

## Smart Action: a Tool to Help Power System Restoration

Jorge Figueiredo, Jacques Sauvé, Pedro Nicolletti, Eloi Rocha, Sérgio Araújo, Frutuoso Amorim, Anderson Feitosa, Ronaldo Agra, Willames Ribeiro

Universidade Federal de Campina Grande, Brazil

{abranter, jacques, peter, eloi}@dsc.ufcg.edu.br, {asergio, fgamorim}@chesf.gov.br,  
{andersonlf, ronaldogra, willamescm}@gmail.com

### Abstract

*This paper presents a tool that assists control room operators during power system restoration. The system is based on a formal representation of restoration plans that can be executed by control room operators through the help of a tool. The plan execution tool receives high-level diagnostics from an alarm processor and can then automatically initiate the execution of a restoration plan. The tool then presents restoration steps to the operator for execution and audits all steps performed. The tool has been tested off line and is set to undergo real-time experimental use in a control room at CHESF, one of Brazil's largest generation and transmission utility.*

### 1. Introduction

Power system restoration includes many activities, among which the two most important are:

- **Producing restoration plans.** Engineers use special tools to perform studies of the electrical system, examine possible contingencies and produce plans that determine how to restore the network under several scenarios.
- **Helping the operator in the control room.** Tools can help the operator analyze a given situation, locate appropriate (previously prepared) plans and execute them.

The Smart Action project was undertaken by CHESF (Companhia Hidro-Elétrica do São Francisco), one of Brazil's largest electric power generation and transmission utility, to deal with the second aspect: helping the *execution* of restoration plans. CHESF's power system consists of 14 hydroelectric plants and 94 substations with an installed capacity of 11,000 MW. CHESF's transmission grid comprises 18,000 kilometers of transmission lines, including 500kV, 230kV, 138kV, 69kV transmission lines.

All previous work that we have identified in the literature and in discussions with restoration experts deal with the first activity: producing restoration plans; see, for example, [1,2]. Several descriptions of restoration implementation strategies practiced internationally are presented in [3] where several references are available to restoration work done in the past, mostly in the 1980s and

1990s. However, only one work (see [4]) has been identified that formalizes restoration plans although it does not aim to be the basis of a tool to be used by control room operators, as we are proposing here.

The rest of this paper is structured as follows: Section 2 presents some reasons to have a tool to assist operators during the execution of restoration plans. Section 3 describes RPML, the language defined to express restoration plans. Section 4 introduces the Smart Action system, showing its architecture and implementation issues. Section 5 describes how experimentation was conducted in order to test the implemented system. Finally, the paper is concluded in Section 6.

### 2. Executing System Restoration Plans

The need for an executing restoration plans tool is of paramount importance, even when restoration plans have been prepared. The execution of restoration plans with quality is problematical for any or all of the following reasons:

- There are **many plans**; for example, CHESF's Eastern Regional Control Center (CROL) has to deal with hundreds of plans. Merely choosing which plan to execute is frequently difficult and prone to error or to wasted time;
- There is very **high coupling** between plans, since they refer to each other. Deciding on the order in which to execute plans, or even parts of plans is not obvious;
- A particular sequence of actions to be performed by an operator may be **spread over many plans**. The operator must therefore hop from plan to plan, in an attempt to execute actions in the right order. This hopping effect is particularly pronounced in systemic plans that cover large regions of the network, a contingency situation that is particularly stressful for operators.
- Plans are executed by **many people**, some of whom are in the regional centers while others are located in substations. Each operator involved in the restoration may have a copy of plans but the only way to keep all actions synchronized is to be on the phone and coordinate actions "manually".
- A plan consists of actions, each of which may be subject to conditions to be executed. Conditions may,

for example, involve tap settings, voltage levels, load levels, etc.) As a result, a plan includes very many conditions that affect the order of actions, choice between alternative actions. Since it is **difficult for an operator to check that all conditions are satisfied**, executing plans is error prone.

- It is difficult to **automatically audit** exactly what an operator did during a particular occurrence.
- Plans are written in natural language and are thus not formal; this leads to **ambiguity** and operating errors.

In the particular case of CHESF power system, there is one additional reason. CHESF used to be responsible for producing restoration plans for its own part of the Brazilian network. The restoration plans were produced by CHESF engineers after performing studies of the electrical systems, considering among other things the voltage level, the network topology and possible contingencies. However, recent changes in the Brazilian model of electric power generation defined a centralized planning. Currently, production of restoration plans is performed by the Brazilian Independent System Operator, called the National System Operator (ONS). There are actually still some restoration plans prepared by CHESF, but they are few and do not involve the core or systemic part of the network. Restoration plans are received by CHESF from ONS and are executed by CHESF personnel. The inadequate execution of restoration plans results in penalties being applied to CHESF's revenue.

For all these reasons, the focus of our work is on making sure that *execution quality* is high. The need to come up with a tool to automate (parts of) restoration plan execution is a non-trivial endeavor.

The following points summarize the aims of the Smart Action project and some high-level requirements for the desired system.

- Produce a system to help operators during restoration.
- Operators should be assisted in performing the required actions for occurrences of any kind, be they small (energize a single line, say) or large (responding to a partial or total blackout).
- Actions must be informed in a detailed fashion, all the way down to switching actions. This includes opening and closing breakers or switches, adjusting transformer taps, checking voltage levels, etc.
- The system must automatically determine that a contingency is occurring, find appropriate restoration plans, and "execute" them. "Executing a plan" means assisting the operator by presenting the list of actions that must be performed. The system itself does not act on its own: it merely suggests what actions should be performed by the operator.
- The system must check all conditions mentioned in plans and inform the operator whether or not they are satisfied; for example, if an action mentioned in a plan

must be performed when the voltage level of a bus is above a certain value, then the system must automatically check the condition. The system does not prevent the operator from performing any desired action, even if conditions seem not to be satisfied.

- The system must provide a log of all actions performed by the operator and conditions that applied at the time.
- The system must impose as little maintenance activity as possible as network topology changes and as plans are modified or new plans are developed.
- The system must be fully integrated with the SCADA system in order to obtain any required information.
- The system must be fully integrated with the Intelligent Alarm Processor (Smart Alarms [5]) system in order to obtain any required information (diagnostics).

### 3. A Language to Express Restoration Plans

As mentioned before, restoration plans used by CHESF are prepared by the Brazilian ISO (ONS). They are presented under form of Operating Instructions (OIs), an informal document created by engineers of the electrical studies team. An OI can be considered as a sequence of actions an operator must follow to bring the system back to normal operation. Each OI describes one or more restoration plans, and consider different scenarios that may occur. OIs are, in general, vague, incomplete and unstructured documents that can be understood only by humans. The Restoration Plan Markup Language (RPML) language that was defined to express restoration plans is an XML-based language intended to make it easy to translate OIs and other natural language documents into a formal representation. Think of RPML as a enabling the formal representation of any kind of OI, allowing its translation to a set of structured restoration plans. After a comprehensive analysis of the OIs adopted by CHESF, we identified that the language to express restoration plans should consider the following key aspects:

- **Basic structure and control mechanisms:** the language should offer ways to describe sequential instructions, loops, choices, conditional instructions and the possibility to set state and equipment properties.
- **Parameters:** Many plans are similar (for equipment of a given type, for example). Plans should therefore be as parameterized as possible to lower maintenance costs. The language must support the definition of generic plans or plan templates as well as a way to instantiate plans from templates.
- **Priority:** when facing systemic occurrences, the course of action for proper restoration is not obvious, even when restoration plans are available. The coupling between plans is high and priority

mechanisms are of utmost importance to decide in which order the plans must be executed. High priority plans should be able to inhibit or preempt the execution of lower priority plans.

- **Audit:** The language should provide ways to audit (some) actions during the execution of a restoration plan.
- **Interaction with operators:** in many situations, it is necessary that operators confirm or answer some questions about properties of system components, for example, the result of a visual inspection. The language must support some form of communication with operators.
- **Reference to switching sequence scripts:** Switching sequence scripts are formal and well-structured documents defined by CHESF engineers that describe some basic consolidated restoration plans for single equipments. A switching sequence script can be viewed as a structured mini restoration plan and, as a result, its execution can be automated. Information from switching sequence scripts can be inserted into restoration plans. Switching sequence scripts are frequently updated and being able to reference such a document (rather than duplicating the information it contains) is important to reduce maintenance overhead.

RPML is based on the Extensible Markup Language (XML). All plan elements and control structures are represented as XML elements. An XML element is described inside a block defined by a pair of tags: a start tag `<element>` and an end tag `</element>`. An XML element may have XML attributes that enrich the element with additional information. XML attributes of an XML element are defined by the assignment of a value to a key (the attribute's name) in the start tag of the XML element `<element key=value>`. An XML element may contain text or other XML elements. If an XML element is very simple, without text or sub-elements, it is denoted by a single tag `</element>`. In the examples used in this section, for the sake of simplicity, sometimes we omit some XML elements. We denote this by an ellipsis (...).

The root tag `<plans>` encloses a set of plans. Each plan inside this set of plans is enclosed by a `<plan>` tag. A plan can be: i) a generic plan that can be instantiated for different equipments and situations; ii) an entire restoration plan; iii) part of a restoration plan that can be invoked by other plans; and iv) a Switching Sequence Script. Figure 1 shows the RPML syntax and elements for a plan to energize the transmission line *04C8-RCD/GNN*. The element `plan` has eight attributes: `title` (a textual description of the plan), `class` (the type of equipment the plan belongs to), `voltage` (equipment voltage), `equipment` (name of the equipment), `FromSide` (the

source substation the transmission line is connected to), `ToSide` (the terminal substation the transmission line is connected to), `source` (indicates the source of the plan; this could be an OI, switching sequence script or even the name of an engineer who supplied the information) and `id` (a unique plan identifier). There are three different XML elements in the language that allows communication with the operator:

- `<step command>`: this element is used to present an action to be executed by the operator. This action is *auditable*, that is, Smart Action can check whether the step has been performed or not and inform the operator of the fact. Also, this information is part of the log produced by the system and can be used for post-operation audit purposes.
- `<step text>`: this element is also used to present an action to be executed by the operator but is *not* auditable.
- `<question>`: this is used to ask the operator a question. The `<text>` attribute is the text of the question put to the operator. The answer is obtained and the plan may proceed differently based on the information provided by the operator. The tag `<answer>` is used to specify what to do for different answers provided.

This is actually a very simple plan and is not representative of what Smart Action can do; it is being shown here to explain RPML syntax.

When the *Energizing-04C8-RCD/GNN* plan is executed, the operator is asked if the first terminal to be closed is on the RCD substation side. If answer is YES, then voltages in terminals RCD and GNN must be regulated to be less than or equal to 242kV and 235kV, respectively. This is an auditable action which means that Smart Action can actually check whether the operator has performed the action (voltage levels can be checked through the SCADA system). If answer is NO, only terminal GNN voltage must be regulated to be less than or equal to 235kV. After regulating voltages, line breakers should be closed in the appropriate sequence.

Figure 2 presents plans *blackout-RLD* and *trip-04T1-RLD*. The *blackout-RLD* plan has two new attributes:

- waitCondition: this attribute defines conditions that must be satisfied before the plan is executed. In this example, the blackout-RLD plan will be executed only when either a blackout in substation RLD occurred (diagnostic RLD blackout) or bus 04BP-RLD is has tripped (04BP-RLD Trip).
- substation: indicates the name of a substation.

```

<plans>
  <plan title="Energizing line 04C8-RCD/GNN" class="TL" voltage="230"
    equipment="04C8-RCD/GNN" FromSide="RCD" ToSide="GNN"
    source="IO-PM.NE.5LE-02" id="Energizing-04C8-RCD/GNN" >
    <question text="The first terminal to be closed is RCD?">
      <answer value="yes">
        <step command="Regulate RCD.voltage <= 242 AND GNN.voltage <= 235"/>
        <step command="Close 14C8-RCD"/>
        <step command="Close 14C8-GNN"/>
      </ answer >
      < answer value="no">
        <step command="Regulate GNN.voltage <= 235"/>
        <step command="Close 14C8-GNN"/>
        <step command="Close 14C8-RCD"/>
      </ answer >
    </question>
  </plan>
</plans>

```

**Figure 1: Simple restoration plan in RPML**

```

<plans>
  <plan id="blackout-RLD" title="Reenergizing, Rio Largo II substation" source="IO-
    RLD-01" class="SS" substation="RLD" waitCondition="diagnostic RLD blackout OR 04BP-
    RLD Trip">
    <inhibitPlans scope="plan.getSubstation = RLD">
      <refPlan ref="checkBlocking-RLD"/>
      <refPlan ref="initialProcedures-RLD"/>
      <refPlan ref="re-RLD">
        <param name="occurrence" value="blackout-RLD"/>
      </refPlan>
    </inhibitPlans>
  </plan>
  <plan id="trip-04T1-RLD" title="Reenergizing 04T1-RLD, Rio Largo II substation"
    class="TRANSFORMER" voltage="230" substation="RLD" waitCondition="Diagnostic 04T1-
    RLD Trip">
    <refPlan if="checkTransformerBlocking">
      <param name="Transf" value="04T1-RLD"/>
    </refPlan>
    <plan skipCondition="04T1-RLD.situation = 'unavailable'">
      <switchseq id="RTM-04T1-RLD" procedure="CLEARANCE" filter="CONTINGENCY"/>
      <step text="Hand over Transf 04T1-RLD to maintenance"/>
      <refPlan id="receiveFromMaintenance">
        <param name="eqpt" value="04T1-RLD"/>
      </refPlan>
    </plan>
    <refPlan ref="re-04T1-RLD"/>
  </plan>
</plans>

```

**Figure 2: More complex RPML plans**

The blackout-RLD plan has two new elements: `<inhibitPlans>` and `<refPlan>`. The `<inhibitPlans>` tag is used by high-priority plans. By using this element, it is possible to inhibit the execution of lower priority plans. Usually, `<inhibitPlans>` has a scope attribute that defines the scope of plans to be

inhibited. For instance, when `<inhibitPlans scope="plan.getSubstation=RLD">` is executed, all the plans related to substation RLD are inhibited. They will resume when execution ends for the block of instructions defined inside `<inhibitPlans>` `</inhibitPlans>` tags. It is possible however to individually free (resume) plans that were inhibited by using the `<freePlan>` tag.

The `<refPlan>` tag is used to invoke a (sub-)plan. The invoked plan must execute to the end before the original plan continues to execute the next instruction. It is therefore similar to a “subroutine call”. Plans may also be invoked through a `<refPlan>` tag together with a list of parameters. In Figure 2, the invocation of the `re-RLD` plan requires parameter occurrence and its value is set to `blackout-RLD`.

The first instruction of the `trip-04T1-RLD` plan in Figure 2 is the invocation of the `checkTransformerBlocking` plan. This instruction is followed by a sub-plan that will be executed if the condition defined by the `skipCondition` attribute is evaluated true. If the condition is false, the sub-plan will be skipped. The `skipCondition` and `waitCondition` attributes have distinct semantics. In the first, if the condition is false, the execution of the respective element is skipped whereas, in the latter, execution waits until the condition is true. The `<switchseq>` tag indicates that a switching sequence script (contained in a separate database) is invoked.

#### 4. The Smart Action System

Figure 3 shows the general organization of the Smart Action architecture which includes the following subsystems:

- **SAGE:** The SCADA system used by CHESF. It is accessed by Smart Action indirectly through the Smart Model subsystem.
- **Smart Alarms:** Intelligent Alarm Processor (IAP) developed to produce diagnostics from alarms and events [5]. It is accessed by Smart Action indirectly through the Smart Model subsystem. It consists of two subsystems – server and gateway; the gateway is used

to isolate and protect the SCADA from external subsystems.

- **Smart Model:** Maintains a view of the electric network entities, their status and attributes, etc. It serves as a basic source of information to Smart Alarms and Smart Action.
- **Smart Action:** The software system that helps operators in restoring the electric system when contingencies or manual interventions occur. It consists of two subsystems: Smart Action itself and SisRTM.
- **SisRTM:** The system that maintains low-level switching sequence scripts. It can also transform switching sequence scripts into restoration plans. Low-level switching sequence scripts for individual equipments (mostly transmission lines and transformers) are currently maintained in a separate database; this database needs to be used by Smart Action in order to reduce maintenance efforts (lessen duplication of information).

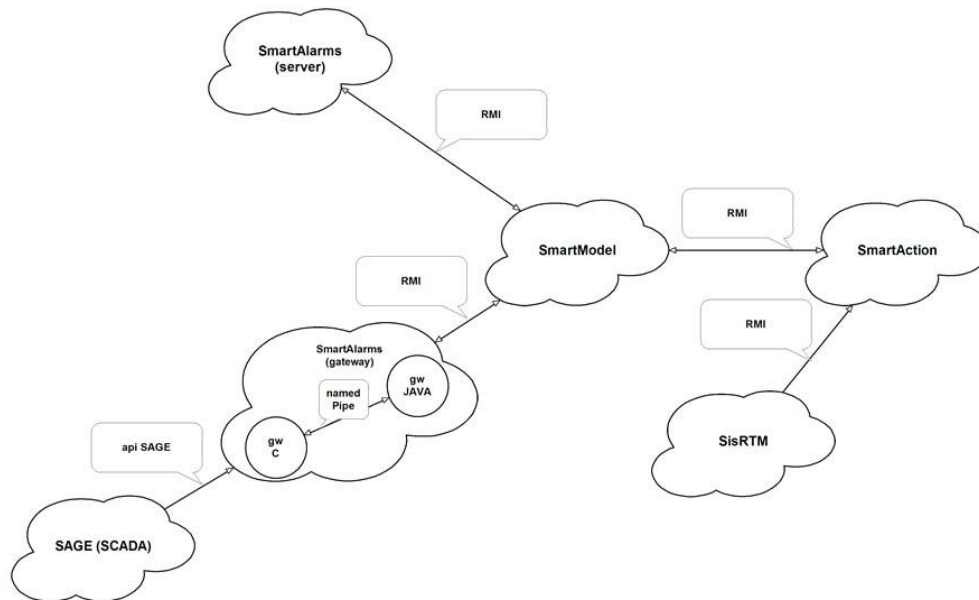
Most communication between subsystems is performed using Remote Method Invocation (RMI), although a named pipe is used to connect two parts of the gateway, one written in C, the other written in Java. The Java part offers a remote API.

#### Organization of the Smart Action system

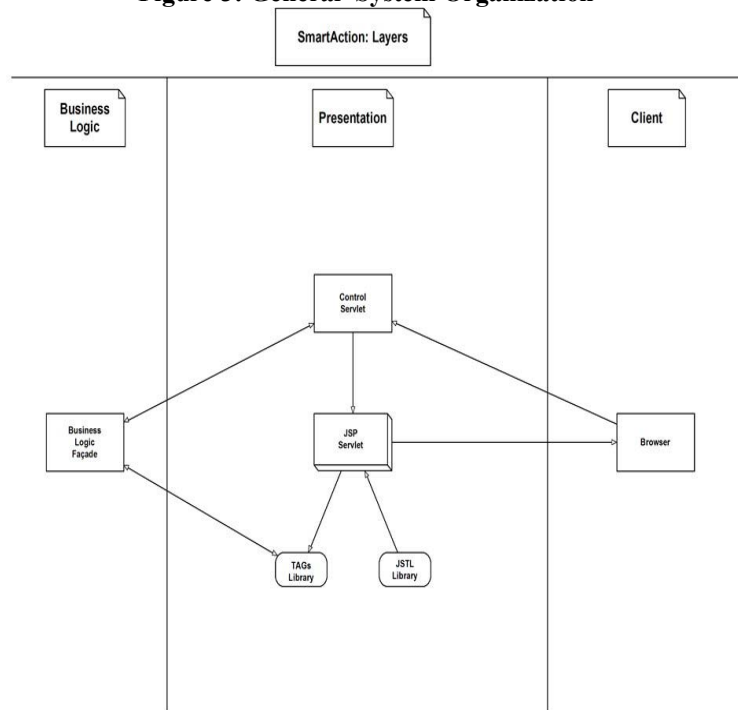
The system is distributed and is organized in 4 tiers (or layers): data, business logic, presentation and client. Please see Figure 4 and Figure 5. The system was developed using a Model-View-Controller (MVC) pattern in Java and XML. It runs in a Tomcat J2EE container under Linux or Windows.

##### Presentation Layer

The system has a web-based interface accessed through a Web browser. JavaScript is used for input data entry control. AJAX technology is used to provide a rich user-interface experience even while using a Web browser. Formatted reports are generated using IReport.



**Figure 3: General System Organization**



**Figure 4: Smart Action Organization (Part 1)**

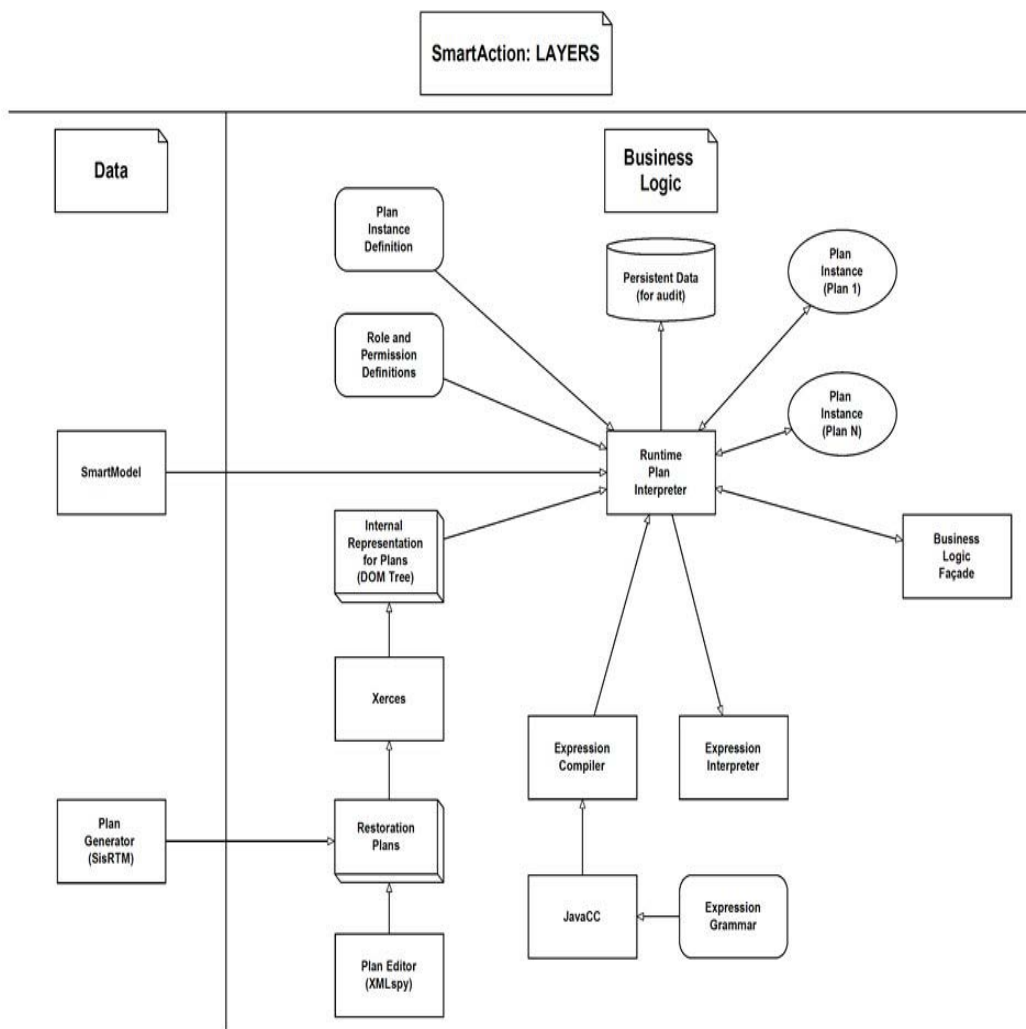


Figure 5: Smart Action Organization (Part 2)

**Business Logic Layer**

This layer implements all system functionality: instantiating and executing restoration plans, helping operators perform restoration actions. The basic inputs used to determine the plans to “fire” are diagnostics from the IAP. Detailed plans are expressed in RPML and low-level details also come from switching sequence scripts prepared through SisRTM.

The restoration plan interpreter instantiates plans according to configuration instructions and executes each possible independent plan in its own thread. Typically, a plan is instantiated for each element that be subject to independent restoration; this includes each transmission line, each bus, each transformer, whole substations, whole subsystems, etc. An executing plan typically waits for a condition that expresses when it should “fire”. When the condition is true, the plan begins execution and, under control of the operator, goes through the steps necessary to perform restoration.

In order to facilitate internationalization of the system, all message strings are defined in a “resource bundle”. Simultaneous access is allowed and controlled by the system. Persistent data is stored in standard files in XML format. System modules are packaged as .war files for deployment in a Tomcat servlet/JSP container. Business rules are integrated in the business logic layer. Security is ensured through the use of user identification, role definition and permissions assigned to roles.

All system activity is logged for audit purposes. All business logic is tested through acceptance tests executed automatically using EasyAccept [7].

**Data Layer**

Plans, log information and configuration information are maintained in standard files, in XML format. No database need be installed.

## 5. Experimenting the Smart Action System

Experiments were conducted by CHESF engineers in order to observe the execution of Smart Action System in a simulation environment that allows the representation of real occurrences in the power system grid. The purpose of such experiments was twofold: on the one hand, it was possible to validate the knowledge obtained from CHESF experts. On the other hand, it was possible to test the system in a simulation environment before placing it in production. Although many acceptance tests have been done during development using the EasyAccept framework, it does not guarantee a perfect functioning in the real time environment.

SIMULOP is a simulation environment provided by SAGE that uses, as engine simulator, the EPRI-OTS (EPRI Operator Training Simulator) [6]. SIMULOP is mainly used for operator training. Smart Action and SIMULOP are integrated in a single working environment. The architecture described in Figure 1 does not need to be modified. Smart Action continues to be connected to SAGE through the integration between Smart Model and Smart Alarms Gateway. The only difference occurs in setting which modules must be loaded when starting the SAGE (in a simulation environment the SIMULOP module must be started).

Using SIMULOP, it is possible to configure any occurrence in the Power System Network through a scenario. A scenario describes, for instance, which circuit breakers have tripped and which protection events occurred. When a scenario is prepared, it is processed by SIMULOP which simulates the occurrence in the SAGE SCADA. As a result, screens, digital and analog measures are updated in SAGE reflecting the new topological state of the network. After a few seconds, Smart Model is updated through the Smart Alarms Gateway; then, the Smart Alarms Server detects alarms and events and emits a diagnostic summarizing the occurrence. As a result, Smart Action receives these diagnostics and fires a Restoration Plan to respond to the occurrence. Figures 6, 7 and 8 present the topological screen of SAGE, the Smart Alarms screen with a diagnostic of an occurrence and the Smart Action screen with a restoration plan being executed, respectively.

Validating the Smart Action system is being performed according to the following four phases.

**Phase 1: Validation of main scenarios using Simulop.** Scenarios were defined to be run offline using Simulop to exercise common situations; the scenarios involve 6 substations, equipment covering several voltage levels (69kV, 138kV, 230kV and 500kV) under the control

of CHESF's Eastern Regional Control Center. More than sixty scenarios were created by CHESF engineers and operators, some of them involving substation blackout and circuit breaker failures and breaker transfer, others involving transmission line, transformer and reactor tripping.

**Phase 2: Observation of Smart Action online performance.** Smart Action will now be connected to a real online SCADA system and will be observed by a set of previously trained operators. However, at this stage, operators will *not* rely on Smart Action output to direct their actions. They will merely gage and report on Smart Action performance.

**Phase 3: Actual use by a single operator.** The use of Smart Action in daily operations will be tested. A single control center operator will be involved and will produce a daily performance report.

**Phase 4: Full scale experimental operation for a single control center.** All operators for CHESF's Eastern Regional Control Center (CROL) will now incorporate the use of Smart Action in their daily routine.

The current project scope does not go beyond this stage. However, if success is attained in Phase 4, it is quite likely that CHESF may adopt the system in the remaining 4 regional control centers and in the centralized control center.

## 6. Conclusion

The Smart Action system as implemented is capable of executing complex plans. The representation of plans for the eastern portion of CHESF's system is completed (about 30 substations and 1000 plans), covering one third of the entire CHESF system.

In this first phase of the project, Smart Action was installed at CHESF's Eastern Regional Control Center (CROL) and is being used for validation and training purposes. Phase 1 ended in August 2007; the CHESF team was in charge of all experiments and gave feedback of the detected problems to the development team. At the time of this writing, Phase 2 was planned to start on September 3rd, 2007 with Smart Action going online for the first time.

CHESF engineers are enthusiastic about the system and aim to put the system in experimental production use.

We are also investigating how to reduce the maintenance effort on restoration plans. Although we have alleviated this problem by defining a way to create generic plans that can be instantiated for different equipments, the production of new plans and modification of existing plans can be a time-consuming activity.



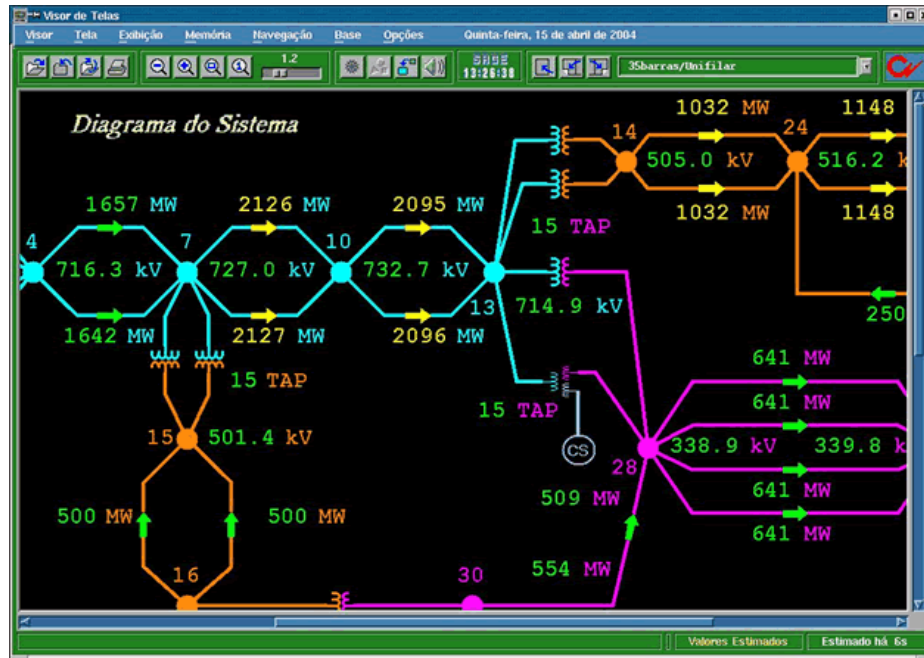


Figura 6: Topological Screen of SAGE SCADA

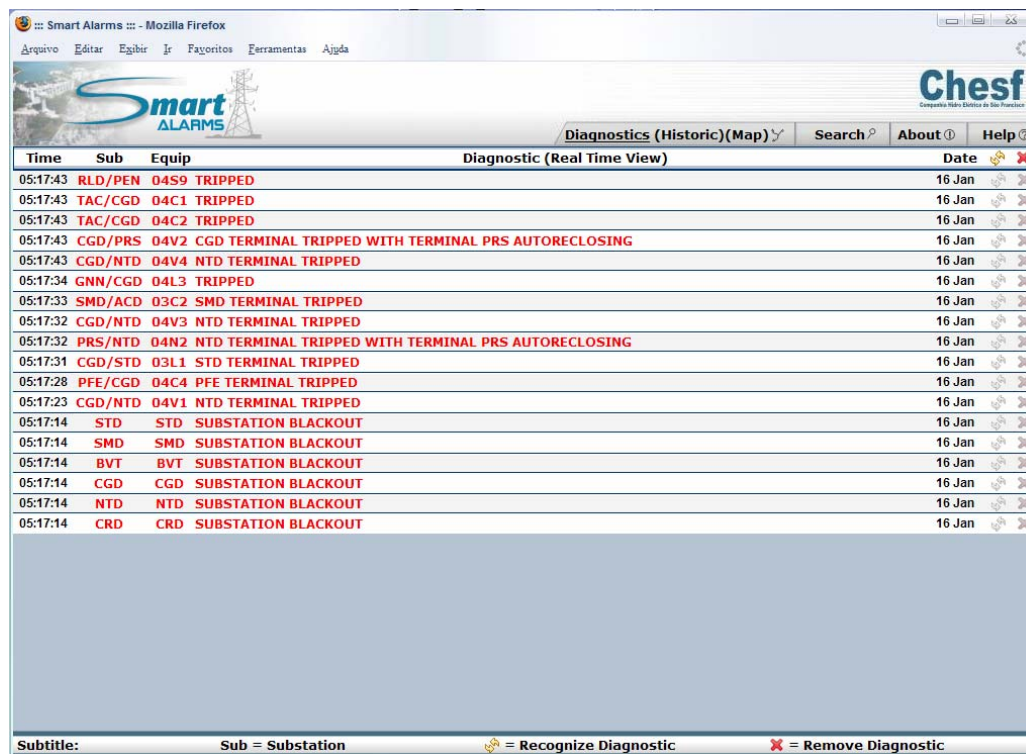


Figure 7: SmartAlarm Screen with a Diagnostic of an Occurrence

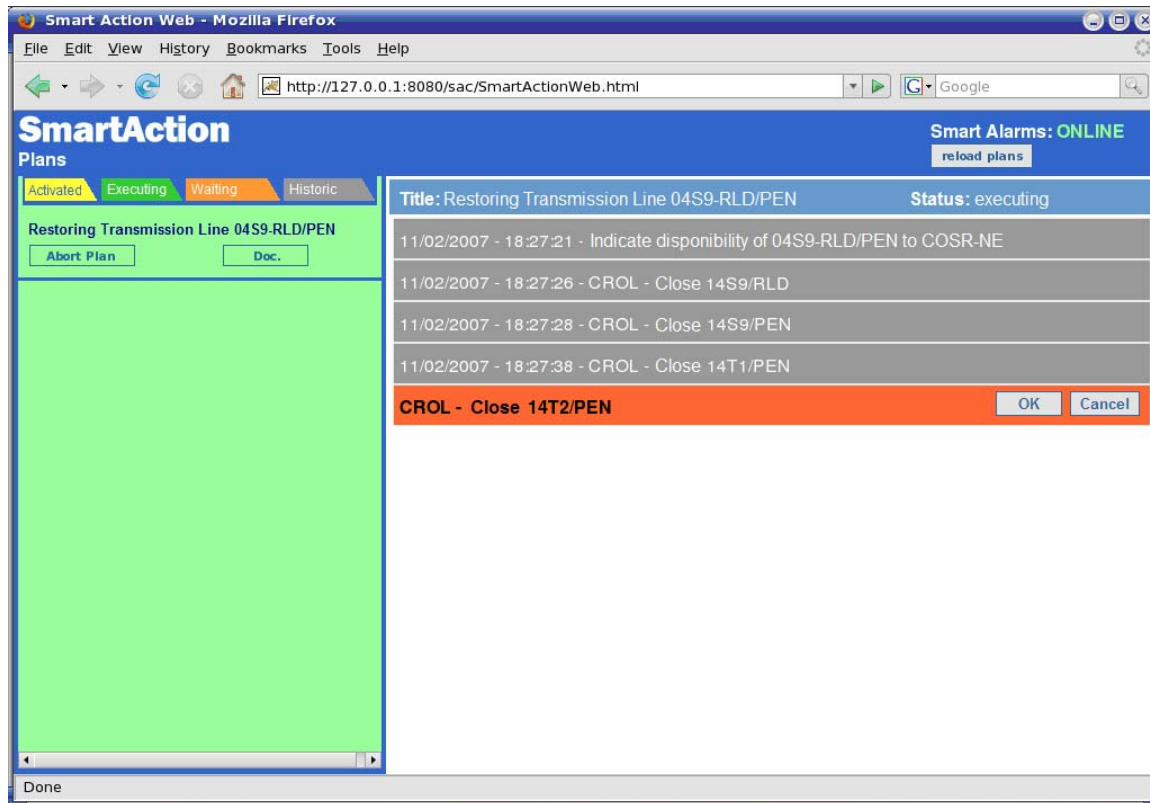


Figure 8: Smart Action Screen with a restoration plan being executed

## 7. References

- 1 Teo , C. Y. and Shen W., Development of an Interactive Rule-Based System for Bulk Power System Restoration, IEEE Transactions On Power Systems, Vol. 15, No. 2, May 2000.
- 2 Kostic, T. Cherkaoui, R., Germond, A., Pruvot, P., Decision Aid Function For Restoration Of Transmission Power Systems: Conceptual Design and Real Time Considerations, IEEE Transactions on Power Systems, Vol. 13, No. 3, August 1998.
- 3 Adibi, M., Power System Restoration: Methodologies and Implementation Strategies, Wiley-IEEE Press, 2000.
- 4 Naggar, R. Cauchon, L. Henault, S. Si Truc Phan, The RECRE System: Restoration Planning For Hydro-Quebec's Power System, Power Engineering Society General Meeting, 2006.
- 5 Sauvé, J.; Rocha Neto,E.; Cirne, W.; Duarte,A.; Figueiredo,J.; Sampaio, M.; Nicolletti, P.; Araújo, A.; Melo, M., Maintenance Techniques For An Intelligent Alarm Processing System. Engineering Intelligent Systems, v. 13, n. 4, p. 213-222, 2005.
- 6 www.epriots.com. Last visit in june/2007.
- 7 http://easyaccept.org/