UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

CENTRO DE CIÊNCIAS E TECNOLOGIA

CURSO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UMA FERRAMENTA ROBUSTA DE TRATAMENTO DE EVENTOS EM REDES ELÉTRICAS

ELOI ROCHA NETO

Campina Grande

Fevereiro – 2004

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

CENTRO DE CIÊNCIAS E TECNOLOGIA

CURSO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UMA FERRAMENTA ROBUSTA DE TRATAMENTO DE EVENTOS EM REDES ELÉTRICAS

ELOI ROCHA NETO

Dissertação submetida à Coordenação de Pós-Graduação em Informática do Centro de Ciências e Tecnologia da Universidade Federal de Campina Grande como requisito parcial para a obtenção do grau de Mestre em Ciências (MSc).

Área de Concentração: Ciência da Computação

Linhas de Pesquisa: Redes de Computadores e Sistemas Distribuídos

Sistemas de Informação e Banco de Dados

Orientadores: Jacques Philippe Sauvé

Marcus Costa Sampaio

Campina Grande

Fevereiro – 2004

FICHA CATALOGRÁFICA

ROCHA NETO, Eloi

R672F

Uma Ferramenta Robusta para o Tratamento de Eventos em Redes Elétricas

Dissertação (mestrado), Universidade Federal de Campina Grande, Centro de Ciências e Tecnologia, Coordenação de Pós-Graduação em Informática, Campina Grande, Paraíba, Fevereiro de 2004.

141 p. Il.

Orientadores: Jacques Philippe Sauvé
Marcus Costa Sampaio

Palavras-chave:
1. Sistemas de Potência
2. Correlação de Eventos Robusta
3. Redes Elétricas
4. Ruído

CDU – 621.316.91

"UMA FERRAMENTA ROBUSTA DE TRATAMENTO DE EVENTOS EM REDES ELÉTRICAS"

ELOI ROCHA NETO

DISSERTAÇÃO APROVADA EM 20.02.2004

PROF. JACQUES PHILIPPE SAUVÉ, Ph.D
Orientador

PROF. MARCUS COSTA SAMPAIO, Dr.
Orientador

PROF. WASHINGTON LUIZ ARAÚJO NEVES, Ph.D
Examinador

PROF. MANOEL AFONSO DE CARVALHO JÚNIOR, Ph.D
Examinador

CAMPINA GRANDE – PB

Aos grandes amores de minha vida:

minha mãe, meu pai e minha namorada.

Agradecimentos

A Deus, por tudo.

Aos meus orientadores, Jacques e Marcus Sampaio, pela confiança de que poderíamos realizar um bom trabalho;

A toda a minha família, por apoiar e incentivar este trabalho, especialmente a minha mãe e meu pai;

A minha namorada Jordana pelo seu amor, carinho e compreensão;

À CAPES, pelo apoio financeiro durante o desenvolvimento desta dissertação;

À CHESF, por incentivar e patrocinar este projeto, especialmente a Sérgio e Socorro, pela colaboração e participação ativa no trabalho;

A toda a equipe do projeto *Smart Alarms* (Walfredo, Jacques, Marcus, Jorge, Alexandre e Michael) por serem uma ótima equipe e por contribuírem com tudo o que foi preciso para que os objetivos deste trabalho fossem atingidos;

Aos meus amigos e amigas, pelo apoio e imensa torcida;

Aos professores e colegas do DSC, pelos valorosos conhecimentos compartilhados durante nossa convivência.

Resumo

Um dos principais problemas encontrados nos centros de supervisão e controle das redes de transmissão e distribuição de energia elétrica consiste na grande quantidade de dados a serem monitorados. Além disso, a dimensão e a complexidade inerente a estas redes tornam esta atividade uma tarefa árdua. Para complicar ainda mais a realização da tarefa, eventos relevantes para o diagnóstico de problemas podem conter ruído, isto é, podem ser perdidos ou gerados espuriamente. Surge portanto a necessidade de ferramentas robustas que considerem, durante o seu processamento, a existência de ruído para auxiliar os operadores destas redes na tomada de decisões. Este trabalho de mestrado teve como objetivo o de desenvolver uma ferramenta para o diagnóstico de falhas em sistemas elétricos, que utilize uma técnica robusta de correlação de eventos, e implantá-la no Centro Regional de Operação Leste da CHESF.

Palavras-chave: correlação de eventos robusta, redes elétricas, ruído

Abstract

One of the most important problems that have been found in power supervision and control centers is a huge mass of data to be monitored. Moreover, the dimension and the complexity of transmission networks make the monitoring task very hard. To complicate matters still more, data can contain noise, in other words, data can be lost or generated spuriously. To cope with noise, robust tools may be considered in order to help operators in power supervision and control centers. This thesis aims to develop a tool for fault diagnosis in electrical systems, that uses a robust event correlation technique, and deploy it in CHESF's Eastern Regional Operations Center.

Keywords: robust event correlation, electrical systems, noise

Sumário

Lista de Siglas

ALR – Alarme

ANEEL – Agência Nacional de Energia Elétrica

CEPEL – Centro de Pesquisas de Energia Elétrica

CHESF – Companhia Hidro Elétrica do São Francisco

CROL – Centro Regional de Operação Leste

DSC – Departamento de Sistemas e Computação

EMS – Energy Management System

FOE – Fase Operacional Experimental

SAGE – Sistema Aberto de Gerência de Energia

SCADA – *Supervisory Control and Data Acquisition*

SDE – Seqüência de Eventos

UFCG – Universidade Federal de Campina Grande

TC – Transformador de Corrente

UTR – Unidade Terminal Remota

Lista de Tabelas

Lista de Figuras

1. Introdução

Os centros de supervisão e controle de redes de transmissão e distribuição de energia elétrica vêm, constantemente, modernizando-se nestas duas últimas décadas. Entretanto, gerenciá-las ainda é uma tarefa árdua; entre as principais razões destacam-se a vasta dimensão geográfica e a complexidade inerente a essas redes. É bem verdade que tais centros dispõem de sistemas computacionais de supervisão, que disponibilizam aos operadores, em tempo real, um conjunto de informações sobre o estado de uma rede, facilitando assim o diagnóstico e a localização de anormalidade na mesma. No entanto, em grandes ocorrências em um sistema elétrico, é muito grande a quantidade de informações disponibilizadas aos operadores por esses sistemas. No dia 10 de outubro de 2002, por exemplo, os operadores receberam, em menos de trinta minutos, mais de cinco mil informações sobre o estado dos equipamentos da rede de transmissão de energia elétrica da Companhia Hidro Elétrica do São Francisco (CHESF).

O grande volume de informações recebidas pelos operadores em situações críticas, muitas vezes, é resultado de um efeito cascata, originado por uma falha — a causa raiz do problema — em um equipamento da rede. Descobrir a causa raiz pode demandar muita análise por parte dos operadores. Convém ressaltar que, nesses momentos críticos, é muito pequeno o tempo disponível aos operadores para tomar as medidas necessárias para a correção do problema, uma vez que, quanto mais tempo uma rede deixar de estar disponível, maior será a insatisfação dos que dela dependem. Além disso, em situações de estresse, operadores podem cometer erros, agravando ainda mais o problema.

Um outro fator extremamente prejudicial para a análise dos operadores está relacionado com o **ruído** nas informações recuperadas de uma rede. Neste contexto, o ruído pode ser uma informação perdida ou gerada espuriamente. Desta forma, além de dispor de um curto espaço de tempo para analisar a grande quantidade de informações — tempo envolve dinheiro —, um operador tem que estar ciente de que muitas das informações em análise podem estar erradas e de que informações importantes para o diagnóstico de problemas podem ter sido perdidas.

A existência de ruído nas informações recuperadas de uma rede é sempre um sério problema. Em momentos críticos, operadores precisam estar em contato com subestações remotas para confirmar a veracidade das informações oriundas dessas subestações. Esse procedimento é fundamental, pois uma ação corretiva equivocada pode danificar um equipamento ou propagar os efeitos de uma falha localizada para outras partes do sistema.

Diante da grande quantidade de informações, do curto espaço de tempo para a realização da análise e da possibilidade de informações com ruído, surge a necessidade de uma solução computacional para auxiliar os operadores de redes elétricas durante grandes ocorrências. Esta solução computacional deve analisar todas as informações recuperadas da rede, considerando a existência de ruído, e informar aos operadores apenas os diagnósticos dos problemas. Desta forma, a quantidade de informações tratadas pelos operadores é reduzida, assim como o tempo de análise e a probabilidade de erros.

À guisa de facilitar a leitura, introduzimos informalmente alguns termos básicos, relacionados com redes elétricas de modo geral. Um **evento** consiste em uma informação recuperada de uma rede elétrica podendo indicar uma situação anormal, possivelmente causada por uma **falha**, isto é, por um problema existente em algum elemento da rede. Em geral, uma falha pode gerar vários eventos. **Correlação de eventos** consiste em recuperar um conjunto de eventos, correlacioná-los com o objetivo de detectar a falha e, no final, emitir um **diagnóstico** da falha ao operador da rede. Uma técnica de correlação de eventos é dita ser **robusta** quando ela considera a existência de informações com ruído, isto é, eventos perdidos ou espúrios. Um

alarme consiste em um evento indicando uma situação de advertência ou de urgência sobre o estado de um equipamento de uma rede elétrica. Na seqüência, utilizaremos apenas o termo **evento**, uma vez que este é mais abrangente.

Devido à importância de correlação de eventos robusta para o diagnóstico de falhas em redes de modo geral, várias técnicas têm sido desenvolvidas. Entre elas: raciocínio baseado em casos (LEWIS, 1999), redes neurais artificiais (BIELER, 1994), lógica nebulosa (LEE, 2000), redes de Bayes (GÜRER, 1996) e *codebooks* (KLIGER, 1995). Apesar de algumas delas já terem sido de algum modo usadas no âmbito de redes elétricas (por exemplo, lógica nebulosa e redes neurais artificiais), nenhuma delas, a nosso ver, é plenamente eficaz para o diagnóstico de falhas em redes elétricas.

Devido à importância de correlação robusta de eventos para o diagnóstico de falhas em redes de modo geral, várias técnicas têm sido desenvolvidas. Entre elas: redes neurais artificiais (BIELER, 1994), lógica nebulosa (LEE, 2000), redes de Bayes (GÜRER, 1996) e *codebooks* (KLIGER, 1995). Duas delas têm sido usadas no âmbito de redes elétricas (lógica nebulosa e redes neurais artificiais).

A principal dificuldade de usar a lógica nebulosa em supervisão de redes elétricas está relacionada com a definição de variáveis lingüísticas e de seus respectivos graus de pertinência.

Apesar de a técnica redes de Bayes ser bastante poderosa para o tratamento de ruído, o seu uso em redes elétricas é dificultado pela complexidade de estimar valores probabilísticos.

O uso de redes neurais no contexto de redes elétricas é muito difícil na prática, devido à complexidade de preparar bases históricas e volumosas de eventos para o treinamento das redes neurais.

A grande vantagem da técnica *codebooks* consiste na sua capacidade de tratar ruído. No entanto, a técnica só mostra-se eficiente no tratamento de falhas que podem ser codificadas como uma conjunção de condições. Infelizmente, é comum em sistemas elétricos a representação de falhas envolvendo tanto conjunção como

disjunção de condições. A disjunção pode tornar combinatória a complexidade da geração e verificação dos códigos, o que inviabilizaria o uso da técnica.

Diante do exposto, faz-se necessário uma nova abordagem para a supervisão de redes elétricas em presença de ruído.

O presente trabalho está inserido em um projeto de P&D, intitulado *Smart Alarms*, entre a UFCG (Universidade Federal de Campina Grande) e a CHESF, sendo financiado por esta última, com o apoio da Agência Nacional de Energia Elétrica (ANEEL).

A CHESF é uma empresa cuja atuação envolve todo o Nordeste do Brasil, gerando e distribuindo energia elétrica. O sistema responsável pelo controle e supervisão do processo de geração e transmissão de energia elétrica chama-se Sistema Aberto de Gerenciamento de Energia (SAGE). O SAGE é um sistema do tipo SCADA (*Supervisory Control and Data Acquisition*) / EMS (*Energy Management System*), baseado em uma arquitetura distribuída e redundante, e organizado em torno de um *software* gerente de banco de dados em tempo real. O SAGE foi desenvolvido pelo Centro de Pesquisas de Energia Elétrica (CEPEL) do Ministério de Minas e Energia, sendo também usado por outras empresas do sistema brasileiro de distribuição de energia (SILVA, 1998).

O objetivo do projeto *Smart Alarms* é a construção de uma ferramenta robusta para o tratamento em tempo real de eventos na rede de transmissão de energia elétrica da CHESF, e integrá-la ao SAGE. Entre os artefatos já produzidos, destaca-se uma ferramenta de correlação de eventos (DUARTE, 2003), intitulada *SmartOne*, que utiliza uma técnica híbrida constituída de raciocínio baseado em regras e de raciocínio baseado em modelos. Entretanto, esta ferramenta não considera, durante seu processamento, a existência de informações com ruído.

1.1. Objetivos da dissertação

Esta dissertação tem como objetivo o de estender o *SmartOne* para uma ferramenta de diagnóstico de falhas — *Robust SmartOne* — que considere, durante o seu processamento, a existência de ruído. Associado a este objetivo, está o de

desenvolver e o de implementar uma técnica robusta de correlação de eventos para o diagnóstico de falhas em redes elétricas.

A ferramenta robusta está sendo implantada no Centro Regional de Operação Leste da CHESF (CROL / CHESF).

1.2. Estrutura da dissertação

A dissertação está dividida em seis capítulos, sendo o capítulo 1 esta introdução.

No capítulo 2, apresentamos alguns exemplos de eventos com ruído encontrados em sistemas de supervisão de redes elétricas, assim como algumas das possíveis razões que explicam a sua existência. No final, estatísticas relacionadas com a presença de ruído, no sistema de supervisão utilizado pela CHESF, são apresentadas.

As principais técnicas de correlação de eventos robustas encontradas na literatura são temas do capítulo 3. Um estudo da aplicabilidade destas técnicas para o problema em estudo também é parte do capítulo.

No capítulo 4, descrevemos nossa ferramenta robusta de tratamento de eventos em redes elétricas. Em maiores detalhes, apresentamos os requisitos que nortearam o desenvolvimento da ferramenta, a técnica robusta de correlação de eventos que ela utiliza e o seu projeto arquitetural.

No capítulo 5, apresentaremos como foi realizada a implementação da ferramenta.

No capítulo 6, mostraremos como foi realizada a validação da ferramenta. Os testes contemplam alguns resultados práticos do uso da ferramenta no CROL / CHESF.

O capítulo 7 fecha o documento, com as conclusões e propostas de trabalhos futuros.

Além destes capítulos, este documento contém um apêndice destinado a leitores que não possuem familiaridade com termos relacionados com sistemas de

potência de modo geral. Nele apresentamos alguns tipos de equipamentos e de arranjos comumente encontrados em redes elétricas.

2. Caracterização dos problemas de ruído em sistemas de supervisão de redes elétricas

Neste capítulo, apresentamos alguns exemplos de ruído que podem ser encontrados em sistemas de supervisão de redes elétricas, assim como algumas das possíveis razões que explicam a sua existência. No final, são apresentadas estatísticas relacionadas com a presença de ruído, em um sistema particular de supervisão de redes elétricas.

Caso o leitor não tenha familiaridade com termos relacionados com sistemas elétricos de modo geral, aconselhamos fortemente a leitura do apêndice que segue esta dissertação.

2.1. Tipos de ruído em sistemas de supervisão de redes elétricas

Sistemas de supervisão de redes elétricas são ferramentas de grande importância para os operadores dessas redes. Tais sistemas atuam tanto ao disponibilizar informações consideráveis para a análise de problemas quanto na reparação deles.

Infelizmente, nem sempre as informações disponibilizadas pelos sistemas de supervisão são confiáveis, isto é, tanto eventos podem ser perdidos como gerados espuriamente. O fato é que qualquer evento de um sistema elétrico pode estar com ruído; por exemplo, um evento relacionado com a abertura de um disjuntor não é uma garantia de que o disjuntor esteja aberto, o mesmo valendo para o fechamento de uma chave, ou a atuação de uma proteção, etc.

Para uma compreensão mais detalhada do assunto, alguns dos possíveis tipos de ruído que podem ser encontrados em sistemas de supervisão de redes elétricas são listados a seguir:

1) Eventos espúrios sinalizando a abertura (fechamento) de disjuntores que já estavam abertos (fechados).
2) Eventos espúrios sinalizando a abertura (fechamento) de chaves que já estavam abertas (fechadas).
3) Eventos espúrios sinalizando a abertura e o fechamento de um disjuntor ou de uma chave em um mesmo instante de tempo.
4) Sinalização de eventos espúrios relacionados com equipamentos inexistentes.
5) Eventos perdidos sinalizando a abertura ou o fechamento de disjuntores e chaves.
6) Eventos sinalizados com grande período de atraso.
7) Eventos perdidos ou espúrios de atuação da proteção.
8) Grandezas Elétricas (tensão, potência, reatância, corrente) de um determinado equipamento apresentando valores incorretos.

Compreender as razões que justificam a presença de ruído em um determinado equipamento não é uma tarefa trivial. Muitas vezes, só é possível descobrir que um evento associado a um equipamento é espúrio ou foi perdido, conversando-se com operadores localizados próximos ao equipamento.

Para facilitar o entendimento dos possíveis motivos responsáveis pela existência de ruído nos eventos recuperados do sistema elétrico, explicaremos, brevemente, como funciona o processo de recuperação destes eventos. Cada subestação de energia elétrica é monitorada por uma unidade terminal remota (UTR) de aquisição de dados. Estas coletam todas as informações relacionadas com os equipamentos da subestação e enviam-nas para o sistema de supervisão da rede elétrica, que as disponibilizam para os operadores na forma de eventos. Para que cada mudança de estado de um equipamento de uma subestação seja coletada por uma

UTR, cada equipamento possui um ponto de supervisão, que consiste em um relé de sinalização, responsável por informar à UTR tais mudanças. Assim, sempre que um disjuntor supervisionado abre, seu relé de sinalização informa à UTR a abertura do disjuntor, e esta, por sua vez, ao sistema de supervisão (veja a Figura 2.1).

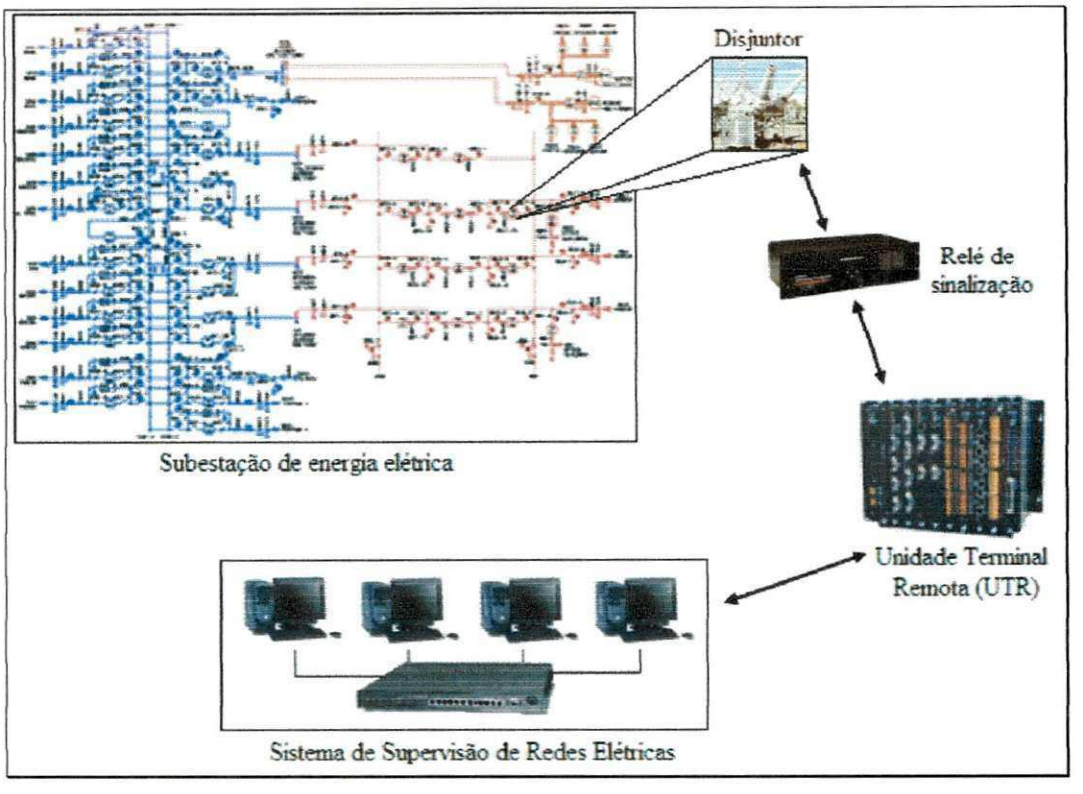

Figura 2.1 – Processo de recuperação de eventos no sistema elétrico

As principais razões que justificam a presença de ruído são:

1) Problemas nos pontos de supervisão dos equipamentos – Quando um ponto de supervisão apresenta problemas, eventos podem ser tanto perdidos como gerados espuriamente. Considere que um disjuntor se abre, e que há um problema em seu ponto de supervisão: neste caso, vários tipos de ruído podem ocorrer, entre eles, o evento de abertura pode ser perdido, ou um evento de fechamento pode ser gerado espuriamente, ou mesmo, vários eventos de abertura e fechamento podem ser gerados espuriamente.

2) Equipamentos sem supervisão – Infelizmente, nem todos os equipamentos em um sistema elétrico são supervisionados; a

conseqüência é que qualquer evento relacionado com um equipamento sem supervisão não poderá ser percebido pelo operador da rede. Isso caracteriza um evento perdido, uma vez que mesmo ocorrendo uma alteração no estado do equipamento, nenhum evento será sinalizado. Chave é um exemplo de equipamento que nem sempre é supervisionado. Assim, toda vez que uma chave sem supervisão for aberta ou fechada, o operador não receberá nenhum evento sinalizando sua mudança de estado. Outro tipo de informação que nem sempre é supervisionada nas redes elétricas são as grandezas elétricas de determinados equipamentos. Nestas circunstâncias, o operador é incapaz de saber, por exemplo, se o equipamento está ou não energizado. Desta forma, a análise fica restrita aos eventos relacionados com os equipamentos, que podem conter ruído.

3) Unidades terminais remotas com problemas – Se uma remota estiver mal configurada ou com outros problemas, vários eventos poderão deixar de ser sinalizados, como também poderão ser gerados espuriamente.

4) Linhas cujo transformador de corrente (TC) localiza-se na bucha do disjuntor – Quando o TC de um terminal de uma linha localiza-se na bucha do disjuntor, toda vez que o disjuntor é *bypassado*, o TC fica isolado e, por conseguinte, as grandezas elétricas informadas aos operadores da rede são iguais a zero. Quando ocorre uma situação deste tipo, o operador é incapaz de saber se a linha está energizada, fazendo-se necessário entrar em contato com o operador da subestação remota, com o intuito de obter tal informação.

5) Canal de comunicação obstruído – Caso o canal de comunicação que liga os equipamentos de uma subestação a sua remota ou o canal que liga as remotas ao sistema de supervisão apresentem problemas, eventos importantes para a análise de problemas deixarão de ser sinalizados.

Apesar de qualquer equipamento estar susceptível a eventos com ruído, podemos afirmar que os eventos relacionados com a abertura e com o fechamento de disjuntores e chaves são os mais vulneráveis à presença deles. Esta vulnerabilidade advém do fato de que disjuntores e chaves representam a maioria absoluta dos equipamentos de um sistema elétrico. Além disso, qualquer manobra (ação realizada pelo operador ou pelo mecanismo de proteção do sistema elétrico), seja ela manual seja automática, baseia-se na abertura e no fechamento de disjuntores ou de chaves. Considerando o fato de que todos os procedimentos que ocorrem no sistema elétrico consistem em manobras, podemos concluir que a maior parte dos eventos com ruído estão associados à abertura e ao fechamento de disjuntores e chaves.

Feitas estas considerações, concentramo-nos no tratamento de eventos com ruído relacionados com a abertura e o fechamento de disjuntores e chaves.

2.2. Ruído em sistemas de supervisão de redes elétricas

Exemplos ilustrativos de ruído em sistemas de supervisão de redes elétricas aparecem nas quatro subseções seguintes, respectivamente para cada tipo de manobra que pode ocorrer em um sistema elétrico. Mais precisamente, os exemplos são sobre manobras e os tipos de ruído que podem estar associados a elas.

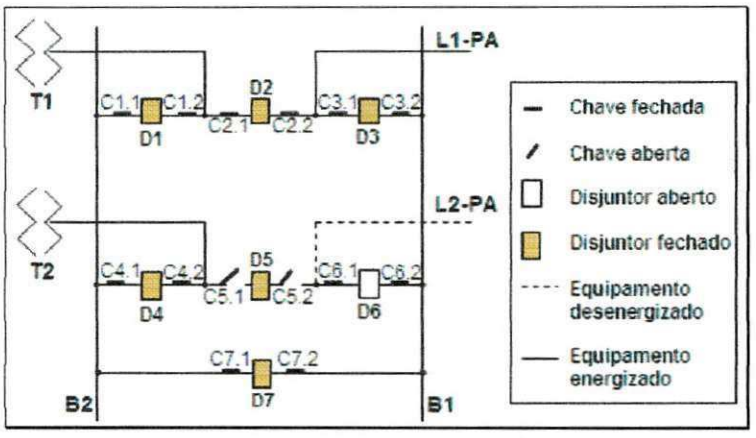

Figura 2.2 – Exemplo de um arranjo

Para facilitar a compreensão dos exemplos, a Figura 2.2 mostra alguns elementos básicos de um arranjo de proteção de um sistema elétrico. O arranjo representa uma parte de uma subestação, composta por dois barramentos (B1 e B2), duas linhas de transmissão (L1-PA e L2-PA – lado PARA), dois transformadores (T1

e T2), alguns disjuntores (representados por caixas batizadas de Di) e algumas chaves (representadas por pequenas barras rotuladas com Ci.j). Um disjuntor fechado é representado por uma caixa marrom, enquanto uma caixa vazia é usada para representar um disjuntor aberto. Uma chave fechada é representada por uma pequena barra horizontal. Uma pequena barra inclinada indica uma chave aberta. Uma linha, um transformador e um barramento, energizados, são indicados por linhas cheias, caso contrário, as linhas são tracejadas.

Na figura, a linha L1 (energizada) está conectada ao barramento B1 pelo disjuntor D3 — que está fechado — e ao barramento B2 pelos disjuntores 1 e 2 — que estão fechados. No entanto, a linha L2 (desenergizada) não está conectada a nenhum dos barramentos, uma vez que tanto a chave C5.2 como o disjuntor D6 estão abertos (respectivamente, desconectando o barramento B2 e o barramento B1).

Cada exemplo que se segue explica uma seqüência possível de estados de equipamentos de um arranjo elétrico.

2.2.1. Ruído em uma manobra de desarme

Um sistema elétrico geralmente é composto por vários equipamentos de valores aquisitivos bastante elevados. Além disso, esses equipamentos são continuamente energizados em determinados níveis de tensão. Se forem submetidos a um nível de tensão superior ao indicado, eles poderão sofrer sérios danos. Para proteger estes equipamentos, o sistema elétrico possui mecanismos de proteção construídos especificamente com este propósito. Desta forma, para evitar que a sobretensão em uma linha danifique outros equipamentos, tais como transformadores, os mecanismos de proteção atuam abrindo os disjuntores necessários para isolar o problema, de forma que, no final, a linha será desenergizada e a sobretensão não afetará nenhum equipamento.

Considere a Figura 2.3, que ilustra um arranjo típico de equipamentos de 500kV. Suponha que a linha L1, em um determinado momento, apresente uma anomalia e sua tensão chegue a 540kV. Note que o terminal da linha L1 está conectado aos dois barramentos e aos dois transformadores e que, nestas condições, uma sobretensão poderia, no pior caso, queimar os dois transformadores. Nestas

circunstâncias, os mecanismos de proteção do sistema elétrico atuam e os disjuntores D2 e D3 abrem, desenergizando a linha no terminal e protegendo os outros equipamentos da sobretensão (observe a Figura 2.4).

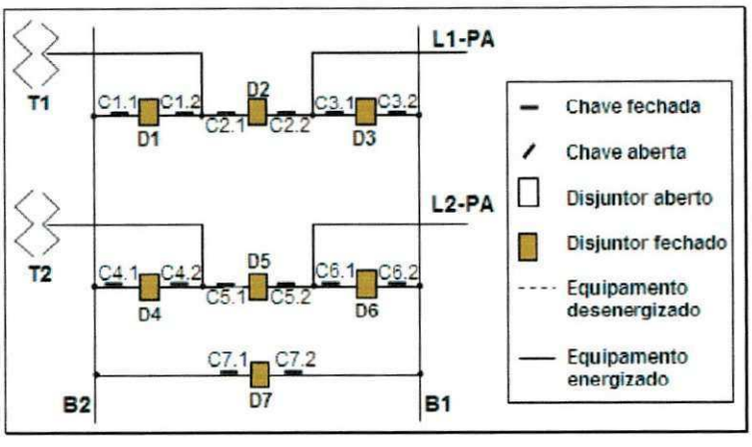

Figura 2.3 - Estado dos equipamentos antes do desarme

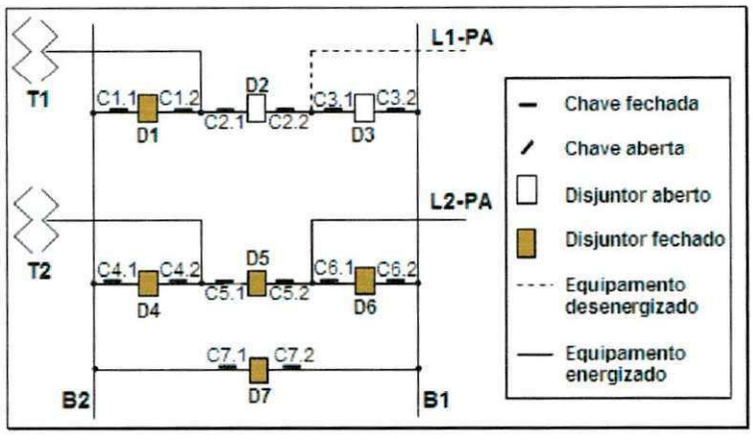

Figura 2.4 – Estado dos equipamentos após a atuação da proteção

Em situações ideais, quando ocorre um cenário desta natureza, pelo menos três eventos devem ser sinalizados ao operador da rede: um sinalizando a abertura do disjuntor D2, outro sinalizando a abertura do disjuntor D3, e outro a atuação da proteção da linha L1 no terminal. Além disso, o operador percebe, com base nos valores das grandezas elétricas apresentadas em seu terminal, que a linha está desenergizada.

Infelizmente, nem sempre as situações são ideais. Ainda com relação ao cenário descrito, pode acontecer que alguns dos eventos não sejam sinalizados, da

mesma forma que podem ser exibidos ao operador outros eventos que não representem bem a realidade do sistema elétrico.

A Figura 2.5 é o mesmo arranjo da Figura 2.3 e da Figura 2.4, e ilustra com mais detalhes a existência de ruído. Suponha que, após a atuação da proteção no terminal da linha L1, apenas dois eventos sejam sinalizados, um sinalizando abertura do disjuntor D3 e o outro, a atuação da proteção, enquanto o evento sinalizando a abertura do disjuntor D2 tenha se perdido. Analisando a figura, podemos observar o suposto estado dos equipamentos após a chegada dos eventos, assim como a presença de algumas inconsistências geradas devido à existência de ruído. Por exemplo: a linha L1 está desenergizada (tracejada), porém está conectada tanto ao barramento 2 como ao transformador 1, que estão energizados. Como podemos ver, a presença de ruído é extremamente prejudicial à análise do operador, uma vez que, ao mesmo tempo, temos um evento sinalizando a atuação da proteção e uma linha conectada a equipamentos energizados, o que é uma incompatibilidade.

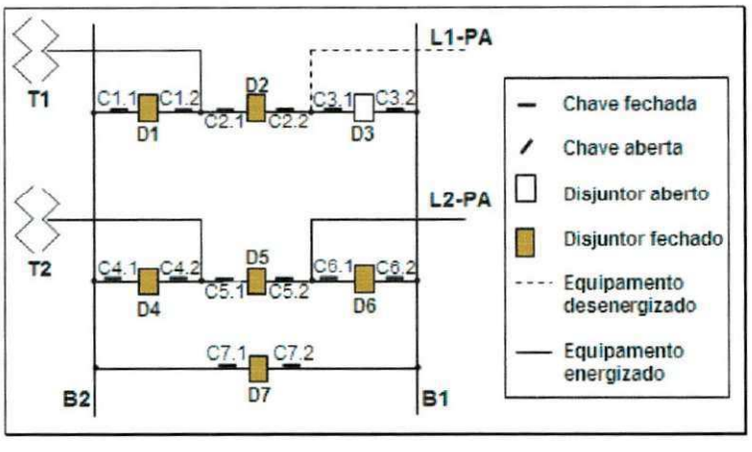

Figura 2.5 – Estado dos equipamentos após a atuação proteção (evento de abertura do disjuntor D2 foi perdido)

Para dificultar ainda mais a interpretação do operador, é possível que seja erroneamente sinalizado que o disjuntor D3 da Figura 2.5, diante de um desarme, abra e feche várias vezes em um mesmo instante de tempo, o que na prática não é possível. Nestas condições, o operador precisa entrar em contato com o operador da subestação remota para saber o estado corrente do disjuntor D3.

2.2.2. Ruído em uma manobra de desligamento

Muitas vezes, um equipamento do sistema elétrico precisa estar desenergizado para poder receber manutenção. O procedimento utilizado para desenergizar um equipamento consiste em abrir os disjuntores necessários para isolá-lo. Observe a Figura 2.6. Suponha que o terminal do lado PARA da linha L1 precisar passar por uma manutenção: o procedimento baseia-se em abrir o disjuntor D1. Neste contexto, a manobra de desligamento da linha consistiu no ato de abrir o disjuntor D1 (Figura 2.7).

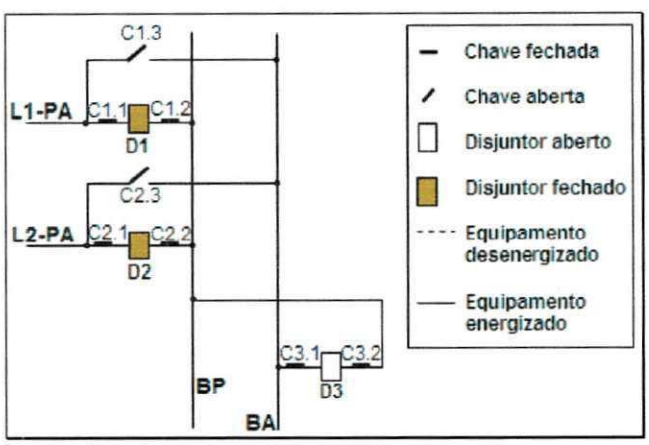

Figura 2.6 – Estado dos equipamentos antes do desligamento da linha L1

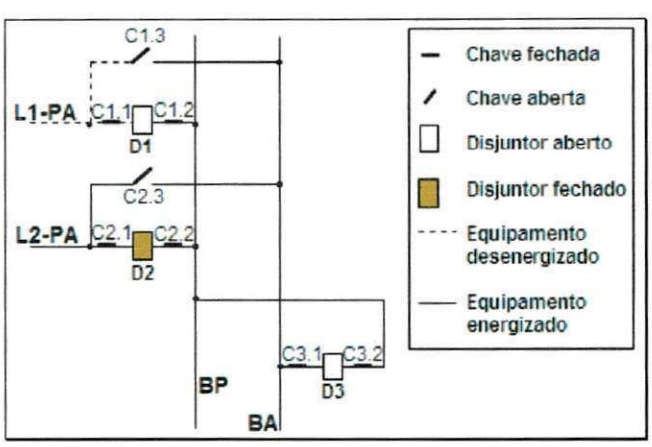

Figura 2.7 – Estado dos equipamentos após o desligamento da linha L1

Do ponto de vista do operador, em situações que uma linha é desligada, pelo menos um evento, informando que o disjuntor da linha foi aberto, deve ser sinalizado. Entretanto, é possível que a linha seja desligada sem que nenhum evento seja sinalizado. Neste caso o evento informando que o disjuntor da linha abriu foi perdido.

A Figura 2.8 ilustra o suposto estado dos equipamentos após o desligamento da linha na presença de eventos com ruído. Note que podemos detectar uma inconsistência, uma vez que a linha está desenergizada, apesar de estar conectada ao barramento BP, que está energizado.

Além da possibilidade de o evento ser perdido, é possível que ele só seja sinalizado muito tempo depois, o que caracteriza um evento atrasado. Analisando-se mais de perto este tipo de evento, trata-se de uma situação que apresenta dois eventos com ruído. O primeiro é que, não sendo sinalizado no momento certo, comporta-se como um evento perdido; o segundo é que, quando o evento finalmente aparece, é considerado como um evento espúrio.

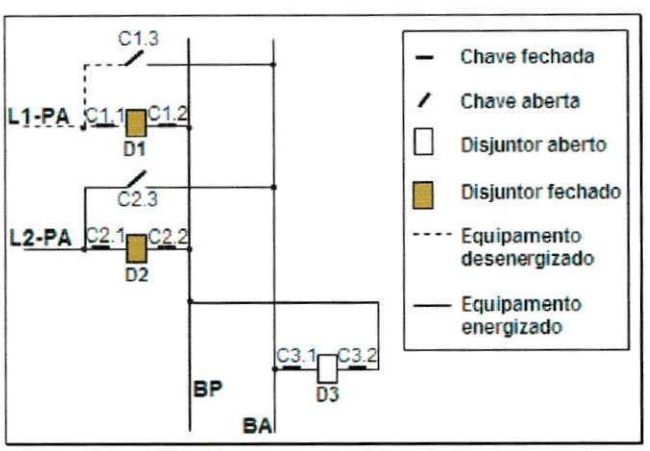

Figura 2.8 – Estado dos equipamentos após o desligamento da linha L1 (evento de abertura do disjuntor D1 foi perdido)

2.2.3. Ruído em uma manobra de religamento

Uma manobra de religamento consiste em reenergizar um equipamento que foi desenergizado devido a um desligamento ou a um desarme. O procedimento baseia-se em fechar os disjuntores necessários para que o equipamento fique conectado a pelo menos um equipamento energizado. A Figura 2.7 pode ser utilizada para ilustrar o estado dos equipamentos antes do religamento.

Em condições normais, quando um equipamento é religado, o operador recebe em seu terminal de operação pelo menos duas informações: a primeira consiste em um evento sinalizando o fechamento do disjuntor; a outra é representada pela

alteração dos valores das grandezas elétricas, indicando que o equipamento foi energizado.

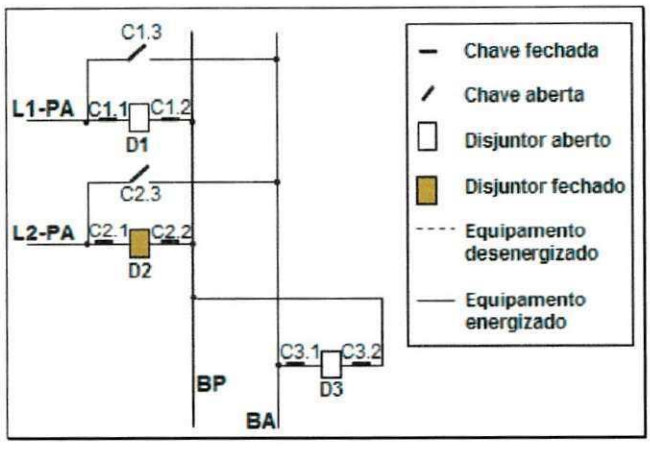

Figura 2.9 – Estado dos equipamentos após o religamento (evento de fechamento do disjuntor D1 foi perdido)

Infelizmente, eventos com ruído podem ocorrer e prejudicar a compreensão do operador sobre o que está ocorrendo no sistema elétrico. A Figura 2.9 apresenta um suposto cenário em que a linha L1 foi religada, apesar de ter sido perdido o evento que sinaliza o fechamento do disjuntor D1. Note que, no final, a linha se encontra energizada sem estar conectada a nenhum equipamento energizado.

2.2.4. Ruído em uma manobra de *bypass*

Assim como quaisquer outros equipamentos do sistema elétrico, disjuntores precisam receber manutenção. O procedimento comumente utilizado para isolar um disjuntor sem comprometer o fornecimento de energia consiste em *bypassar* o disjuntor (manobra de *bypass*). Observe a Figura 2.6 para facilitar a compreensão de uma manobra de *bypass*. Suponha que desejamos efetuar uma manutenção no disjuntor D1. Desta forma, a manobra de *bypass* consiste em fechar a chave de *bypass* (C1.3), em seguida, fechar o disjuntor de transferência (D3) e, finalmente, abrir o disjuntor da linha (D1). Uma vez realizado o *bypass*, as chaves dos disjuntor D1 podem ser abertas com o intuito de isolá-lo (veja a Figura 2.10). Observe que a linha continua energizada, pois tanto o disjuntor D3 como as chaves C1.3, C3.1 e C3.2 estão fechadas. Note que a energia que passava pelo disjuntor D1 passa, agora, pelo disjuntor D3.

Entre todos os tipos de ruído, podemos afirmar que os relacionados com manobras de *bypass* são os mais freqüentes. A principal razão para tal fundamenta-se no fato de que a grande maioria das chaves de *bypass* não são supervisionadas. Assim, toda vez que uma chave deste tipo abre ou fecha, nenhum evento informando a mudança de estado dela é sinalizado.

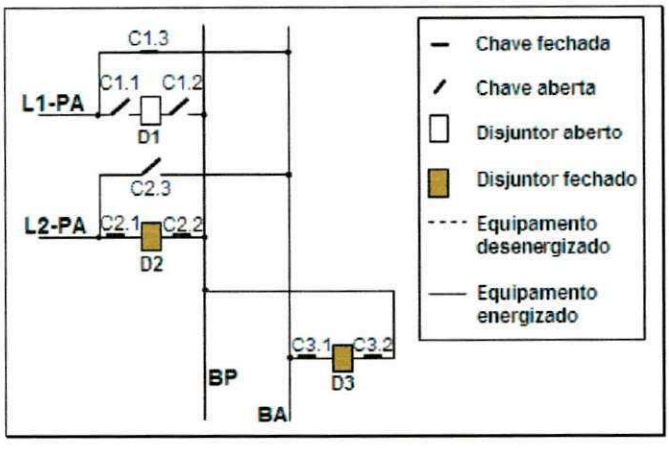

Figura 2.10 – Estado dos equipamento após a realização do *bypass*

A Figura 2.11 ilustra o suposto estado dos equipamentos, após a realização da manobra sem a sinalização da chave de *bypass*. Observe que, no final, temos uma linha energizada sem estar conectada a nenhum equipamento energizado, o que é uma situação anormal.

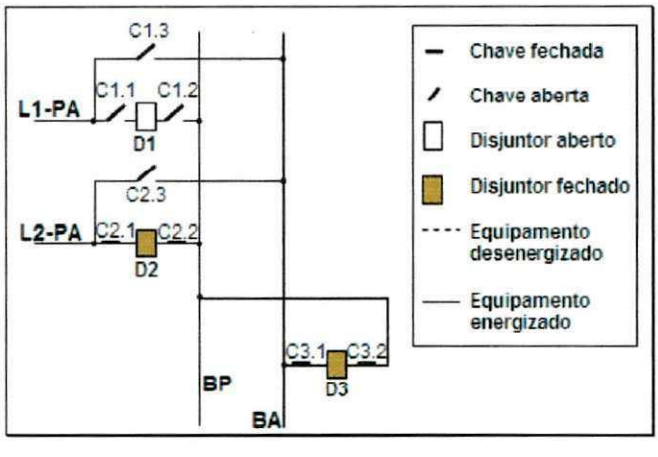

Figura 2.11 – Estado dos equipamentos após a realização da manobra de *bypass*, em uma linha cuja chave de *bypass* não é supervisionada

As conseqüências deste tipo de ruído podem ainda ser mais graves quando a linha não possui suas grandezas elétricas supervisionadas. Neste caso, as informações

disponíveis para o operador consistem basicamente nos eventos que sinalizam a mudança de estado dos disjuntores, que, por sua vez, podem ser perdidos.

Uma outra fonte de ruído ocorre quando o TC da linha localiza-se na bucha do disjuntor. Quando isto ocorre, toda vez que o disjuntor é *bypassado*, as grandezas elétricas zeram. Nestas circunstâncias, o operador, além de não receber todas as informações necessárias para a verificação da realização do *bypass*, também receberá informações incorretas, uma vez que a linha está energizada.

2.3. Freqüência de eventos com ruído em sistemas de supervisão de redes elétricas

Nesta seção, apresentamos algumas estatísticas relacionadas com a presença de eventos com ruído em sistemas de supervisão de redes elétricas. Os dados levantados foram retirados do sistema de supervisão da CHESF — SAGE — durante um período de aproximadamente cinco dias. A Tabela 2.1 resume os intervalos para a análise estatística.

	Início	Fim
Intervalo 1	06/11/2003 - 11:29:32	06/11/2003 - 21:56:59
Intervalo 2	07/11/2003 - 09:47:49	11/11/2003 - 07:45:35
Intervalo 3	11/11/2003 - 14:42:53	13/11/2003 - 11:24:15
Intervalo 4	13/11/2003 - 11:36:18	15/11/2003 - 08:23:10
Intervalo 5	18/11/2003 - 12:03:39	18/11/2003 - 23:59:59

Tabela 2.1 – Intervalos de tempo utilizados para a análise estatística

A Tabela 2.2 trata da freqüência de alguns tipos de ruído, dentro dos intervalos da Tabela 2.1.

Número	Tipo de ruído	Freqüência
1	Eventos perdidos sinalizando a abertura ou o fechamento de disjuntores	14
2	Eventos espúrios sinalizando a abertura (fechamento) de disjuntores que já estavam abertos (fechados)	15
3	Eventos espúrios sinalizando a abertura (fechamento) de chaves que já estavam abertas (fechadas)	12

4	Eventos espúrios sinalizando a abertura e o fechamento de disjuntores em um mesmo instante de tempo	16
5	Eventos espúrios sinalizando a abertura ou o fechamento de disjuntores	36
6	Eventos perdidos sinalizando a abertura ou o fechamento de chaves	12
7	Eventos espúrios sinalizando a abertura ou o fechamento de chaves	5
8	Eventos perdidos sinalizando a abertura ou o fechamento de chaves de *bypass*	59
9	Eventos espúrios sinalizando a abertura ou o fechamento de chaves de *bypass*	15
10	Grandezas elétricas (tensão, potência, reatância, corrente) de um determinado equipamento apresentando valores incorretos	0

Tabela 2.2 – Estatísticas da presença de ruído em sistemas de supervisão de redes elétricas

As freqüências da Tabela 1.2 foram plotadas em um gráfico (Figura 1.12), para efeito de visualização.

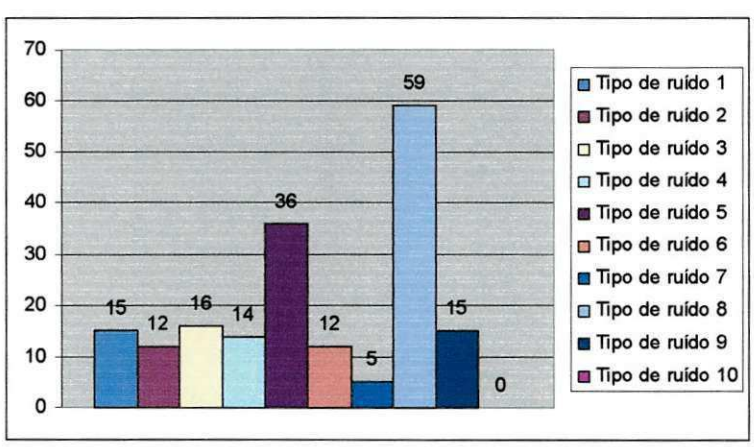

Figura 2.12 – Tipo de ruído X freqüência

Observando o gráfico, podemos notar que eventos com ruído relacionados com chaves de *bypass* são os mais freqüentes. A principal razão para esta exacerbada freqüência deve-se ao fato de que a maior parte das chaves do sistema elétrico CHESF não são supervisionadas. Isto pode ser comprovado ao notar-se que das 74 ocorrências de eventos com ruído, 59 estão relacionadas com eventos perdidos. Quando aos eventos com ruído relacionados com outros tipos de chave, podemos perceber que eles não aparecem com tanta freqüência (12 eventos perdidos e 5 eventos espúrios). No entanto, a maioria está relacionada com eventos perdidos, o que é justificado pela falta de supervisão nestes equipamentos.

Outro tipo de ruído bastante freqüente está relacionado com disjuntores. Eles correspondem ao segundo tipo de ruído mais freqüente no gráfico (36 eventos espúrios e 14 eventos perdidos).

Algo que deve ser observado nestes dados consiste na ausência de ruído relacionado com as grandezas elétricas. Todavia, em outras amostras, percebemos que estas medidas também são susceptíveis a ruído.

Finalmente, eventos com ruído relacionados com impossibilidades — três primeiros tipos de ruído — em um sistema elétrico devem ser mencionados, uma vez que eles ocorrem com uma freqüência significativa (43 eventos). Contudo, é importante salientar que muitos destes eventos estão associados a um único equipamento devido, provavelmente, a um defeito no relé de sinalização do equipamento.

3. Técnicas robustas de correlação de eventos

Este capítulo tem o objetivo de descrever, de forma geral, as principais técnicas de correlação de eventos robustas encontradas na literatura. Ele está dividido em duas grandes seções: a primeira apresenta as técnicas; a segunda apresenta a aplicabilidade da utilização das técnicas abordadas para o problema em estudo.

3.1. Técnicas robustas de correlação de eventos

3.1.1. Redes de Bayes

Redes de Bayes é uma técnica computacional fundamentada em um modelo matemático probabilístico, o qual permite expressar tanto os elementos de um determinado domínio como os relacionamentos de dependência entre eles. O modelo consiste em um grafo acíclico, cujos nodos são conectados por arcos, que definem as relações de causa e efeito entre eles. Estes relacionamentos são quantificados por probabilidades condicionais que expressam a probabilidade de ocorrência de um dado nodo (nodo efeito), com base na probabilidade de ocorrência de seus predecessores (nodos causa) (GÜRER, 1996).

Cada nodo pertencente ao modelo possui uma variável que pode ser discreta ou contínua. Uma variável é dita ser **discreta** quando o número de estados possíveis é finito; quando não, ela é dita ser **contínua**. Para exemplificar tais conceitos, observe a Tabela 3.1, que ilustra a variável Tensão modelada tanto na forma de uma variável discreta como na forma de uma variável contínua. Note que ela, quando modelada como uma variável discreta, pode assumir apenas três valores: "Tensão 69kV", "Tensão 138kV" e "Tensão 500kV". Já na forma de uma variável contínua, ela pode assumir os infinitos valores compreendidos nos intervalos especificados.

Variável discreta	Variável contínua
Tensão de 69kV	$65kV \leq V \leq 73kV$
Tensão de 138kV	$134kV \leq V \leq 141kV$
Tensão de 500kV	$497kV \leq V \leq 503kV$

Tabela 3.1 – Variáveis discretas e contínuas

A Figura 3.1 ilustra uma rede de Bayes constituída por quatro nodos: "Desarme", "Proteção", "Bloqueio" e "Sobretensão". Observe que todos os nodos estão modelados na forma de variáveis discretas; por exemplo: o nodo "Desarme" pode assumir apenas dois valores: "total" e "parcial". Note também que todos os nodos estão relacionados entre si, através de arcos, e que as probabilidades condicionais referentes às relações causais entre os nodos estão localizadas dentro deles.

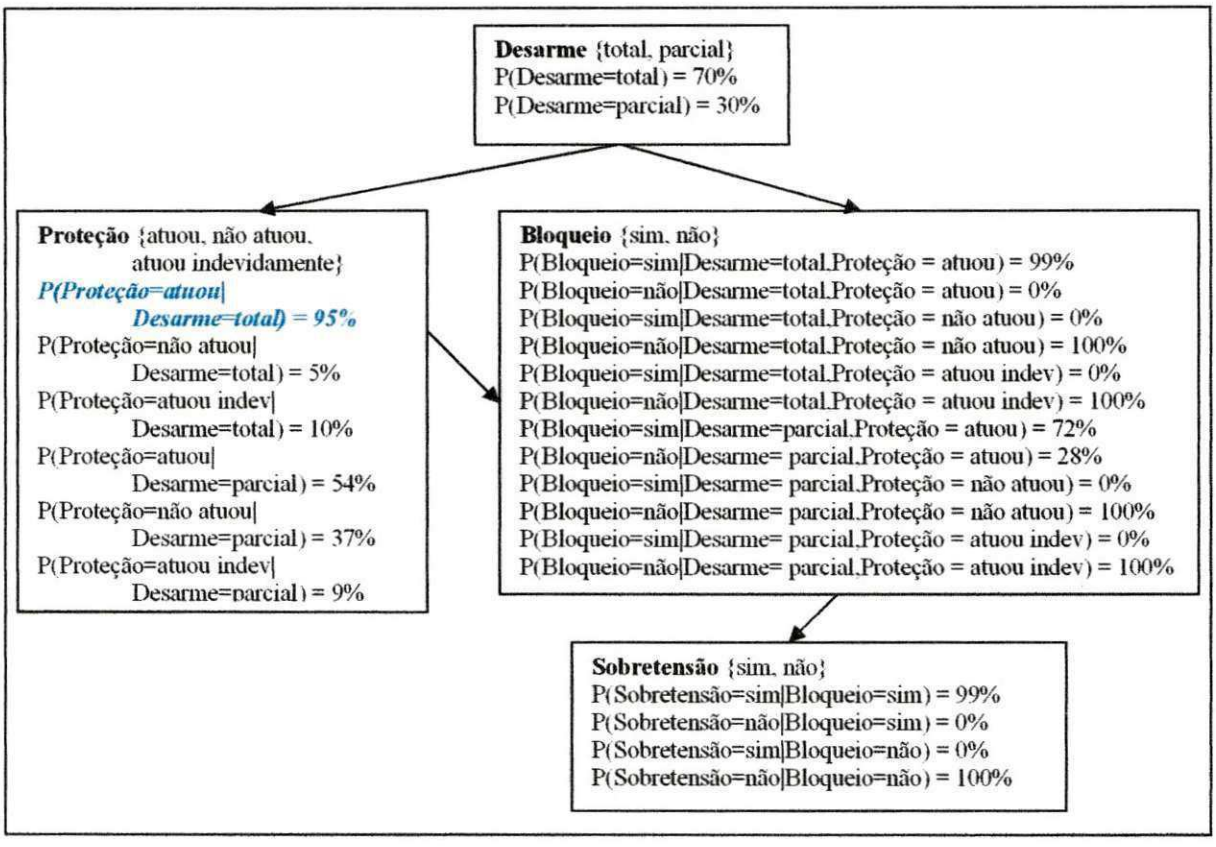

Figura 3.1 – Exemplo de uma rede de Bayes

Uma rede de Bayes corretamente modelada consiste em um ferramental extremamente importante na tomada de decisões. Com base nela, é possível observar o grau de dependência entre quaisquer nodos do modelo. Se quisermos saber qual a probabilidade de o nodo "Proteção" assumir o valor "atuar", dado que o nodo "Desarme" assumiu o valor "total", poderemos concluir, observando a Figura 3.1 (parte em azul), que é a de 95%.

No entanto, se quisermos saber a partir da rede de Bayes ilustrada na Figura 3.1, qual a probabilidade de o nodo "Desarme" assumir o valor "total" e, simultaneamente, os nodos "Proteção", "Bloqueio" e "Sobretesão" assumirem os valores "atuou", "sim" e "sim", respectivamente, poderemos notar que o valor procurado não está explicitamente representado na rede. Nestas circunstâncias, para se encontrar o valor, faz-se necessário efetuar um cálculo, que consiste no produto das probabilidades condicionais de cada nodo individualmente, dado que todos os nodos ocorram juntamente com seus antecedentes. Mais formalmente, temos:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Antecessores(X_i))$$

onde,

X_i – Probabilidade do nodo X_i ocorrer.

n – Número de nodos.

$Antecessores(X_i)$ – Probabilidade de todos os antecessores do nodo X_i tenham ocorrido.

Equação 3.1 - Cálculo da probabilidade em uma rede de Bayes

Tomando como exemplo a Equação 3.1, considere o cálculo da probabilidade para que os nodos "Desarme", "Proteção", "Bloqueio" e "Sobretensão" assumam os seguintes valores, respectivamente: "total", "atuou", "sim" e "sim".

= P(Desarme = total, Proteção = atuou, Bloqueio = sim, Sobretensão = sim)

= P(Desarme = total) * P(Proteção = atuou | Desarme = total) * P(Bloqueio = sim | Desarme = total, Proteção = atuou) * P(Sobretensão = sim | Bloqueio = sim)

= 0.70 * 0.95 * 0.99 * 0.99

= 0.6517 (65,17%)

No contexto da correlação de eventos, cada nodo representa um tipo de evento. O nodo "Bloqueio", por exemplo, pode tanto representar um evento que sinaliza a atuação do bloqueio (quando o nodo assume o valor "sim"), como a ausência do evento (quando o nodo assume o valor "não"). Os relacionamentos entre os nodos expressam a probabilidade de acontecimento de um evento supondo-se que os seus antecedentes tenham ocorrido; e, finalmente, uma falha é definida como um conjunto de eventos associados entre si pelos seus relacionamentos causais. Para exemplificar, a falha "desarme total de uma linha de transmissão com atuação da proteção com bloqueio por sobretensão" só ocorrerá quando os nodos "Desarme", "Proteção", "Bloqueio" e "Sobretensão" assumirem os valores "total", "atuou", "sim" e "sim", respectivamente.

A correlação de eventos em um sistema que utiliza uma rede de Bayes consiste em avaliar as probabilidades associadas a uma ou mais falhas, com base nos eventos recuperados da rede supervisionada, com o intuito de encontrar a falha mais provável (MEIRA, 1997).

Quando eventos são perdidos ou gerados espuriamente, a probabilidade de ocorrência da falha tende a ser muito baixa (próxima a zero). Como exemplo, considere que sejam recuperados da rede três eventos: "desarme total", "atuação do bloqueio" e "sobretensão". Nestas condições, o módulo de inferência, ao analisar os eventos juntamente com a rede de Bayes, verificará que a probabilidade de acontecimento destes eventos é zero. Veja o cálculo abaixo:

= P(Desarme = total, Proteção = não atuou, Bloqueio = sim, Sobretensão = sim)

= P(Desarme = total) * P(Proteção = não atuou | Desarme = total) * P(Bloqueio = sim | Desarme = total, Proteção = não atuou) * P(Sobretensão = sim | Bloqueio = sim)

= 0.70 * 0.05 * 0.0 * 0.99

= 0.0 = 0%

Com base no valor encontrado, o módulo de inferência constata que eventos foram perdidos ou gerados espuriamente. Para solucionar o problema, uma busca é efetuada, tentando-se encontrar a falha mais provável com base nos eventos recuperados da rede. Desta forma, várias falhas podem existir; todavia, a falha "desarme total de uma linha de transmissão com atuação da proteção com bloqueio por sobretensão" é a mais provável (65%) com o número mínimo de alterações nos eventos que compõem a falha (supondo-se que o evento, sinalizando a atuação da proteção, tenha sido perdido).

A principal desvantagem da utilização de redes de Bayes para fazer correlação de eventos consiste na dificuldade de construir uma rede com uma distribuição probabilística adequada. Este trabalho é feito consultando-se dados de correlações passadas e de especialistas da área. Além disto, a construção do algoritmo que rege o módulo de inferência não é uma atividade trivial, uma vez que precisa existir um equilíbrio entre a falha mais provável para o problema em análise e o número de alterações nos eventos que compõem a falha.

Análises mais detalhadas sobre redes de Bayes para o diagnóstico de falhas podem ser encontradas em OHSIE (1998) e MEIRA (1997).

3.1.2. Lógica nebulosa

Lógica nebulosa é uma técnica de Inteligência Artificial destinada à representação de um conhecimento impreciso. O conceito básico por trás da lógica nebulosa é o de conjuntos nebulosos, enquanto, na lógica clássica, a propriedade de pertinência entre um elemento X e um conjunto A sempre assume dois possíveis valores (verdadeiro ou falso); nos conjuntos nebulosos, cada elemento X possui, em relação ao conjunto A, um grau de pertinência μ, que pode assumir qualquer valor entre 0 (o elemento definitivamente não está contido no conjunto) e 1 (o elemento definitivamente está contido no conjunto).

Como exemplo, considere dois conjuntos: um contendo pessoas altas e outro, pessoas baixas. Na lógica clássica, um indivíduo X com 1,60m de altura ou se encontra no conjunto das pessoas altas ou no das pessoas baixas, dependendo do critério de classificação. Já na lógica nebulosa, este indivíduo X pertence a ambos os

conjuntos com graus de pertinência diferentes. Observando a Figura 3.2, podemos notar que ele pertence ao conjunto dos altos com grau de pertinência, $\mu(X) = 0,6$, e ao conjunto dos baixos com $\mu(X) = 0,8$. Perceba que, na lógica nebulosa, a classificação não é precisa; no entanto, é possível constatar que o indivíduo X está mais relacionado com o conjunto dos baixos do que com o dos altos.

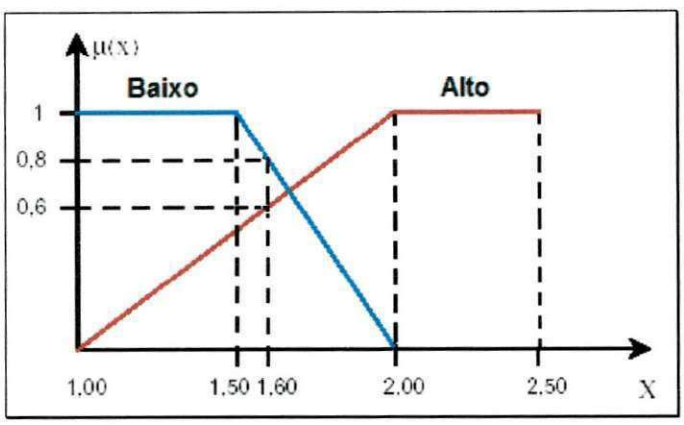

Figura 3.2 – Conjuntos nebulosos

Muitas vezes, não é possível construir uma função matemática que informe o grau de pertinência de um determinado elemento em um conjunto nebuloso. Nestas circunstâncias, faz-se necessário dividir um conjunto nebuloso em variáveis lingüísticas[1]. Por exemplo: muito baixo, baixo, pouco baixo, alto e muito alto. Cada variável lingüística em um conjunto nebuloso está associada a um determinado grau de pertinência (observe a Tabela 3.2).

Variável lingüística	**Grau de pertinência (μ)**
Muito baixo	1,0
Baixo	0,8
Pouco baixo	0,6
Alto	0,2

[1] Variáveis lingüísticas são valores nominais expressos na forma de palavras ou sentenças em linguagem natural (ABOELELA, 1999).

Muito alto	0,0

Tabela 3.2 – Variáveis lingüísticas para o conjunto nebuloso das pessoas baixas

Um sistema que utiliza lógica nebulosa é composto por um conjunto de regras definidas por um especialista. Cada regra é uma implicação do tipo SE-ENTÃO, cujos termos que a constituem consistem em variáveis lingüísticas de um determinado conjunto nebuloso (SABINO, 1999). Na Figura 3.3, v_1, v_2 e v_3 são variáveis lingüísticas que recebem como valores: "Disjuntor aparenta estar aberto", "Linha está com tensão muito baixa" e "Linha aparenta ter sido desligada", respectivamente. O resultado de cada regra ativada é uma variável lingüística associada a um conjunto nebuloso de saída. No mesmo exemplo, o resultado da ativação da regra foi a variável lingüística "Linha aparenta ter sido desligada".

SE v_1 = *"Disjuntor aparenta estar aberto"* ***E*** v_2 = *"Linha está com tensão muito baixa"* ***ENTÃO*** v_3 = *"Linha aparenta ter sido desligada"*

Figura 3.3 – Exemplo de uma regra utilizando lógica nebulosa

Como várias regras podem ser ativadas ao mesmo tempo, o módulo de inferência é responsável por escolher as regras que levem a valores lingüísticos com maior grau de pertinência. Além disso, ele pode combinar regras com o intuito de encontrar uma saída nebulosa com maior grau de pertinência (SABINO,1999). Entre os métodos de inferência mais conhecidos, podemos citar o de Mamdani (MAMDANI,1975) e o de Sugeno (SUGENO,1995).

É importante notar que o tratamento de ruído é inerente à técnica, uma vez que as próprias variáveis lingüísticas já expressam incerteza. Por exemplo, se um evento sinalizando a abertura de um disjuntor for perdido e, desta forma, o disjuntor aparenta estar fechado, a regra ilustrada na Figura 3.3 ainda poderá ser ativada, pois a variável lingüística "Disjuntor aparenta estar aberto" se aplica quando o disjuntor aparenta estar tanto aberto como fechado; entretanto, com diferentes graus de pertinência.

Entre as vantagens da utilização da lógica nebulosa em sistemas de correlação de eventos, podemos ressaltar a capacidade de representar um conhecimento impreciso, assim como o tratamento de ruído. No entanto, o grande problema da

utilização dela consiste na dificuldade associada à definição das variáveis lingüísticas e dos seus respectivos graus de pertinência, que são escolhidos a partir de dados estatísticos e de informações levantadas por especialistas na área.

No domínio de sistemas de potência, um excelente exemplo desta prática pode ser encontrado nas redes elétricas coreanas (LEE, 2000). Um outro trabalho interessante envolvendo lógica nebulosa temporal foi desenvolvido por ABOELELA (1999). Uma análise do estado atual das aplicações desta técnica no domínio de sistemas de potência pode ser encontrada em HIYAMA (1999).

Um estudo mais detalhado da técnica lógico nebulosa pode ser encontrado em GIARRATANO (1989).

3.1.3. Redes neurais artificiais

Redes neurais artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes, cuja principal característica é a capacidade de aprender com experiências passadas. A estrutura de uma rede neural artificial, da mesma forma que em uma rede neural biológica, é constituída por neurônios — unidades de processamento —, assim como dendritos e axônio — canais de comunicação de entrada e saída —, respectivamente, que interligam as unidades de processamento. Associados aos canais de comunicação de entrada, existem pesos, que são responsáveis pelo aprendizado destas redes. A Figura 3.4 ilustra um neurônio biológico e um neurônio artificial.

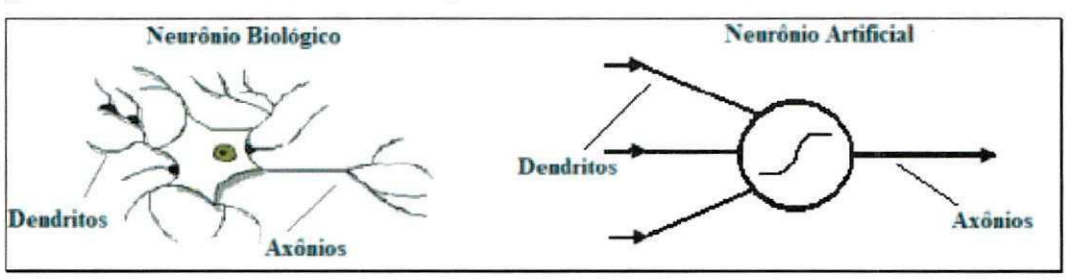

Figura 3.4 – Neurônio biológico e neurônio artificial

As redes neurais artificiais tiveram origem em três grandes publicações: (McCULLOCH, 1943), (HEBB, 1949), e (ROSENBLATT, 1958), as quais introduziram o primeiro modelo de redes neurais simulando "máquinas", o modelo básico de rede de auto-organização e o modelo *perceptron*, respectivamente. No

entanto, o interesse pela área foi intensificado no começo da década de 80, quando a performance dos computadores começou a permitir implementações práticas de alto custo computacional. Além disto, o surgimento do poderoso método *backpropagation* (RUMELHART, 1986) aumentou ainda mais a capacidade de aprendizado das redes neurais.

O comportamento inteligente de um neurônio artificial está relacionado com a sua capacidade de classificar padrões de informações. Um padrão quando apresentado ao neurônio, é dividido em vários sinais. Cada sinal X_i é multiplicado pelo peso W_i relativo ao canal de comunicação do sinal. Em seguida, calcula-se a soma ponderada entre os sinais e os pesos, com o objetivo de encontrar o nível de atividade, que é avaliado por uma função f(a). Se este exceder um certo limiar, o neurônio produz uma determinada resposta "y" de saída. A Figura 3.5 ilustra os elementos envolvidos neste processamento.

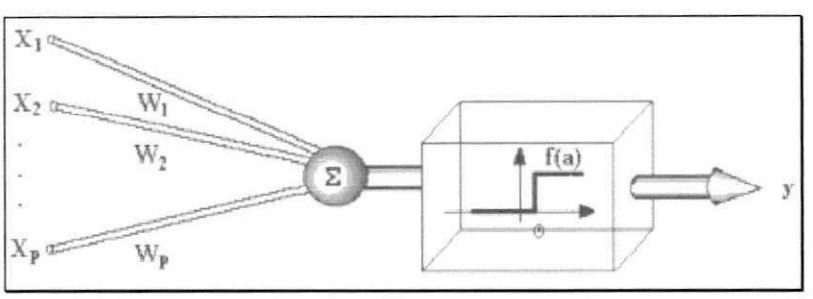

Figura 3.5 – Elementos envolvidos no processamento de um neurônio artificial

Da mesma forma que uma rede neural biológica é constituída por vários neurônios biológicos, uma rede neural artificial também é constituída por vários neurônios artificiais. A Figura 3.6 apresenta uma rede neural artificial. Observe que a rede está organizada em quatro camadas, cada uma contendo um número diferente de neurônios. Um padrão, quando é apresentado à rede, é dividido em vários sinais, que são distribuídos entre os neurônios da camada de entrada. O processamento destes alimenta os neurônios das camadas intermediárias que, por sua vez, alimentam os da camada de saída. No final do processamento destes últimos, uma saída é produzida para o padrão apresentado.

No contexto da correlação de eventos, cada padrão representa uma falha, enquanto cada sinal apresentado a um canal de comunicação consiste de um evento da

rede supervisionada. Desta forma, cada padrão é representado por um conjunto de eventos.

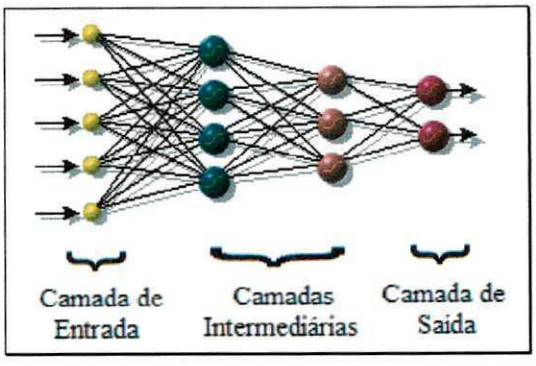

Figura 3.6 – Rede neural artificial

É importante salientar que, para uma rede neural poder simular um comportamento inteligente, os pesos associados aos canais de comunicação de entrada dos neurônios precisam estar bem ajustados. O processo que objetiva ajustar estes pesos é denominado de treinamento. Este processo consiste em apresentar à rede uma série de padrões juntamente com suas respectivas saídas desejadas. Em cada padrão apresentado, os pesos são ajustados. O objetivo da etapa de treinamento consiste em ajustar os pesos, de forma que, quando um novo padrão for apresentado à rede, a saída deste será equivalente à do padrão mais próximo a este apresentado durante a etapa de treinamento.

Redes neurais artificiais têm sido utilizadas em diversas áreas; entre elas, podemos citar: processamento digital de imagens, processamento de linguagem natural, jogos e sistemas de apoio à decisão. No âmbito desta área no domínio de sistemas de potência, alguns trabalhos devem ser mencionados; entre eles: (COUTTO, 1999) e (JOYA, 2000).

Entre as principais vantagens da utilização de redes neurais em sistemas de correlação de eventos, podemos citar:

1) Excelente performance – A etapa de treinamento, que é a mais lenta, é feita *off-line*, enquanto na etapa realizada em tempo de execução, o processamento é extremamente rápido.

2) Tratamento de ruído – A rede é treinada para aprender não apenas padrões idênticos, mas também similares. Desta forma, mesmo na presença de ruído, a rede neural ainda poderá efetuar diagnósticos corretos.

No entanto, assim como quaisquer outras técnicas, redes neurais artificiais possuem algumas desvantagens:

1) Dificuldade na etapa de treinamento – Existem dois problemas relacionados com a etapa de treinamento: o primeiro consiste na necessidade de uma base de dados grande e bem representativa; o segundo consiste na dificuldade de definir tanto os pesos dos canais de comunicação quanto o número de camadas e a disposição dos neurônios nestas últimas.

2) Não tolerante a mudanças topológicas – Todas as vezes que a topologia é alterada, uma nova etapa de treinamento precisa ser feita.

Uma análise mais detalhada de outros paradigmas de redes neurais em sistemas de correlação de eventos no âmbito de sistemas de potência pode ser encontrada em BIELER (1994).

3.1.4. *Codebooks*

Codebooks é uma técnica computacional simples que provê um mecanismo de correlação de eventos com uma excelente performance para redes de escalas e complexidades arbitrárias (YEMINIa, 1996). No contexto de *codebooks*, uma falha é modelada em um código binário, cujos elementos representam um determinado evento que pode ocorrer na rede monitorada.

Todas as falhas que podem acontecer na rede monitorada são armazenadas em uma tabela denominada *codebook*, onde cada coluna representa uma falha e cada linha representa um evento que pode ocorrer. Se n eventos distintos são apresentados no *codebook*, cada elemento do vetor $f_i=(e_1, e_2,..., e_n)$ contém uma medida da causalidade da falha f_i para o evento correspondente. Portanto, se no vetor f_1, $e_1=0$, o

evento e_1 nunca irá ocorrer como conseqüência da falha f_1; por outro lado, se e_1 é igual a 1, o evento e_1 sempre ocorre como conseqüência da falha f_1.

O processo de geração de um *codebook* consiste em mapear todas as informações relativas a cada falha a ser monitorada em uma matriz de correlação. Após a geração desta matriz, esta passa por uma etapa de redução, onde redundâncias são eliminadas, dando origem ao *codebook*.

Considere a seguinte matriz de correlação contendo duas falhas e quatro eventos:

	f1	**f2**
e1	1	1
e2	0	1
e3	0	0
e4	1	0

Figura 3.7 – Matriz de correlação

De acordo com a matriz de correlação ilustrada na Figura 3.7, para que a falha 2 seja diagnosticada, é necessário que os eventos 1 e 2 ocorram. Da mesma forma, para que a falha 1 seja detectada, os eventos 1 e 4 devem ocorrer. É importante perceber que as falhas 1 e 2 podem ser diferenciadas utilizando-se apenas os eventos 1 e 2. Observe a nova matriz de correlação (ou *codebook*) na Figura 3.8.

	f_1	**f_2**
e_1	1	1
e_2	0	1

Figura 3.8 – *Codebook*

O código binário que representa uma falha também é conhecido como a **assinatura** da falha. É importante que cada falha só possua uma única assinatura, uma vez que precisamos diferenciar uma falha de outra para efetuar corretamente o diagnóstico delas durante a correlação.

O processo de correlação consiste em procurar no *codebook* o código que mais se aproxima ou os códigos que mais se aproximam do código que representa o estado da rede. Utiliza-se a distância Hamming[2] para calcular a proximidade entre dois códigos.

Este método matemático possibilita detectar e até corrigir "erros" existentes no código que representa o estado da rede. Para um *codebook* de raio **r**, onde **r** é definido como a metade da distância Hamming mínima entre dois códigos quaisquer do *codebook*, pode-se detectar **r** erros, e corrigir **r-1** erros. Um erro significa um evento não detectado ou gerado espuriamente.

	$\mathbf{f_1}$	$\mathbf{f_2}$	$\mathbf{f_3}$	$\mathbf{f_4}$	$\mathbf{f_5}$	$\mathbf{f_6}$
$\mathbf{e_1}$	1	0	0	1	0	1
$\mathbf{e_2}$	1	1	0	1	0	0
$\mathbf{e_3}$	1	0	1	0	1	0
$\mathbf{e_4}$	1	1	1	0	0	1
$\mathbf{e_5}$	0	1	0	0	1	1
$\mathbf{e_6}$	0	1	1	1	0	0

Figura 3.9 – *Codebook* de raio 1.5

Considere um *codebook* de raio 1.5 da Figura 3.9 contendo 6 falhas e 6 eventos. Suponha que, durante o processo de monitoração de uma rede, dois eventos sejam recuperados: $\mathbf{e_1}$ e $\mathbf{e_3}$. O algoritmo de correlação de eventos procurará, no *codebook*, a falha que contiver apenas estes dois eventos ou a mais próxima desta.

Observando novamente o *codebook* da Figura 3.9, podemos concluir que não existe nenhuma falha que contenha apenas estes dois eventos. Se considerarmos que é muito mais provável um evento ser perdido do que ser gerado espuriamente, notaremos que a falha 1 ($\mathbf{f_1}$) é a mais próxima da falha que representa os eventos e_1 e e_3, uma vez que apenas 2 eventos foram perdidos (e_2 e e_4). Por outro lado, se

[2] Distância Hamming é o número de bits diferentes entre dois códigos binários.

considerarmos que é muito mais provável um evento ser gerado espuriamente do que ser perdido, a falha 5 ($\mathbf{f_5}$) será a mais próxima da falha em análise, uma vez que apenas um evento foi perdido (e_5) e outro foi gerado espuriamente (e_2).

Analisando sob um outro prisma, *Codebooks* pode ser considerado como uma técnica baseada em regras, representada em uma matriz comprimida utilizando teoria da codificação para tratar informações com ruído.

Entre as vantagens de Codebooks, podemos citar:

1) Resiliência a ruído – Incorpora durante o seu processamento a existência de eventos perdidos ou espúrios.
2) Adaptabilidade – Uma vez a topologia ter sido alterada, o novo *codebook* é gerado automaticamente.
3) Escalabilidade – A velocidade do diagnóstico é alta, podendo ser gerados *codebooks* com milhares de falhas e eventos sem degradar, de forma considerável, a performance de correlação.
4) Generalidade – Da mesma forma que se utilizam valores binários, podem-se utilizar valores probabilísticos e até mesmo temporais para representar a ocorrência de um evento.

Desvantagens:

1) Não permite detectar múltiplos problemas em uma única correlação.
2) Não existe uma fase de treinamento; faz-se necessário um especialista para definir os possíveis problemas da rede monitorada.
3) Dificuldade na geração do *codebook* – Existem algoritmos que efetuam a redução na matriz de correlação. Todavia, melhorias ainda podem ser feitas nestes algoritmos.
4) Existência de patentes que inviabilizam financeiramente a utilização da técnica (YEMINIb, 1996).

A técnica de Codebooks foi desenvolvida e patenteada pela SMARTS (*System Management ARTS Inc*). Ela é utilizada pelo sistema *InCharge* para o diagnóstico de

falhas em redes de telecomunicação e redes de computadores (KLIGER, 1995). Uma extensão deste trabalho, a qual permite identificar múltiplos problemas em uma única correlação com uma performance superior, foi desenvolvida por LO (1998).

3.1.5. Raciocínio baseado em casos

Raciocínio baseado em casos (RBC) é uma técnica de Inteligência Artificial que utiliza experiências adquiridas no passado para a tomada de decisões no futuro. Cada experiência é armazenada na forma de um caso, que é registrado em uma base de casos. Quando um novo problema é apresentado ao sistema, este recupera, na base de casos, o caso mais *similar*. A experiência obtida pelo sistema, quando este soluciona o problema, é utilizada para gerar um novo caso, que é adicionado à base de casos. Desta forma, o sistema é capaz de, sozinho obter conhecimento, sem necessitar da intervenção de um especialista.

A medida que expressa matematicamente quanto dois casos são similares entre si é chamada **medida de similaridade**. Ela é representada por um número real que varia entre 0 (não similar) e 1 (muito similar).

Uma outra característica de sistemas que utilizam esta técnica é a sua habilidade de modificar seu comportamento futuro, de acordo com erros cometidos. Além disso, soluções podem ser construídas para problemas novos, com a adaptação de casos passados para a nova situação.

CASO 1
Eventos: Disjuntor 15L8-RCD Aberto Disjuntor 15L8-AGD Aberto Tensão(05L8,RCD) = 2kV Tensão(05L8,AGD) = 1kV Potência(05L8,RCD) = 3MW Potência(05L8,AGD) = 1.5MW
Diagnóstico: Desligamento da Linha de Transmissão 05L8-RCD/AGD

Figura 3.10 – Exemplo de um caso

No contexto da correlação de eventos, cada caso armazena tanto os eventos como o diagnóstico de uma falha solucionada anteriormente com sucesso. O processamento do sistema consiste em recuperar o caso mais similar ao da falha a ser diagnosticada. Desta forma, o diagnóstico da falha em análise passa a ser similar ao do caso recuperado. A Figura 3.10 ilustra um exemplo de um caso no domínio de sistemas de potência.

Para compreender como é feito o cálculo da similaridade, observe a Figura 3.11, que ilustra dois casos: NOVO CASO e CASO 1. Note que cada evento possui a ele associado um peso, que expressa a importância do evento no cálculo da similaridade; o evento "Disjuntor 15L9-RCD Fechado", por exemplo, possui peso 4. Perceba que entre os eventos existem medidas de similaridades locais que expressam quanto os eventos são similares entre si. Dependendo do valor dos eventos, a medida similaridade local pode variar; o evento "Tensão(05L9,AGD)=1.1kV", por exemplo, é muito similar ao evento "Tensão(05L8,AGD)=1.0kV", uma vez que a similaridade local é 0.9. O cálculo da similaridade é a soma ponderada entre os pesos e as medidas de similaridade locais entre cada evento. O valor 76% expressa a similaridade global entre os dois casos.

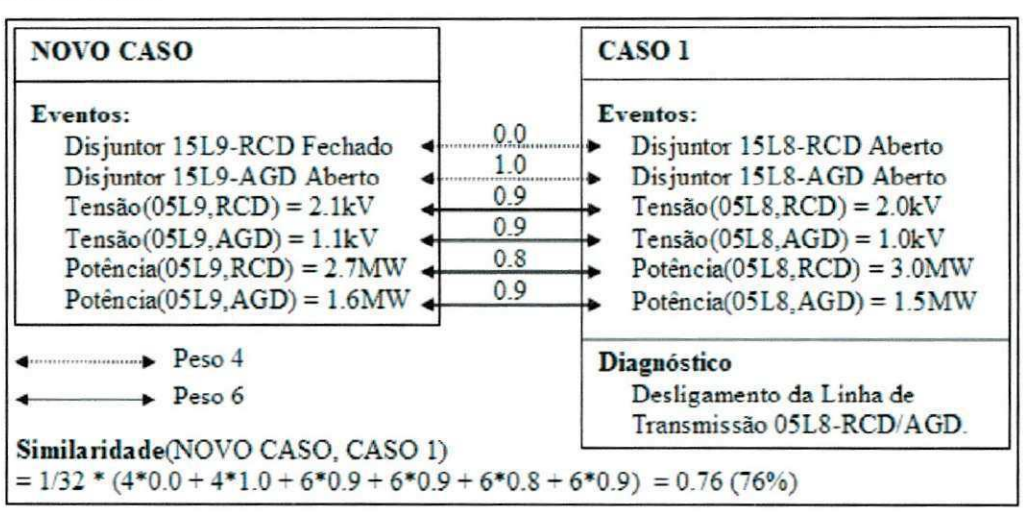

Figura 3.11 – Cálculo da similaridade

A medida de similaridade permite tratar informações com ruído, uma vez que os casos não precisam ser idênticos para serem considerados similares. A similaridade entre dois casos depende de quão similares os eventos são entre si.

Se, entre todos os casos da base de casos, o CASO 1 for o mais similar ao NOVO CASO, este último será adaptado e, em seguida, incorporado à base de casos. A Figura 3.12 ilustra o NOVO CASO ao final do processo.

NOVO CASO
Eventos: Disjuntor 15L9-RCD Fechado Disjuntor 15L9-AGD Aberto Tensão(05L9,RCD) = 2.1kV Tensão(05L9,AGD) = 1.1kV Potência(05L9,RCD) = 2.7MW Potência(05L9,AGD) = 1.6MW
Diagnóstico: Desligamento da Linha de Transmissão 05L9-RCD/AGD

Figura 3.12 – Estrutura do caso antes de ser adicionado à base de casos

A principal desvantagem da utilização desta técnica consiste na preparação de uma base de casos adequada com pesos e métricas de similaridade bem distribuídos. Nem sempre é desejável que o sistema recupere o caso com maior número de campos coincidentes como o mais similar, uma vez que alguns campos podem ser irrelevantes (LEWIS, 1999). Além disso, faz-se necessário a presença de um especialista para avaliar os casos e validar os novos casos. A existência de um caso com ruído na base de casos compromete tanto o diagnóstico da falha como o futuro da base de casos, uma vez que novos casos com ruído podem ser incorporados à base.

3.2. Aplicabilidade das técnicas robustas para o problema em estudo

Nesta seção apresentaremos as principais razões que impossibilitam a **plena eficácia** da utilização das técnicas elucidadas para o problema em estudo. Entretanto, para facilitar a compreensão desta seção, algumas dificuldades relacionadas com o problema em estudo serão enumeradas. São elas:

1) Diversidade de eventos – Aproximadamente trezentos tipos diferentes de eventos importantes para o diagnóstico de falhas podem ser sinalizados na rede elétrica da CHESF.

2) Diversidade de equipamentos – A rede elétrica da CHESF é composta por aproximadamente doze mil equipamentos (envolvendo chaves, disjuntores, transformadores, linhas de transmissão, reatores e banco de capacitores).

3) Base de dados existente – Ela é composta por medidas digitais e analógicas. As medidas digitais consistem nos eventos e nos alarmes recuperados da rede elétrica durante um determinado período de tempo; enquanto as medidas analógicas são representadas por um conjunto de coletas — recuperadas do sistema elétrico a cada cinco minutos — contendo as grandezas elétricas de todos os equipamentos supervisionados.

Redes de Bayes

Apesar de a técnica redes de Bayes ser bastante poderosa para o tratamento de ruído, a sua utilização para o problema em estudo é dificultada por algumas razões. A principal razão consiste na dificuldade de elaborar uma rede com valores probabilísticos corretamente distribuídos. A dificuldade advém do fato de que muitas vezes os especialistas não têm segurança suficiente nem mesmo para afirmar o que é mais provável.

Outro problema diz respeito à inexistência de uma base de dados bem representativa, na qual fosse possível recuperar os valores probabilísticos a partir de uma análise estatística. O grande problema da base de dados existente está relacionado com a carência informações relacionadas com as grandezas elétricas, uma vez que as coletas encontradas na base de dados foram realizadas a cada cinco minutos, o que não representa bem a realidade no sistema elétrico, pois uma grandeza elétrica pode variar a cada dois segundos, por exemplo.

Além disso, a utilização desta técnica levaria à elaboração de uma rede com aproximadamente 100 nodos, devido à quantidade de tipos diferentes de equipamento e de evento. Este problema se torna ainda mais complexo, quando a rede precisa receber uma manutenção, onde a inserção de um valor probabilístico mal estimado pode levar a grandes efeitos colaterais.

codificadas como uma conjunção de condições. Entretanto, é comum em sistemas elétricos a representação de falhas envolvendo tanto conjunção como disjunção de condições. A disjunção leva a um conjunto de códigos para uma mesma falha, podendo tornar combinatorial a complexidade da geração e verificação do *codebook*.

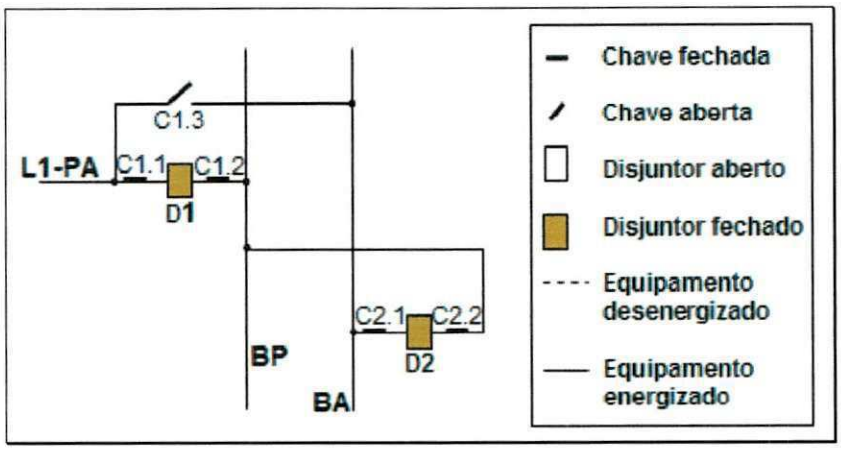

Figura 3.13 – Exemplo de uma linha conectada a seus barramentos

Para exemplificar uma disjunção de condições, considere uma falha que consiste na desconexão de uma linha de seus barramentos. Para facilitar a compreensão deste exemplo, observe a Figura 3.13 que ilustra uma forma de os equipamentos poderem estar dispostos em uma rede elétrica. Para que a linha L1 (lado PARA) esteja conectada a um de seus barramentos, existem três possibilidades: a de todos os equipamentos estarem fechados, a de o disjuntor D1 e as chaves C1.1 e C1.2 estarem fechados, e a de o disjuntor D2 e chaves C1.3, C2.1 e C2.2 estarem fechados[3]. Desta forma, qualquer outra possibilidade implicará a não-conexão da linha a seus barramentos. Se formos totalizar a quantidade de possibilidades, chega-se a 105 combinações. Este valor pode ser encontrado da seguinte forma: total de todas as combinações possíveis com exceção das possibilidades que implicam a conexão da linha a um dos barramentos. Observe o cálculo abaixo:

---

[3] Note que a opção "chave C1.3 fechada" não foi mencionada. Apesar de ser uma possibilidade do ponto de vista de um grafo de conectividade, não o é do ponto de vista de um sistema elétrico (a linha ficaria sem proteção).

Total de combinações:

$$P_7^{0,7} + P_7^{1,6} + P_7^{2,5} + P_7^{3,4} + P_7^{4,3} + P_7^{5,2} + P_7^{6,1} + P_7^{7,0} = 128$$

Total de combinações que implicam a conexão da linha:

$$A + B - A \cap B = 23$$

$$A = P_4^{0,4} + P_4^{1,3} + P_4^{2,2} + P_4^{3,1} + P_4^{4,0} = 16$$

$$B = P_3^{0,3} + P_3^{1,2} + P_3^{2,1} + P_3^{3,0} = 8$$

$A \cap B = 1$ (combinação na qual todos os equipamentos estão fechados)

onde:

A é o número de combinações, no qual o disjuntor D1 e chaves C1.1 e C1.2 estão fechados

B é o número de combinações, no qual o disjuntor D2 e chaves C1.3, C2.1 e C2.2 estão fechados

Resultado: *total de combinações exceto as combinações que implicam a conexão da linha = 128 - 23 = 105*

É importante salientar que esta é uma das falhas com o menor número de disjunções que pode ser encontrado em um sistema elétrico. Existem falhas, por exemplo, onde o número de disjunções pode chegar a 30. Considerando o fato de o número de eventos possíveis ultrapassar 100, de o número de falhas possíveis estar em torno de 100 e que o número de equipamentos na rede supera 3000, a utilização de *codebooks* implicaria um problema de explosão combinatorial.

4. Uma ferramenta robusta de tratamento de eventos em redes elétricas: requisitos, técnica robusta de correlação de eventos e projeto

Neste capítulo, temos o propósito de apresentar uma ferramenta robusta de tratamento de eventos, intitulada *Robust SmartOne*, desenvolvida para auxiliar os operadores de redes elétricas no diagnóstico de problemas. Esta ferramenta é fruto de um projeto de P&D entre a CHESF e a UFCG, intitulado *Smart Alarms*. Ela foi desenvolvida com o objetivo de ser utilizada nos cinco centros de operação da CHESF e de ser integrada ao sistema de supervisão desta empresa — o SAGE. Atualmente, ela está sendo utilizada no Centro Regional de Operações Leste da CHESF.

Antes de detalharmos a ferramenta *Robust SmartOne*, é necessário salientar que ela é uma extensão de uma ferramenta já existente desenvolvida pelo projeto *Smart Alarms* — o *SmartOne*. Entretanto, essa ferramenta não considera durante seu processamento o tratamento de eventos com ruído.

Para facilitar a compreensão deste capítulo, inicialmente apresentaremos o *SmartOne*; em seguida, descreveremos os requisitos que nortearam o desenvolvimento da ferramenta robusta, a técnica robusta de correlação de eventos utilizada por ela e, finalmente, seu projeto arquitetural.

4.1. Uma ferramenta de tratamento de eventos em redes elétricas: *SmartOne*

Nesta seção, explicaremos, brevemente, o funcionamento do *SmartOne*, o ambiente físico no qual ele está inserido e o seu projeto arquitetural.

O *SmartOne* utiliza uma técnica híbrida constituída de raciocínio baseado em regras e de raciocínio baseado em modelos. A Figura 4.1 ilustra as principais entidades responsáveis pelo seu funcionamento: a primeira é o modelo da rede, o qual consiste em um grafo, onde os nodos representam os elementos da rede e seus estados correntes e os arcos indicam como os elementos se conectam eletricamente; a segunda é o módulo de diagnóstico de falhas que utiliza uma base de regras, definidas por especialistas, com o intuito de efetuar possíveis diagnósticos de problemas no modelo da rede. O seu funcionamento consiste em atualizar o modelo da rede com base nos eventos oriundos do sistema elétrico e, em seguida, analisá-los com o objetivo de efetuar possíveis diagnósticos de problemas.

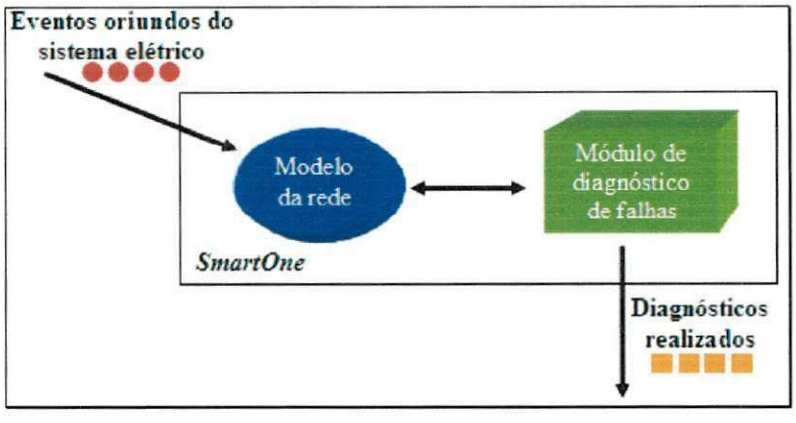

Figura 4.1 – Principais entidades que compõem o SmartOne

A principal deficiência do *SmartOne* está relacionada com a sua dependência do modelo da rede. Desta forma, se eventos forem perdidos ou gerados espuriamente, o modelo da rede deixará de estar representando a realidade no sistema elétrico e, por conseguinte, o módulo de diagnóstico de falhas será incapaz de realizar diagnósticos corretamente.

4.1.1. Ambiente físico

O ambiente físico em que o *SmartOne* está inserido pode ser observado através da Figura 4.2. Nela, podemos encontrar as subestações de energia elétrica, as unidades terminais remotas, a rede do SAGE, o *gateway* e, finalmente, o *SmartOne*.

O *gateway* consiste em um componente de *software*, desenvolvido pelo projeto *Smart Alarms*, responsável por prover ao *SmartOne* todas as informações

necessárias para o seu funcionamento. Ele foi desenvolvido com o intuito de evitar que o processamento do *SmartOne* interfira no escalonamento em tempo-real do SAGE e que possíveis defeitos de *software* comprometam o funcionamento deste sistema. Desta forma, o *SmartOne*, em vez de acessar diretamente o SAGE, acessa o *gateway* e este, por sua vez, o SAGE.

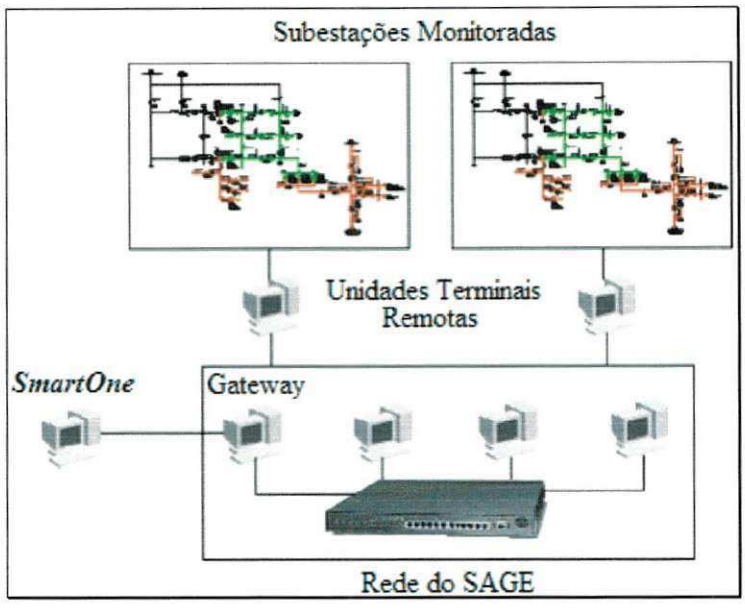

Figura 4.2 – Ambiente físico no qual o SmartOne está inserido

4.1.2. Projeto arquitetural

A arquitetura do *SmartOne* foi desenvolvida usando uma abordagem baseada em componentes. A linguagem de programação utilizada foi Java, permitindo que a ferramenta funcione em máquinas com sistemas operacionais diferentes.

A Figura 4.3 apresenta a arquitetura básica do *SmartOne*. Nela, é possível observar o *gateway*, que é responsável por recuperar do SAGE todos os eventos oriundos da rede elétrica, e inseri-los no barramento[4] de eventos.

O módulo de diagnóstico de falhas e o modelo da rede recebem os eventos recuperados pelo *gateway* através do barramento de eventos. O primeiro analisa os

---

[4] Note, que o termo **barramento** se refere a um barramento de *software* e não a um barramento elétrico.

eventos com base no estado do segundo e os possíveis diagnósticos são inseridos no barramento de diagnósticos.

Os diagnósticos gerados pelo módulo de diagnóstico de falhas são recuperados pelo apresentador gráfico através do barramento de diagnósticos.

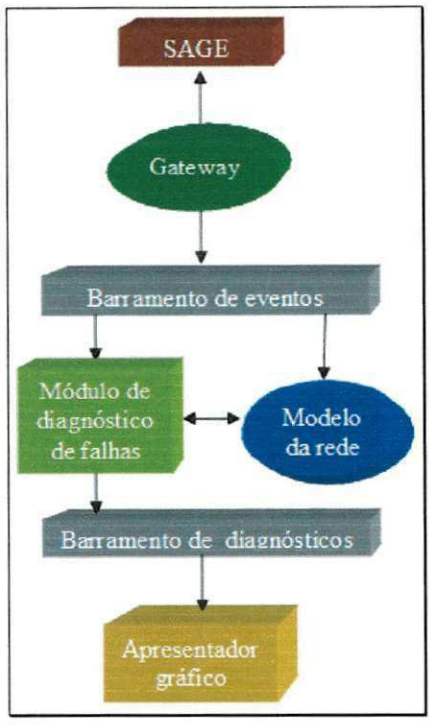

Figura 4.3 – Arquitetura do *SmartOne*

O barramento de eventos e o barramento de diagnósticos são responsáveis por prover flexibilidade à arquitetura. Eles servem para isolar os **produtores de dados** dos **consumidores de dados**. Na arquitetura básica, o *gateway* não se comunica com o módulo de diagnósticos de falhas nem com o modelo da rede, e sim com o barramento de eventos; da mesma forma, o módulo de diagnóstico de falhas não se comunica com o apresentador gráfico para inserir os diagnósticos efetuados. Ele se comunica com o barramento de diagnósticos, que, por sua vez, fornece os diagnósticos ao apresentador gráfico.

4.2. Levantamento de requisitos

Os requisitos que nortearam o desenvolvimento deste trabalho podem ser analisados tanto do ponto de vista do *SmartOne*, como da ferramenta robusta, uma

vez que esta última é uma extensão do primeiro. Nesta seção descreveremos apenas os requisitos funcionais e não-funcionais que foram aplicados no desenvolvimento da ferramenta robusta. Os requisitos aplicados no desenvolvimento do *SmartOne* podem ser encontrados em DUARTE (2003).

4.2.1. Requisitos funcionais

A principal dificuldade enfrentada, na etapa de levantamento destes requisitos, consistiu na carência de conhecimento dos eventos com ruído presente no sistema elétrico. Esta dificuldade advém do fato de que nem mesmo os próprios operadores da rede detêm tal conhecimento de forma completa e explícita.

Em conseqüência, fizemos um levantamento de todos os tipos de ruído conhecidos, tanto consultando os operadores da rede, como utilizando o *SmartOne*, objetivando descobrir quando este funcionava incorretamente devido à presença de ruído. No final deste levantamento, concluiu-se que o ruído relacionado com eventos de disjuntores e chaves eram os mais freqüentes (o capítulo 2 descreve com detalhes exemplos de eventos com ruído relacionados com estes equipamentos). Desta forma, os requisitos funcionais que nortearam o desenvolvimento da ferramenta robusta foram:

1) Diagnósticos – A ferramenta robusta deve ser capaz de efetuar diagnósticos corretos, mesmo que eventos de abertura e fechamento de disjuntores e chaves sejam perdidos ou gerados espuriamente.

2) Correção topológica – A ferramenta robusta deve recuperar o estado de abertura de todos os disjuntores e chaves da rede elétrica. Além disso, quando não for possível recuperar tais estados, ela deve ser capaz de estimá-los.

3) *Logs* – A ferramenta robusta deve armazenar em arquivos de dados históricos todas as informações necessárias para a compreensão das ações por ela tomadas. Com isso, tem-se um histórico do processamento, o que permite uma localização mais fácil de defeitos de *software*.

4.2.2. Requisitos não-funcionais

Os seguintes requisitos não-funcionais também devem ser atendidos pela ferramenta robusta:

1) Facilidade de uso – O uso da ferramenta robusta deve consistir apenas em observar os diagnósticos efetuados por ela. Em nenhum momento o operador deve intervir no funcionamento dela, para que esta efetue diagnósticos corretos.

2) Manutenção – A manutenção da ferramenta robusta deve estar relacionada apenas com a atualização do modelo da rede, quando equipamentos forem inseridos ou removidos da rede elétrica.

3) Desempenho – A janela de tempo (conjunto de eventos recuperados da rede durante um determinado período de tempo) utilizada pela ferramenta robusta não deve ser superior a 10 segundos, isto é, no máximo, a cada 10s, uma análise dos eventos recuperados da rede deve ser efetuada com o intuito de efetuar possíveis diagnósticos.

4.3. Uma nova técnica robusta de correlação de eventos

Nesta seção descrevemos a técnica robusta de correlação de eventos, desenvolvida para solucionar o problema em estudo. Para facilitar a compreensão da técnica, primeiramente explicaremos os motivos que nos levaram ao seu desenvolvimento e, em seguida, apresentá-la-emos.

Após a etapa de levantamento dos requisitos, uma análise das técnicas encontradas na literatura foi realizada com o intuito de descobrir aquela que poderia ser utilizada para solucionar o problema em estudo. Entretanto, no final da análise, constatamos que nenhuma delas, ao nosso ver, resolveria o problema com plena eficácia (veja a última seção do capítulo anterior, para maiores detalhes). Nestas circunstâncias, decidimos desenvolver uma nova técnica que pudesse ser aplicada ao problema em estudo. O principal fator que contribuiu para a concepção desta técnica foi a existência de uma ferramenta capaz de realizar diagnósticos corretos, na ausência de ruído. Desta forma, se o modelo da rede atualizado com os eventos

oriundos da rede elétrica sempre fosse correto, a ferramenta sempre iria efetuar diagnósticos corretos.

O fundamento da técnica robusta desenvolvida consiste em remover todo o ruído presente nos eventos recuperados da rede elétrica, de forma que o modelo da rede, quando atualizado, sempre estará o mais próximo possível do estado real da rede elétrica; por conseguinte, o módulo de diagnóstico de falhas sempre será capaz de realizar diagnósticos corretos.

A concepção da ferramenta *Robust SmartOne* consistiu em estender a ferramenta *SmartOne* com a incorporação de um novo sistema baseado em conhecimento, o **filtro de ruído**. Ele é responsável por analisar os eventos recuperados da rede, detectar possíveis inconsistências nos eventos e, caso existam, removê-las. Neste contexto, uma **inconsistência** é caracterizada por pelo menos um evento com ruído. Desta forma, para que inconsistências sejam eliminadas, os eventos espúrios precisam ser removidos e os perdidos, gerados. No final do processamento do filtro de ruído, os eventos filtrados atualizarão o modelo da rede, para que, no futuro, este seja analisado pelo módulo de diagnóstico de falhas. A Figura 4.4 ilustra as principais entidades que compõem a ferramenta robusta.

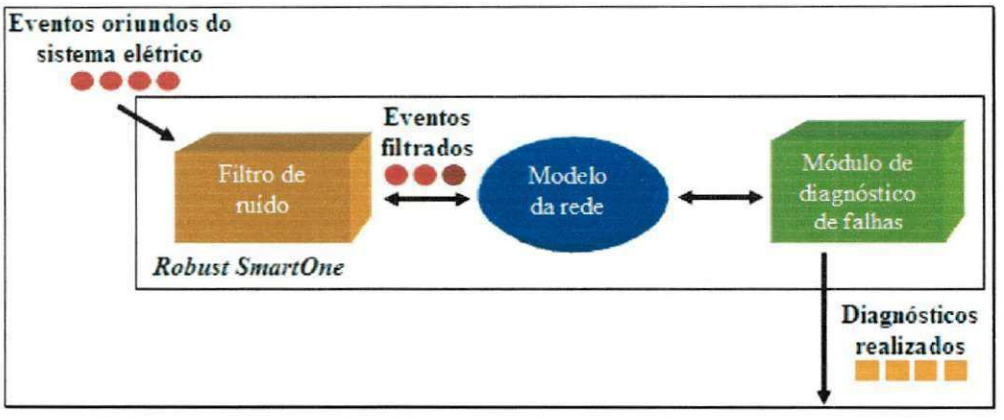

Figura 4.4 – Principais entidades que compõem a ferramenta robusta

Como podemos perceber, com a incorporação do filtro de ruído ao *SmartOne* existente, a ferramenta *Robust SmartOne* passou a ser constituída por dois sistemas baseados em conhecimento, organizados em série, um alimentando o outro. O primeiro sendo representado pelo filtro de ruído, enquanto o segundo, pelo módulo de diagnóstico de falhas.

O processamento do filtro de ruído é divido em duas fases: detecção e correção. Na primeira, o filtro analisa os eventos oriundos da rede juntamente com o modelo da rede em busca de inconsistências nos eventos; na segunda, as possíveis inconsistências detectadas na fase anterior são eliminadas, seja mediante a remoção de eventos, seja mediante a adição de novos eventos (veja a Figura 4.5). As seções seguintes descrevem estas duas fases com mais detalhes.

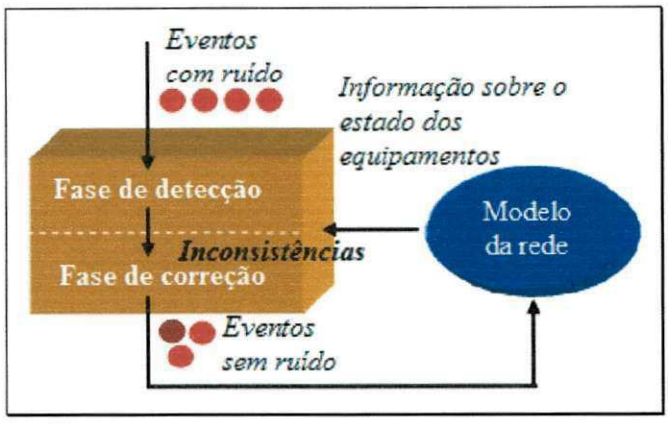

Figura 4.5 – Fases do filtro de ruído

4.3.1. Fase de detecção

A fase de detecção tem o objetivo de detectar possíveis inconsistências nos eventos recuperados da rede. A detecção de ruído nos eventos de uma janela de tempo é feita com o auxílio de heurísticas sobre fatos da rede e de regras de consistência entre estados dos elementos da rede.

1) Heurísticas – São padrões comportamentais que, embora não tenham sua corretude provada matematicamente, aplicam-se bem na grande maioria dos casos. Elas são descobertas pelos operadores do sistema elétrico da CHESF, com base em suas experiências. Por exemplo: "grandezas elétricas são mais confiáveis que as variáveis de estado[5]".

2) Regras de consistência entre estados dos elementos da rede – Essas regras definem associações entre estados dos elementos da rede; por exemplo, "se a tensão de uma linha for zero, ela não poderá estar

[5] Estado de abertura de disjuntores e chaves.

> conectada a nenhum barramento energizado". A maioria das regras de consistência é obtida a partir da base de regras, a qual é utilizada pelo módulo de diagnóstico de falhas, enquanto as demais são obtidas do conhecimento de especialistas do sistema elétrico.

As regras de consistência se apóiam em heurísticas. Para ficar mais claro, seja o seguinte exemplo: se as grandezas elétricas indicarem que a tensão em uma linha é zero e as variáveis de estado indicam que ela está conectada a um barramento e as grandezas elétricas indicam que este barramento está energizado, pode-se concluir que a linha não está conectada ao barramento (regra de consistência). A conclusão está apoiada no fato de que grandezas elétricas são mais confiáveis que variáveis de estado (heurística).

Para facilitar a compreensão dos exemplos que descrevem o funcionamento da fase de detecção, considere a Figura 4.6, que ilustra alguns elementos básicos de um modelo da rede; a figura representa uma pequena parte de uma subestação, composta por dois barramentos (B1 e B2), dois terminais de linhas de transmissão (L1-PA e L2-PA – lado PARA), dois transformadores (T1 e T2) e alguns disjuntores. Para que um terminal de uma linha esteja energizada, é preciso estar conectado a pelo menos um equipamento energizado, seja ele um transformador seja um barramento. Por exemplo, o terminal do lado PARA da linha L1 está energizada, pois ela está conectado ao barramento B1 pelo disjuntor D3 — que está fechado —, ao barramento B2 pelos disjuntores D1 e D2 — que estão fechados — e ao transformador T1 pelo disjuntor D2 — que também está fechado.

Seja o modelo da rede no estado I ilustrado na Figura 4.6, com todos os disjuntores fechados. Surge uma janela de tempo contendo o evento: "Disjuntor D2 abriu"; quando o procedimento da fase de detecção atuar, notará que, se o modelo da rede fosse atualizado com o evento, passaria para um estado II inconsistente (Figura 4.7), uma vez que o terminal da linha L1 do lado PARA está desenergizado; porém, está conectado por meio do disjuntor D3 ao barramento energizado B1. Portanto, o procedimento de detecção conclui que os eventos recuperados da rede estão inconsistentes, isto é, contêm ruído.

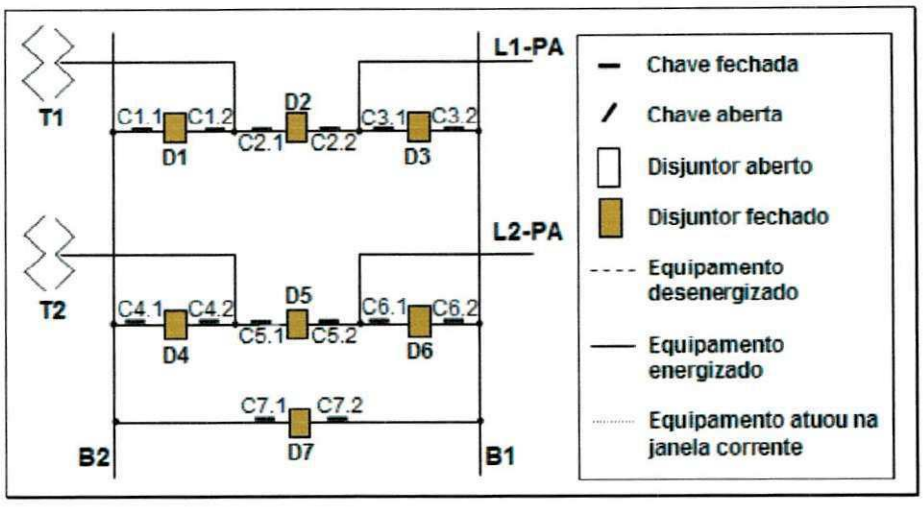

Figura 4.6 – Modelo da rede (estado I)

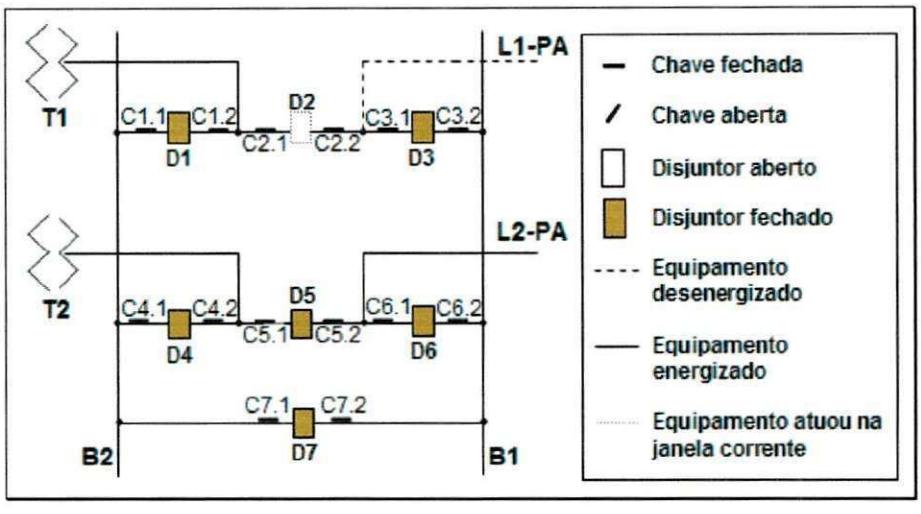

Figura 4.7 – Modelo da rede (estado II)

Observe que, com o modelo da rede no estado I, o terminal da linha L1 do lado PARA está energizado; entretanto no estado II, está desenergizado. Esta mudança de estado está relacionada com alterações nos valores das grandezas elétricas após o término da janela de tempo.

Maiores detalhes relacionados com a fase de detecção do filtro de ruído serão fornecidos mais adiante.

4.3.2. Fase de correção

A fase de correção é caracterizada pela tentativa de corrigir as inconsistências levantadas na fase de detecção. Assim como o procedimento da fase anterior, o

funcionamento do procedimento de correção é baseado nas regras de consistência e nas heurísticas.

Continuando o exemplo utilizado para ilustrar a fase de detecção, o procedimento de correção, pela análise do estado II, conclui que o terminal da linha deve estar desconectado dos equipamentos energizados (regra de consistência). Desta forma, o procedimento conclui que o disjuntor D3 deveria estar aberto e, em seguida, gera um novo evento: "Disjuntor D3 abriu". No final do processamento da fase de correção, os eventos "Disjuntor D2 abriu" e "Disjuntor D3 abriu" atualizam o modelo da rede (veja a Figura 4.8).

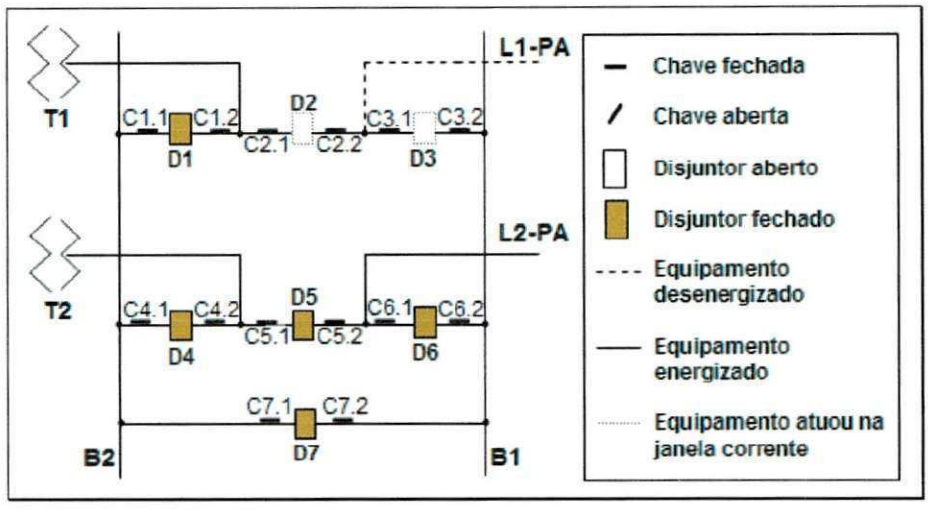

Figura 4.8 – Estado do modelo da rede atualizado com os eventos filtrados

Assim como na fase anterior, maiores detalhes relacionados com a fase de correção do filtro de ruído serão fornecidos mais adiante.

4.4. Projeto arquitetural da ferramenta robusta de tratamento de ruído

Nesta seção, abordaremos o projeto arquitetural da ferramenta robusta desenvolvida. Inicialmente, apresentaremos o projeto arquitetural; e, em seguida, o projeto detalhado do módulo de tratamento de ruído existente na ferramenta.

4.4.1. Projeto arquitetural

A ferramenta robusta herdou a arquitetura do *SmartOne*. A extensão da arquitetura para a inserção do filtro de ruído foi realizada com a adição de dois novos componentes (veja a Figura 4.9): o filtro de ruído e o barramento de eventos filtrados.

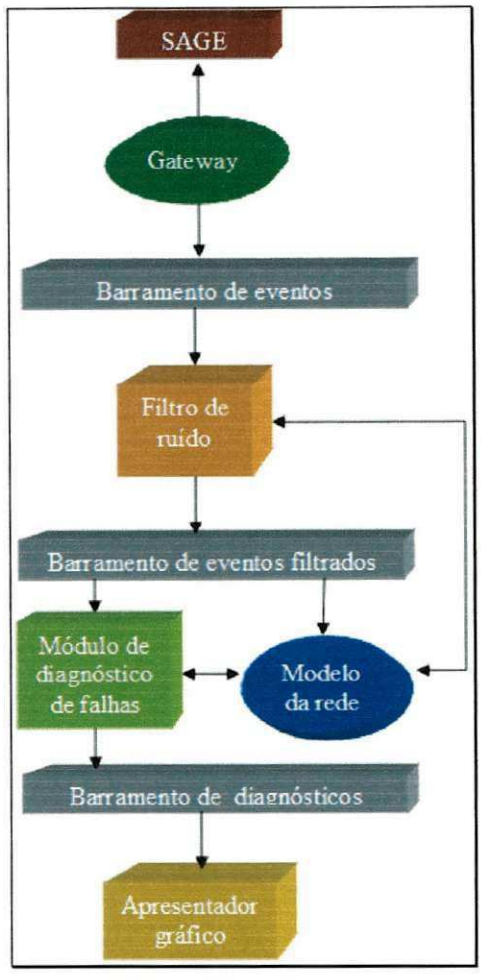

Figura 4.9 – Arquitetura da ferramenta robusta

O filtro de ruído recebe os eventos recuperados pelo *gateway* através do barramento de eventos. Em seguida, ele analisa os eventos recuperados com base no estado atual do modelo da rede, com o objetivo de encontrar ruído nas seqüências de eventos. Após a análise, os possíveis eventos com ruído são corrigidos. No final, os eventos filtrados são inseridos no barramento de eventos filtrados.

A flexibilidade da arquitetura permitiu que o filtro de ruído fosse inserido sem causar alteração a qualquer outro componente. Da mesma forma, que na arquitetura do *SmartOne*, o módulo de diagnóstico de falhas não conhece o *gateway*, uma vez que

os eventos recuperados por este último são obtidos através do barramento de eventos. Na arquitetura atual, o módulo de diagnóstico de falhas também não conhece o filtro de ruído, pois os eventos que o módulo de diagnóstico de falhas analisa são recuperados através do barramento de eventos filtrados.

4.4.2. Projeto detalhado

Nesta seção, descreveremos em maiores detalhes os principais módulos que compõem o filtro de ruído.

O filtro de ruído, internamente, é composto por quatro filtros, diferenciando-se pelo tipo de ruído que é tratado. Além disso, cada filtro é dividido em duas fases (detecção e correção) e seu funcionamento é baseado em heurísticas e regras de consistência (veja a Figura 4.10).

Arquiteturalmente, os filtros estão dispostos em série, de forma que os eventos filtrados por um filtro consistem na entrada do filtro seguinte. Note que, à medida que os eventos passam pelos filtros, a quantidade de ruído presente nos eventos é reduzida.

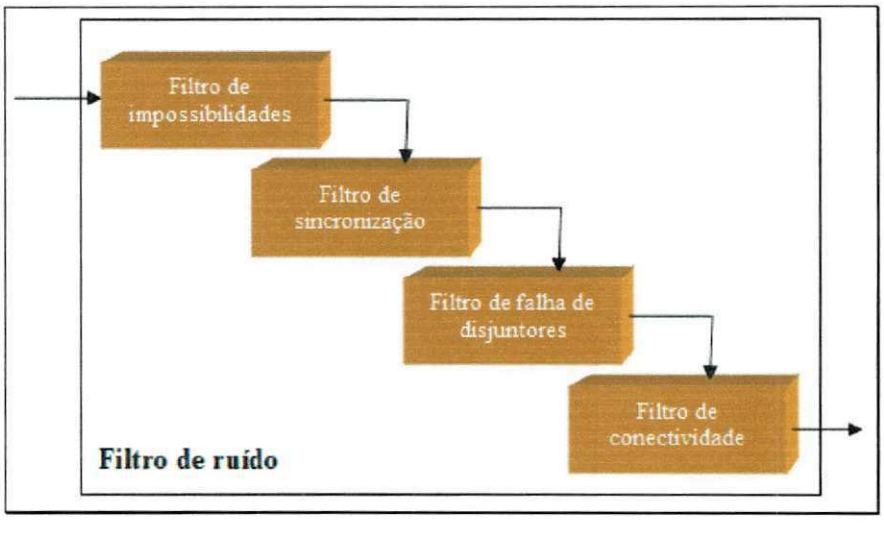

Figura 4.10 – Projeto detalhado do filtro de ruído

Os quatro filtros que compõem o filtro de ruído são os seguintes:

1) filtro de impossibilidades;

2) filtro de sincronização;

3) filtro de falha de disjuntores;

4) filtro de conectividade.

No restante desta subseção, descrevemos com mais detalhes cada um destes filtros.

Filtro de impossibilidades

O intuito deste filtro consiste em detectar e corrigir impossibilidades nos eventos recuperados da rede elétrica. Neste contexto, uma impossibilidade é caracterizada como uma inconsistência, que é detectada na fase de detecção e removida na de correção.

Este filtro é extremamente importante para remover eventos com ruído relacionados com problemas nos relés de sinalização, que são responsáveis pela sinalização de vários eventos espúrios; dentre eles: eventos sinalizando a abertura de um disjuntor que está aberto; ou mesmo, vários eventos de abertura e fechamento de um disjuntor em um mesmo instante de tempo.

Durante a fase de detecção, as regras de consistências utilizadas para localizar impossibilidades nos eventos recuperados da rede elétrica são as seguintes:

1) Um disjuntor ou uma chave não pode abrir e fechar várias vezes em um mesmo instante de tempo[6].

2) Um disjuntor ou uma chave aberta não pode abrir novamente[6].

3) Um disjuntor ou uma chave fechada não pode fechar novamente[6].

4) Um disjuntor ou uma chave inexistente no sistema elétrico não pode abrir ou fechar[7].

---

[6] Esta regra de consistência é importante para detectar eventos com ruído relacionados com problemas em relés de sinalização.

[7] Esta regra de consistência é importante para detectar eventos com ruído relacionados com UTRs cujas bases de dados encontram-se desatualizadas.

Na fase de correção, os eventos detectados como inconsistentes pela fase anterior são removidos e, em seguida, os equipamentos relacionados com eles são marcados, uma vez que esta informação poderá ser utilizada como uma heurística durante a fase de correção de outro filtro (posteriormente, descreveremos em maiores detalhes esta heurística).

Seja um exemplo de uma janela de tempo contendo os seguintes eventos: "Disjuntor D1 abriu no instante de tempo t_1", "Disjuntor D1 fechou em t_2", "Disjuntor D1 abriu em t_2" e "Disjuntor D2 fechou em t_3". Considerando que o disjuntor D1 estava aberto antes do início da janela de tempo e que o disjuntor D2 não existe, concluímos que três inconsistências serão detectadas durante a fase de detecção: a primeira, devido à sinalização da abertura do disjuntor (primeiro evento) que já estava aberto; a segunda, devido ao fechamento e à abertura do disjuntor D1 no mesmo instante de tempo (t_2); a terceira, devido à sinalização de um evento relacionado com um equipamento que não existe na rede elétrica (último evento). Quando o procedimento de correção entrar em execução, todos os eventos serão eliminados e, em seguida, o disjuntor D1 será marcado (note que o disjuntor D2 não o foi, pois ele não existe). Este último passo tem o intuito de facilitar o trabalho de correção de outro filtro, pois é mais provável um evento relacionado com um disjuntor marcado ser inconsistente que outro relacionado com outro disjuntor que nunca apresentou uma inconsistência no passado (uma heurística).

Filtro de sincronização

Para facilitar o entendimento deste filtro, faz-se necessário explicarmos onde são recuperados os eventos oriundos da rede elétrica, e as inconsistências existentes relacionadas com esta recuperação.

Os eventos recuperados da rede elétrica são apresentados pelo sistema SAGE em duas fontes distintas: SDE (Seqüência de Eventos) e ALR (Alarmes). A primeira fornece quase todos os eventos, enquanto a segunda, uma menor parte; no entanto, existe um conjunto de eventos que é fornecido em ambas as fontes, tais como aqueles relacionados com a abertura ou fechamento de disjuntores. O fato é que, devido à presença de ruído, nem sempre todos os eventos deste conjunto aparecem nestas duas

fontes de dados. Por exemplo, o evento "Disjuntor D1 abriu" pode se encontrar no SDE, mas não no ALR; da mesma forma, o inverso também pode ocorrer.

O objetivo do filtro de sincronização é o de receber os eventos provenientes do SDE e do ALR, detectar as possíveis inconsistências entre as duas fontes e, em seguida, corrigi-las. No contexto deste filtro, uma inconsistência é caracterizada pela falta de sincronia entre os eventos oriundos destas duas fontes. Um outro propósito deste filtro é o de evitar que tanto os filtros seguintes quanto o módulo de diagnóstico de falhas utilizem informação duplicada, uma vez que, para cada abertura de um disjuntor, por exemplo, dois eventos serão gerados: um no SDE e outro no ALR.

Durante a fase de detecção, a regra de consistência utilizada para localizar inconsistências é a seguinte:

1) Para cada evento de abertura ou de fechamento de um disjuntor existente no SDE, deve existir um correspondente no ALR (vice-versa).

Na fase de correção, as possíveis inconsistências descobertas na fase anterior são removidas, tanto inserindo-se novos eventos, de forma que, no final do procedimento de correção, os eventos oriundos das duas fontes estarão sincronizados. Para evitar que filtros posteriores utilizem informação duplicada, a saída da fase de correção deste filtro consiste em um dos fluxos de eventos, seja ele SDE, seja ALR, uma vez que, após a filtragem, serão idênticos.

Tomemos como exemplo a Tabela 4.1, a qual ilustra os eventos sinalizados no SDE e no ALR em uma determinada janela de tempo. Durante a fase de detecção, poderemos detectar duas inconsistências: uma, ao analisarmos o disjuntor D1, pois este tem sua abertura sinalizada no SDE e tanto a abertura como o fechamento no ALR; outra, ao analisar o disjuntor D2, pois este tem um fechamento seguido de uma abertura no SDE e apenas uma abertura no ALR. Quando o procedimento de correção entra em execução, o seguinte fluxo de eventos é gerado: "Disjuntor D1 abriu", "Disjuntor D2 fechou", "Disjuntor D2 abriu" e "Disjuntor D1 fechou".

	SDE	ALR
Eventos sinalizados	Disjuntor D1 abriu	Disjuntor D1 abriu
	Disjuntor D2 fechou	Disjuntor D2 abriu
	Disjuntor D2 abriu	Disjuntor D1 fechou

Tabela 4.1 – Exemplo das fontes SDE e ALR

Da mesma forma que no filtro anterior, todos os equipamentos detectados como inconsistentes são marcados.

Filtro de falha de disjuntores

Antes de entrarmos em detalhes sobre o funcionamento deste filtro, faz-se necessário detalhar um pouco o mecanismo de proteção do sistema elétrico.

Em geral, sistemas elétricos são compostos por diversos equipamentos caros que precisam ser protegidos, por exemplo, contra altas tensões. Uma forma de proteger estes equipamentos, quando situações desta natureza ocorrem, consiste em abrir automaticamente os disjuntores necessários para desenergizá-los. Todavia, para complicar a situação, disjuntores podem falhar ao abrir e, por conseguinte, comprometer vários equipamentos caros. Nestas circunstâncias, a manobra automática utilizada pelo sistema elétrico consiste em isolar estes equipamentos caros, abrindo todos os disjuntores necessários, com exceção do que falhou. Quando isto ocorre, vários disjuntores abrem levando à desenergização do barramento associado ao disjuntor que falhou, podendo causar um blecaute na subestação.

O objetivo deste filtro é o de detectar e corrigir inconsistências relacionadas com falhas de disjuntores. Neste contexto, uma inconsistência pode ser a falta da sinalização da abertura de um disjuntor que abriu, devido a uma falha de disjuntor; ou ainda a sinalização espúria da abertura do disjuntor que falhou.

Durante a fase de detecção, as seguintes regras de consistência são utilizadas para localizar inconsistências:

1) O estado de um disjuntor que falhou sempre é "fechado".

2) Quando um disjuntor falha, abrem todos os disjuntores necessários para desenergizar o barramento a ele associado.

É importante salientar que a segunda regra de consistência está apoiada na heurística segundo a qual as grandezas elétricas são mais confiáveis que as variáveis de estado, uma vez que, se ocorrer um evento sinalizando que um disjuntor falhou e o barramento a ele associado estiver energizado, o procedimento de detecção acreditará que a falha foi espúria.

Já na fase de correção, as inconsistências detectadas na fase anterior são corrigidas, seja eliminando um evento espúrio seja adicionando um perdido.

Seja a Figura 4.11 o modelo da rede após o término de uma janela de tempo contemplada pelos eventos "Disjuntor D3 falhou" e "Disjuntor D7 abriu". Quando o procedimento de detecção entrar em operação, uma inconsistência será detectada, uma vez que ocorreu uma falha no disjuntor D3 e o barramento B1 está desenergizado; entretanto, nem todas as aberturas dos disjuntores foram sinalizadas, como é o caso do disjuntor D6. O procedimento de correção elimina esta inconsistência adicionado na janela de tempo evento perdido "Disjuntor D6 abriu", veja a Figura 4.12.

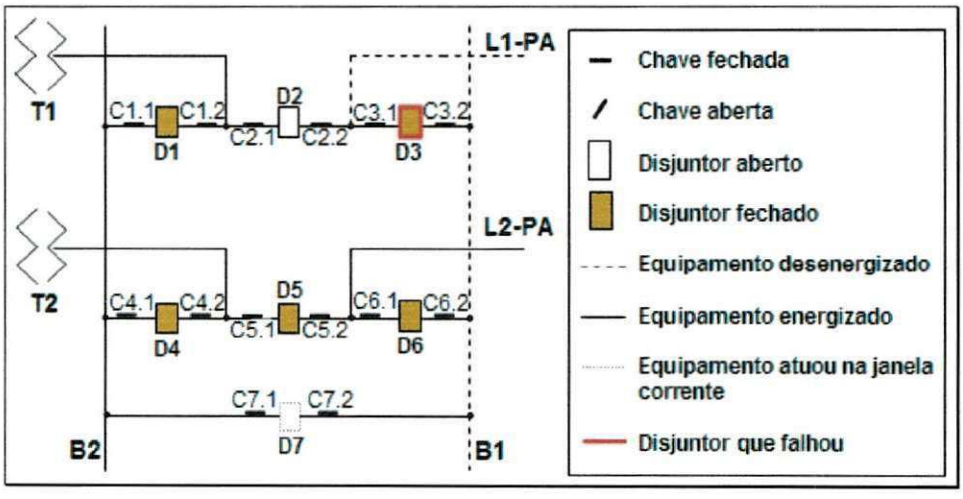

Figura 4.11 – Estado do modelo da rede se fosse atualizado com os eventos com ruído

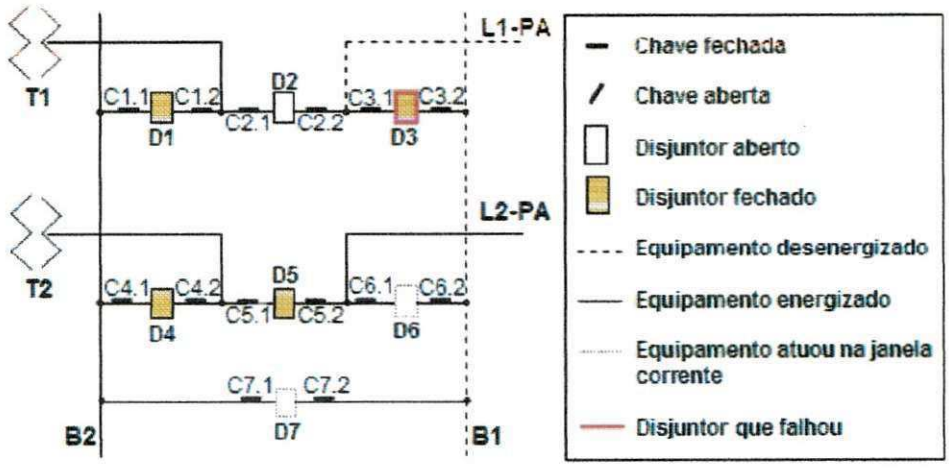

Figura 4.12 – Estado do modelo da rede atualizado com os eventos filtrados

Filtro de conectividade

O filtro de conectividade tem a responsabilidade de garantir a consistência do modelo da rede. Este é dito ser consistente, na ótica deste filtro, quando os relacionamentos de conectividade entre os equipamentos do modelo estão de acordo com todas as regras de consistência utilizadas por ele. Desta forma, se não for consistente o estado do modelo da rede atualizado com os eventos oriundos da rede, inconsistências serão detectadas durante a fase de detecção e corrigidas durante a de correção.

Cabe considerar que o filtro de conectividade garante apenas os relacionamentos de conectividade de linhas de transmissão, transformadores e barramentos. Demais equipamentos, tais como: banco de capacitores, reatores, compensadores síncronos e estáticos, que consistem na minoria dos equipamentos da rede elétrica, não são tratados por este filtro. Esta restrição deve-se ao fato de que estes equipamentos não possuem supervisão suficiente para a detecção de inconsistências por parte deste filtro. Por exemplo, dentre as informações fornecidas pelo sistema de supervisão, não existe nenhuma informação que possa ser utilizada para verificar se um evento de abertura do disjuntor de um reator está correto; no entanto, para se confirmar a veracidade um evento de abertura do disjuntor de uma linha, pode-se verificar se ele está correto, mediante uma análise das grandezas elétricas relacionadas com a linha em questão.

As regras de consistência utilizadas, durante a fase de detecção, por este filtro podem ser divididas em duas partes: uma associada com os relacionamentos de conectividade de terminais de linhas de transmissão, e outra com os relacionamentos de conectividade de transformadores.

As regras de consistências utilizadas para avaliar os relacionamentos de conectividade de um terminal de uma linha de transmissão são as seguintes:

1) Se um terminal de uma linha estiver energizado (potência ativa ou reativa forem diferentes de zero), estará conectado a pelo menos um equipamento energizado. Esta regra está apoiada na heurística segundo a qual as grandezas elétricas são mais confiáveis que as variáveis de estado.

2) Se o disjuntor de uma linha estiver aberto, o disjuntor de transferência desta mesma linha estiver fechado e nenhuma outra linha estiver utilizando o disjuntor de transferência, o terminal da linha estará sendo *bypassado*; logo, ele estará conectado a pelo menos um equipamento energizado. Note que esta regra se aplica mesmo quando as grandezas elétricas da linha indicarem que ela está desenergizada. Neste caso, as redundâncias associadas às variáveis de estado levam a acreditar que elas são mais confiáveis que as grandezas elétricas.

3) Se um terminal de uma linha estiver desenergizado e o estado de abertura dos disjuntores indicarem que o terminal da linha não está *bypassado*, ele não estará conectado a equipamentos energizados. Observe que, nesta regra, as redundâncias relacionadas com as grandezas elétricas e com as variáveis de estado estão convergindo para a mesma conclusão.

Com relação às regras de consistência utilizadas para avaliar os relacionamentos de conectividade de transformadores, podemos enumerá-las:

1) Se um transformador estiver energizado (potência ativa ou reativa forem diferentes de zero), estará conectado pelo menos a dois equipamentos energizados de tensões diferentes. Esta regra está

apoiada na heurística, segundo a qual, as grandezas elétricas são mais confiáveis que as variáveis de estado.

2) Se um transformador estiver desenergizado, ele estará conectado a apenas um ou nenhum equipamento energizado. Assim como a regra anterior, esta regra está apoiada na heurística, segundo a qual, as grandezas elétricas são mais confiáveis que as variáveis de estado.

É importante ressaltar que as regras de consistência utilizadas para garantir os relacionamentos de conectividade associados aos barramentos já estão inseridas nas de linhas de transmissão e transformadores, uma vez que os relacionamentos de conectividade associados a estes equipamentos são os mesmos que os associados aos barramentos.

Para ilustrar o funcionamento do procedimento de detecção, considere o modelo da rede da Figura 4.13 e uma janela de tempo contendo o evento "Disjuntor D3 abriu". O procedimento de detecção quando entrar em ação, detectará uma inconsistência, uma vez que, se o modelo da rede fosse atualizado com este evento, a linha L1 ficaria desconectada no terminal do lado PARA, o que não é verdade segundo as regras de consistência, pois a linha está energizada.

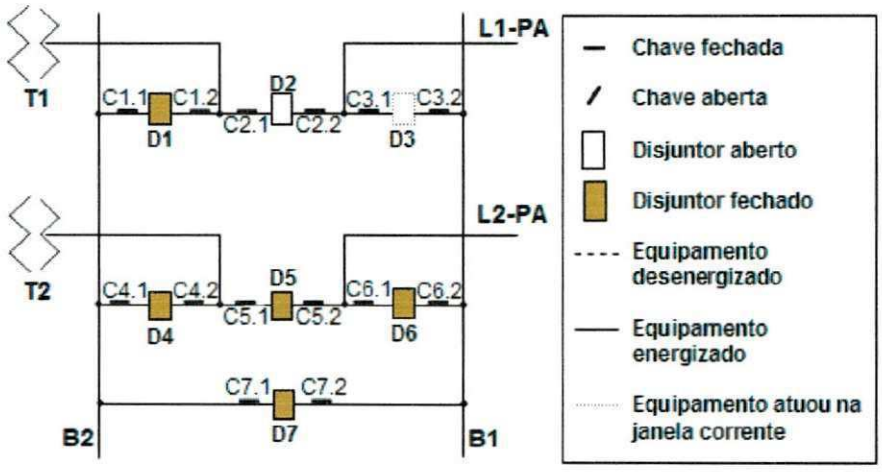

Figura 4.13 – Estado do modelo se fosse atualizado com os eventos com ruído

Note que, no exemplo anterior, a inconsistência detectada na linha L1 pode ser corrigida de duas formas: seja removendo-se o evento de abertura do disjuntor D3,

seja adicionando-se o evento de fechamento do disjuntor D2. Na primeira alternativa, o modelo da rede deixaria de ser atualizado, uma vez que o único evento da janela de tempo seria removido; na segunda alternativa, quando o modelo da rede fosse atualizado, a linha L1 ficaria conectada apenas ao barramento B2, uma vez que o disjuntor D2 ficaria fechado e o disjuntor D3, aberto.

Como podemos observar, para se corrigir uma inconsistência, podem existir várias alternativas. Desta forma, o procedimento de correção deve ser capaz de analisar cada uma delas e escolher a mais provável, isto é, a que mais se aproxima do estado real dos equipamentos da rede elétrica. Este processo de decisão é feito com base em um conjunto de heurísticas. Exemplos de algumas heurísticas que podem ser utilizadas durante a fase de correção para descobrir qual evento foi perdido ou gerado espuriamente são descritas a seguir:

1) É mais provável estar com ruído um evento relacionado com um equipamento que apresentou uma inconsistência detectada pelo filtro de impossibilidades ou de sincronização, do que um relacionado com outro equipamento que nunca apresentou uma inconsistência.

2) É mais provável estar com ruído um evento relacionado com um equipamento que atuou na janela corrente, do que em um relacionado com outro equipamento que não possui eventos presentes na janela de tempo.

3) É mais provável existir um evento com ruído do que dois ou mais.

4) Quando uma linha ou um transformador é desenergizado, é bem mais provável um evento de abertura de um disjuntor ter sido perdido do que um de abertura de uma chave.

Continuando o exemplo anterior, verificamos que, quando o procedimento de correção atuar, constatará, através de suas heurísticas, que é mais provável o ruído estar relacionado com o disjuntor D3 do que com o disjuntor D2, que não atuou na janela corrente. Desta forma, o evento sinalizando a abertura do disjuntor D3 será removido da janela corrente e o modelo da rede, por não ser mais atualizado, continuará consistente.

Um tipo de ruído bastante comum na rede elétrica da CHESF, por exemplo, está relacionado com linhas que não possuem supervisão na chave de *bypass* e cujos TCs localizam-se na bucha do disjuntor. Nestas circunstâncias, toda vez que uma linha desta natureza for *bypassada*, não haverá eventos sinalizando o fechamento da chave de *bypass* e as grandezas elétricas informadas pelo sistema de supervisão da rede estarão zeradas.

Tomando este problema como um exemplo para ilustrar o funcionamento do filtro de conectividade, considere a Figura 4.14, que apresenta o estado do modelo da rede após a janela de tempo contendo os eventos "Disjuntor D1 abriu" e "Disjuntor D3 fechou". Observe que o evento sinalizando o fechamento da chave C1.3 foi perdido e que o terminal do lado PARA da linha L1 está desenergizado. Diante deste cenário, o procedimento de detecção do filtro de conectividade, quando entrar em ação, notará que o terminal da linha está *bypassado* através de suas regras de consistência; portanto, deveria estar conectado a algum equipamento energizado. O procedimento de correção, quando for executado, adicionará na janela de tempo o evento "Chave C1.3 fechou".

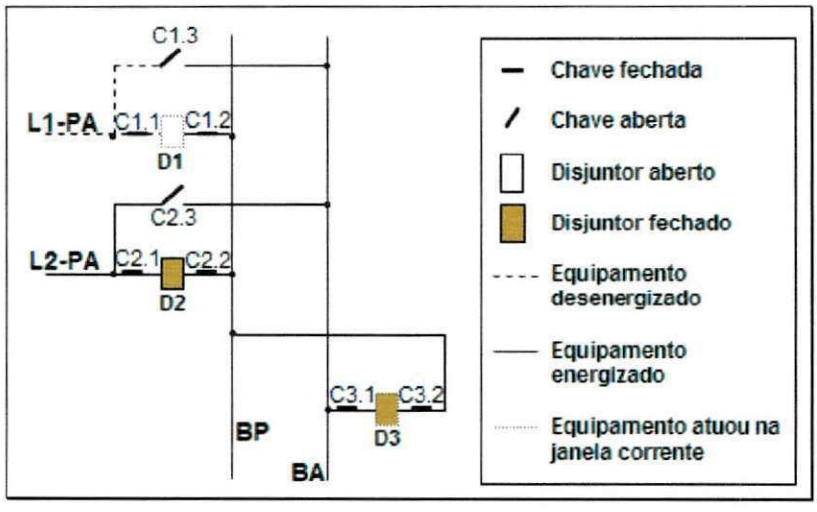

Figura 4.14 – Estado do modelo diante de eventos com ruído relacionados com uma manobra de *bypass*

5. Uma ferramenta robusta de tratamento de eventos em redes elétricas: implementação

Uma das maiores dificuldades encontradas durante o desenvolvimento desta ferramenta foi a falta de conhecimento sobre o funcionamento do sistema elétrico. Desta forma, à medida que este conhecimento se aprimorava, heurísticas e regras de consistência eram definidas ou modificadas; por conseguinte, tanto o código da ferramenta quanto os seus testes eram alterados com bastante freqüência. Além disso, colocar uma ferramenta desta natureza em operação é sempre um grande desafio. Na prática, é que, realmente, podemos observar as particularidades do sistema elétrico. A implementação desta ferramenta foi caracterizada por grandes descobertas e aprendizado, que só vieram a valorizar o trabalho desenvolvido.

Para facilitar a compreensão sobre a implementação desta ferramenta, este capítulo foi dividido em três seções. Primeiramente, descreveremos sucintamente a organização dela; em seguida, abordaremos detalhes associados à implementação do filtro de ruído; na última, apresentaremos como foi realizada a etapa de verificação da ferramenta.

5.1. Organização da ferramenta

Esta seção tem o objetivo de apresentar, de forma geral, a organização da ferramenta. Os componentes responsáveis pelo funcionamento da ferramenta foram organizados em pacotes de *software* de acordo com suas funcionalidades. Os principais pacotes são descritos na Tabela 5.1.

A Figura 5.2 mostra como estes pacotes estão organizados. As setas existentes na figura ilustram os relacionamentos de dependência entre dois pacotes.

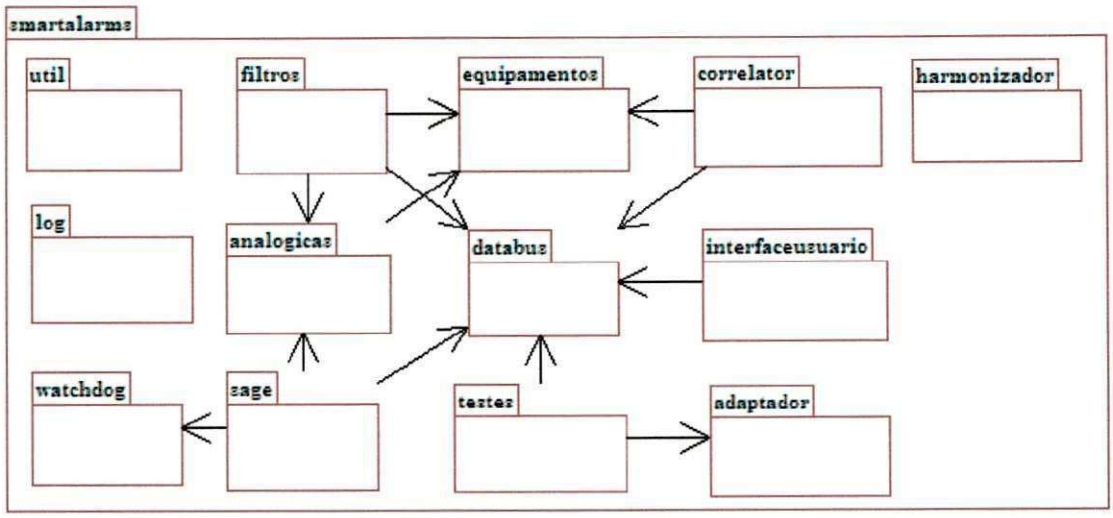

Figura 5.1 – Principais pacotes que compõem a ferramenta[8]

Pacote	Descrição	Desenvolvedor	Estatísticas
smartalarms. watchdog	Garante a estabilidade da ferramenta. Se a ferramenta parar de funcionar — por algum problema de rede ou de *software* —, é automaticamente reiniciada.	Alexandre[9]	6 classes 201 linhas de código
smartalarms.log	Realiza a persistência das informações geradas pela ferramenta.	Alexandre / Eloi	1 classe 107 linhas de código
smartalarms. filtros	Elimina o ruído presente nos eventos recuperados da rede. Este é o cerne deste capitulo.	Eloi	61 classes 3462 linhas de código
smartalarms.util	Oferece classes de utilidade geral para todas as outras classes que compõem a ferramenta.	Alexandre / Eloi	16 classes 775 linhas de código

[8] Os pacotes **smartalarms.util** e **smartalarms.log** são utilizados por todos os outros pacotes. Para evitar uma grande quantidade de setas na figura, estes pacotes aparecem isolados.

[9] Alexandre Nóbrega Duarte foi o responsável pelo desenvolvimento do *SmartOne*.

smartalarms. analogicas	Atualiza o modelo da rede com todas as grandezas elétricas recuperadas da rede, como também informa que equipamentos estão ou não energizados.	Eloi	6 classes 359 linhas de código
smartalarms.sage	Recebe todos os eventos e grandezas elétricas do sistema de supervisão, e os distribui com os demais componentes da ferramenta. Além disso, permite simular eventos recebidos do SAGE para permitir testes da ferramenta.	Alexandre / Eloi	8 classes 922 linhas de código
smartalarms. equipamentos	Mantém o modelo da rede elétrica, isto é, todos os equipamentos da rede e os relacionamentos de conectividade entre eles.	Alexandre	46 classes 1826 linhas de código
smartalarms. databus	Realiza a comunicação entre os diversos componentes da ferramenta.	Alexandre	4 classes 44 linhas de código
smartalarms. testes	Testa os componentes responsáveis pelo funcionamento da ferramenta, por meio de testes de unidade, de aceitação e de regressão.	Alexandre / Eloi	280 classes 3932 linhas de código
smartalarms. correlator	Realiza o diagnóstico de falhas na rede elétrica.	Alexandre	208 classes 3005 linhas de código
smartalarms. interfaceusuario	Fornece uma interface gráfica que permite simular um conjunto de eventos, como também apresenta os diagnósticos efetuados pela ferramenta.	Alexandre / Eloi	32 classes 2884 linhas de código
smartalarms. adaptador	Fornece uma fonte transparente de dados para o funcionamento do módulo de simulação da ferramenta, permitindo que os dados possam ser recuperados de repositórios diferentes, como arquivos e banco de dados.	Alexandre / Eloi	11 classes 586 linhas de código

smartalarms. harmonizador	Remove os problemas de padronização dos eventos recuperados da rede.	Alexandre / Eloi	4 classes 993 linhas de código

Tabela 5.1 – Pacotes que compõem a ferramenta robusta

A seção seguinte descreve em maiores detalhes a implementação do filtro de ruído.

5.2. Implementação do filtro de ruído

No capítulo anterior, apresentamos a arquitetura da ferramenta robusta desenvolvida, descrevendo os seus principais componentes, com destaque para o filtro de ruído, que recebe os eventos da rede elétrica através do barramento de eventos, remove o ruído presente nesses eventos e, no final, insere os eventos filtrados no barramento de eventos filtrados. Nesta seção, explicaremos como o filtro de ruído foi implementado.

A implementação do filtro de ruído levou nove meses. Foi utilizado o *software* Eclipse 2.1.1 (ECLIPSE, 2003) como ambiente de desenvolvimento. Foram implementadas 163 classes Java, incluindo as classes feitas para testar o filtro, totalizando aproximadamente 4432 linhas de código.

As principais classes e interfaces responsáveis pelo funcionamento do filtro de ruído se encontram dentro do pacote **smartalarms.filtros**. Internamente, este pacote é composto por seis outros, organizados de acordo com o funcionamento dos quatro filtros que compõem o filtro de ruído (veja a Tabela 5.2).

Pacote	Descrição	Estatísticas
smartalarms.filtros. filtro_impossibilidades	Contém todas as classes responsáveis por filtrar o ruído relacionado com impossibilidades existente nos eventos recuperados do sistema elétrico.	4 classes 202 linhas de código
smartalarms.filtros. filtro_sincronizacao	Contém a implementação do filtro de sincronização, que é responsável por sincronizar os eventos de abertura e fechamento de disjuntores oriundos das fontes SDE e ALR.	3 classes 105 linhas de código

smartalarms.filtros. filtro_fldj	Contém a implementação do filtro de falha de disjuntores, que é responsável por eliminar ruído relacionado com eventos de abertura e fechamento de disjuntores, em ocorrências envolvendo falhas de disjuntores.	2 classes 114 linhas de código
smartalarms.filtros. filtro_conectividade	Contém todas as classes e interfaces responsáveis por garantir a consistência do modelo da rede, quando este é atualizado com os eventos filtrados.	21 classes 1136 linhas de código
smartalarms.filtros. estimador	As classes existentes dentro deste pacote são utilizadas tanto pelo filtro de falha de disjuntores quanto pelo de conectividade. Elas são responsáveis por estimar a tensão de alguns equipamentos que não possuem supervisão na rede elétrica, como também de informar, diante de ruído, se determinadas linhas de transmissão estão ou não *bypassadas*.	2 classes 149 linhas de código
smartalarms.filtros. simulador	As classes existentes dentro deste pacote são utilizadas tanto pelo filtro de falha de disjuntores quanto pelo de conectividade. Elas permitem verificar se dois equipamentos estão conectados entre si com base em um conjunto de alterações topológicas inerente aos eventos recuperados da rede elétrica.	15 classes 1234 linhas de código

Tabela 5.2 – Pacotes que compõem o filtro de ruído

A Figura 5.2 ilustra como estes pacotes estão relacionados.

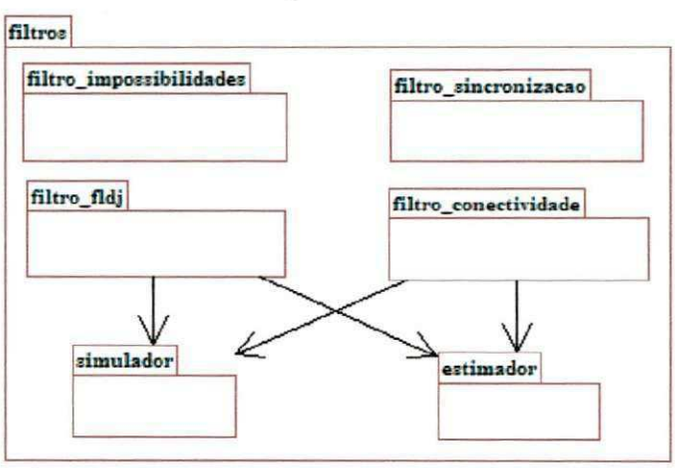

Figura 5.2 – Organização interna do pacote smartalarms.filtros

Antes de detalharmos os pacotes que compõem o pacote **smartalarms.filtros**, faz-se necessário explicar as principais classes e interfaces existentes neste pacote, uma vez que elas são importantes para o entendimento dos demais.

Para facilitar a compreensão do pacote **smartalarms.filtros**, dividiremos sua apresentação em três partes. Na primeira parte, descrevemos as principais classes e interfaces responsáveis pela comunicação do filtro de ruído com os barramentos que compõem a arquitetura da ferramenta; na segunda, apresentaremos as classes e interfaces que compõem a implementação do filtro de ruído; na terceira, descreveremos a implementação das fases de detecção e correção do filtro de ruído.

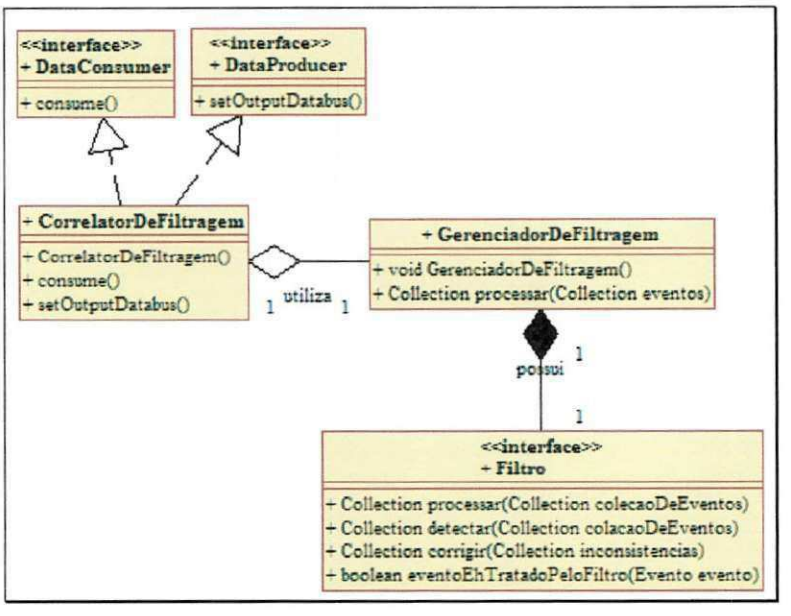

Figura 5.3 – Diagrama de classes do pacote smartalarms.filtros (parte I)

Observe a Figura 5.3 para compreender as principais classes e interfaces responsáveis pela comunicação do filtro de ruído com os barramentos que compõem a arquitetura da ferramenta. A classe **CorrelatorDeFiltragem** é responsável por recuperar os eventos oriundos da rede elétrica do barramento de eventos, filtrar o ruído presente nos eventos e, no final, inserir os eventos filtrados no barramento de eventos filtrados. Para permitir que a classe **CorrelatorDeFiltragem** receba eventos de um barramento e insira eventos em outro, ela implementa duas interfaces: **DataConsumer** e **DataProducer** (ambas pertencentes ao pacote **smartalarms.databus**). O mecanismo de filtragem é iniciado quando a instância da classe **CorreladorDeFiltragem** executa o método **processar** da classe

GerenciadorDeFiltragem, que, por sua vez, comunica-se com o filtro de ruído, representado pela interface **Filtro**.

O filtro de ruído é constituído por quatro filtros seqüenciais: filtro de impossibilidades, filtro de sincronização, filtro de falha de disjuntores e filtro de conectividade. A filtragem do ruído presente nos eventos recuperados do gerenciador de filtragem, realizada pelo filtro de ruído, é feita seqüencialmente, isto é, os eventos são processados, sucessivamente, pelos quatro filtros que compõem o filtro de ruído (observe a Figura 5.4). Note que o processamento de **cada** um consiste em remover o ruído relacionado com a sua especialidade e, em seguida, enviar os eventos filtrados para o filtro seguinte (se este existir), para que este realize a filtragem do ruído relacionado com a sua especialidade. Desta forma, este processo repete-se até que não exista mais nenhum filtro seguinte. Quando isto ocorre, os eventos filtrados são retornados sucessivamente pela cadeia de filtros, até chegar ao gerenciador de filtragem.

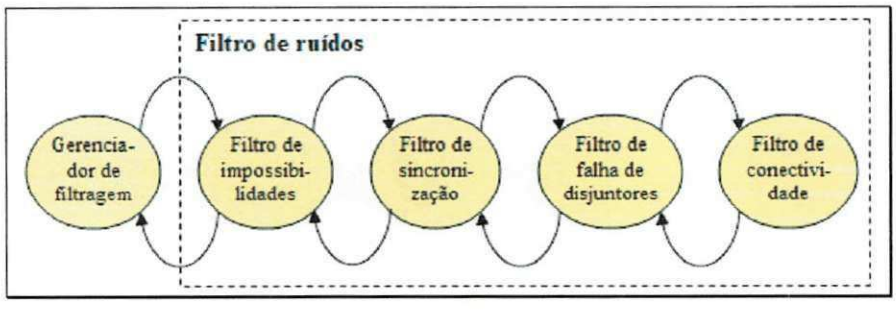

Figura 5.4 – Comunicação entre o gerenciador de filtragem e o filtro de ruído

A implementação do filtro de ruído foi baseada no padrão *Decorator* descrito em GAMMA (1999). A utilização possibilitou a transparência entre os filtros que compõem o filtro de ruído, isto é, eles comunicam-se entre si, sem saber com quem se estão comunicando. Além disso, do ponto de vista de um observador externo, como, por exemplo, para uma instância da classe **GerenciadorDeFiltragem**, os quatro filtros podem ser observados como um único, uma vez que o observador externo se comunica apenas com um filtro e este, com o filtro seguinte e assim, sucessivamente.

Em termos de *design*, cada filtro estende a classe abstrata **AbstractFiltro**, que, por sua vez, implementa a interface **Filtro** (veja a Figura 5.5). Note que cada filtro, ao estender esta classe abstrata, herda todos os métodos necessários para o

funcionamento do filtro, com exceção dos relacionados com a sua especialidade. Além disso, estendendo esta classe abstrata, ele pode estar conectado a qualquer outro. Desta forma, quando uma instância da classe **GerenciadorDeFiltragem**, por exemplo, envia um conjunto de eventos para uma instância da classe **FiltroImpossibilidades**, o ruído dos eventos relacionados com impossibilidades do sistema elétrico é removido e, em seguida, os eventos filtrados são enviados para o filtro seguinte, no caso, uma instância da classe **FiltroDeSincronizacao**.

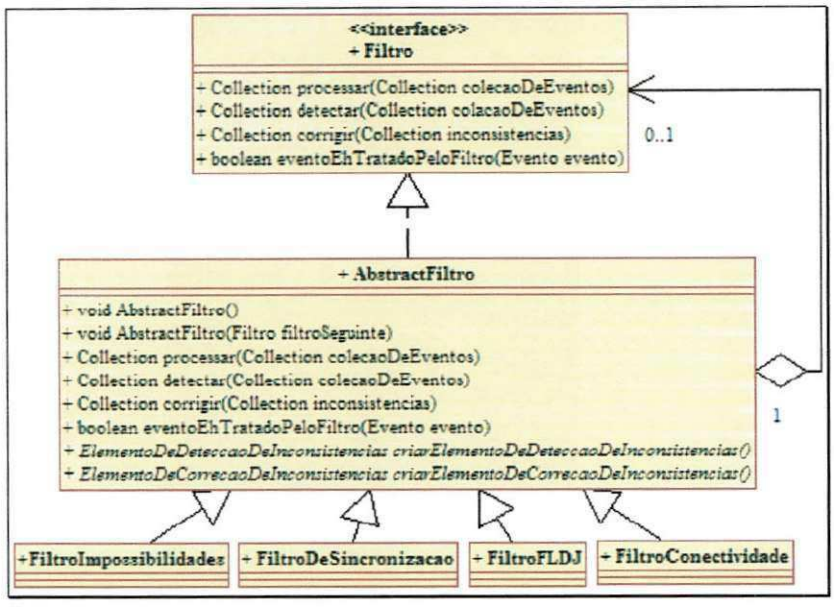

Figura 5.5 – Diagrama de classes do pacote smartalarms.filtros (parte II)

Antes de explicarmos como foram implementadas as fases de detecção e correção de cada filtro, primeiramente devemos explicar com clareza como elas funcionam. Cada filtro possui pelo menos um elemento de detecção de inconsistências e pelo menos um elemento de correção de inconsistências. Os primeiros são responsáveis por detectar todas as inconsistências relacionadas com a especialidade do filtro em que atua, enquanto os últimos corrigem as inconsistências detectadas pelos primeiros, seja removendo eventos espúrios seja adicionando eventos perdidos.

Duas interfaces são utilizadas para implementar os elementos de detecção e correção de inconsistências (veja a Figura 5.6):

1) **ElementoDeDeteccaoDeInconsistencias** – Esta interface é implementada por todos os elementos de detecção de inconsistências.

Quando uma inconsistência é detectada por um elemento desta natureza, uma instância de uma classe que implementa a interface **Inconsistencia** é gerada. Além disso, cada inconsistência está relacionada pelo menos com um evento com ruído, cada um consistindo de uma instância da classe **EventoInconsistente**.

2) **ElementoDeCorrecaoDeInconsistencias** – Todo elemento de correção de inconsistências implementa esta interface. As inconsistências corrigidas por estes elementos são detectadas pelos elementos de detecção de inconsistências, e consistem em implementações da interface **Inconsistencia**.

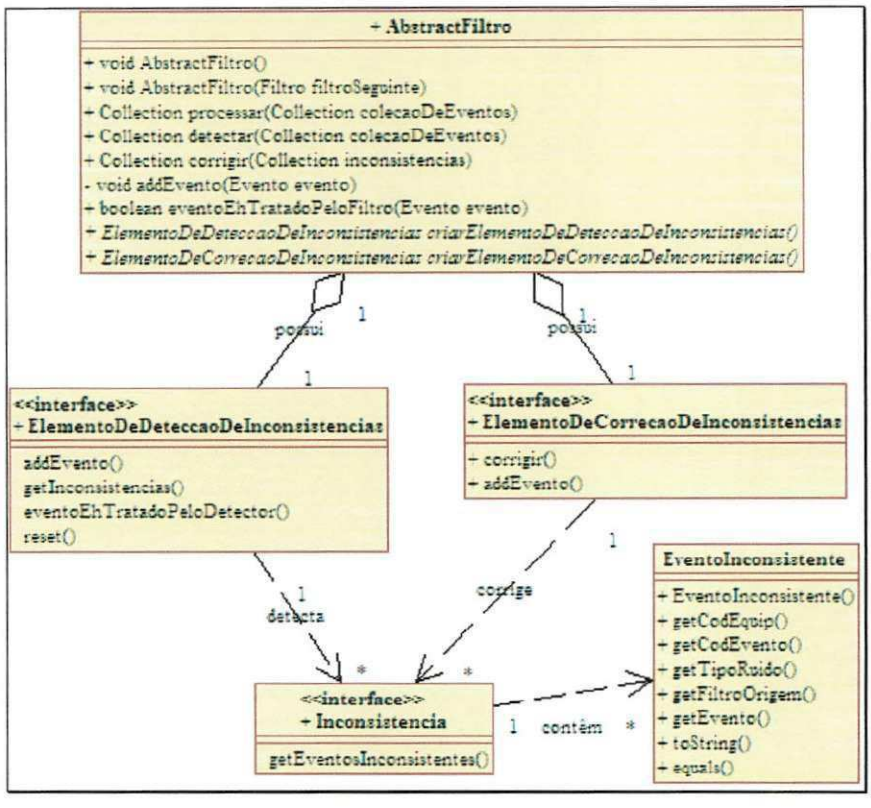

Figura 5.6 – Diagrama de classes do pacote smartalarms.filtros (parte III)

É importante notar que cada filtro pode possuir vários elementos de detecção de inconsistências; no entanto, a classe abstrata **AbstractFiltro** ilustrada na Figura 5.6 só está relacionada com um único elemento de detecção de inconsistências (representado pela interface **ElementoDeDeteccaoDeInconsistencias**). Para permitir que esta classe abstrata possua vários elementos de detecção de inconsistências, foi

utilizada uma solução baseada no padrão *Composite*, descrito em GAMMA (1999). Esta solução consistiu em implementar uma classe abstrata chamada **AbstractElementoCompostoDeDeteccaoDeInconsistencias**, a qual implementa a interface **ElementoDeDeteccaoDeInconsistencias** e, ao mesmo tempo, possui um conjunto de elementos de detecção de inconsistências, que também implementam a interface **ElementoDeDeteccaoDeInconsistencias**. Desta forma, cada filtro que possuir um elemento de detecção de inconsistências, estenderá a classe abstrata **AbstractElementoCompostoDeDeteccaoDeInconsistencias**; logo, possuirá também um conjunto de elementos de detecção de inconsistências (veja a Figura 5.7).

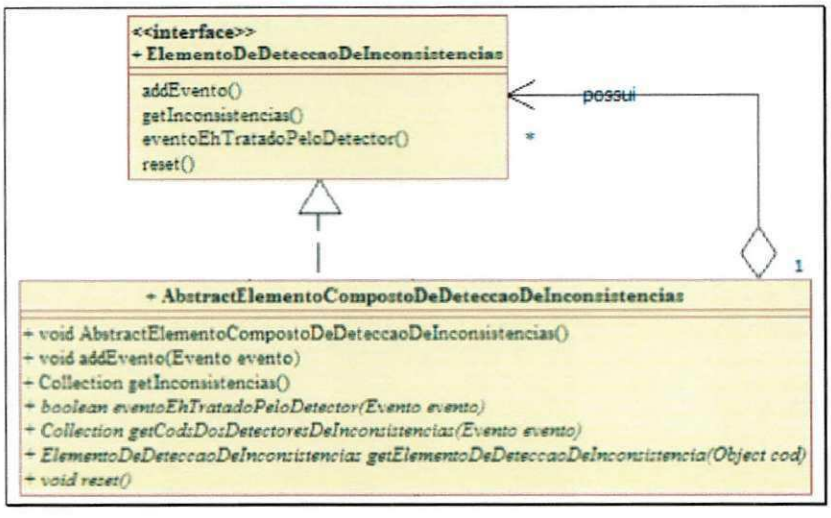

Figura 5.7 – Diagrama de classes de uma classe abstrata que possui vários elementos de detecção de inconsistências

Durante as fases de detecção e correção de inconsistências, existem duas classes no pacote **smartalarms.filtros,** que são utilizadas por alguns filtros (observe a Figura 5.8) São elas:

1) **InconsistenciaSimples** – Esta classe consiste em uma implementação extremamente simples de uma inconsistência. Cada instância desta classe está associada apenas a um evento com ruído, que é passado como parâmetro pelo seu construtor.

2) **ElementoSimplesDeCorrecaoDeInconsistencias** – Esta classe consiste em uma implementação de um elemento de correção de inconsistências. Seu funcionamento consiste em receber uma coleção

de inconsistências, e, em seguida, corrigir o ruído relacionado com estas inconsistências, seja removendo eventos espúrios seja adicionando eventos perdidos.

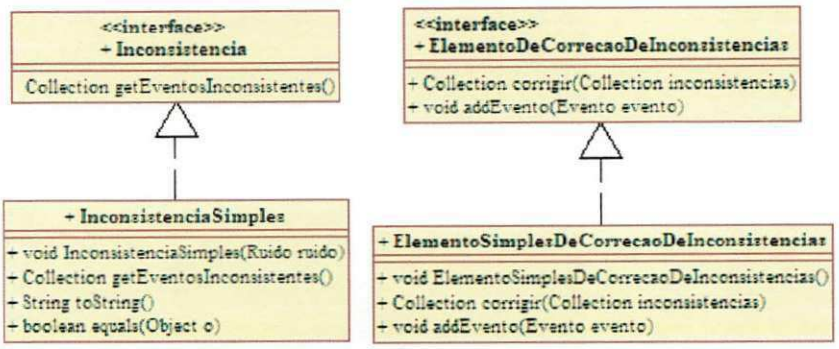

Figura 5.8 – Classes existentes no pacote smartalarms.filtros que são utilizadas em outros pacotes

As subseções seguintes explicam em maiores detalhes a implementação de cada um dos filtros que compõem o filtro de ruído.

5.2.1. Implementação do filtro de impossibilidades

O filtro de impossibilidades é composto por vários elementos de detecção de inconsistências e por um elemento de correção de inconsistências. Cada elemento de detecção de inconsistências é responsável por analisar os eventos oriundos da rede elétrica associados, com um determinado disjuntor ou chave, com o intuito de detectar possíveis inconsistências relacionadas com impossibilidades no sistema elétrico. No contexto deste filtro, cada inconsistência está relacionada com um evento espúrio. O processo de detecção destas inconsistências é baseado em um conjunto de regras de inconsistências definidas no capítulo anterior. As inconsistências detectadas pelos elementos de detecção são enviadas aos elementos de correção, que são responsáveis por corrigi-las, isto é, remover da janela de tempo os eventos espúrios relacionados com as inconsistências.

A Figura 5.9 ilustra o funcionamento do filtro de impossibilidades. De um lado, temos duas janelas de tempo contendo um conjunto de eventos, antes e depois do processamento do filtro de impossibilidades, como também o estado de abertura dos disjuntores associados a estes eventos, antes do início da janela de tempo. De outro temos os elementos de detecção e correção, utilizados durante o processamento

do filtro de impossibilidades, assim como as inconsistências por eles detectadas. Note que os sete eventos existentes na janela de tempo analisada pelo filtro estão associados a três disjuntores. Desta forma, durante a fase de detecção, três elementos de detecção de inconsistências são ativados para tratar destes eventos, um para cada disjuntor. Observe, por exemplo, que o elemento de detecção ED_2 está relacionado com o disjuntor D2; logo, todos os eventos associados com este disjuntor são analisados por este elemento de detecção. As inconsistências detectadas durante a fase de detecção consistem no conjunto de todas as inconsistências detectadas pelos elementos de detecção de inconsistências. O elemento de detecção ED_1, por exemplo, não encontrou nenhuma inconsistência relacionada com o disjuntor D1; já o elemento de detecção ED_2 encontrou uma inconsistência em um evento relacionado com o disjuntor D2, uma vez que ele estava fechado e surgiu um evento, dentro da janela de tempo, sinalizando o seu fechamento. Por último, o elemento de detecção ED_3 detectou quatro inconsistências relacionadas com os eventos associados ao disjuntor D3, uma vez que ocorreram várias sinalizações de abertura e fechamento do disjuntor no mesmo instante de tempo. No final do processamento dos elementos de detecção, todas as inconsistências detectadas pelos elementos de detecção serão enviadas para o elemento de correção EC_1, que se encarrega de corrigi-las. Observe, no final da fase de correção, que as cinco inconsistências detectadas durante a fase de detecção resultaram na remoção de cinco eventos da janela de tempo.

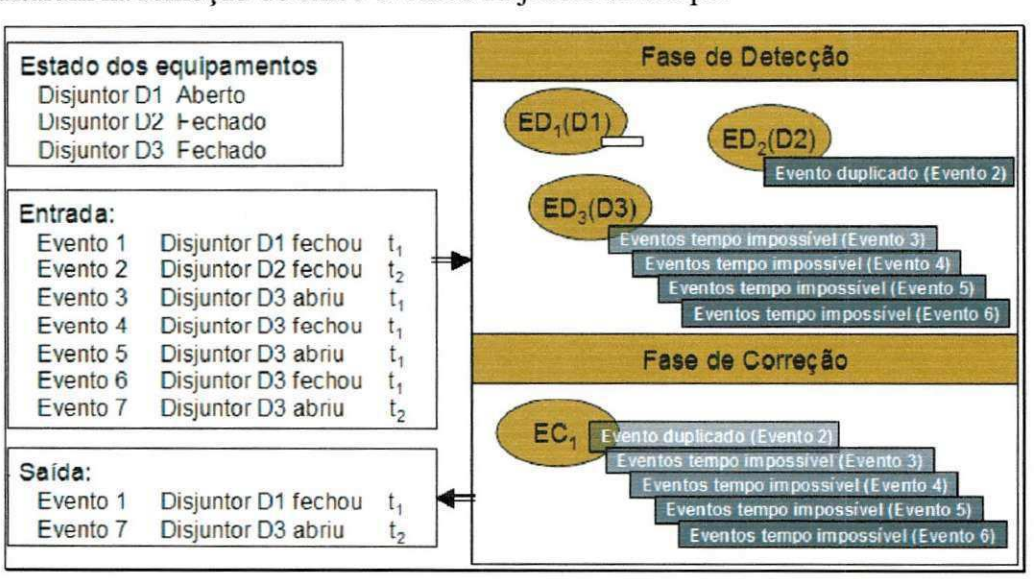

Figura 5.9 – Elementos de detecção de correção de inconsistências do filtro de impossibilidades

Em termos de implementação, a classe **FiltroImpossibilidades** ilustrada na Figura 5.5 é responsável por instanciar as classes responsáveis pelo processamento dos elementos de detecção e correção de inconsistências.

A Figura 5.10 ilustra as classes responsáveis pelo processamento dos elementos de detecção de inconsistências. São elas:

1) **ElementoDeDeteccaoDeInconsistenciasFiltroImpossibilidades** – Esta classe é responsável por analisar os eventos de abertura e fechamento relacionados com um determinado disjuntor ou chave da rede elétrica, com o intuito de descobrir possíveis inconsistências nestes eventos.

2) **ElementoCompostoDeDeteccaoDeInconsistenciasFiltroImpossibilidades** – Esta classe estende a classe abstrata **AbstractElementoCompostoDeDeteccaoDeInconsistencias**. Ela é responsável por instanciar todos os elementos de detecção de inconsistências necessários para a detecção de inconsistências nos eventos recuperados da rede elétrica. Estes elementos são instâncias da classe **ElementoDeDeteccaoDeInconsistenciasFiltroImpossibilidades**.

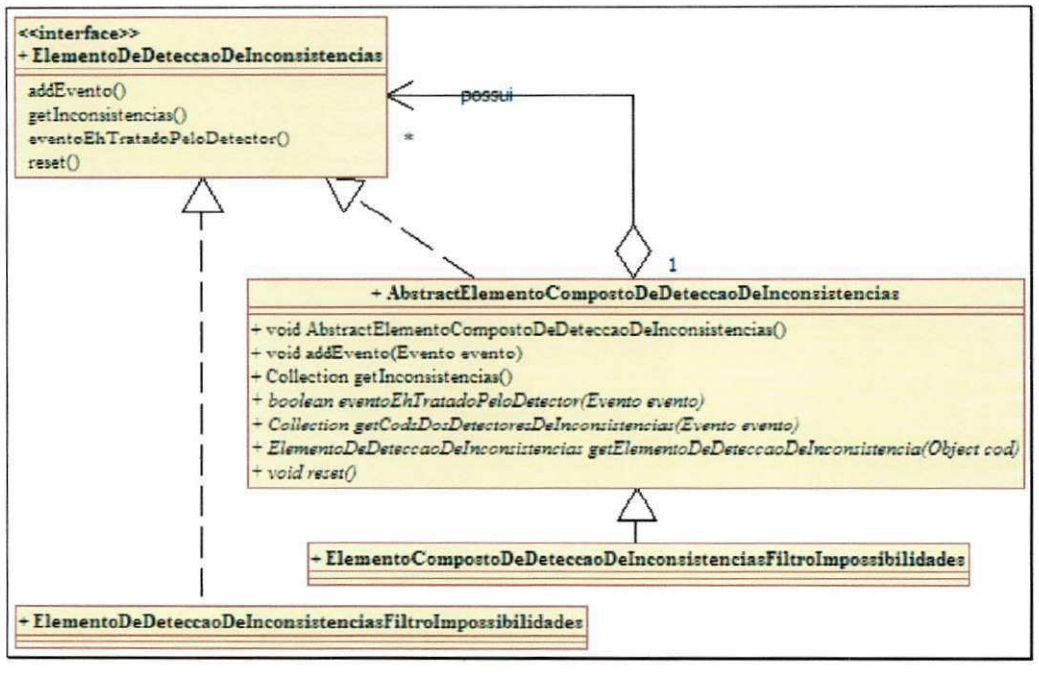

Figura 5.10 – Diagrama de classes da fase de detecção do filtro de impossibilidades

As inconsistências detectadas pelos elementos de detecção de inconsistências consistem em instâncias da classe **InconsistenciaSimples**, localizada no pacote **smartalarms.filtros**. Já o elemento de correção de inconsistências, responsável pela execução da fase de correção, consiste em uma instância da classe **ElementoSimplesDeCorrecaoDeInconsistencias**, também localizada no pacote **smartalarms.filtros**.

5.2.2. Implementação do filtro de sincronização

Assim como o filtro de impossibilidades, o filtro de sincronização é composto por um conjunto de elemento de detecção de inconsistências e um elemento de correção de inconsistências. Cada elemento de detecção de inconsistências é responsável por analisar um conjunto de eventos — oriundos tanto da fonte SDE quanto da fonte ALR — associados com um determinado disjuntor, com o objetivo de detectar possíveis inconsistências relacionadas com a falta de sincronização entre essas duas fontes. Já o elemento de correção de inconsistências corrige as inconsistências detectadas pelos elementos de detecção.

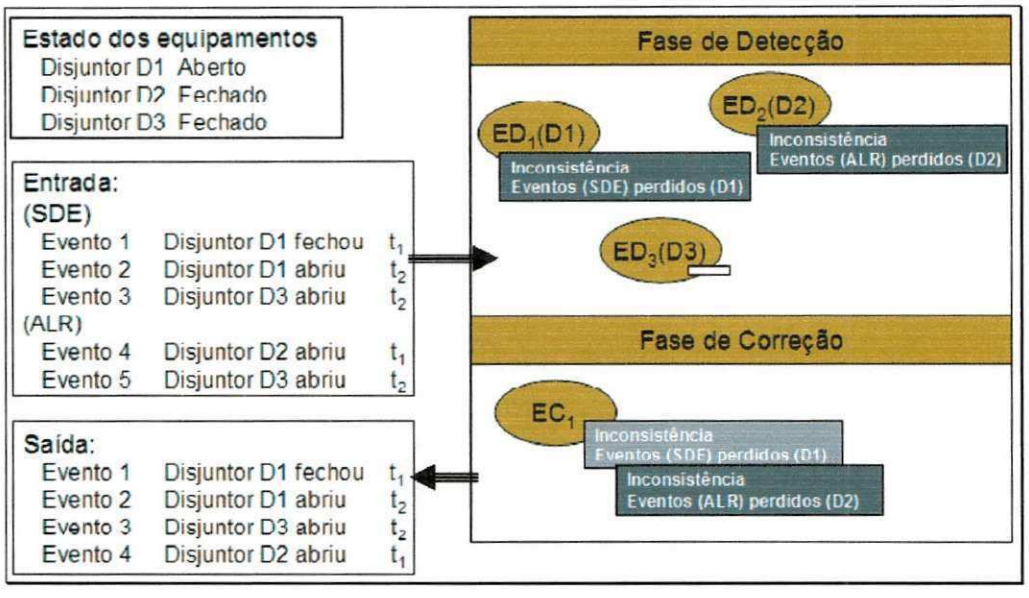

Figura 5.11 – Elementos de detecção e correção do filtro de sincronização

A Figura 5.11 ilustra o funcionamento deste filtro. Do lado esquerdo, encontram-se duas janelas de tempo, uma contendo os eventos a serem filtrados pelo filtro de sincronização e outra contendo os eventos filtrados por ele; do lado direito,

os elementos de detecção e correção de inconsistências utilizados por este filtro durante o seu processamento. Note que os eventos existentes na janela de tempo estão relacionados com dois tipos distintos de fonte; além disso, eles estão associados com três diferentes disjuntores. Quando a fase de detecção iniciar, três elementos de detecção de inconsistências serão ativados para o tratamento dos eventos relacionados com estes disjuntores. Observe que o elemento de detecção ED_2 está associado ao disjuntor D2, além disso, quando entrar em ação, ele detectará uma inconsistência, uma vez que existe um evento na fonte ALR sinalizando a abertura do disjuntor, enquanto o mesmo não ocorre na fonte SDE, o que caracteriza uma inconsistência relacionada com a falta de sincronização entre as fontes com relação aos eventos deste disjuntor em particular. O processamento do elemento de correção de inconsistências consiste em receber todas as inconsistências detectadas pelos três elementos de detecção de inconsistências e, em seguida, corrigi-las. Observando novamente a figura, podemos notar que foram detectadas duas inconsistências, uma pelo elemento de detecção ED_1 e outra pelo elemento de detecção ED_2. O processamento do filtro de sincronização termina quando o elemento de correção corrige estas duas inconsistências. Note que os eventos existentes na janela de tempo, que antes pertenciam a fontes distintas, agora consistem em um único fluxo **sincronizado** de eventos.

A classe **FiltroSincronizacao**, ilustrada na Figura 5.5, é responsável por instanciar todas as classes necessárias para o funcionamento dos elementos de detecção e correção de inconsistências.

Com relação aos elementos de detecção de inconsistências, as classes responsáveis pelo funcionamento destes elementos são as seguintes (veja a Figura 5.12):

1) **ElementoDeDeteccaoDeInconsistenciasFiltroSincronizacao** – Uma instância desta classe analisa conjunto de eventos (oriundos tanto da fonte SDE, como ALR) relacionados com um determinado disjuntor, com o intuito de detectar possíveis inconsistências. Cada inconsistência detectada consiste em uma instância da classe **InconsistenciaSimples**, localizada no pacote **smartalarms.filtros**.

2) **ElementoCompostoDeDeteccaoDeInconsistenciasFiltroSincronizacao** – Esta classe estende a classe abstrata **AbstractElementoCompostoDeDeteccaoDeInconsistencias**. Ela é responsável por instanciar todos os elementos de detecção de inconsistências necessários para a detecção de inconsistências nos eventos recuperados da rede elétrica. Estes elementos são instâncias da classe **ElementoDeDeteccaoDeInconsistenciasFiltroSincronizacao**.

O elemento de correção de inconsistências utilizado pelo filtro de sincronização, assim como ocorre no filtro de impossibilidades, consiste em uma instância da classe **ElementoSimplesDeCorrecaoDeInconsistencias**, localizado no pacote **smartalarms.filtros**.

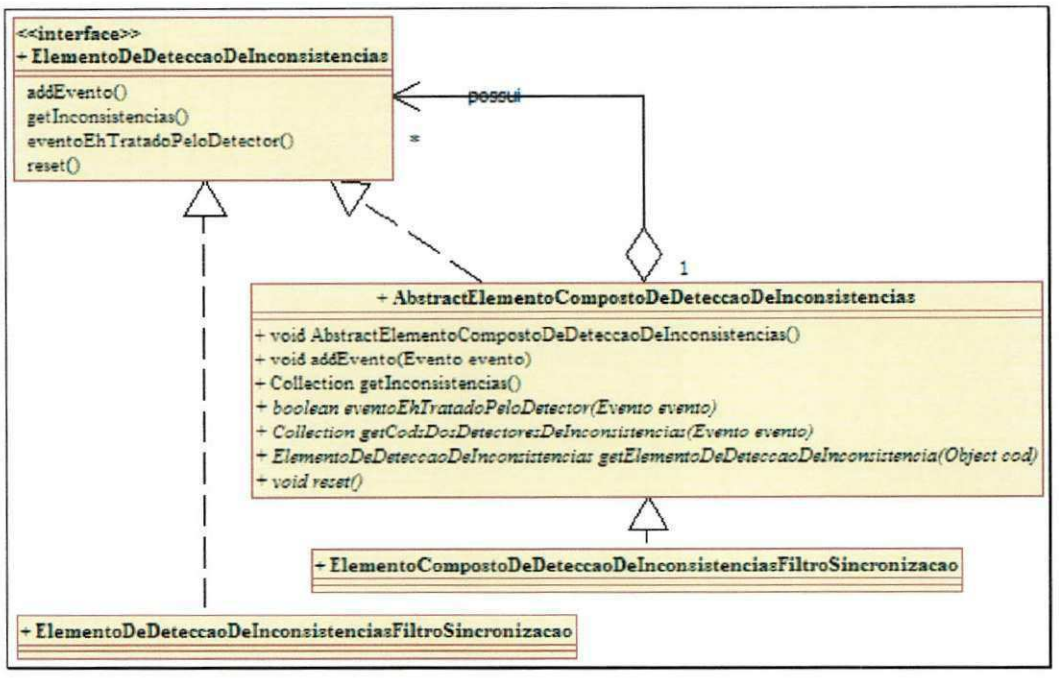

Figura 5.12 – Diagrama de classes da fase de detecção do filtro de sincronização

5.2.3. Implementação do filtro de falha de disjuntores

O filtro de falha de disjuntores é composto apenas por um elemento de detecção de inconsistências e um elemento de correção de inconsistências. Juntos, estes elementos são responsáveis pelo funcionamento deste filtro. O elemento de detecção de inconsistências analisa os eventos recuperados da rede elétrica com intuito de detectar possíveis inconsistências relacionadas com falhas de disjuntores.

As regras de consistência utilizadas durante a fase de detecção foram abordadas no capítulo anterior. Já o elemento de correção de inconsistências corrige o ruído relacionado com as inconsistências detectadas pelo elemento de detecção.

A Figura 5.13 ilustra o modelo da rede após o término de uma janela de tempo contendo os eventos "Disjuntor D3 falhou" e "Disjuntor D7 abriu". Note que o cenário exposto na figura referida está relacionado com a tentativa, sem sucesso, de abertura do disjuntor D3, diante de um desarme da linha L1. Como conseqüência, todos os disjuntores necessários para isolar a linha L1, com exceção do que falhou, abriram. O elemento de detecção ED_1, ilustrado na figura, quando entrar em ação, detectará uma inconsistência, pois nem todos os disjuntores necessários para isolar o terminal da linha L1 abriram, como é o caso, por exemplo, do disjuntor D6. A fase de correção é caracterizada pela atuação do elemento de correção de inconsistências, que recebe a inconsistência detectada pelo elemento de correção e, em seguida, adiciona à janela de tempo o evento "Disjuntor D6 abriu".

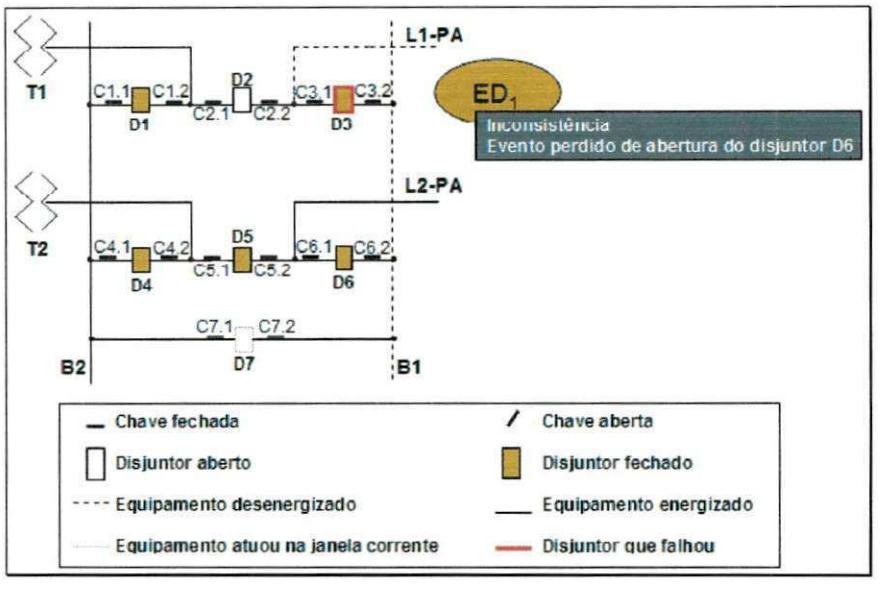

Figura 5.13 – Elemento de detecção de inconsistências do filtro de falha de disjuntores

Com relação à implementação deste filtro, a classe **FiltroFLDJ**, ilustrada na Figura 5.5, é responsável por instanciar todas as classes necessárias para o funcionamento dos elementos de detecção e correção de inconsistências.

O elemento de detecção de inconsistências utilizado por este filtro foi implementado na classe **ElementoDeDeteccaoDeInconsistenciasFiltroFLDJ** (veja a

Figura 5.14). As inconsistências detectadas por este filtro são instâncias da classe **InconsistenciaSimples**, localizada no pacote **smartalarms.filtros**. Já o elemento de correção de inconsistências, assim como ocorre nos filtros anteriores, consiste em uma instância da classe **ElementoSimplesDeCorrecaoDeInconsistencias**, também localizada no pacote **smartalarms.filtros**.

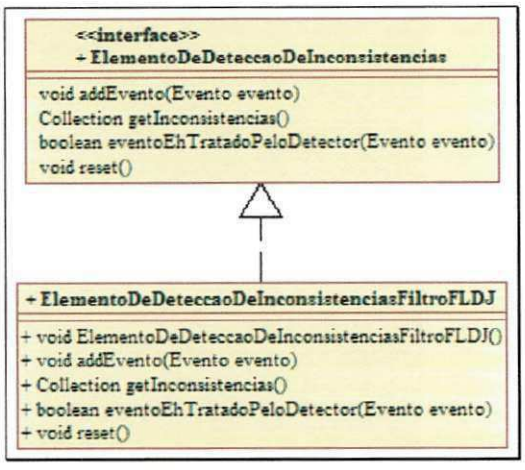

Figura 5.14 – Diagrama de classes do elemento de detecção de inconsistências utilizado pelo filtro de conectividade

5.2.4. Implementação do filtro de conectividade

Antes de entrarmos em detalhes sobre as classes e interfaces que compõem a implementação do filtro de conectividade, explicaremos a solução utilizada por este filtro para garantir a consistência do modelo da rede.

O funcionamento deste filtro consiste em garantir que o modelo da rede, ao ser atualizado com os eventos recuperados da rede elétrica, seja consistente. O modelo da rede é dito consistente, quando os relacionamentos de conectividade entre os elementos do modelo estão de acordo com as regras de consistência definidas. É importante lembrar, que o compromisso deste filtro está relacionado apenas com os relacionamentos de conectividade associados a linhas de transmissão e transformadores.

Para garantir a consistência destes relacionamentos de conectividade, o filtro de conectividade utiliza, durante a fase de detecção, um conjunto de elementos de detecção de inconsistências. Existem dois tipos de elementos de detecção utilizados

na fase de detecção: os simples e os compostos. Já na fase de correção, um elemento de correção de inconsistências é utilizado por este filtro.

Um elemento de detecção de inconsistências simples é responsável por garantir os relacionamentos de conectividade associados a um único equipamento, seja ele um terminal de uma linha de transmissão seja um transformador. Estes relacionamentos consistem de caminhos em um grafo de conectividade que interligam um equipamento de origem aos seus equipamentos de destino. Para facilitar a compreensão, observe na Figura 5.15 o elemento de detecção ED_1. Ele é responsável pelos relacionamentos de conectividade associados ao terminal PARA da linha L1. Desta forma, ele conhece todos os caminhos que interligam o equipamento de origem — terminal do lado PARA da linha L1 — até seus equipamentos de destino: barramentos BP e BA. Neste contexto, um caminho é uma seqüência de nós e arcos, entre um nó-origem e um nó-destino. Um nó-origem ou destino pode ser uma linha, um barramento ou um transformador. Note que na Figura 5.15, L1-PA → C1.1 → D1 → C1.2 → BP e L1-PA → C1.3 → C3.1 → D3 → C3.2 → BP são exemplos de caminhos.

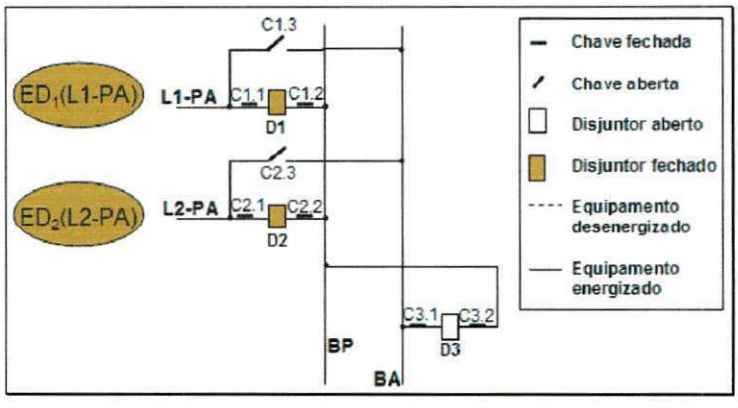

Figura 5.15 – Elementos de detecção de inconsistências simples

Para compreender o funcionamento do filtro de conectividade, utilizando um elemento de detecção de inconsistências simples e um elemento de correção de inconsistências, considere ainda a Figura 5.15. Suponha que, em determinado momento, surja uma janela de tempo contendo o evento "Disjuntor D1 abriu". Nestas circunstâncias, o detector ED_1 será ativado, uma vez que o disjuntor D1 está contido em seus relacionamentos de conectividade. Supondo ainda que o estado do modelo da

rede após o término da janela de tempo seja o representado na Figura 5.15, o elemento de detecção ED_1 detectará uma inconsistência, pois, caso o modelo da rede fosse atualizado com o evento de abertura do disjuntor D1, o terminal da linha L1, que está energizado de acordo com as grandezas elétricas, não ficaria conectado a nenhum equipamento energizado, o que contraria as regras de consistência. O elemento de correção de inconsistência, quando receber a inconsistência detectada, concluirá que o evento foi espúrio, uma vez que o terminal está energizado e, por conseguinte, está conectado ao barramento BP. Em seguida, o elemento de correção remove o evento e o modelo da rede continua consistente.

Em determinados arranjos do sistema elétrico, existem equipamentos que não podem ser analisados por um único elemento de detecção de inconsistências, uma vez que os disjuntores e chaves responsáveis pelos seus relacionamentos de conectividade são comuns a vários terminais de linhas de transmissão e transformadores. Desta forma, a detecção de ruído por parte dos elementos de detecção associados a estes equipamentos precisa ser feita em conjunto, com o intuito de evitar que uma possível correção de uma inconsistência, detectada por um elemento de detecção de um equipamento, leve ao surgimento de uma nova inconsistência, sob o ponto de vista de um outro equipamento. Para evitar efeitos colaterais desta natureza, elementos de detecção de inconsistências compostos são utilizados para sincronizar a detecção de inconsistências por parte dos elementos de detecção simples associados a esses equipamentos.

Para facilitar o entendimento de um elemento de detecção de inconsistências composto, observe o modelo da rede ilustrado na Figura 5.16, após o término de uma janela de tempo contendo os eventos "Disjuntor D1 abriu" e "Disjuntor D3 abriu". Note que os dois disjuntores que atuaram estão associados tanto ao elemento ED_1(T1) como ao ED_2(L1-PA). Desta forma, o elemento ED_1(T1), quando entrar em ação, detectará uma inconsistência, uma vez que o transformador T1 está energizado, porém não está conectado a nenhum equipamento energizado (regra de consistência). Da mesma forma, o detector ED_2(L1-PA) também detectará uma inconsistência, pois o terminal da linha L1 está desenergizado e está conectado ao transformador T1, que está energizado. É importante notar que estas duas inconsistências detectadas estão

relacionadas, pois os possíveis eventos com ruído estão associados a equipamentos analisados pelos dois elementos de detecção. No primeiro caso, tanto é possível que o evento de abertura do disjuntor D1 seja espúrio — o transformador T1 estaria conectado ao barramento B2 — como o de abertura do disjuntor D3 — o transformador T1 estaria conectado ao barramento B1. No segundo caso, é possível que o evento de abertura do disjuntor D2 tenha se perdido — o terminal da linha L1 deixaria de estar conectado ao transformador T1. Note que, se o elemento $ED_1(T1)$ concluir que o evento de abertura do disjuntor D3 for espúrio, levará ao surgimento de uma nova inconsistência — o terminal da linha L1 ficaria conectado ao barramento B1. Neste momento, entra em ação o elemento de detecção de inconsistências composto EDC_1, que analisa todas as inconsistências detectadas pelos elementos de detecção com o intuito de encontrar uma combinação (ou um conjunto) que, se corrigida, não leve ao surgimento de uma nova inconsistência. No final, ele conclui que a única combinação possível se trata de uma inconsistência caracterizada pelo evento espúrio de abertura do disjuntor D1 e pelo evento perdido de abertura do disjuntor D2. Quando o elemento de correção de inconsistência receber a inconsistência detectada, analisará qual a correção mais provável (neste caso só existe uma possível), em seguida, ele removerá o evento de abertura do disjuntor D1 e adicionará o evento perdido de abertura do disjuntor D2 na janela de tempo.

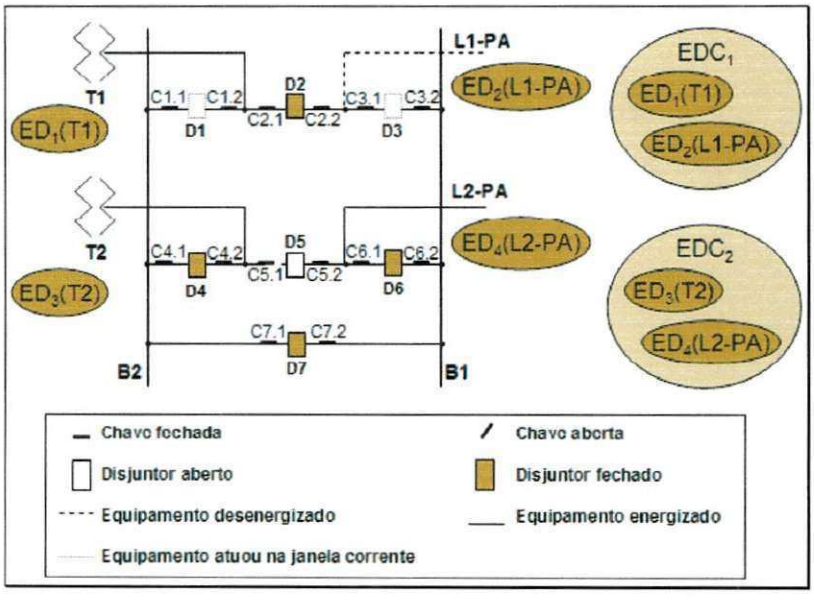

Figura 5.16 – Estado no modelo da rede após o término da janela de tempo

Assim como nos filtros anteriores, a implementação do filtro de conectividade utiliza algumas classes e interfaces do pacote **smartalarms.filtros**. Primeiramente, explicaremos como foram implementados os elementos de detecção de inconsistências e, no final, a implementação dos elementos de correção de inconsistências.

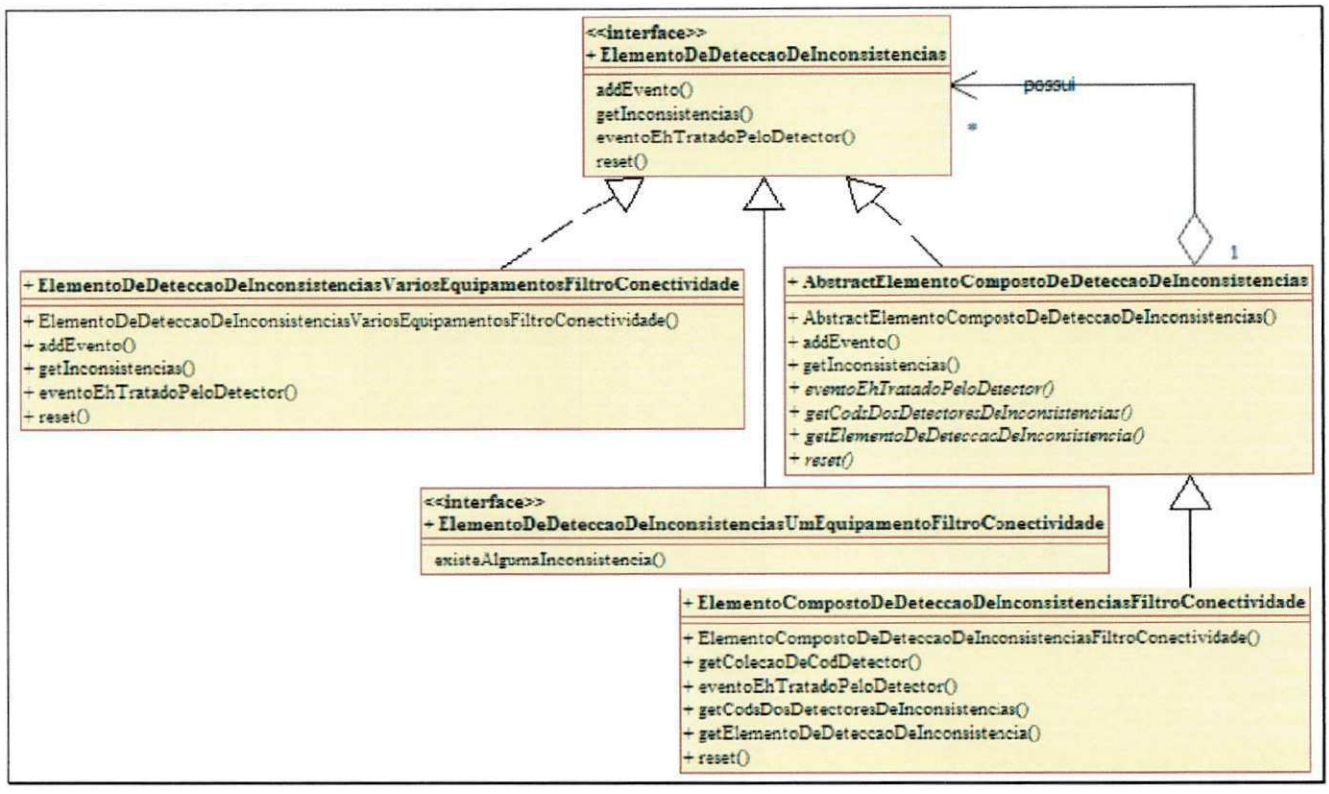

Figura 5.17 – Diagrama de classes do filtro de conectividade

A Figura 5.17 ilustra as principais classes e interfaces responsáveis pela implementação dos elementos de detecção de inconsistências deste filtro. São elas:

1) **ElementoDeDeteccaoDeInconsistenciasUmEquipamentoFiltroConectividade** – Esta interface é responsável por padronizar a implementação de todos os elementos de detecção simples associados a transformadores e a terminais de linhas de transmissão.

2) **ElementoDeDeteccaoDeInconsistenciasVariosEquipamentosFiltroConectividade** – Esta classe é responsável pela implementação dos elementos de detecção de inconsistências compostos.

3) **ElementoCompostoDeDeteccaoDeInconsistenciasFiltroConectividade** – Esta classe tem a função de instanciar todos os elementos de detecção de inconsistências simples e compostos; além disso, ela recebe os eventos oriundos da rede elétrica e os distribui para os elementos de detecção de inconsistências responsáveis por analisá-los.

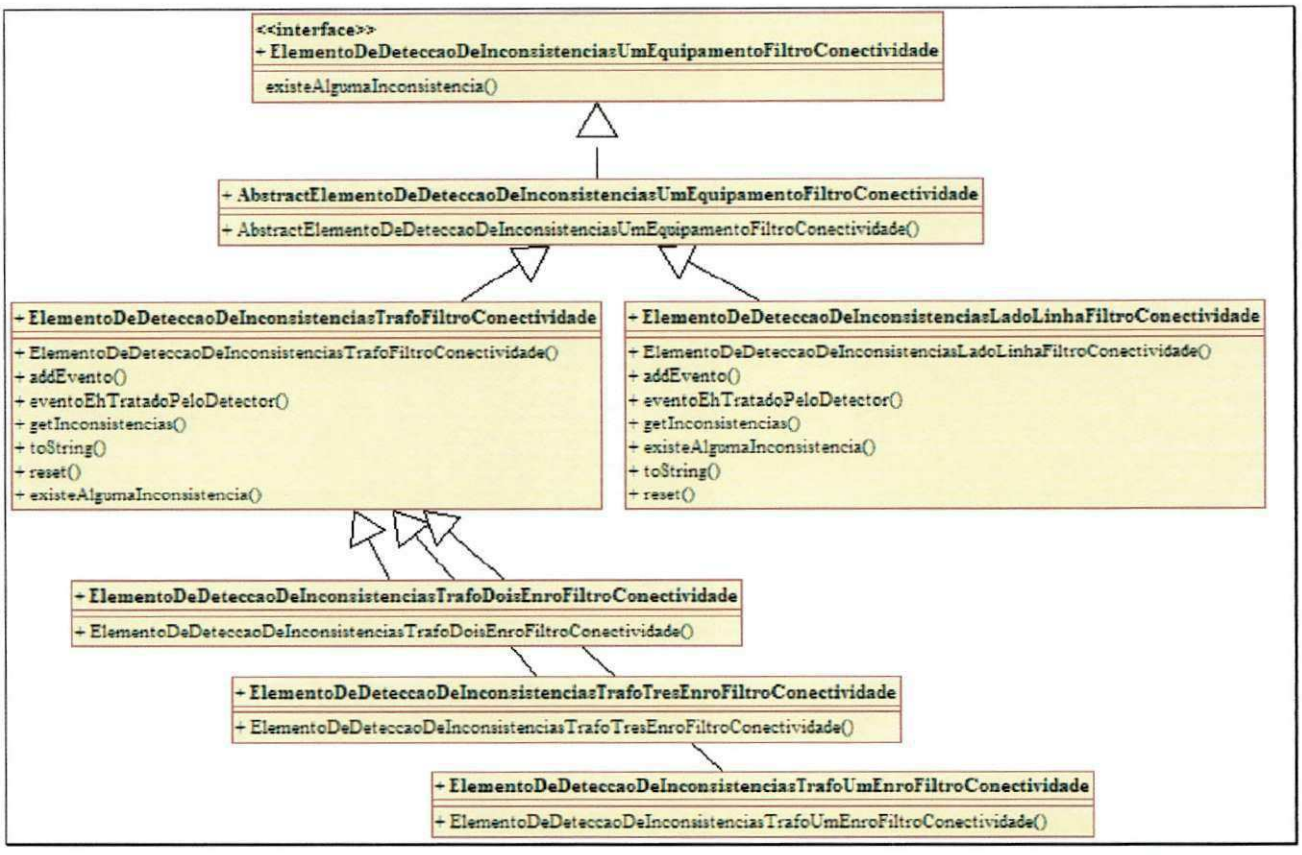

Figura 5.18 – Elementos de detecção de inconsistências simples

As classes que implementam a interface, responsável por padronizar a implementação dos elementos de detecção de inconsistências simples, estão ilustradas na Figura 5.18. São elas:

1) **ElementoDeDeteccaoDeInconsistenciasLadoLinhaFiltroConectividade** – Esta classe é responsável por detectar inconsistências relacionadas com terminais de linhas de transmissão.

2) **ElementoDeDeteccaoDeInconsistenciasTrafoUmEnroFiltroConectividade** – Esta classe é responsável por detectar inconsistências

relacionadas com transformadores que possuem um único enrolamento.

3) **ElementoDeDeteccaoDeInconsistenciasTrafoDoisEnroFiltroConectividade** – Esta classe é responsável por detectar inconsistências relacionadas com transformadores que possuem dois enrolamentos.

4) **ElementoDeDeteccaoDeInconsistenciasTrafoTresEnroFiltroConectividade** – Esta classe é responsável por detectar inconsistências relacionadas com transformadores que possuem três enrolamentos.

As inconsistências detectadas pelos elementos de detecção implementam a interface **Inconsistência** do pacote **smartalarms.filtros**. Existem cinco implementações para esta interface (observe a Figura 5.19). São as seguintes:

1) **InconsistenciaEquipEnergizadoNaoConectadoAEquipsEnergizados** – Inconsistência caracterizada por um equipamento energizado que não está conectado a nenhum equipamento energizado.

2) **InconsistenciaEquipEnergizadoConectadoAEquipsDesenergizados** – Inconsistência caracterizada por um equipamento energizado conectado a pelo menos um equipamento desenergizado.

3) **InconsistenciaEquipDesenergizadoConectadoAEquipsEnergizados** – Inconsistência caracterizada por um equipamento desenergizado conectado a equipamentos energizados.

4) **InconsistenciaEquipsComEstadosIncorretos** – Uma instância desta classe é utilizada, quando o elemento de detecção de inconsistências sabe exatamente quais disjuntores ou chaves estão associados a eventos com ruído.

5) **InconsistenciaGenerica** – Uma instância desta classe é criada quando um elemento de detecção de inconsistência composto analisa um conjunto de inconsistência e gera uma que seja comum a todos os elementos de detecção simples.

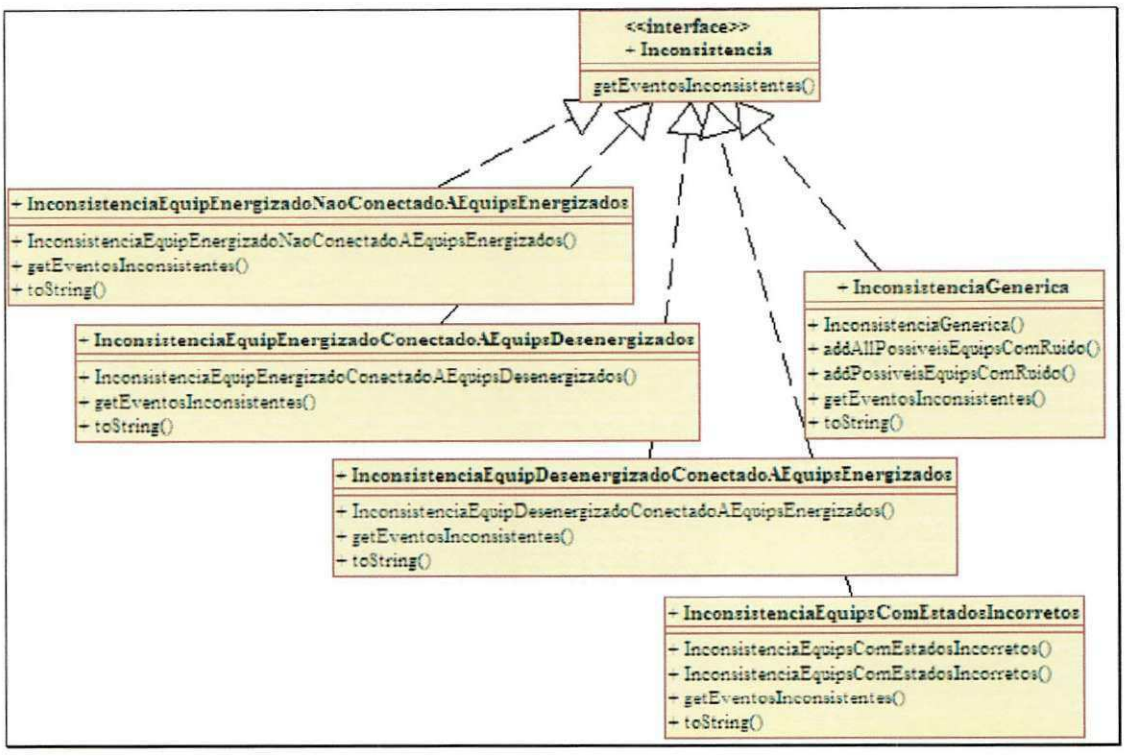

Figura 5.19 – Diagrama de classes das inconsistências do filtro de conectividade

A implementação do elemento de correção de inconsistências foi baseada na interface **ElementoDeCorrecaoDeInconsistencias**. A Figura 5.20 apresenta o diagrama de classes da classe **ElementoDeCorrecaoDeInconssitenciasFiltro Conectividade**, que foi utilizada pelo filtro de conectividade para corrigir as inconsistências levantadas pelos elementos de detecção de inconsistências.

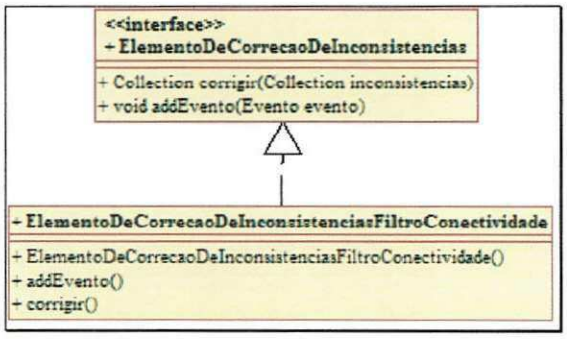

Figura 5.20 – Elemento de correção de inconsistências

5.3. Verificação

Desenvolver *software* não é uma atividade trivial. Faz-se necessário executar um conjunto de testes para verificar se o funcionamento do *software* é o esperado.

Muitas vezes uma única alteração, realizada com o objetivo de corrigir um erro, pode levar ao surgimento de novos erros. Desta forma, testar o *software* em desenvolvimento, toda vez que ele for alterado, é fundamental para verificar se nenhum novo erro foi inserido. Além disso, é importante salientar que esta atividade deve ser realizada a baixo custo, para que possa ser executada com freqüência e, portanto, ser **automatizada**.

Outra grande vantagem associada com a atividade de fazer testes, percebida durante o desenvolvimento desta ferramenta, está relacionada com a garantia de que a ferramenta está funcionando corretamente. Assim, toda vez que o código da ferramenta era refatorado, tínhamos a garantia de que se o código passassem nos testes, ele estaria correto. Desta forma, os testes também serviram como um incentivo para o constante refatoramento do código da ferramenta.

A etapa de verificação desta ferramenta foi caracterizada pela implementação de testes de unidades, que consistem em programas escritos para testar pedaços de códigos de outros programas. Neste contexto, um pedaço de código consiste em uma unidade, que pode ser, por exemplo, um método. Para implementação destes testes de unidades, empregamos o *framework* JUnit (BECK, 1999), que auxilia tanto na construção como na execução de testes de unidade para códigos escritos em Java. Com a utilização deste *framework*, os testes de unidade são implementados em classes desenvolvidas em Java, cujos métodos, desprovidos de parâmetros, invocam os métodos da classe testada, comparando-se o resultado obtido com o esperado. Utilizando JUnit, é possível ainda executarmos automaticamente todos os testes construídos através de uma linha de comando ou com a ajuda de ferramentas visuais, que facilitam a localização e correção de erros identificados. Foram criados 50 testes de unidade para as principais classes que compõem a ferramenta.

6. Uma ferramenta robusta de tratamento de eventos em redes elétricas: validação

No capítulo anterior, apresentamos como foi realizada a implementação da ferramenta robusta desenvolvida durante este trabalho, assim como a metodologia de testes utilizada para verificar o seu funcionamento. No entanto, apesar dos testes de unidade serem bastante eficazes quando utilizados para verificar o funcionamento de pedaços de códigos, não são apropriados para validar a ferramenta, isto é, para constatar se os requisitos levantados foram alcançados, ou mesmo, se a ferramenta atende as necessidades do cliente.

Neste capítulo, apresentaremos como foi realizada a validação da ferramenta, mostrando a satisfação dos requisitos, os testes de aceitação e de regressão utilizados e alguns resultados práticos da sua utilização no CROL.

6.1. Satisfação dos requisitos

Requisitos funcionais:

4) Qualidade nos diagnósticos

   Requisito – A ferramenta deve ser capaz de efetuar diagnósticos corretos, mesmo que eventos de abertura e fechamento de disjuntores e chaves sejam perdidos ou gerados espuriamente.

   Satisfação do requisito – Vários testes de unidade, de aceitação e regressão foram realizados confirmando a satisfação deste requisito (veja a seção 6.2).

5) Correção topológica

Requisito – A ferramenta deve recuperar o estado de abertura de todos os disjuntores e chaves da rede elétrica. Além disso, quando não for possível recuperar tais estados, ela deve ser capaz de estimá-los.

Satisfação do requisito – Quando a ferramenta entra em execução, todos os elementos de detecção de inconsistências do filtro de conectividade são ativados. Em seguida, as possíveis inconsistências detectadas por estes elementos de detecção são corrigidas pelo elemento de correção de inconsistências do mesmo filtro. Testes práticos confirmam satisfação deste requisito.

6) *Logs*

Requisito – A ferramenta deve armazenar em arquivos de dados históricos todas as informações necessárias para a compreensão das ações por ela tomadas.

Satisfação do requisito – O pacote **smartalarms.log** desta ferramenta foi desenvolvido pela equipe do projeto *Smart Alarms* com tal propósito.

Requisitos não-funcionais:

4) Facilidade de uso

Requisito – O uso da ferramenta deve consistir apenas em observar os diagnósticos efetuados por ela.

Satisfação do requisito – A interface gráfica da ferramenta consiste apenas em uma janela contendo os diagnósticos realizados pelo mecanismo de diagnóstico de falhas (veja a Figura 6.1).

5) Manutenção

Requisito – A manutenção da ferramenta deverá estar relacionada apenas com a atualização do modelo da rede, quando equipamentos forem inseridos ou removidos da rede elétrica.

Satisfação do requisito – A satisfação deste requisito foi observada na prática. Existe um arquivo XML — *Extensible Markup Language* —, que contém todos os relacionamentos de conectividade entre os equipamentos da rede elétrica. Para se atualizar a topologia da rede, basta atualizar este arquivo e reiniciar a ferramenta.

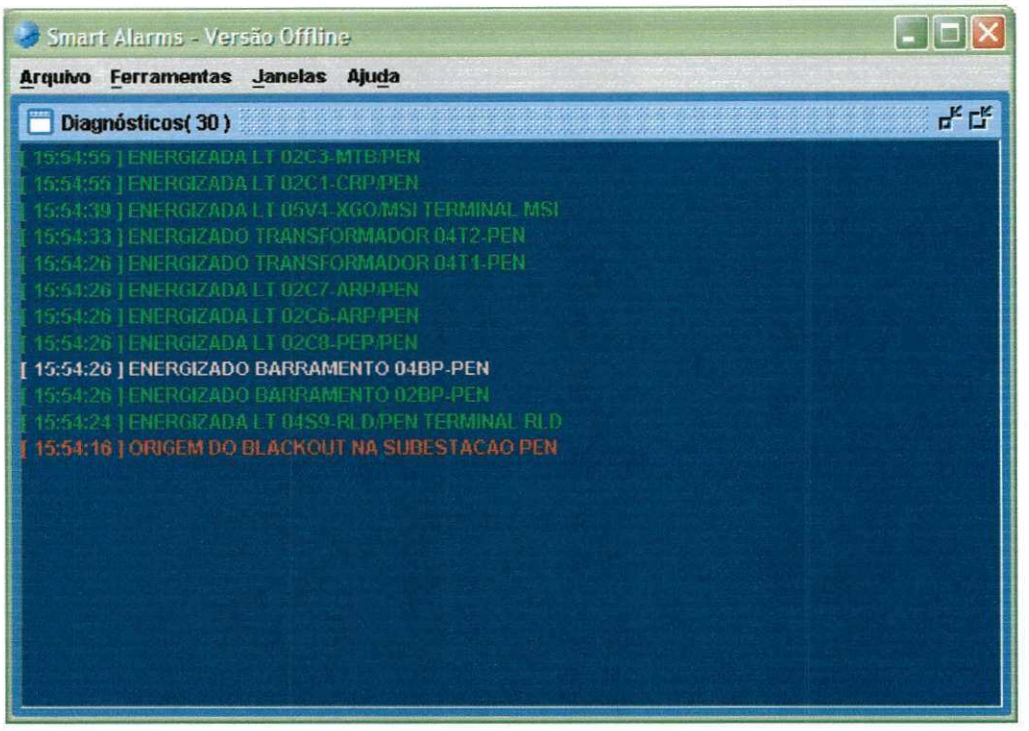

Figura 6.1 – Interface gráfica da ferramenta

6) Desempenho

Requisito – A janela de tempo utilizada pela ferramenta não deve ser superior a 10s, isto é, no máximo, a cada 10s, uma análise dos eventos recuperados da rede deve ser efetuada com o intuito de efetuar possíveis diagnósticos.

Satisfação do requisito – O tempo médio necessário para se recuperarem todas as informações do sistema de supervisão, adicionado com o tempo de processamento do filtro de ruído e do módulo de diagnóstico de falhas, é, aproximadamente, o de 2 segundos. O processamento do filtro de ruído está em torno de

150ms. Estes dados foram obtidos utilizando-se um computador *Celeron* 2.4GB com 512MB de RAM. No entanto, é importante ressaltar que o tempo de processamento da ferramenta pode ser superior a 10s, embora isto não tenha sido observado até então. Estudos estão sendo realizados pelos membros do projeto *Smart Alarms*, com o objetivo de encontrar um algoritmo no qual tamanho de cada janela de tempo seja definido dinamicamente.

6.2. Testes de aceitação e de regressão

Nesta seção, descreveremos os testes utilizados para validar o funcionamento da ferramenta. Foram realizados dois tipos de testes: de aceitação e de regressão.

Os testes de aceitação elaborados durante o desenvolvimento deste trabalho tiveram como objetivo avaliar se o resultado do processamento da ferramenta está de acordo com o esperado, mais precisamente, se os diagnósticos efetuados pela ferramenta estão corretos diante da presença de ruído. Estes testes foram definidos com o auxílio de especialistas e foram classificados em três subtipos:

1) **Testes de aceitação do filtro de ruído** – O objetivo destes testes é o de avaliar o processamento das heurísticas e regras de consistências utilizadas na elaboração do filtro de ruído. Estes testes foram concebidos à medida que as heurísticas e regras de consistência foram sendo definidas. É importante salientar que eles evoluíram e foram modificados à medida que nosso conhecimento sobre o funcionamento do sistema elétrico se aprimorava. Cada um destes testes consiste em dois cenários: um contém eventos com ruído e outro, sem ruído. Ambos sendo formados por uma seqüência de eventos, correspondentes a uma janela de tempo, e por um conjunto de grandezas elétricas referentes aos equipamentos que sofreram alterações nos valores de suas grandezas elétricas. A Figura 6.2 ilustra dois cenários utilizados por em um teste de aceitação deste subtipo. Note que o ruído associado ao primeiro cenário está relacionado com a ausência do evento de abertura do disjuntor 14V1-RCD (em vermelho,

no cenário sem ruído). O teste consiste em verificar se o filtro de ruído, recebendo como entrada as informações contidas no primeiro cenário, é capaz de gerar uma seqüência de eventos idêntica à do segundo cenário. Foram implementados 83 testes deste tipo durante o desenvolvimento desta ferramenta.

Seqüência de eventos com ruído

Data e hora do evento	Equipamento	Código do evento
2002-10-23 00:00:00	14V1-BGI	ABER
2002-10-23 00:00:00	04V1-RCD/BGI	ATPRDE
2002-10-23 00:00:00	04V1-RCD/BGI	ATPRPA
2002-10-23 00:00:00	04V1-RCD/BGI	DE-STTT
2002-10-23 00:00:00	04V1-RCD/BGI	DE-ATRB

Grandezas Elétricas

Data e hora do evento	Equipamento	KV	MW	MVAR
2003-10-23 00:00:00	04V1-RCD/BGI-DE	0	0	0
2003-10-23 00:00:00	04V1-RCD/BGI-PA	0	0	0

Seqüência de eventos sem ruído

Data e hora do evento	Equipamento	Código do evento
2002-10-23 00:00:00	14V1-RCD	ABER
2002-10-23 00:00:00	14V1-BGI	ABER
2002-10-23 00:00:00	04V1-RCD/BGI	ATPRDE
2002-10-23 00:00:00	04V1-RCD/BGI	ATPRPA
2002-10-23 00:00:00	04V1-RCD/BGI	DE-STTT
2002-10-23 00:00:00	04V1-RCD/BGI	DE-ATRB

Grandezas Elétricas

Data e hora do evento	Equipamento	KV	MW	MVAR
2003-10-23 00:00:00	04V1-RCD/BGI-DE	0	0	0
2003-10-23 00:00:00	04V1-RCD/BGI-PA	0	0	0

Figura 6.2 – Cenários utilizados durante os testes de aceitação do filtro de ruído

2) **Testes de aceitação da ferramenta sem ruído** – Durante o desenvolvimento do *SmartOne*, um conjunto de testes de aceitação foram criados, juntamente com os especialistas, para avaliar o seu funcionamento. Estes testes consistem em verificar se os diagnósticos efetuados pelo *SmartOne*, com base em um conjunto de eventos, são os esperados. Para cada regra de diagnóstico existente, um teste desta natureza foi elaborado, cada um consistindo em um cenário composto

por uma seqüência de eventos e por uma seqüência de diagnósticos[10]. É importante lembrar, que a ferramenta desenvolvida é uma extensão do *SmartOne*; logo, ela também deve ser capaz de passar por todos estes testes. Além disso, a execução destes testes é importante para verificar se a incorporação do filtro de ruído ao *SmartOne* não levou ao surgimento de erros de *software*. Para executarmos estes testes, uma adaptação foi realizada, uma vez que, em sua versão original, os cenários utilizados pelos testes não continham as grandezas elétricas dos equipamentos relacionados com os eventos. Esta adaptação consistiu em analisar os eventos existentes nos cenários com o intuito de descobrir os equipamentos que ficaram desenergizados e, no final, incorporar as grandezas elétricas relacionadas e estes equipamentos aos cenários. Note que esta adaptação gerou um novo conjunto de testes de aceitação, que verificam o funcionamento da ferramenta com o todo, com base em um conjunto de eventos **sem ruído**. Foram criados 158 testes desta natureza.

3) **Testes de aceitação da ferramenta com ruído** – Estes testes de aceitação são semelhantes aos anteriores. Enquanto estes últimos utilizavam um conjunto de eventos sem ruído para avaliar o funcionamento da ferramenta, estes avaliam-na com base em um conjunto de evento **com ruído**. Assim como ocorre nos testes de aceitação da ferramenta sem ruído, estes também consistem em uma adaptação dos testes de aceitação utilizados pelo *SmartOne*; no entanto, esta adaptação, além de incorporar as grandezas elétricas aos cenários, remove todos os eventos de abertura e fechamento de disjuntores e chaves existentes nos cenários. Desta forma, a ferramenta, para passar nestes testes, deve ser capaz de realizar

[10] É importante ressaltar que estes testes foram desenvolvidos por Alexandre Nóbrega Duarte (desenvolvedor do *SmartOne*)

diagnósticos corretos com base em um conjunto de eventos **com ruído**. Foram elaborados 158 testes de aceitação desta natureza.

Os testes de aceitação, embora tentem cobrir todos os aspectos da ferramenta, relacionados com as heurísticas e as regras de consistência do filtro de ruído ou com o módulo de diagnóstico de falhas do *SmartOne*, não são capazes de garantir uma cobertura total da ferramenta. Além do mais, desenvolver uma técnica computacional que garanta o funcionamento de um *software* grande e complexo, como esta ferramenta, ainda é um dos grandes desafios da Engenharia de *Software*. No entanto, para minimizar este problema, um outro tipo de teste, chamado teste de regressão, foi adotado. Este teste consiste em simular, de forma acelerada, todos os eventos juntamente com todas as grandezas elétricas recuperadas da rede elétrica durante vários dias, e verificar, ao final, se os diagnósticos efetuados pela ferramenta são os esperados. Note que este teste é bastante interessante, pois ele consiste em um cenário real, onde é muito grande a diversidade de informações relacionadas com os equipamentos da rede elétrica. Desta forma, é possível analisar o comportamento da ferramenta diante de ruído relacionado com manobras manuais e automáticas do sistema elétrico. Devido a vários problemas relacionados com a forma de *logar* os eventos e as grandezas elétricas recuperadas da rede elétrica, os testes de regressão só começaram a ser elaborados em dezembro de 2003. Foi realizado apenas um teste de regressão, contendo aproximadamente nove dias; no entanto, a utilização dele foi extremamente importante para a descoberta e correção de erros de *software*.

Nos três diferentes subtipos de testes de aceitação a margem de acerto foi 100%. Enquanto que nos testes de regressão, a margem de acerto foi de 85%.

6.3. Resultados de implantação da ferramenta

Colocar uma ferramenta desta natureza em operação é sempre um enorme desafio. Primeiramente, devido à responsabilidade, uma vez que se trata de uma ferramenta destinada a auxiliar os operadores durante a análise de grandes ocorrências do sistema elétrico, e caso o operador tome alguma decisão errada devido a um possível diagnóstico incorreto efetuado pela ferramenta, a situação do sistema elétrico pode se agravar ainda mais. Além disso, sempre que um *software* pioneiro, como esta

ferramenta, é colocado em operação, vários problemas não imaginados durante o seu desenvolvimento são descobertos. Outro ponto importante que deve ser mencionado, está relacionado com as expectativas tanto dos dirigentes da CHESF como dos operadores, sobre a precisão da ferramenta. Desta forma, gerenciar as expectativas destas pessoas é fundamental durante a fase inicial de implantação de uma ferramenta como esta.

Para contornar este problema, o procedimento de implantação da ferramenta foi dividido em duas fases: pré-FOE[11] e FOE. Na primeira, os diagnósticos efetuados pela ferramenta não podem ser utilizados pelos operadores, na tomada de suas decisões. Já na segunda, o sistema é utilizado pelos operadores na sua rotina diária de operação e passa a ser alvo de relatórios de operação.

O principal objetivo da fase de pré-FOE foi o de levantar os problemas que só são descobertos quando a ferramenta entra em operação. Desta forma, o operador tem o direito de criticar a ferramenta apenas de forma construtiva, uma vez que a ferramenta está sendo construída para auxiliá-lo no futuro; logo, qualquer contribuição é desejável.

Inicialmente, a ferramenta está sendo implantada no Centro Regional Leste de Operação da CHESF (CROL). Posteriormente, ela será implementada nos demais centros regionais desta Companhia. Atualmente, o CROL supervisiona 23 subestações de energia elétrica, com uma demanda máxima por dia de 2800MW.

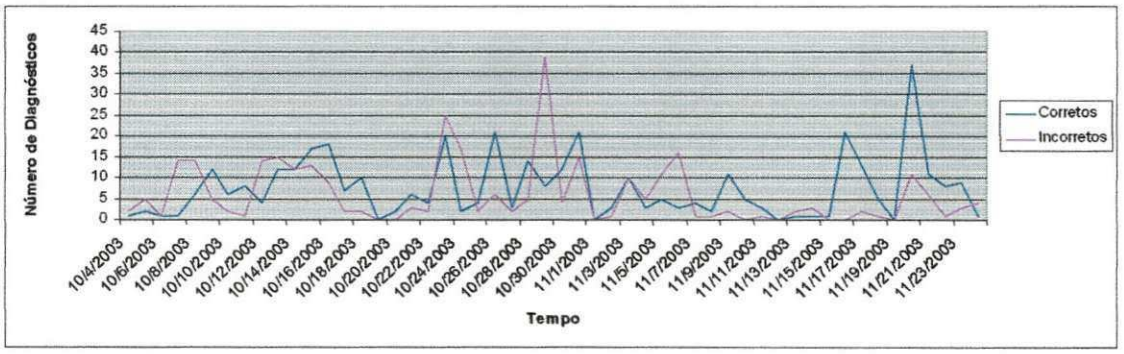

Figura 6.3 – Evolução da qualidade dos diagnósticos da ferramenta

[11] O termo FOE significa "Fase Operacional Experimental".

A ferramenta está em fase pré-FOE desde o dia 15 de setembro de 2003. Deste então, vários problemas já foram detectados e corrigidos. A estratégia utilizada durante esta fase consistiu em construir um mapa contendo todos os problemas detectados e, com base nele, corrigir os problemas mais freqüentes. A Figura 6.3 ilustra a evolução da qualidade dos diagnósticos da ferramenta durante o período compreendido entre 4 de outubro a 24 de novembro de 2003[12]. Note que, no início desta fase, a ferramenta emitiu vários diagnósticos incorretos, enquanto, a partir do dia 7 de novembro, os problemas mais freqüentes foram corrigidos.

À guisa de mostrar o quanto é difícil colocar uma ferramenta desta natureza em operação, apresentaremos alguns dos motivos que levaram a ferramenta a realizar diagnósticos incorretos durante o período exposto na Figura 6.3. Estes motivos podem ser assim classificados:

1) Erros de *software* – São erros inseridos de forma acidental durante o desenvolvimento da ferramenta.

2) Falta de conhecimento – A falta de conhecimento está relacionada com um conhecimento incorreto ou incompleto levantado com os especialistas. Quando um conhecimento desta natureza é implementado na ferramenta, diagnósticos incorretos podem ser efetuados.

3) Falta de supervisão – A falta de supervisão leva a não sinalização de eventos importantes para o diagnóstico de falhas, o que na prática consiste em uma fonte de ruído. Problemas desta natureza devem ser tratados pela ferramenta, entretanto, os diagnósticos incorretos efetuados pela ferramenta foram provenientes de um tipo de falta de supervisão, que só foi descoberto durante a fase de pré-FOE.

---

[12] Não utilizamos dados mais recentes, pois os operadores do CROL não nos fornecerem a análise de todos os diagnósticos efetuados pela ferramenta durante o mês de dezembro de 2003 e janeiro de 2004.

4) Problema no ponto de supervisão – Quando um equipamento apresenta um problema em seu ponto de supervisão, tanto eventos como grandezas elétricas podem estar com ruído. Os diagnósticos incorretos realizados pela ferramenta devido a este motivo foram oriundos de um tipo de ruído descoberto apenas quando a ferramenta entrou em operação.

5) Problema de configuração – Um problema de configuração ocorre quando a topologia utilizada pela ferramenta não representa bem a realidade do sistema elétrico.

A Figura 6.4 ilustra a freqüência relacionada com os motivos que levaram a ferramenta a efetuar diagnósticos incorretos.

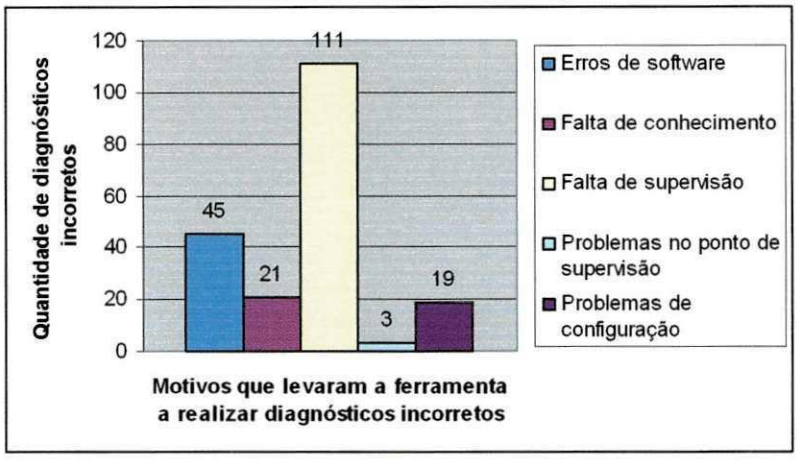

Figura 6.4 – Freqüência relacionada aos principais motivos que levaram a ferramenta a realizar diagnósticos incorretos durante a fase de pré-FOE

Na Tabela 6.1, descrevemos em mais detalhes os motivos que levaram a ferramenta a realizar diagnósticos incorretos durante o período exposto na Figura 6.3. Nela, também podemos observar o status relacionado com cada motivo — se ele foi ou não corrigido —, a quantidade de diagnósticos incorretos (QDI) que ele está associado, e o responsável pela sua correção. É importante lembrar que a implantação da ferramenta envolveu toda a equipe do projeto *Smart Alarms*, desta forma, os motivos a serem listados estão relacionados não apenas com os módulos desenvolvidos neste trabalho.

Classificação	**Motivos**	**Status**	**QDI**	**Responsável**
Erros de *software*	Problema de sincronização entre o filtro de ruído e o módulo de diagnóstico de falhas.	Corrigido	6	Alexandre
	Erro de *software* na implementação do método que detecta a presença de eventos com ruído durante manobras de *bypass* em linhas de transmissão.	Corrigido	11	Eloi
	Erro de *software* na implementação das regras de desarme.	Corrigido	2	Alexandre
	Erro de *software* inserido no filtro durante a correção de um problema.	Corrigido	9	Eloi
	Equipamento abriu e fechou na mesma janela de tempo (este é um problema relacionado com o módulo de diagnóstico de falhas, uma vez que, quando isto ocorre, o módulo é incapaz de perceber que o equipamento abriu e fechou).	Não corrigido	7	Alexandre
	Momento de requisitar as grandezas elétricas (isto é um problema, pois, as grandezas elétricas não são atualizadas no sistema de supervisão de forma instantânea, e se a ferramenta requisitar uma grandeza elétrica e ela vier desatualizada, diagnósticos incorretos poderão ser realizados).	Não corrigido	5	Alexandre / Eloi
	Erro de *software* na implementação das regras de religamento.	Corrigido	5	Alexandre
Falta de conhecimento	Evento de falha de disjuntor sendo tratado incorretamente.	Corrigido	4	Alexandre
	Falta de conhecimento sobre o funcionamento de um transformador com três enrolamentos.	Corrigido	4	Eloi
	Problema na regra de desligamento de linhas conectadas a arranjos de disjuntor e meio.	Corrigido	2	Alexandre
	Problema nos intervalos das grandezas elétricas que caracterizam quando um equipamento está desenergizado.	Corrigido	6	Eloi

	Falta de conhecimento dos identificadores dos eventos que caracterizam a atuação de determinadas proteções do sistema elétrico.	Corrigido	5	Alexandre
Falta de supervisão	Ruído em manobras de *bypass* em linhas que não possuem disjuntores de transferência e cujas chaves de *bypass* não são supervisionadas.	Corrigido	1	Eloi
	Ruído em eventos associados a barramentos sem supervisão de grandezas elétricas.	Corrigido	26	Eloi
	Ruído em eventos associados a linhas sem supervisão de grandezas elétricas.	Corrigido	14	Eloi
	Ruído em manobras associadas a equipamentos que não foi possível recuperar seu estado de abertura quando a ferramenta entrou em execução (quando a ferramenta entra em execução, ela requisita ao SAGE o estado de abertura de todos os equipamentos, uma vez que esta informação é importante durante a análise de ruído).	Corrigido	21	Eloi
	Ruído em manobras de *bypass* em linhas que não possuem grandezas elétricas supervisionadas e cujas chaves de *bypass* não são supervisionadas.	Corrigido	34	Eloi
	Ruído em manobras associados a linhas sem disjuntor próprio, que utilizam um disjuntor de transferência e cujas chaves que a ligam ao barramento não é supervisionada.	Não corrigido	3	Eloi
	Eventos não sinalizados diante de manobras de *bypass* em linhas cujas chaves de *bypass* não são supervisionadas e que possuem o TC localizado na bucha do disjuntor.	Corrigido	12	Eloi
Problema de configuração	Equipamento não modelado na topologia.	Corrigido	7	Jacques
	Equipamento modelado incorretamente na topologia.	Corrigido	9	Jacques
	Equipamento inexistente no sistema elétrico, porém modelado na topologia.	Corrigido	3	Jacques

Problemas no ponto de supervisão	Grandezas elétricas incorretas (transformador desligado apresentando 54.8MW de potência).	Não corrigido	3	Eloi

Tabela 6.1 – Motivos que levaram a ferramenta a realizar diagnósticos incorretos durante a fase de pré-FOE

É fácil perceber, na Figura 6.4, que a falta de supervisão em vários equipamentos da rede elétrica foi a principal razão para os diagnósticos incorretos efetuados pela ferramenta. Esta exacerbada freqüência está relacionada com a nossa carência de conhecimento sobre a realidade do sistema elétrico, pois quando o filtro de ruído foi inicialmente concebido, acreditávamos que eventos com ruído ocorriam esporadicamente e que as grandezas elétricas, por serem extremamente confiáveis, resolveriam praticamente todo o problema de ruído nos eventos recuperados da rede elétrica. No entanto, quando a ferramenta entrou em operação, descobrimos que as grandezas elétricas de muitas linhas de transmissão não eram supervisionadas, ou, quando possuíam, o TC estava localizado na bucha[13] do disjuntor. Além disso, percebemos que muitas chaves de *bypass* não eram supervisionadas e que as grandezas elétricas não eram tão confiáveis quanto imaginávamos.

Com relação aos problemas de configuração, é importante salientar que os equipamentos modelados incorretamente, durante o período em análise, foram corrigidos, entretanto, isto não significa que outros equipamentos não possam vir a ter problemas de modelagem novamente. Procedimentos criados pela equipe do projeto *Smart Alarms* estão sendo utilizados para evitar que este tipo de problema volte a ocorrer. Por exemplo: toda vez que um equipamento sinalizado pelo SAGE não se encontra na topologia da rede da ferramenta, os especialistas da CHESF são avisados para que possam tomar as devidas providências.

É importante salientar que durante todo o período analisado foi caracterizado por situações normais de operação do sistema elétrico, não acontecendo, portanto, nenhuma grande ocorrência. Entretanto, no mês de janeiro ocorreram duas grandes

[13] No capítulo 2, descrevemos os eventos com ruído relacionados com uma linha quando o seu TC localiza-se na bucha do disjuntor.

ocorrências que levaram ao blecaute de várias subestações supervisionadas pelo CROL. Problemas foram detectados na regra de blecaute utilizada pelo módulo de diagnóstico de falhas. As devidas correções estão sendo feitas.

Um dos motivos responsáveis tanto pela dificuldade como pelas grandes descobertas que caracterizaram a fase de implantação da ferramenta, podemos destacar a ausência de dados históricos que pudessem ser utilizados em simulações durante a concepção da ferramenta. Tomando como exemplo o caso dos blecautes, antes das ocorrências, não conhecíamos como os eventos e as grandezas elétricas se comportariam em situações de estresse do sistema elétrico. Desta forma, a implantação da ferramenta foi caracterizada por um constante aprendizado, que só veio a engrandecer o trabalho realizado.

Acreditamos que, em abril de 2004, a ferramenta inicie a fase FOE. O término do projeto *Smart Alarms* está previsto para maio de 2004, quando a ferramenta deve entrar em operação. Convém deixar claro que esta ferramenta consiste em um *software* crítico e de bastante responsabilidade. Desta forma, para que ela seja dita extremamente confiável, é preciso muito tempo de amadurecimento. Esperamos que a CHESF contrate a equipe de desenvolvimento para prestar um serviço de manutenção na ferramenta.

7. Conclusão

Neste documento, descrevemos uma ferramenta de diagnóstico automático de falhas em redes de transmissão de energia elétrica, que utiliza uma técnica robusta de correlação de eventos.

Discutimos os requisitos que nortearam o desenvolvimento da ferramenta, a técnica robusta de correlação de eventos usada por ela, seu projeto arquitetural, sua implementação e também sua validação.

Foi realizada uma análise dos principais tipos de ruído que podem ser encontrados em sistemas de supervisão de redes elétricas, assim como algumas das possíveis razões que explicam a sua existência.

Além disso, foi realizado um estudo das técnicas robustas de tratamento de eventos encontradas na literatura; no entanto, nenhuma delas, a nosso ver, poderia ser adotada, em sua plena eficácia, para tratar o ruído relacionado com eventos oriundos de redes elétricas. Em conseqüência, desenvolvemos uma nova técnica robusta de correlação de eventos, capaz de tratar os principais tipos de ruído presentes em redes elétricas.

A nova técnica é baseada no conceito de **filtro de ruído**, o qual é responsável por detectar e corrigir o ruído inerente aos eventos recuperados da rede, seja removendo eventos espúrios, seja adicionando eventos perdidos. Em seguida, os eventos filtrados são enviados ao módulo de diagnóstico de falhas o qual usa uma técnica híbrida combinando raciocínio baseado em regras com raciocínio baseado em modelos. Este módulo de diagnóstico de falhas foi herdado de uma ferramenta já existente, denominada *SmartOne*, desenvolvida pelo projeto *Smart Alarms*. Desta forma, a ferramenta desenvolvida — *Robust SmartOne* — consiste em uma extensão

desta primeira, sendo constituída, portanto, por um filtro de ruído e por um módulo de diagnóstico de falhas.

A ferramenta *Robust SmartOne* foi implantada no Centro Regional de Operação Leste da CHESF. A partir de alguns resultados obtidos, pudemos constatar que a utilização do filtro de ruído foi fundamental para a qualidade dos diagnósticos efetuados pela ferramenta. Sua importância se explica pela grande quantidade de ruído presente nos eventos recuperados pelo sistema de supervisão da CHESF. Na ausência do filtro de ruído, praticamente para cada evento com ruído oriundo da rede elétrica seria gerado um diagnóstico incorreto pelo módulo de diagnóstico de falhas.

Com a implantação da ferramenta na CHESF, foi possível verificar que a freqüência de ruído nos eventos recuperados da rede elétrica é muito alta, diferentemente do que a própria CHESF havia imaginado ao iniciarmos este trabalho. É importante também ressaltar que eventos com ruído não ocorrem apenas durante grandes ocorrências, mas também em dias normais de operação do sistema elétrico. A principal razão para esta exacerbada freqüência está relacionada com a falta de supervisão nos equipamentos. Desta forma, sempre que um equipamento sem supervisão tem seu estado de normalidade alterado, o evento sinalizando a mudança de estado do equipamento não é enviado aos operadores da rede elétrica, o que caracteriza um evento perdido.

Por se tratar de um trabalho teórico-prático, o seu desenvolvimento foi caracterizado por descobertas de situações novas ou não previstas inicialmente, e por um constante aprendizado, que só vieram a engrandecer o trabalho. Inicialmente, nem ao menos compreendíamos o funcionamento do sistema elétrico. À medida que conhecíamos mais sobre a natureza de cada equipamento e os tipos de ruído inerentes aos eventos recuperados da rede elétrica, a ferramenta sofria várias mudanças. É essencial salientar que vários tipos de ruído eram desconhecidos pelos próprios especialistas da CHESF, que assim puderam também aprender com o sistema.

Outro aspecto interessante associado com o lado prático do trabalho diz respeito à forma em que a ferramenta foi concebida, isto é, aprendendo com especialistas da área, e com o próprio sistema. Se, ao contrário, a concepção do

sistema tivesse sido eminentemente teórica, teríamos, por certo, optado por uma solução probabilística, isto é, os diagnósticos efetuados pela ferramenta seriam acompanhados com uma probabilidade de certeza. No entanto, tal solução baseada em incerteza é indesejada por operadores de sistema elétrico.

Para explicar o porquê da desconfiança de operadores com resultados probabilísticos, uma analogia pode ser feita com uma arma de guerra nas mãos de um soldado durante uma batalha. Para o soldado, uma arma deve inspirar confiança, pois, em circunstâncias vitais, quando precisar utilizá-la, ele a utilizará. Da mesma forma, o operador precisa de uma ferramenta confiável, pois, durante uma ocorrência no sistema elétrico, quando precisar de informações que o ajudem em sua análise, ele utilizará os diagnósticos efetuados pela ferramenta. Se eles estiverem incorretos, o problema em análise poderá agravar-se ainda mais. Desta forma, assim como o soldado não deseja uma arma que só atire nove balas das dez que podem ser disparadas, o operador não deseja um diagnóstico com 90% de certeza de estar correto.

A ferramenta, apesar de estar em operação, precisa de mais tempo de amadurecimento, uma vez que se trata de um *software* crítico e de grande responsabilidade para a CHESF.

7.1. Trabalhos futuros

Embora a ferramenta tenha sido implementada e esteja em operação no Centro Regional de Operação Leste da CHESF, ela ainda carece de refinamentos, com o intuito tanto de melhorar a qualidade dos seus diagnósticos como de lhe acrescentar novas funcionalidades. Entre os refinamentos, podemos enumerar:

1) Construção de um módulo de sugestão de ações corretivas para os problemas diagnosticados pela ferramenta. Este módulo analisaria os diagnósticos realizados pela ferramenta e proporia um conjunto de ações corretivas para normalizar a situação da rede elétrica.

2) Implementação de um módulo de explicação de diagnósticos, que informaria ao operador que eventos levaram o módulo de diagnóstico

de falhas a realizar um determinado diagnóstico. Esse módulo seria útil para dar uma maior confiança aos operadores nos diagnósticos efetuados pela ferramenta, uma vez que eles poderiam observar como a ferramenta chegou ao diagnóstico.

3) Elaboração de um módulo que seria encarregado de gerar relatórios contendo estatísticas sobre falhas nos equipamentos. Com ele, seria possível, por exemplo, descobrir as linhas que apresentaram defeitos durante os últimos três meses, ou mesmo, os disjuntores que mais mudaram de estado na última semana. Estes relatórios seriam de extrema valia para as equipes responsáveis por fazer a manutenção do sistema elétrico.

4) Construção de um módulo que avaliaria o grau de urgência dos diagnósticos realizados pela ferramenta. Este módulo seria extremamente importante para os operadores durante grandes ocorrências, pois lhes informaria os diagnósticos que devem ser atendidos com maior urgência.

5) Implementação de um módulo que identificaria a causa raiz dos problemas no sistema elétrico durante uma ocorrência. Este módulo seria importante, pois a ferramenta informaria ao operador os diagnósticos dos problemas de uma forma mais organizada, apontando a causa raiz dos problemas e suas conseqüências.

6) Criação de um módulo que se comunicaria com o estimador de estado do SAGE para melhorar o tratamento de ruído da ferramenta. O estimador de estado consiste em um novo componente do SAGE destinado a estimar tanto os valores analógicos de determinados equipamentos como o estado de abertura de disjuntores e chaves. Assim, a utilização das informações disponibilizadas pelo estimador seria muito importante para o funcionamento do filtro de ruído, uma vez que a maior quantidade de ruído existente nos eventos recuperados

da rede elétrica está associada com a falta de supervisão nos equipamentos.

Apêndice

I. Geração, Transmissão e Distribuição de Energia Elétrica

Este apêndice foi retirado, com algumas alterações, da dissertação de mestrado de DUARTE (2003). Ele tem a função de prover os conhecimentos sobre engenharia elétrica e gerenciamento de sistemas de potência necessários para um bom entendimento do trabalho como um todo. Não foram considerados detalhes técnicos, tampouco fundamentações matemáticas ou físicas dos assuntos abordados uma vez que tal conhecimento não foi necessário para o desenvolvimento do trabalho proposto. Ao invés disso, o conhecimento é apresentado de forma intuitiva e, muitas vezes, auto-explicativa.

I.1. Sistemas de Geração, Transmissão e Distribuição de Energia Elétrica

Na Figura I-1, é possível observar um esboço de um sistema de geração, transmissão e distribuição de energia elétrica, a qual é produzida nas unidades de geração e transmitida em linhas de alta tensão e de extra-alta tensão até subestações de transformação que reduzem a tensão de operação para transmissões de curta e média distâncias. Neste patamar de tensão, a energia chega aos consumidores industriais de grande porte, que possuem subestações próprias, e às subestações de distribuição. Nas subestações de distribuição, a tensão é novamente reduzida e, por meio de alimentadores primários, supri consumidores industriais de pequeno porte, que não possuem subestações próprias de transformação, e para o sistema de distribuição secundária. Este último, por sua vez, encarrega-se de levar a energia até os consumidores residenciais e comerciais.

Nesta seção, são discutidas as três funções dos sistemas de potência: geração, transmissão e distribuição de energia elétrica, e são apresentados os principais equipamentos utilizados para o seu funcionamento.

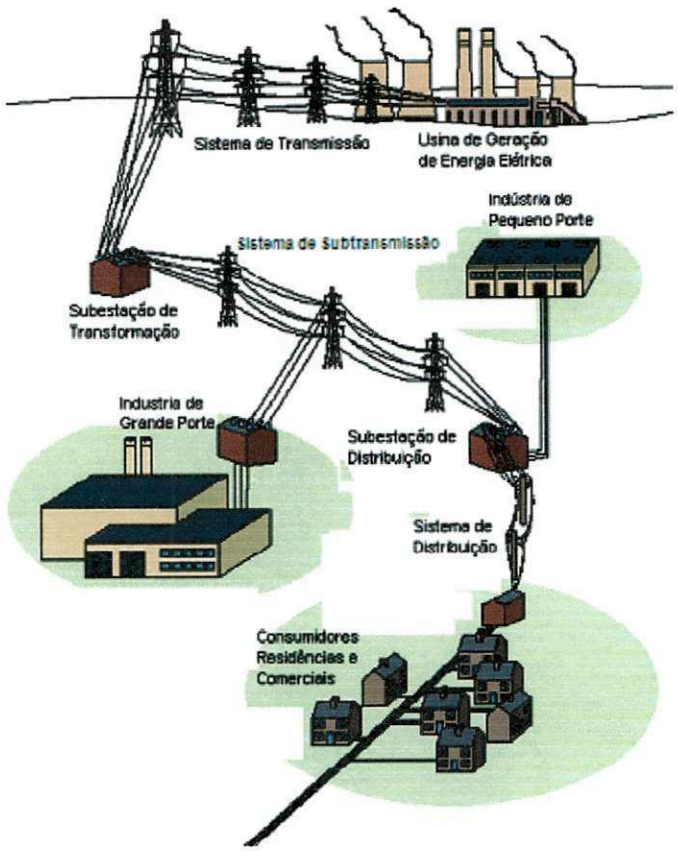

Figura I-1: Sistema de Geração, Transmissão e Distribuição de Energia Elétrica

I.1.1. Geração

A geração de energia elétrica consiste na transformação de algum tipo de energia em energia mecânica que é utilizada para fazer funcionar os geradores elétricos. A seguir é apresentado o principal método de geração de energia elétrica utilizado no Brasil: geração hidrelétrica.

A geração de energia hidrelétrica envolve o armazenamento de um fluido (Figura I-2), normalmente água de rios, a conversão da energia hidráulica de água em energia mecânica, em uma turbina hidráulica, e a conversão da energia mecânica em energia elétrica por um gerador elétrico. Apesar do alto custo inicial para a construção de usinas hidrelétricas, os custos baixos de manutenção, o alto tempo de serviço e a alta confiabilidade dos equipamentos as tornam uma fonte de energia flexível e com uma relação custo/benefício muito boa.

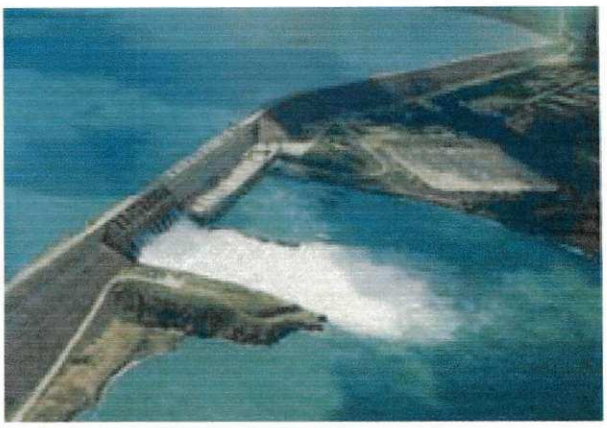

Figura I-2: Reservatório da Usina Hidrelétrica Luiz Gonzaga da CHESF

Usinas hidrelétricas são localizadas em áreas nas quais é possível realizar um uso econômico das fontes de energia hidráulica. A energia hidráulica está presente em qualquer lugar onde há um fluxo de um fluído e um desnível. O desnível denota a energia potencial e é proporcional a diferença de altura vertical entre as turbinas e o nível de água represada. A maioria das usinas hidrelétricas utiliza a energia potencial obtida com o represamento dos rios, porém outros líquidos, como a água do mar e resíduos de esgotos tratados, também têm sido utilizados. A implantação de usinas hidrelétricas requer um estudo minucioso de fatores técnicos, econômicos, ambientais e sociais. Uma parte significante dos recursos gastos para construir uma usina hidrelétrica é utilizada para mitigar os efeitos ambientais na vida selvagem e na re-locação da infra-estrutura e população afetadas pela inundação da região do reservatório da usina (RAMAKUMAR, 1998).

I.1.2. Transmissão e Distribuição

O propósito dos sistemas de transmissão e distribuição de energia elétrica é levar a energia elétrica das usinas de geração até os consumidores. Um sistema de corrente alternada trifásico é utilizado para a maioria das linhas de transmissão de energia.

A Figura I-3 ilustra os conceitos de um sistema típico de transmissão e distribuição de energia elétrica. As usinas de geração produzem energia com uma tensão entre 5 e 25 kV. Esta tensão relativamente baixa não é adequada para a transmissão de energia por longas distâncias. Na saída das usinas de geração há

subestações de transmissão, que empregando transformadores elevam a tensão de operação para valores de ordem de centenas de quilovolts (230, 500kV).

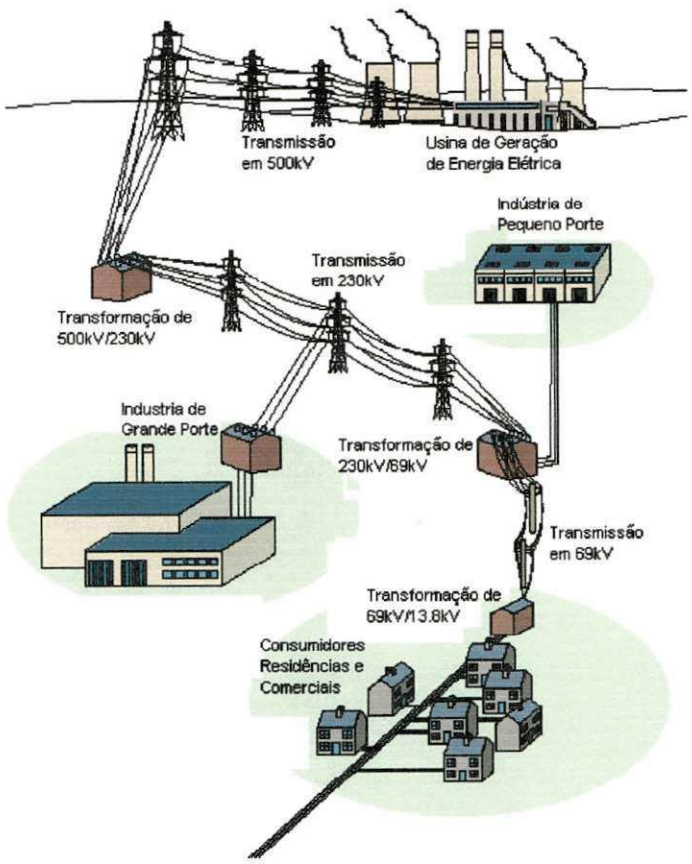

Figura I-3: Esquema típico de transmissão e distribuição de energia

Na Figura I-3, a tensão foi elevada para 500kV, e uma linha de transmissão de extra-alta tensão é utilizada para transmitir a energia gerada para uma subestação distante. Nessa subestação, a tensão é reduzida de 500kV para 230kV, e a energia é retransmitida utilizando linhas de alta tensão para subestações de alta tensão próximas das cidades. Nas subestações de alta tensão, a tensão é novamente reduzida, agora, de 230kV para 69kV. Depois disso, linhas de sub-transmissão (média tensão) levam a energia até as subestações de distribuição, onde a tensão é mais uma vez reduzida de 69kV para 13.8kV. Várias linhas de distribuição saem das subestações de distribuição em postes ou através de dutos subterrâneos e levam a energia até as ruas e avenidas. Antes de chegar até os consumidores residenciais, a energia passa por mais uma transformação de tensão de 13.8kV para 230/115 V para só então ser utilizada pelos clientes.

Existem clientes industriais que podem receber a energia em tensões mais altas que os clientes residenciais e comerciais. Geralmente esses clientes possuem as próprias subestações para reduzir a tensão a níveis desejáveis.

O sistema de transmissão precisa ser bastante robusto para permanecer funcionando mesmo que várias linhas de transmissão deixem de transmitir energia por algum motivo. Para tanto, são criadas redundâncias no sistema, fazendo com que as subestações se interconectem por vários caminhos diferentes.

I.1.3. Principais Equipamentos

Nesta seção, são apresentados alguns dos principais equipamentos utilizados nos sistemas de potência.

Chave

Figura I-4: Representação gráfica de uma chave em um circuito

Chaves, cuja representação gráfica é exibida na Figura I-4, são dispositivos mecânicos utilizados para alterar as conexões de um circuito. São normalmente utilizadas em subestações para isolar os equipamentos durante os períodos de manutenção; para manobrar circuitos, permitindo a transferência de carga entre barramentos de uma subestação; e para propiciar *bypass* de equipamentos, notadamente os disjuntores da subestação. Também são utilizadas em redes aéreas de distribuição urbana e rural com o propósito de seccionar alimentadores para manutenção ou para manobras diversas (MAMEDE, 1994).

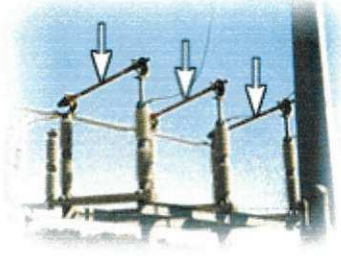

Figura I-5: Chave seccionadora

Na Figura I-5, podem ser observadas algumas chaves em uma subestação.

Disjuntor

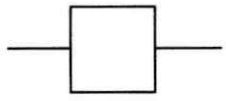

Figura I-6: Representação gráfica de um disjuntor em um circuito

Disjuntores, cuja representação gráfica é exibida na Figura I-6, são dispositivos mecânicos capazes de interromper ou restabelecer a passagem da corrente em um curtos espaços de tempo durante a operação normal do circuito, ou em casos de defeito, sendo comandados por sensores de falha. A operação de um disjuntor se faz reparando os seus respectivos contatos, o que ocasiona o surgimento de um arco elétrico. Para extinguir o arco elétrico, propiciando a interrupção da chave, é necessário que se provoque o alongamento e resfriamento do mesmo e que substitua-se o meio ionizado entre os contatos por um meio isolante e eficiente (MAMEDE, 1994). Os disjuntores são geralmente classificados de acordo com o meio de isolante utilizado, sendo os principais tipos: disjuntores a óleo, disjuntores a vácuo e disjuntores a gás SF6 (hexafluorido de enxofre). Um disjuntor pode ser aberto automaticamente pela proteção do sistema elétrico durante uma falha no sistema, como um curto circuito, para evitar que os efeitos do problema se propaguem para o restante da rede elétrica (MCDONALD,1998) e para evitar danos no sistema. Na Figura I-7, podem ser observados alguns disjuntores a gás instalados em uma subestação.

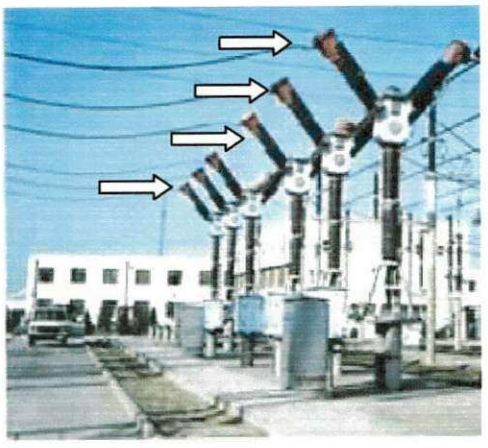

Figura I-7: Disjuntores a gás

Linha de Transmissão

Linhas de transmissão são o meio físico utilizado para transportar a energia elétrica das usinas de geração até as áreas consumidoras. Existem vários tipos de linhas de transmissão, classificadas de acordo com sua tensão de operação. Os principais tipos de linhas são (KARADY, 1998):

- Linhas de alta tensão: são utilizadas, juntamente com as linhas de extra-alta tensão, para conectar usinas de geração a subestações de alta tensão. A tensão de uma linha de alta tensão pode variar entre 100 e 230 kV; considera-se extra-alta as tensões acima de 230kV. O comprimento máximo de uma linha de alta tensão é de 320 quilômetros, e o de uma linha de extra-alta tensão é de 800 quilômetros.
- Linhas de sub-transmissão: geralmente são utilizadas para interconectar as subestações de alta tensão com as subestações de distribuição dentro de uma cidade. A tensão das linhas de sub-transmissão é sempre inferior a 115kV. O comprimento máximo de uma linha de sub-transmissão é de cerca de 90 quilômetros. A maioria das linhas de sub-transmissão está localizada em ruas ou avenidas.

Na Figura I-8, pode ser observada uma torre de sustentação de uma linha de transmissão de alta tensão.

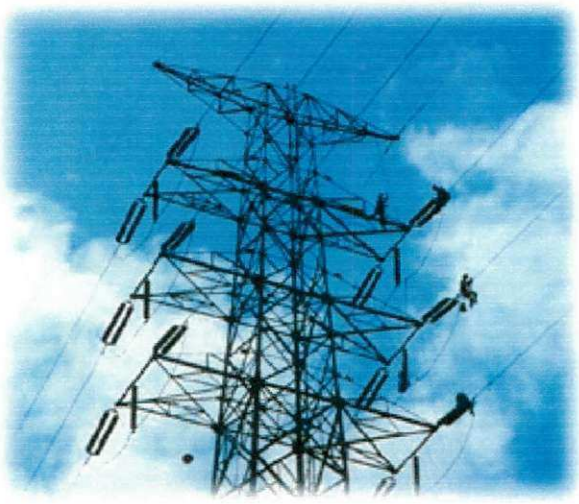

Figura I-8: Torre de alta tensão

Barramento

Os barramentos são elementos estruturais das subestações. Geralmente, a subestação possui um barramento de entrada, em que se conectam mecanicamente os condutores das linhas de transmissão e sub-transmissão, e um barramento de saída, no qual estão conectados os circuitos de saída da subestação. Há carias configurações possíveis para o arranjo dos barramentos e dispositivos de interrupção e manobra (chaves e disjuntores), denotando diferentes ajustes nos critérios de segurança, confiabilidade, economia e simplicidade. A seguir, são apresentados os tipos de arranjos mais comuns (MCDONALD,1998):

- Barramento simples (Figura I-9): neste arranjo há um barramento principal com todos os circuitos diretamente conectados a ele, com baixa confiabilidade, já que uma única falha no barramento afeta todo o sistema, porém o custo é muito baixo e necessita de pouco espaço na subestação.

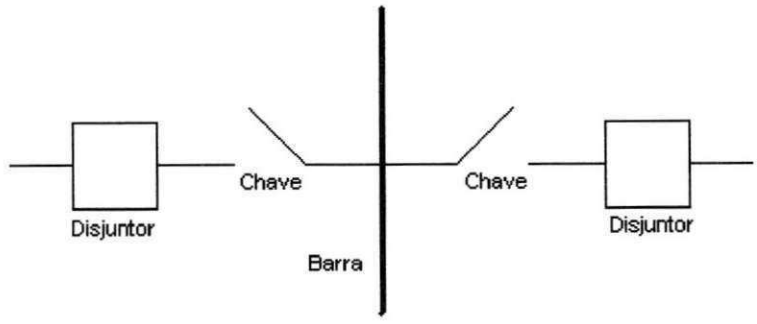

Figura I-9: Arranjo de barramento simples

- Barramento duplo, disjuntor duplo (Figura I-10): este arranjo fornece um alto nível de confiabilidade por possuir dois disjuntores distintos para cada circuito. Adicionalmente, com dois barramentos diferentes, falhas em apenas um deles, não compromete todo o sistema. A manutenção de um barramento ou de um disjuntor pode ser realizada sem interromper nenhum dos circuitos, com um custo muito alto e necessita do dobro do espaço de um arranjo com barramento simples.

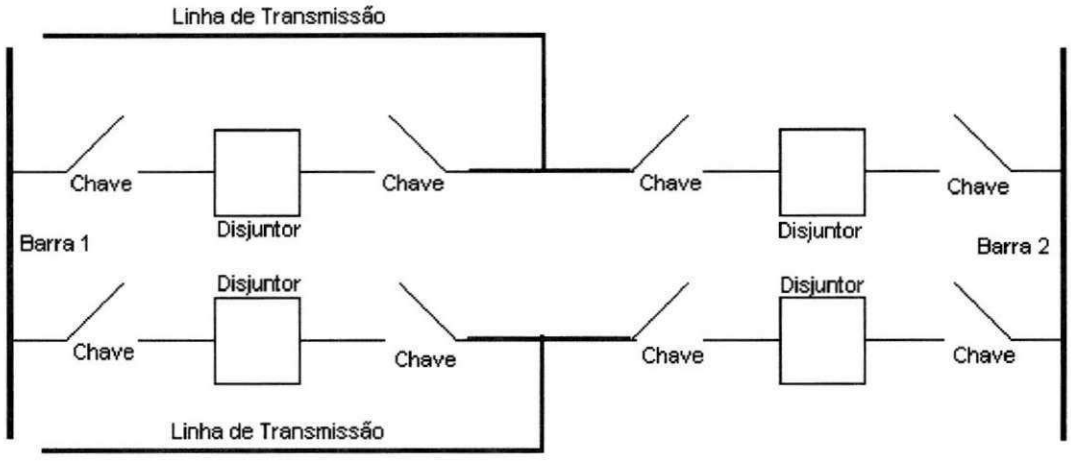

Figura I-10: Arranjo de barramento duplo, disjuntor duplo

- Barramento principal e barramento de transferência (Figura I-11): este arranjo apresenta todos os circuitos conectados entre um barramento principal e um barramento de transferência. Mantendo custos relativamente baixos, esse arranjo permite que qualquer disjuntor saia de serviço para manutenção sem prejudicar o circuito correspondente.

Contudo, requer um disjuntor extra para interligação dos barramentos e é ineficaz em caso de defeito no barramento ou em mais de 1 disjuntor.

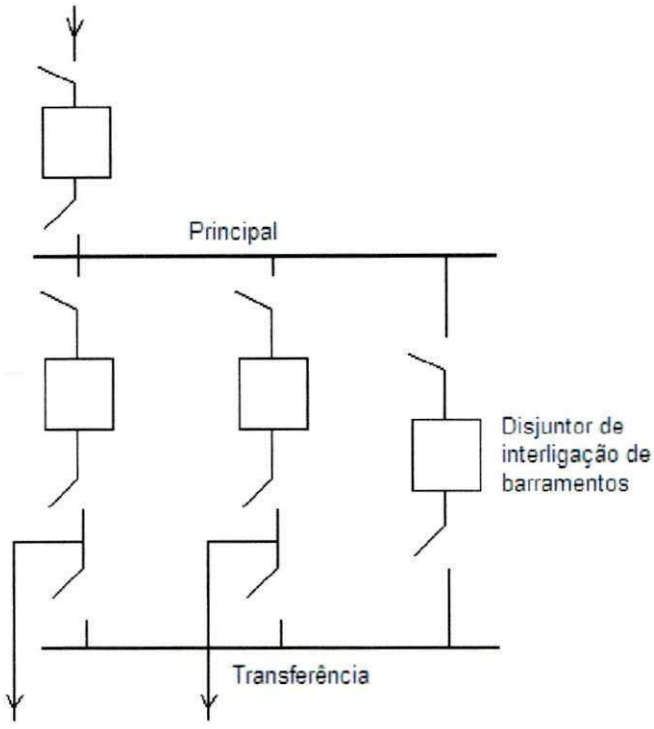

Figura I-11: Arranjo com barramento principal e de transferência

- Barramento duplo, disjuntor simples (Figura I-12): Este arranjo possui dois barramentos principais conectados a cada uma das linhas de transmissão e a um disjuntor de desempate. Utilizando o disjuntor de desempate na posição fechada permite a transferência de circuitos de um barramento para o outro através das chaves. Este arranjo permite que os circuitos operem utilizando qualquer um dos barramentos. Uma falha em um barramento não irá afetar o outro mas uma falha no disjuntor de desempate irá causar um desligamento de todo o sistema. Com o disjuntor de desempate operando na posição aberta são perdidas as vantagens de se utilizar dois barramentos principais. Nesse caso o sistema possui dois barramentos simples, descritos anteriormente, como baixo nível de confiabilidade.

- Barramento em Anel (Figura I-13): Nesse arranjo, como o nome indica, todos os disjuntores são organizados em um anel. Se um circuito falhar, seus dois disjuntores adjacentes abrirão e o restante do sistema não será afetado. Da mesma forma, uma falha em um barramento afetará apenas os disjuntores adjacentes e o restante do sistema permanecerá energizado. A manutenção de um disjuntor nesse arranjo pode ser realizada sem a interrupção de nenhum circuito.

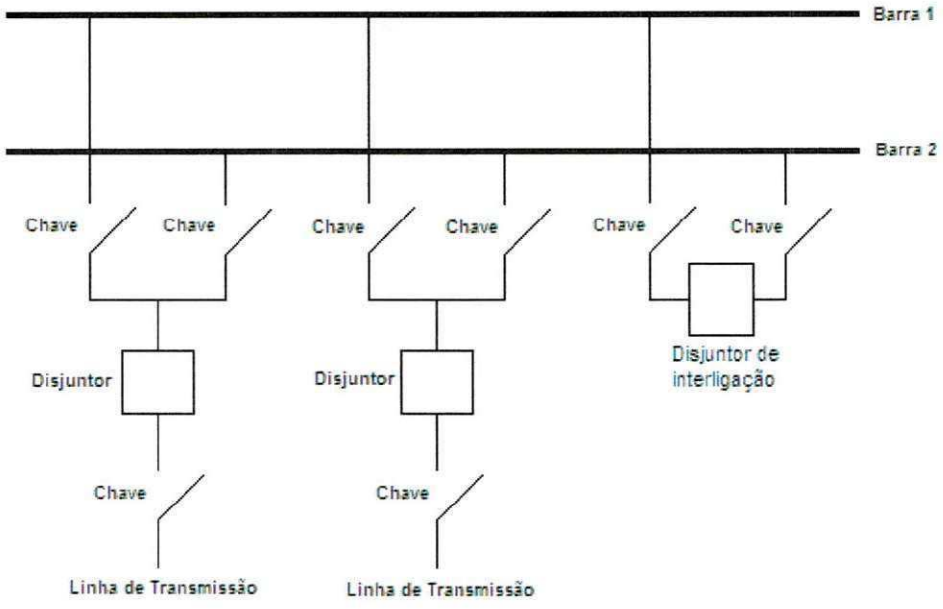

Figura I-12: Arranjo com barramento duplo e disjuntor simples

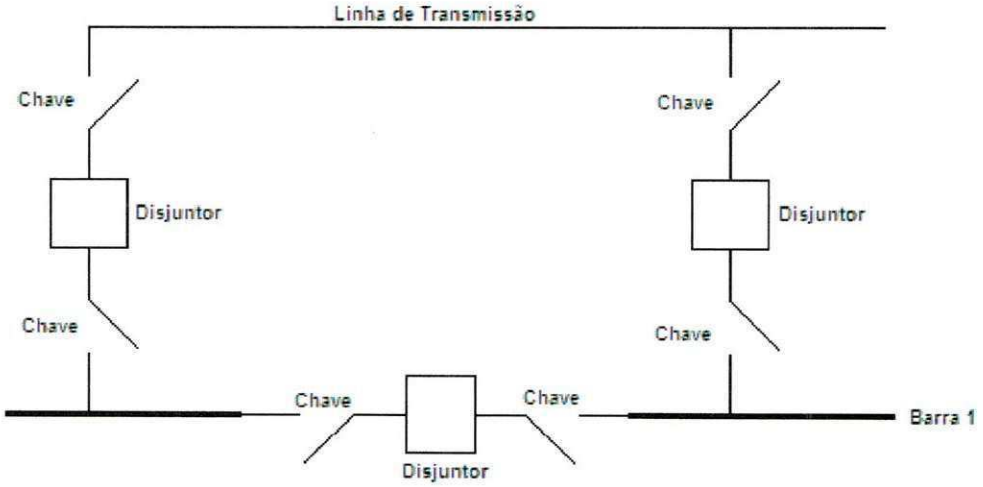

Figura I-13: Arranjo com barramento em anel

- Disjuntor e meio (Figura I-14): o arranjo com disjuntor e meio apresenta a característica de que cada circuito está entre dois disjuntores e possui dois barramentos principais. Uma falha em um dos circuitos provocará uma abertura nos dois disjuntores e não interferirá em quaisquer outros circuitos. A manutenção de qualquer disjuntor pode ser realizada sem a interrupção de nenhum dos circuitos. Este é um dos arranjos com maior confiabilidade, pode ser expandido facilmente e apresenta custo e necessidade de espaço inferiores aos do arranjo com barramento duplo.

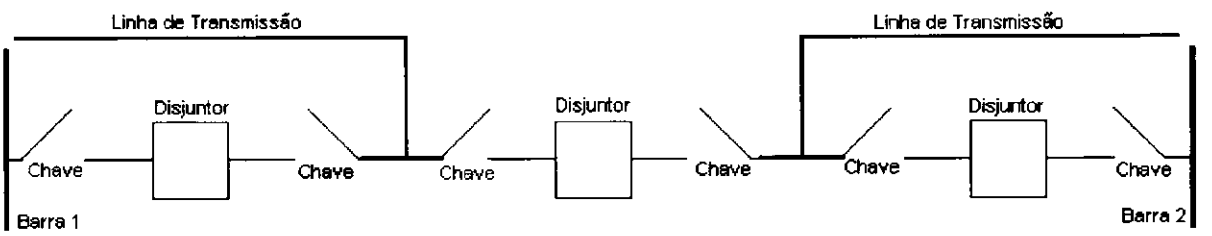

Figura I-14: Arranjo com disjuntor e meio

A Tabela I-1 apresenta uma comparação em termos de confiabilidade, custo e área necessária para instalação dos arranjos apresentados.

Tabela I-1: Comparação dos arranjos de barramentos

Arranjo	Confiabilidade	Custo	Área
Barramento simples	Baixa	Baixo	Pequena
Barramento duplo	Alta	Alto	Grande
Barramento principal e de transferência	Baixa	Médio	Pequena
Barramento duplo, disjuntor simples	Média	Médio	Média
Barramento em Anel	Alta	Médio	Média
Disjuntor e meio	Alta	Médio	Grande

Transformador

Um transformador é definido como um dispositivo elétrico estático, ou seja, sem partes constantemente em movimento, que por meio de indução eletro-magnética

transfere energia de um circuito, chamado primário, para um ou mais circuitos, denominados, respectivamente, secundário e terciário. Nesta transformação é mantida a mesma freqüência, porém as tensões e correntes são alteradas (MAMEDE, 1994). Os sistemas de potência tipicamente dispõem de um grande número de locais de geração, pontos de distribuição e interconexões com o próprio sistema e com outros sistemas. A complexidade do sistema acarreta em uma variedade de tensões de transmissão e distribuição. Os chamados transformadores de potência devem ser utilizados em cada um dos pontos onde há uma transição entre os níveis de tensão do sistema, e podem realizar dois tipos de operação: elevar ou diminuir a tensão. A elevação da tensão geralmente só é realizada na usina de geração; já a diminuição da tensão é usada para alimentar os diversos circuitos de transmissão e distribuição (HARLOW, 1998).

A Figura I-15 exibe um transformador de grande porte instalado em uma subestação.

Figura I-15: Transformador em uma subestação

Reator

Os reatores são utilizados para prover reatância indutiva aos circuitos de potência para uma grande variedade de propósitos, cujos aspectos técnicos fogem ao escopo deste trabalho. Podem ser utilizados em qualquer nível de tensão (HARLOW, 1998).

Gerador

Geradores elétricos são dispositivos capazes de converter alguma forma de energia em energia elétrica. Os mais comuns são os geradores eletromecânicos que convertem a energia mecânica em energia elétrica. Exemplos de geradores eletromecânicos são os utilizados em usinas hidrelétricas, que convertem a energia mecânica gerada pelas turbinas, devido ao movimento de um fluido, em energia elétrica.

Referências Bibliográficas

(ABOELELA, 1999) ABOELELA E.; DOULEGERIS C., **Fuzzy Temporal Reasoning Model for Event Correlation in Network Management**, 24th Conference on Local Computer Networks, LCN'99, Lowell, Massachusetts, USA, pp.150-159, October 1999.

(BECK, 1999) BECK, K; GAMMA, E. **Junit: A cook's tour**. Java Report, 27-38, May 1999.

(BIELER, 1994) BIELER, K.; GLAVITSCH, H. **Evaluation of different AI-methods for fault diagnosis in power systems**. In:. International Conference on Intelligent System Application to Power Systems, 1994, Nanterre Cedex, France, v. 1, p. 209-216, 1994.

(COUTTO, 1999) COUTTO, B.; RODRIGUES, P. **Processamento de Alarmes e Localização de Defeitos em Sistemas de Potência Utilizando Redes Neurais**, VII SEPOPE, Junho, 2000.

(DUARTE, 2003) DUARTE A. N.; **Tratamento de Eventos em Redes Elétricas: Uma Ferramenta,** Dissertação de Mestrado, COPIN/ CCT/UFCG, Fevereiro, 2003.

(ECLIPSE, 2003) Copyright IBM Corp. and others 2003; Versão 2.1.1; http://www.eclipse.com,

(LEWIS, 1999) LEWIS, L; FREY, J. **Multi-level reasoning for managing distributed enterprises and their networks**. In: Integrated Network Management, p. 5-16 1999.

(GAMMA, 1999) GAMMA E.; HELM R.; JOHNSON R.; VLISSIDES J.; **Design Patterns: Elements of Object-Oriented Software**. Addison-Wesley, 1995.

(GIARRATANO, 1989) GIARRATANO J. C.; **Expert systems: principles and programming**, PWS-KENT; ISBN: 0-87835-335-6, 1989.

(GÜRER, 1996) GÜRER, D. W., KHAN I., OGLER R., KEFFER R., **An Artificial Intelligence Approach to Network Fault Management**, SRI International, 1996.

(HARLOW, 1998) HARLOW, J. H. Transformers. In: GRIGSBY, L. L. **The Eletric Power Engineering Handbook**. Auburn, Alabama: CRC Press/IEEE Press, 3, p3.1-3.268, 1998.

(HEBB, 1949) HEBB, D.O. **The organization of behavior**. Science Editions, New York, NY, 1949.

(HIYAMA, 1999) HIYAMA, T. **Current Status of Fuzzy System Applications in Power Systems**. Department of Electrical and Computer Engineering. Kumamoto University, Kumamoto, Japan. 1999.

(JOYA, 2000) JOYA, G., **Connectionist Solutions for Energy Management Systems**. ESQNN'2000 proceedings – European Sysmposium on Artificial Neural Networks, Bruges, Belgica. Abril 2000.

(LEE, 2000) LEE H.; PARK D.;AHN B.; PARK Y.; **A Fuzzy Expert System for the Integrated Fault Diagnosis**, IEEE TRANSACTIONS ON POWER DELIVERY, VOL. 15, NO. 2, Abril, 2000.

(LO, 1998) LO C.; CHEN S.; LIN **B. Coding-based schemes for fault identification in communication networks**. John Wiley & Sons, Inc. New York, NY, USA, 1998.

(KARADY, 1998) KARADY, G. G. Transmission System. In: GRIGSBY, L. L. **The Eletric Power Engineering Handbook**. Auburn, Alabama: CRC Press/IEEE Press, p4.1-4.169, 1998.

(KLIGER, 1995) KLIGER, S.; YEMINI, S.; YEMINI, Y.; OHSIE, D.; STOLFO, S. **A coding approach to event correlation**. In IFIP/IEEE Internation Symposium on Integrated Network Management, 4, p. 266-277. 1995.

(MAMDANI, 1975) MAMDANI, E. H. **An experiment in linguistic synthesis with a fuzzy logic controller**. International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

(MAMEDE, 1994) MAMEDE, J. F.; **Manual de Equipamentos Elétricos**, LTC; 8521610246; 2ª Edição; 1994.

(MCDONALD,1998) MCDONALD, J. D. Substations. In: GRIGSBY, L. L. The Eletric Power Engineering Handbook. Auburn, Alabama: CRC Press/IEEE Press, 5, 5.1-5.134, 1998.

(MEIRA, 1997) MEIRA, D. **A Model for Alarm Correlation in Telecommunications Networks**. Tese de Doutorado em Ciência da Computação. Instituto de Ciências Exatas (ICEx) da UFMG. Belo Horizonte, Brazil, 1997.

(McCULLOCH, 1943) MCCULLOCH, W. S., PITTS, W. H., **A logical calculus of the ideas immanent in nervous activity.** Bulletin of Mathematical Biophysics, 1943.

(OHSIE, 1998) OHSIE, D. A., **Modeled Abductive Inference for Event Management and Correlation**. Ph.D. Thesis. Graduate School of Arts and Sciences. Columbia University, 1998.

(ROSEMBLATT, 1958) ROSEMBLATT, F., **The Perceptron: A probabilistic model for information storage and organization in the brain**. Psychological Review, 1958.

(RAMAKUMAR, 1998) RAMAKUMAR, R. **Eletric Power Generation: Conventional Methods**. In: GRIGSBY, L. L. The Eletric Power Engineering Handbook. Auburn, Alabama: CRC Press/IEEE Press, p. 2, 2.1-2.27, 1998.

(RUMELHART, 1986) RUMELHART, D. E.; McClelland, J. L.. **Parallel Distributed Processing: Explorations on the Microstructure of Cognition**, vol.I. Foundations, MIT Press, Cambridge, MA, 1986.

(SABINO,1999) SABINO, L. R. **Redes Neurais Artificiais, Lógica Nebulosa e Sistemas Neuro-Fuzzy na Previsão de Carga Elétrica em Curto Prazo**. DEE-PUC/RJ, Outubro,1999.

(SILVA, 1998) SILVA, A. J. S; **SAGE Architecture for Power System Competitive Environments**, *VI SEPOPE* Salvador/BA, Maio, 1998.

(SUGENO, 1985) SUGENO, M. **Industrial applications of fuzzy control**, Elsevier Science Pub. 1985.

(YEMINIa, 1996) YEMINI, S.; KLIGER, S.; MOZES, E.; YEMINI, Y.; OHSIE, D. **High Speed and Robust Event Correlation**. IEEE Communications Magazine, p. 82-90, Maio, 1996.

(YEMINIb, 1996) YEMINI, Y.; YEMINI, S.; KLIGER, S. **Apparatus and Method for Event Correlation and Problem Reporting**, United States Patent 5,528,516, 1996.