A Robust and Maintenance-Free Alarm Processing Solution for Transmission System Operations Control Centers

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE – UFCG
(Brazil)

A. S. de Araújo, M. S. M. Cavalcante
COMPANHIA HIDRO ELÉTRICA DO SÃO FRANCISCO – CHESF
(Brazil)

ABSTRACT

Large and complex systems (such as computer networks and power transmission systems) are difficult to operate and manage. Even when good remote managing capabilities are available, emergency situations prompt “alarm avalanches” that make diagnosis more difficult. This situation has prompted the development of intelligent alarm processing systems (IAP systems). Although it has been shown that IAP systems can be highly effective, most of them fail in practice due to the maintenance burden. In fact, most IAP systems require maintenance whenever the underlying monitored system changes. This paper presents the simplification and enhancement in maintainability of IAP systems made possible by separating topological information from the knowledge base (referred to as topology rule base). We investigate this idea in the context of a power transmission system, submitting to the same maintenance operations both a topology rule base IAP system and an identical traditional production system (i.e., one without topological information and with a topology-specific rule base). Our results confirm that the topology rule base substantially curtails the effort expended for typical maintenance procedures. Additionally, field experience in debugging an IAP system is described and the quality of the final system is characterized.

1.0. INTRODUCTION

A typical power system control center is responsible for managing a system that comprises several substations, monitors large extensions of transmission lines and an impressive number of different types of devices, and is equipped with thousands of measuring points. Under critical situations, a huge number of alarms arrive at the control center indicating potential abnormal situations. A single root fault in the system can lead to an avalanche of alarms. For example, regional control centers of Hydro-Québec, Canada registered up to 2000 alarms for a general substation fault or up to 15000 alarms per regional center during the first five seconds of a complete system collapse [3]. For its part, a regional control center of Companhia Hidro-Elétrica do São Francisco (CHESF), Brazil, once registered more than 1500 alarms in one
second during a partial system black-out [2]. The interpretation of this massive volume of alarms with the aim of diagnosing faults constitutes a real challenge for human operators, since they have to make correct and efficient decisions about the actions to be triggered to deal with the emergency situation.

In order to cope with these situations, some modern power system control centers are equipped with computer-based intelligent alarm processing systems (IAP) [5] in order to assist operators in analyzing acquired real-time information about power systems. These applications are of utmost importance when a system works under abnormal condition. In such a situation, the operator in charge of the system must act promptly to recover and pull the system back to a safe operational state. If the operator does not succeed in recovering the system in time, catastrophic consequences can result. For instance, a blackout situation or even the entire collapse of the system may occur, causing unpredictable social and economic consequences.

Due to the symbolic nature of the reasoning associated with alarm processing, knowledge-based systems (KBS) appear as a promising solution. In fact, many successful knowledge-based systems to assist human operators at power system control centers have been developed. SPARSE [10] is an example of a KBS installed at Portuguese power system control centers that helps operators to interpret massive numbers of alarm messages during incidents. Smart One [2] is another example of a KBS installed at CHESF power system control centers.

KBS have many advantages when compared to traditional computer applications. However, they present some well-known disadvantages, mainly related to knowledge acquisition and knowledge maintenance [11].

1.1 Knowledge Acquisition and Maintenance

Due to the difficulty in extracting knowledge from the experts, the knowledge acquisition phase is very time-consuming and constitutes one of the main problems of KBS development. On the other hand, the maintenance of knowledge bases is also not a trivial task. According to [13], the main difficulties in maintaining KBS are: i) consistency and homogeneity of knowledge; ii) predictability of rules; iii) organization of knowledge base; and, iv) reusability of previously encoded knowledge. Considering power systems in which changes in the topology occur frequently, the difficulties mentioned above are more evident.

To address the problem of maintaining KBS one must 1) keep the knowledge base as small as possible; 2) separate control knowledge from rules; and 3) provide methods to validate and verify the consistency of the knowledge base when new rules and facts are added.

To illustrate knowledge acquisition and maintenance problems, Soloway et. al. reported that 50% of XCON’s rules are updated every year [9]. For the whole CHESF transmission network, well over 20000 specific production rules are needed to generate diagnostics. Implicit procedural controls are embedded in the rules, creating implicit links between rules even if there is no data dependency. Therefore, even a small rule base upgrade for maintenance purposes may disturb a large part of the rule base. Moreover, the CHESF network topology changes several times a month. For each change in the topology, it would be necessary to rewrite or update the rule base. In view of the size of the rule base, the rule base maintenance task becomes very difficult to perform.
Consequently, alternative approaches must be investigated with the aim of reducing the size of the rule base, in order to simplify and improve its maintenance.

1.2 Objectives and Organization of the Paper

The purpose of this paper is to demonstrate the simplification and enhancement in maintainability of IAP systems made possible by introducing knowledge bases composed of topological information and generic rule bases (topology rule bases). An IAP system currently in operation at CHESF (Smart One [2]) employing a topology rule base has been developed and studied in operation over a period of many months in order to characterize the effort required in maintaining the knowledge base as engineers perform typical changes to the electrical network. We confirm that the topology rule base substantially curtails the effort expended for typical maintenance procedures.

A second goal of the paper is to describe field experience of what has been a difficult system to develop. We focus on our maintenance methodology for evaluating the quality of diagnostics and debugging the system. Data is provided concerning the growing accuracy of diagnostics, the causes of errors and the quality of the system in its current version.

The rest of the paper is structured as follows: section II describes related work; section III presents the IAP system, Smart One; section IV discusses the maintenance of the Smart One rule base; field experience with Smart One over several months are discussed in section V; the conclusions close the paper in section VI.

RELATED WORK

Although KBS are difficult to maintain, there has been little research on rule base maintenance. Madeo et. al. proposed a method to evaluate a rule utility based on the frequency of rule firing [7]. Debenham proposed knowledge constraints as a tool for constructing maintainable expert systems [1]. However, neither work measured the effort expended during the maintenance.

[12] describes the design and implementation of an IAP system integrated in a real-time environment for power distribution networks. Among its main features, the authors cite the automatic creation and update of the knowledge base. However they do not explain how that feature has been achieved.

Yoshizawa et. al. have proposed a learning method named HCL (Hierarchical Concept-based Learning) to acquire knowledge for switching sequence generation [14]. HCL can simplify knowledge acquisition and maintenance through the learning process from experiences of human experts, and it enables the development of switching sequence generation of power systems with ease. On the other hand the authors themselves recognize that the method needs to be further developed to enhance its practical use.

In a concise paper, Lee and O’Keefe discuss the effect of knowledge representation schemes on the maintainability of knowledge-based systems [6]. Results show that an object-oriented system, compared to a structured rule-based system, is easier to maintain in terms of the time to perform maintenance tasks, but not necessarily in terms of the accuracy of the changes.
Contrasting the above efforts, some approaches have been studied that describe procedural control knowledge in languages that are independent of the production rules (i.e. meta-level control architectures). This architecture separates a rule base into control plans and pure declarative domain rules.

Ishida et. al. proposed a method that is emblematic of such approaches [4]. They separated a rule base into control plans and pure declarative domain rules, and showed that the architecture has no representation overhead and no significant run-time overhead. However, they did not demonstrate maintainability enhancement.

Sasaki et. al. expanded Ishida’s work to demonstrate maintainability enhancements with the separation of a rule base into control plans and pure declarative domain rules [8]. They measured the maintenance effort by counting the number of basic operations (e.g., reference to rules) performed during an upgrade. Contrary to the measurement of the time spent – an accurate measure but that may depend on personal ability – the operation count is only a rough measure of effort, but does not depend on personal ability.

Unfortunately, it is not clear that Ishida et. al.’s method of separating a rule base into control plans and pure declarative domain rules may be easily applied to the power system domain. The same holds for Sasaki et. al.’s experiments: what is the meaning of control plans in the context of power systems? Lastly, neither [4] nor [8] inform the size of the rule bases, with or without control plans.

**EVENT CORRELATION IN SMART ONE**

*Smart One* [2] is a software system that performs problem diagnosis for fault treatment in power generation and transmission systems. *Smart One* is currently deployed at a CHESF regional control center in Brazil. Successful operation has prompted its deployment to 4 other regional centers over the next year. Problem diagnosis is performed through the correlation between events generated by the Remote Terminal Units and received by the SCADA systems installed in substations and Control Centers. Events from the SCADA systems are sent to an Event Databus, as illustrated in Figure 1.

The Event Correlation module implements a hybrid event correlation technique consisting of, on the one hand, generic production rules related to CHESF transmission lines, busses, capacitor banks, compensators, reactors, generators and transformers (*Generic Rule Base*) and, on the other hand, CHESF network topological models (*Topological Model*). The two main advantages of this hybrid approach are: (1) the generic rule base is small, since a same rule can be reused by several devices; and (2) changes into the network topology do not cause changes to the generic rule base. In this manner, the effort needed to maintain the rule base up to date is minimal: this only happens on the rare occasions when new diagnostics need to be emitted leading to new rules or when errors are discovered in the rules.

The Event Correlation module retrieves *dynamic event windows* from the Event Databus, which are time related event sequences. Each event in a sequence specifies a state change for the topological model. Then all of the generic rules are parameterized with topological information, and the specific rules that were activated compose the diagnostic that is sent to the Diagnostic Databus.
In the two next sections, we detail the generic rule base and the topological model, respectively.

1.3 Generic Rule Base

In the system, 125 generic transmission line rules currently synthesize 10731 topology-specific transmission line rules\(^1\); furthermore, some predicates appear in all of the generic rules.

For example, the topological primitive predicate DISCONNECTED(line, substationId) currently appears in 476 specific 500kV transmission line rules, 3780 specific 230 kV transmission line rules, 300 specific 138kV transmission line rules, 5980 specific 69kV transmission line rules and 195 specific 13.8kV transmission line rules; a total of over 10000 specific transmission line rules.

Currently, the whole rule base is composed of 212 generic rules, about 60% of which are generic transmission line rules while the rest are generic rules for busses, transformers, reactors, capacitors, compensators and generators. By contrast, the equivalent topology-specific rule base is composed of around 12,800 rules. In order to use generic rules, these must be parameterized. A typical rule is given below:

\[
\text{TOTALLY}\_\text{DISCONNECTED}(\text{line}, \text{from}, \text{to}) \land \\
(\text{PROTECTION}\_\text{ACTIVATED}(\text{from}) \lor \text{PROTECTION}\_\text{ACTIVATED}(\text{to})) \land \\
(\text{BLOCKING}\_\text{RELAY}\_\text{ACTIVATED}(\text{from}) \lor \text{BLOCKING}\_\text{RELAY}\_\text{ACTIVATED}(\text{to})) \land \\
(\text{TIMED}\_\text{OVERVOLTAGE}(\text{from}) \lor \text{INSTANTANEOUS}\_\text{OVERVOLTAGE}(\text{from})) \land \\
\neg (\text{TIMED}\_\text{OVERVOLTAGE}(\text{from})) \lor \neg (\text{INSTANTANEOUS}\_\text{OVERVOLTAGE}(\text{from}))
\]

\[\Rightarrow \text{Automatic}\_\text{total}\_\text{line}\_\text{disconnection}\_\text{with}\_\text{over-voltage}\_\text{in}\_\text{source}\_\text{substation}(\text{line}, \text{from}, \text{to})\]

A rule is a Boolean expression. One can see that the primitives used in the rule (e.g., DISCONNECTED)\(\text{(line, substationId)}\) accept one or more parameters. By being evaluated with particular parameters, the generic rule essentially represents a specific rule (as present in typical IAPs); the expression evaluates as true if the specific rule “fires”. Although the generic rules were gathered in the context of a CHESF network, we believe that they are suitable for many electrical generation and transmission networks.

The ubiquitous primitives DISCONNECTED are evaluated with the aid of the topological model. Before explaining how these primitives are evaluated (section 1.5), we need to answer the question: what does the topological model consist of?

---

\(^1\) By a topology-specific rule (or simply specific rule), we mean a rule as it would appear in a rule base in which network topology has not been factored out. It is to be contrasted with a generic rule.
1.4 The Topological Model and Its Maintenance

The topological model is responsible for maintaining the state of the devices of the CHESF electrical network, and the connections among the devices. Conceptually, the topological model is a graph, in which each node represents a device and each edge represents a connection between two devices. Information in a node includes the activity vector, a bit vector signalling electrical measurements (bit “KV ↓0”, bit “MW ↓0”, bit “MVAR ↓0”, etc.) for transmission lines, protection operations (BREAKER OPENED, etc) for protection devices, and so on. For all devices for which generic rules exist, an additional vector – the rule vector – references the generic rules associated with the device. By navigating through the connections, all device nodes related to a particular device can be reached from the original device node.

An event occurrence on the Event Databus (Figure 1) informs a state change in one of the network devices; consequently, the graph representing the network is updated each time an event occurs.

It is easy to see that the graph is very large (currently around 50,000 nodes for about a quarter of the total CHESF network used in our pilot project). It is clear that the maintenance problem is real, and consists of:

- device additions; we consider as “devices” transmission lines, transformers, generators, reactors, compensators, capacitors banks, busses, breakers, switches, substations, etc;
- device removals;
- device configuration changes (whether a breaker is normally open or closed, for instance).

In order to simplify the maintenance procedure, part of the procedure is realized manually (raw topology maintenance), while the other part is realized automatically (structured topology maintenance).

Raw topology maintenance consists of adding devices and their connections, deleting devices and their connections, or changing device attributes. The task is performed by experienced CHESF maintenance professionals: on average, these changes are performed quickly. Raw topology maintenance data is stored in a text file, which is the first implementation of the conceptual graph. This implementation suffers from four main problems: (p1) low readability: it is difficult to recognize a transmission line or transformer, say, with all of its related protection devices (essentially, switches and different types of breakers), or to recognize busses and related breakers, etc.; also, it is not easy to recognize all of the devices and connections in a substation, etc.; (p2) poor description of the devices: for instance, the role each breaker assumes (transformer breakers, central breakers, transfer breakers, synchronous compensator breakers and capacitor bank breakers) are absent in the file; (p3) the impossibility of directly extracting parameters for the generic rules (recall that rules are parameterized); and (p4) the absence of device relationships in the file (most rules depend on the existence of relationships between devices, providing for higher-level diagnostics). For example, a generator is associated with a specific elevating transformer.

2 Near zero.
In order to reduce topology maintenance efforts, our technique automatically extracts information from the raw topology file and produces a structured topology file. The module that performs this transformation is called Topogiggio (see Figure 2). The raw topology file is Topogiggio’s input. The output is an XML file refining the topology in the input file, producing the XML structured topology file. Unlike the input file, the XML file is highly readable: user-defined tags easily identify substations, transmission lines inside substations, protection devices for transmission lines and transformers, as well as the connections among the devices. Also, the XML file provides a semantically rich description of the devices, such as the role played by each breaker, as well as the topological arrangements for each transmission line, capacitor, compensator, reactor, or bus. Also, and more importantly, the structured topology discriminates the parameters for the generic rules (Figure 1); this is done by extracting the relationships between devices. Finally, the XML structured topology file is generic, being therefore suitable for describing any electrical network topology.

![Figure 2 – Topological Analyzer Topogiggio](image)

### 1.5 Generic Rules Activation

Recall that each generic rule is implemented by a parameterized Boolean expression; when parameters are instantiated, the expression may evaluate as true if that instance of the rule fires, for a given model state; otherwise it returns false (section III.A). On the other hand, an event window in Event Databus (Figure 1) can contain events from one or more transmission lines or other devices; events from a transmission line can be the opening of a line breaker, information about overvoltage on the line, etc.

The Event Correlation module (Figure 1) works as follows. At any given time, a set of *event closure sets* are kept by the system. Whenever an event occurs on the Event Databus, it is added to the appropriate event set. This set is chosen to maintain correlated events together. Events are considered correlated when they involve related equipment. Each set of events is timestamped according to the time of occurrence of the last event that joined the set. Whenever an event set times out – that is, no further events have been added to the set for a period called the *affinity period* – then this set of events is then examined by the module. Each event specifies a state change for the model and this change is applied. Then all rules are checked. Since rules are parameterized with topological information, and since these parameters are contained in the structured topology file, rule firing can be verified.

As an example, consider the case of a transmission line. Each node of the topology graph (created during the processing of the XML structured topology file at system initialization time) represents a certain device of the electric system. The state changes in the device are represented by events stored in the activity vector of the corresponding node. The rules applicable to the device are stored in the rules vector. The nodes that represent transmission lines and busses also possess electrical measures (KV, MW, MVAR). Fault diagnosis for a transmission line consists of iterating over the rules vector of the corresponding node in the topology graph to verify which rules evaluate as true. The specific rules that were activated compose the diagnostic, which is sent to the Diagnostic Databus (Figure 1).
In next section, we illustrate through an example how the *Smart One* knowledge base is updated; we will consider the addition of a transmission line, event occurrences for the line, and how the *Smart One* Event Correlation module diagnoses the problem.

### 1.6 An Example of Smart One Operation

This section aims to clarify *Smart One*’s operation through an example, presented in the following subsection. The case study considers a particular transmission line, with id 05L9-AGD/RCD, from substation AGD to substation RCD, for which the one-wire diagram is shown in Figure 3.

![Figure 3 – Line 05L9-AGD/RCD One-Wire Diagram (Partial)](image)

Figure 4 shows the XML file that describes (part of) the power system topology. Recall that such XML topological description was generated by the *Topogiggio* module using the raw topology file prepared by technical CHESF personnel.

```xml
<ElectricalSystem id="CHESF">
  <Region id="CROL">
    <Substation id="AGD" name="ANGELIM II">
      <Bus500KV id="05B1-AGD" connections="35D1-1-AGD,35L5-4-AGD,35L6-4-AGD" breakers="15L5-AGD,15L6-AGD,15L8-AGD,15L9-AGD" busNumber="1" associatedBus="05B2-AGD"/>
      <Bus500KV id="05B2-AGD" connections="35D1-1-AGD,35L8-4-AGD,35L9-4-AGD" breakers="15L5-AGD,15L6-AGD,15L8-AGD,15L9-AGD" busNumber="2" associatedBus="05B1-AGD"/>
      <Switch id="35D3-1-AGD" connections="35E3-8-AGD,35L6-5-AGD,35L6-8-AGD,15D3-AGD" normalState="closed"/>
      <Switch id="35D3-2-AGD" connections="15D3-AGD,35L9-5-AGD,35L9-8-AGD"/>
    </Substation>
  </Region>
</ElectricalSystem>
```
Now suppose that all of the protection devices for the line are closed, or that the line is operating normally. Suddenly, some devices open, producing the event window shown in Figure 5.

Entirely based on the structured topology XML file in Figure 4, and taking into account the event window on the Event Databus, the first action of the Event Correlation module is to determine whether the line is disconnected or not, either from the From substation (From side), or from the To substation (To side), or from both substations, through the topological primitive DISCONNECTED(). As discussed in section III.A, this important primitive is present in all of the generic rules for transmission line, sometimes encapsulated into the primitive TO_TALLY_DISCONNECTED() or the primitive PARTIALLY_DISCONNECTED().

The primitive DISCONNECTED() is implemented as follows by the Event Correlation module: first, at system initialization time, the connection graph in Figure 6 is constructed from the XML file of Figure 4; second, the opened breakers in the event window of Figure 5 are processed, and then the graph is updated with the deletion of the breakers and their respective connections (Figure 7); third, the updated graph is analyzed, with the conclusion that line 05L9-AGD/RCD reaches neither the busses 05B1-AGD and 05B2-AGD (the two busses on the From side with which the line was previously connected), nor the busses 05B1-RCD and

---

**Figure 4: Structured Topology for Line 05L9-AGD/RCD**

**Figure 5: Event Window**
05B2-RCD (the two busses on the To side with which the line was previously connected); finally and consequently, both the primitive `DISCONNECTED()` for the substation AGD and the primitive `DISCONNECTED()` for the substation RCD, are evaluated as true.

![Figure 6: Partial topology (normal)](image)

![Figure 7: Partial topology (after failure)](image)

The next step is to verify the electrical measures (KV, MW, MVAR) for the line, with the aid of the activity vector for the line (section III.B). The Event Correlation module then concludes that the following generic (or parameterized) topological primitive, instantiated with the parameters for the line 05L9-AGD/RCD:

\[
\text{DISCONNECTED}(05L9, \text{AGD}) \\land \\text{DISCONNECTED}(05L9, \text{RCD}) \\land \text{KV}(\text{AGD}, ↓0) \& \text{KV}(\text{RCD}, ↓0) \\land \text{MW}(\text{AGD}, ↓0) \& \text{MW}(\text{RCD}, ↓0) \\land \text{MVAR}(\text{AGD}, ↓0) \& \text{MVAR}(\text{RCD}, ↓0) \Rightarrow \text{TOTALLY\_DISCONNECTED}(05L9, \text{AGD, RCD})
\]

is true.

Finally, the Event Correlation module scans the rule vector for line 05L9-AGD/RCD (sections III.B and III.1.5), in order to verify which of the transmission line generic rules evaluate as true. Only the rule ‘Automatic total line disconnection with overvoltage in substation From’

\[
\text{TOTALLY\_DISCONNECTED(line, from, to)} \\land \text{(PROTECTION\_ACTIVATED(from) OR PROTECTION\_ACTIVATED(to)) AND (BLOCKING\_RELAY\_ACTIVATED(from) OR BLOCKING\_RELAY\_ACTIVATED(to)) AND (TIMED\_OVERVOLTAGE(from) OR INSTANTANEOUS\_OVERVOLTAGE(from)) AND NOT (TIMED\_OVERVOLTAGE(to)) OR NOT (INSTANTANEOUS\_OVERVOLTAGE(to))} \Rightarrow \text{Automatic\_total\_line\_disconnection\_with\_over-voltage\_in\_substation\_From(line, from, to)}
\]

evaluates as true. Thus, the rules fires with parameter values `(line=05L9, from=AGD, to=RCD)`. It is as if the specific rule below (commonly found in traditional IAP systems) had fired:

\[
\text{TOTALLY\_DISCONNECTED}(05L9, \text{AGD, RCD}) \\land
\]

\[
\text{TOTALLY\_DISCONNECTED}(05L9, \text{AGD, RCD}) \\land
\]
\((\text{PROTECTION\_ACTIVATED(AGD)} \text{ OR PROTECTION\_ACTIVATED(RCD)}) \text{ AND} \)  
\((\text{BLOCKING\_RELAY\_ACTIVATED(AGD)} \text{ OR BLOCKING\_RELAY\_ACTIVATED(RCD)}) \text{ AND} \)  
\((\text{TIMED\_OVERVOLTAGE(AGD)} \text{ OR INSTANTANEOUS\_OVERVOLTAGE(AGD)}) \text{ AND} \)  
\((\text{NOT(TIMED\_OVERVOLTAGE(RCD)) OR NOT(INSTANTANEOUS\_OVERVOLTAGE(RCD))}) \)  
\(\implies\)  
\(\text{Automatic\_total\_line\_disconnection\_with\_over\_-\_voltage\_in\_substation\_From(05L9, AGD, RCD)}\)

With this specific rule, the Event Correlation module can finally update the Diagnostics Databus (section III.1.5) with the diagnostic shown in Figure 8.

**Table 1: Summary of Devices Configuration and Connection Changes**

<table>
<thead>
<tr>
<th>Devices</th>
<th>Number of Changes</th>
<th>Specific Rule Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.8 kV busses</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>69 kV busses</td>
<td>29</td>
<td>203</td>
</tr>
<tr>
<td>138 kV busses</td>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>230 kV busses</td>
<td>23</td>
<td>115</td>
</tr>
<tr>
<td>13.8 kV lines</td>
<td>22</td>
<td>88</td>
</tr>
<tr>
<td>69 kV lines</td>
<td>86</td>
<td>1032</td>
</tr>
<tr>
<td>138 kV lines</td>
<td>6</td>
<td>66</td>
</tr>
<tr>
<td>230 kV lines</td>
<td>121</td>
<td>2662</td>
</tr>
<tr>
<td>500 kV lines</td>
<td>2</td>
<td>48</td>
</tr>
<tr>
<td>69 kV transformers</td>
<td>20</td>
<td>120</td>
</tr>
<tr>
<td>138 kV transformers</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>230 kV transformers</td>
<td>31</td>
<td>393</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>361</strong></td>
<td><strong>4767</strong></td>
</tr>
</tbody>
</table>

In order to verify our claim of reduced maintenance efforts, we examined the Smart One system in actual pilot operation over a period of four months. The system was installed at CROL (one of the four CHESF Regional Centers). During this time, tens of changes were performed on the network topology, renaming devices, including new devices, new transmission lines, etc. Even substantial changes such as the addition of two whole new substations were performed over this period of time.
More than 5000 changes were performed on the topology over this four-month period at CROL. For example, more than 1000 changes were observed for switches and breakers. Considering only busses, transmission lines and transformers, the number of device configuration and connection changes was 361. This number would require 4767 changes in specific rules. In the same period, the total number of device removals and additions was 48, corresponding to 543 specific rules. Table 1 presents a summary about those device configuration and connection changes and the respective number of related topology-specific rules. Table 2 details the addition and removal of busses, transmission lines and transformers performed on the network topology.

<table>
<thead>
<tr>
<th>Devices</th>
<th>Additions and Removals</th>
<th>Specific Rule Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.8 kV busses</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>69 kV busses</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>230 kV busses</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>13.8 kV lines</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>69 kV lines</td>
<td>18</td>
<td>216</td>
</tr>
<tr>
<td>230 kV lines</td>
<td>8</td>
<td>176</td>
</tr>
<tr>
<td>69 kV transformers</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>230 kV transformers</td>
<td>6</td>
<td>78</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>48</strong></td>
<td><strong>543</strong></td>
</tr>
</tbody>
</table>

Table 2 – Summary of Devices Additions/Removals

The experiments conducted with this pilot operation have demonstrated that the organization of the knowledge base defined in the Smart One system is a good approach to enhance maintainability of intelligent alarm processing systems. Even considering the high rate of changes in the topology (over 5000 changes over 4 months), no update was required on the generic rule base.

The experience with this pilot operation can be summarized as follows:

- Raw topology information was maintained up to date by network engineers; this requires a substantial portion of a single engineer’s time, several hours a week.
- Producing a new structured topology file and importing it into Smart One requires executing a single program (Topogiggio), with no further human intervention. This program includes a module that detects possible inconsistencies in the rule base.
- Thus, at no time were changes to the generic rule base required, due to topological changes.

An important point to stress is that updating the raw topology must be performed even when no intelligent alarm processing system is in use. In other words, other Energy Management Systems, such as the “state estimator” require this topological information in any case. Thus, over a period of several months, no additional maintenance efforts were required to maintain the rule base up to date due to topological changes, whether maintenance work is estimated using time or number of operations.

**Smart One Quality: A Report from the Field**

Smart One has been tested in the field over a period of 18 months (September 2003 to February 2005). Improving the quality of diagnostic activity over time has been a major challenge. The main sources of errors in diagnostic activity are described below.
Noise. *Smart One* receives its input data from a SCADA system operating in a hostile environment. The end result is that the input event stream fed into the system (the so-called *sequence of events and alarms*) is quite noisy. There are frequently spurious events and many events are never received by the SCADA system. This was such a problem at first that the *Smart One* development team had to be refocused to handle this fact. Several techniques were investigated that could reduce noise levels. We finally decided on a noise filter that uses heuristics to remove events thought to be spurious and add events thought to be missing. Many heuristics examine the electrical measures (voltage, active and reactive power) to infer device states and hence guess what events are spurious or missing. We have been quite successful with this approach, as will be seen when numerical results are discussed below.

Rules. We have said in the last section that rules were not changed during the test period due to topological changes. However, many errors in rules were discovered by examining the cause of wrong diagnostics. Fortunately, this is a once-only effort. Of a total of 212 rules, approximately 50 have had to suffer changes after experts examined the cause of wrong diagnostic activity.

Time correlation. Many types of event correlation algorithms are used in the system. The first type of correlation performed is time correlation (discussed above in terms of event closure sets). Basically, a time window is used to correlate events in time. Choosing when to close a time window is critical to the proper correlation of events and this algorithm has suffered many changes over time. Currently, our algorithm is quite effective, although late events (up to 30 or more seconds late) can sometimes force bad diagnostics to be emitted. Our solution is to emit early diagnostics and, when a late event affecting the diagnostic reaches the system, to emit a new diagnostic preempting the earlier one. This has been deemed satisfactory by system operators.

Synchronizing state with the SCADA system. Network state is maintained separately by *Smart One* and by the SCADA system. These two network views must be kept synchronized. Our first synchronizing algorithms were weak and produced many bad diagnostics. This should have been discovered and fixed earlier but an oversight on our part kept using the weak algorithm for much too long. Today, this source of errors has completely vanished.

Miscellaneous causes. Many other causes of errors include software bugs, unsupervised devices, etc.

In order to evaluate the quality of diagnostics emitted by *Smart One* over time, a special evaluation process was used, together with tools specially developed to that end.

1. Diagnostics are shown to system operators but are also kept in a log. This log is examined daily by a CHESF engineer who also receives reports from post-operation analysts. These two sources are compared and a specially formatted Web page describing expected diagnostic activity is published using a Twiki collaborative tool [15].
2. Daily, a development team member then accesses the Twiki to examine error reports. This person uses a special tool that shows all events that occurred on this particular day and, using the error report produced by CHESF engineers, marks up the event log with the expected diagnostics.
3. With this mark up information, the tool automatically prepares an XML file describing a regression test for each expected diagnostic that was in error. These regression tests are added to a test database.
4. Before a new version of Smart One is deployed, all regression tests are expected to run perfectly, emitting the correct diagnostics. Our test database currently has many hundreds of regression tests.

As can be seen, the main aim of our method is to *automate* the testing activity.

The evolution of the Smart One system over an 13-month period is shown in Figures 9 and 10. Several points can be observed from the figures. First, diagnostic accuracy was really low at first and is only now over 90%. While this may seem very low, we are confident that our next version (to be installed shortly) will have better than 95% accuracy since the main source of errors (errors in rules) seems to have been eliminated, according to our regression tests. The remaining errors are due to noise, unsupervised devices and communication issues with the SCADA system causing late events.

Secondly, the accuracy has become better over time, although version 1.12 was subject to particularly harsh contingencies that uncovered several errors and bugs. The last version for which we have definite results (v1.13.3) no longer exhibits time correlation errors, very few noise-related errors and SCADA synchronization problems.

Regional center operators (the harshest critics!) are starting to have confidence in the system and this has prompted the decision to proceed with deployment to 4 other regional centers over the next year. A second regional center has recently started experimental operation and is using a new Web-based interface.

### 2.0 CONCLUSIONS

We have studied the advantages of the adoption of a topological model in maintaining rule-based systems. The topological model is responsible for maintaining the state of the devices an electrical network, the connections among the devices and, very importantly, several device parameters and device relationships that allow generic rules to be expressed parametrically. Conceptually, the topological model is a graph, in which each node represents a device and each edge represents a connection between two devices. It is quite general, being suitable for any electrical network, as well as telecommunication networks, computer networks, etc. The goal was to make the topological model easy to maintain: the structured part of the maintenance task is performed automatically, and in general changes in the topology do not require changes in the rule base.

The rule base is generic. To illustrate the advantages of our generic approach, for a system in use at a Brazilian generation and distribution company, 212 generic rules currently synthesize...
around 12800 specific rules. Parameters and device relationships are automatically extracted from the topology in order to activate generic rules.

Experiments with a pilot project operated over several months demonstrated the superiority of topology rule bases with regard to simplifying and enhancing the maintenance task of knowledge bases. Diagnostic accuracy has been improving with time and is currently above 90% and we expect it to reach 95% in the next few months.

We believe that our research on rule-based system can be widely utilized in production systems containing topological architectures.

3.0 REFERENCES


