database in order to identify the new events. When a new event is detected, its ID (identifier) and the ID of the corresponding IED are transmitted to the module

that acquires events. If this module fails for whatever reason (failure to acquire the data package for the event from the IED, failure to save in the database etc.) there are previous steps implemented in the software where the EMCSIT Server application can return and restart the process of acquiring events.

3.2.3.2. Using duplication techniques

The duplication techniques are used in the following situations:

- If an IED stops working or the communication with it fails, the software that communicates with the equipment must warn the user and try to obtain equivalent data from other equipment;
- If the connection between the local servers that acquire and record the data from the monitoring equipment is not possible, after the situation is solved, the *EMCSIT Server* application will acquire and display the data regarding the monitored parameters registered in the IEDs' internal memory when the communication was not possible.

3.2.3.3. Using reconfiguration and rejuvenation techniques

The EMCSIT system can be configured in such a manner, that every local server or the central server has, at some point and depending on several factors, the possibility to reinitiate. The reconfiguration takes into account the processor's load status, insufficient free internal memory, the blocked communication ports, network interface problems.

The solution consists in allocating some additional resources to ensure the system's continuous run. If the EMCSIT system runs on a MS Windows platform, an extreme solution is programming the server, on which the software is running, to restart at a certain moment. Rejuvenation is a technique that takes into account the fact that the *EMCSIT SuperServer* or the *EMCSIT Station* software may enter an infinite loop or may exhaust its resources. If a deliberate or an accidental restart takes place, the software will run automatically on the server or on the client computer after the system has loaded.

4. IEC61850 and Bayesian networks

4.1. SCL Language

The last years have produced significant progress for standards that define communication in electrical substations.

Probably the most important step in this direction was the publication of the IEC61850 standard. Substation Configuration Language is the language and representation format specified by IEC61850 for describing the equipment configuration of an electrical substation.

The complete representation of data in the SCL language enables the possibility of interoperability between equipment from an electrical substation via file exchange. The great advantage of using IEC61850 is the interoperability between IEDs, which belong to different manufacturers. The language has a standard naming convention for data, equipment is automatically described via the SCL language, it allows virtual modeling for logical equipment and offers a common language for configuration of equipment.

4.2. The mathematical model of Markov chain

In the probability theory, a Markov model is a stochastic process possessing the Markov property.

This property means that future states of a system modeled via a Markov chain depend on the present states and are independent of the ones from the past.

Based on our experience in developing and using some monitoring and control software systems, such a system should be associated at a certain point with one of the following states:

1. Good State (S1): overall, the system works according to specifications;
2. Acceptable State (S2): there are some missing data packages (they are not received by the server software) in certain time intervals; the errors that appear sporadically do not call for intervention by the substation personnel;
3. Bad State (S3): the system is unavailable most of the time; data packages are often missed, delay in displaying the acquired data; the user has to interfere in order to restart one or more software systems; it may become necessary to restart the client computer, one or more IEDs;
4. Unacceptable State (S4): the majority or even all system components are not functioning or are not functioning according to the specifications; the system is no longer available;

The proposed method for estimating the current state and for estimating the transition time between states is based on the analysis of log files.

In the case of the EMCSIT monitoring and control system used as case study and presented in chapter 4, the log files created by the server software contain data about the receiving date of every data package, the ID of the package, whether the package was valid or not and the state of the monitored electrical equipment. By analyzing the log file, it can be observed whether there are missing packages between those that were supposed to be received at pre-established moments of time.

By observing the system behavior for a sufficient period of time, we discovered that transition from one state to another is connected with the number of lost data packages (not received).

Therefore, the system can be classified in four discrete states:

- If less than one data package per day is missed (not received by the server software), then the system is in the S1 state;
- If no more than one data package is missed every six hours or an average of four data packages are missed per day, then the system is in the S2 state. The system is also in this state if maximum two data packages are missed consecutively;
- If there are maximum 24 missed data packages per day (an average of one data package every hour) or more than three consecutively missed packages, then the system is in the S3 state;
- In worse situations than the aforementioned ones, the system is in the S4 state.

The Markov model can be used to estimate transition times between the states of the monitoring and control system, respectively when the system will reach the unacceptable state [3].

4.3. Bayesian networks

A Bayesian network (BN) is a directed graph without cycles representing probabilistic relationships between the variables of an array.

Variables are represented as nodes: $\{X_1, X_2, \dots, X_n\}$ and the relations between them are represented by arcs:

- parent node is the variable that determines the change of the child variable;
- child node is the variable that is influenced by the parent variable;

The intensity and the way in which the influences between variables show, are given by the conditional probability table of each node separately.

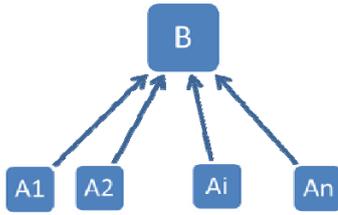


Fig. 3. Example of a Bayesian network (BN)

The conditional probabilities are based on the information from the past. A conditional probability is mathematically written as $P(x|p_1, p_2, \dots, p_n)$ and it is likely that X variable is in x state if the parent P1 is in p1 state, P2 is in p2 state, ..., respectively Pn is in pn state.

In the case of a BN (Fig. 3), in order to calculate the fault probability of the parent nodes marked with Ai, Bayes formula is applied for the calculation of the conditioned probability:

$$P(A_i | E = \text{defect}) = \frac{P(A_i) \times P(E = \text{defect} | A_i)}{\sum_{j=1}^n P(A_j) \times P(E = \text{defect} | A_j)}$$

where n is the node that directly influence the node i, $i=1, \dots, n$.

The probability that the child node B defects is:

$$P(B) = \sum_{i=1}^n P(A_i) \times P(E = \text{defect} | A_i)$$

The BNs can estimate the probability of errors for a software system based on the history of the fault data for the constituent modules.

Our purpose is to create a monitoring and control system for an electrical substation through a BN [4]. This will allow us to calculate the fault probability of each node and for the entire system. For the calculations regarding the fault probability of the nodes (IEDs) and of the monitoring and control system, information regarding the number of faults must be acquired on a certain period of time for each IED. Each child node is characterized by a certain fault rate (number of faults in a given period of time). The BN (as orientated graph) supports any kind of relationship between the nodes. Depending on the relationships between the nodes, the fault probability of the node is calculated [6].

5. Developed software

5.1. Software simulator

This IED simulator sends on a serial interface, data packages that are acquired by the EMCSIT server applications.

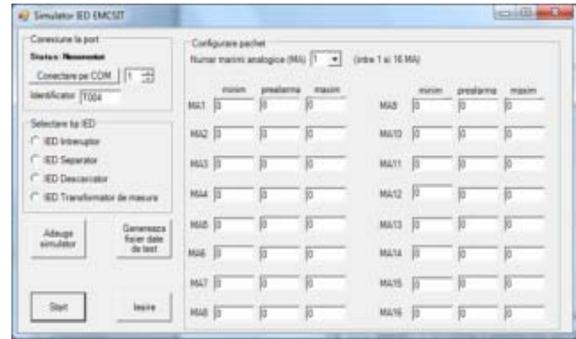


Fig. 4. The configuration window of IED simulator

The data packages sent by the simulator are generated based on input data specific to electrical equipment that is connected to the IED. The input data differ depending on the type of IED.

A configuration is required, in which the type of IED to be simulated, the number of parameters monitored by IED, the domain of values for every monitored parameter, limits for alarm are defined (Fig. 4). The simulator can add more virtual IEDs thanks to the implementation of Daisy Chain communication protocol between IEDs. The IEDs are connected to a server computer in a Daisy Chain configuration by connecting each IED to another IED, not by connecting each IED directly to the server. Only the last IED in this chain connects directly to the server. Each IED has an identifier, the communication being made on the same bus, depending on the identifier's value.

When the simulator is started, at certain moments of time, it will generate data packages that will be sent to the serial interface, from where they are received by the server.

The simulator generates log files that contain information regarding the date when the data package was generated, its identifier and if it was successfully received by the server application.

5.2. Parser SCL and BN Calculus

For modeling an electrical substation according to IEC61850 standard, SCL (Substation Configuration Language) language and Visual SCL application were used [5]. This software allows the user to save the substation schema in the XML standard format type specific to IEC61850.

For the purpose of using the information from .SCD files, *Parser SCL* software module was developed. It reads the information from the .SCD files written in the configuration language of an electrical substation - SCL and delivers as output, data structure that can be later defined as nodes of a Bayesian network (BN).

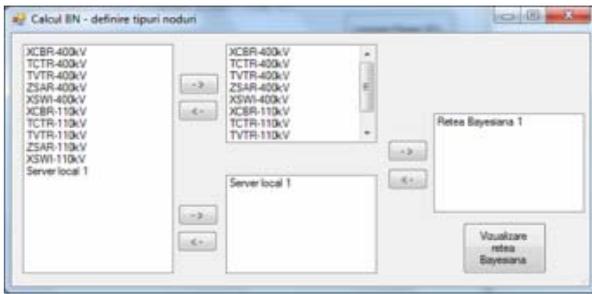


Fig. 5. Define the node types for the BN

The output data is later used by the *BN Calculus* module for estimating the system's reliability (Fig. 5). The nodes and the information associated to it that represents the fault data of IEDs will be used for estimating the fault probability of the monitoring and control system. The information obtained (electrical equipment and the associated IEDs) is used for the mathematical model of the BN associated to the monitoring and control system. After the connections between the IEDs and the electrical equipment have been made, *BN Calculus* reads the IEDs list and offers the user the possibility of completing the fault data.

5.3. Diagrams

The *Diagrams* software application was created for creating virtual entities in a substation that can be later used for BN modeling. The user can connect different types of entities - electrical equipment, save the configuration in xml files or load other substation configurations.

Each entity has specific attributes such as: name, voltage level, type (transformer, circuit breaker, disconnecter, etc.), neighbors, physical coordinates, etc. (Fig. 6).

Equipment can be lately integrated in electrical cells. This feature along with the possibility to define neighbors is of great help in modeling the electrical substation using Bayesian networks.

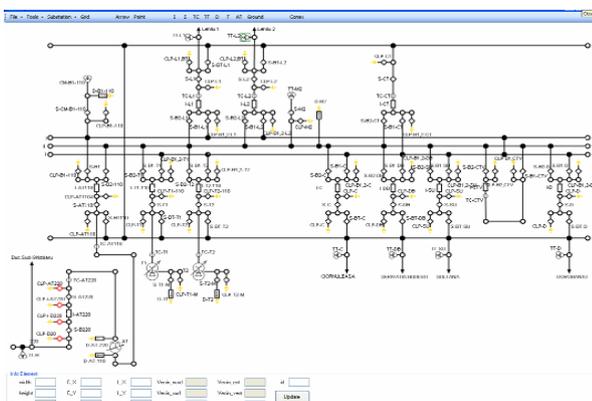


Fig. 6. *Diagrams* application

In future development of this application, the *BN Calculus* module will be integrated in order to calculate the failure probability for electrical cells and further on for the entire electrical substation.

6. Conclusion

Using as case study, a system that was developed and installed in an electrical substation from Romania (EMCSIT), solutions and examples for applying these solutions were described, in order to ensure the quality of this system. A software simulator of IEDs was developed and using pairwise method, the modules of EMCSIT system were tested.

Some mathematical models for estimating the reliability of monitoring and control systems for the electrical substations are proposed: Markov Chains, Bayesian networks.

The EMCSIT system is modelled as a Bayesian network using information from SCL file (IEC61850) and the failure probability of the system is calculated using software developed by the authors.

The obtained results were used by the developer of EMCSIT system to improve its hardware and software modules.

References:

- [1] C. Moldoveanu, V. Ursianu, M. Avramescu and others, Intelligent electronic system for continuous monitoring and diagnostic of high voltage substations, *International Conference on Condition Monitoring, Diagnosis and Maintenance*, Bucharest, 2011.
- [2] K. Naik, P. Tripathy, Software testing and Quality assurance, Wiley Ed., 2008.
- [3] C. Moldoveanu, R. Ursianu, E. Ursianu, V. Ursianu and others, Determine the optimal moments for investigating technical state of primary equipments for the purpose of assuring the safety levels imputed by the National Company Transelectrica – România, *International Conference on Condition Monitoring and Diagnosis*, Beijing, 2008.
- [4] V. Ursianu, R. Ursianu, E. Ursianu, Bayesian Networks to predict Software Quality, *SPSR Conference of Statistics and Probabilistics Romanian Society*, Bucharest, 2011.
- [5] Visual SCL, Applied Systems Engineering website: <http://www.ase-systems.com>
- [6] V. Ursianu, Quality assurance solutions for electrical substations monitoring and control software systems, Ph.D. Thesis, 2011.