SISTEMAS ESPECIALISTAS APLICADOS AO PROCESSAMENTO

DE ALARMES EM CENTROS DE CONTROLE

Luiz Corrêa Lima

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.) EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO

Aprovada por:

Prof. Nelson Maculan Filho
(presidente)

Dr. Mário Veiga Ferraz Pereira

Prof. Antônio Almeida Pinho

RIO DE JANEIRO - RJ, BRASIL
JANEIRO DE 1988

LIMA, LUIZ CORRÊA

Sistemas Especialistas Aplicados ao Processamento de Alarmes em Centros de Controle [Rio de Janeiro] 1988.

VIII, 124 p. 29,7 cm (COPPE/UFRJ, M.Sc.,Engenharia de Sistemas e Computação, 1988)

Tese - Universidade Federal do Rio de Janeiro, COPPE.

1. Sistemas Especialistas, Inteligência Artificial, Centros de Controle, Alarme. I. COPPE/UFRJ II. Título (série).

Aos meus pais

AGRADECIMENTOS

Ao dr. Mário Veiga F. Pereira, pela orientação segura, críticas, sugestões e apoio à elaboração deste trabalho.

Ao prof. Nelson Maculan Filho, pela orientação acadêmica.

À Elizabeth Nemer Moysés, pelo apoio, estímulo e motivação para que este trabalho fosse realizado, além da revisão e crítica do texto.

Ao dr. Leslie Terry, pelas discussões iniciais que levaram à definição do assunto da tese.

A Mario C.S. Oliveira e Carlos E. Iencarelli pela sugestão do tema da tese e indicação de material bibliográfico.

A J.A. Fabiano Mendes e Luiz A.C. Pereira pela indicação de material bibliográfico.

Resumo de Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SISTEMAS ESPECIALISTAS APLICADOS AO PROCESSAMENTO DE ALARMES EM CENTROS DE CONTROLE

Luiz Corrêa Lima

Janeiro/1988

Orientador: Mário Veiga Ferraz Pereira

Programa: Engenharia de Sistemas e Computação

Este trabalho apresenta o problema do processamento de alarmes em centros de controle de energia elétrica, seus aspectos e dificuldades operativas, frente às necessidades dos operadores do centro de controle.

São identificadas as características desejáveis que um sistema de alarmes deve apresentar de modo a tornar-se um auxílio efetivo na tarefa operativa, melhorando o desempenho e reduzindo o "stress" do operador.

Procura-se mostrar que com o uso de técnicas de Inteligência Artificial os objetivos acima podem ser alcançados.

Um Sistema Especialista, desenvolvido para esta finalidade com o emprego da linguagem PROLOG é apresentado .

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Sciences (M.Sc.)

EXPERT SYSTEMS APPLIED TO ALARM PROCESSING IN CONTROL CENTERS

Luiz Corrêa Lima

January/1988

Chairman: Mário Veiga Ferraz Pereira

Department: Engenharia de Sistemas e Computação

This work presents the problem of alarm processing in electric energy control centers, its various aspects and operational difficulties due to operator needs.

An identification is made of the desirable features an alarm system should possess in order to be of effective help in the operative duty, improving operator performance as well as alleviating operator stress.

It is shown that the above objectives can be achieved with the employment of Artificial Intelligence techniques.

An example is presented of an Expert System, constructed especially with this objective, with the use of the PROLOG language.

--- INDICE ---

I - INTRODUÇÃO

Os centros de controle de energia elétrica têm passado por uma grande evolução nas últimas décadas. Esta evolução tem-se apresentado na forma de algumas "gerações" que podem ser identificadas [1]. Cada nova geração traz consigo novos conceitos e novas ferramentas que procuram corrigir deficiências anteriores ou aumentar o escopo das funções do centro.

A razão básica para o aumento da complexidade no conceito dos novos centros de controle advém do próprio processo objeto do controle, qual seja o processo de produção, transmissão e entrega de energia elétrica. As redes elétricas tendem a uma contínua expansão, a um aumento das distâncias das fontes de energia aos centros consumidores e do número de interligações entre as redes de companhias vizinhas. Por outro lado há necessidade de operação com carregamentos mais altos de equipamentos, o que reduz as margens operativas de segurança.

Hoje pode-se dizer que os centros de controle de energia elétrica estão à frente, em termos técnicos , daqueles de outros processos industriais [2].

Apesar deste constante aperfeiçoamento técnico, com a introdução de novas funções e atribuições, uma das funções básicas do centro de controle, o processamento de alarmes, presente já nas primeiras gerações, não tem recebido

tratamento adequado de modo a evoluir de acordo com as necessidades.

Este trabalho apresenta uma análise do problema do tratamento de alarmes em centros de controle e da sua integração com a rotina de operação. Procura identificar características ideais de um sistema de processamento de alarmes e demonstrar que com o uso de técnicas da área de Inteligência Artificial muito pode ser feito para se desenvolver sistemas que contribuam para a melhoria do desempenho e redução do "stress" do operador.

Um sistema especialista desenvolvido com a finalidade de mostrar a viabilidade de aplicação deste tipo de técnica é apresentado em detalhe.

O capítulo II faz uma apresentação do centro de controle e das suas características. São analisados os ambientes operativos, as diferenças entre esses ambientes em termos de informações presentes, processo de decisão e carga de trabalho sobre o operador (despachante).

O capítulo III apresenta o conceito de alarme, as razões para problemas na interface do sistema de alarme com o operador e ainda uma análise das características gerais que um sistema de alarme deve possuir de modo a auxiliar e agilizar a tarefa do despachante.

O capítulo IV apresenta conceitos de Inteligência Artificial relevantes para este trabalho, em especial os Sistemas Especialistas, suas características e organização

típica. Uma descrição sucinta da linguagem PROLOG é apresentada.

O capítulo V descreve um sistema especialista desenvolvido com o uso da linguagem PROLOG, com a finalidade de mostrar a possibilidade de aplicação prática desse tipo de técnica para o problema proposto. Um caso-teste é apresentado.

Os apêndices apresentam diversos dados concernentes ao caso-teste discutido no capítulo V.

II - AMBIENTE OPERACIONAL

A atividade fim de uma empresa de energia elétrica é o suprimento de eletricidade aos consumidores de uma região dentro de padrões de qualidade, confiabilidade e economia.

As redes elétricas são usualmente espalhadas por grandes regiões geográficas e são compostas basicamente por Usinas, onde a eletricidade é gerada; Subestações, que permitem realizar as manobras necessárias no sistemas elétrico; e ainda Linhas de Transmissão que interligam usinas e subestações e destinadas a efetuar o transporte da energia até os centros consumidores. Cada usina ou subestação dispõe localmente de recursos limitados para manter o sistema operando em boas condições. Alguns fenômenos elétricos ou ações efetuadas sobre a rede elétrica têm no entanto efeitos que afetam a rede mais globalmente, daí a necessidade de uma coordenação e supervisão centralizadas da operação. Muitas vezes a rede elétrica de uma companhia é de tal magnitude que se torna conveniente a divisão do sistema em regiões, cada uma coberta por um centro de controle de menor porte. Esses centros, algumas vezes chamados de "Despachos Regionais", possuem a sua própria responsabilidade no processo operativo e são por sua vez coordenados por um centro de controle hierarquicamente superior, no caso o "Despacho Central" da mesma companhia.

Por outro lado, há uma tendência constante para o aumento do número de interligações entre as redes de companhias vizinhas, o que permite uma melhoria global nas condições

operativas das redes individuais, tais como economia e segurança. Isto no entanto introduz complicações adicionais na tarefa operativa do ponto de vista de cada empresa. Passa então a ser importante a existência de mais um nível na hierarquia dos centros de controle, correspondendo à atividade de coordenação dos centros das várias empresas. No Brasil esta tarefa será desempenhada pelo Centro Nacional de Supervisão e Coordenação da Operação, da Eletrobrás.

Os conceitos apresentados neste trabalho aplicam-se aos centros de controle de energia elétrica em geral, independente da sua colocação na hierarquia dos centros de controle, apresentada na figura II.1.

Os centros de controle utilizam-se de informações recebidas em tempo-real provenientes diretamente de Unidades Terminais Remotas (UTRs) localizadas nas subestações e usinas ou transmitidas de outros centros de controle. Estas informações correspondem, em geral, a valores de grandezas elétricas e conexão de equipamentos, representativos da condição operativa de rede elétrica no momento.

O sistema de um centro de controle de energia elétrica de certo porte, qualificado na geração atual como um "Sistema de Gerência de Energia" pode ser visto como sendo composto por um certo número de "Resolvedores de Problemas" (RP).

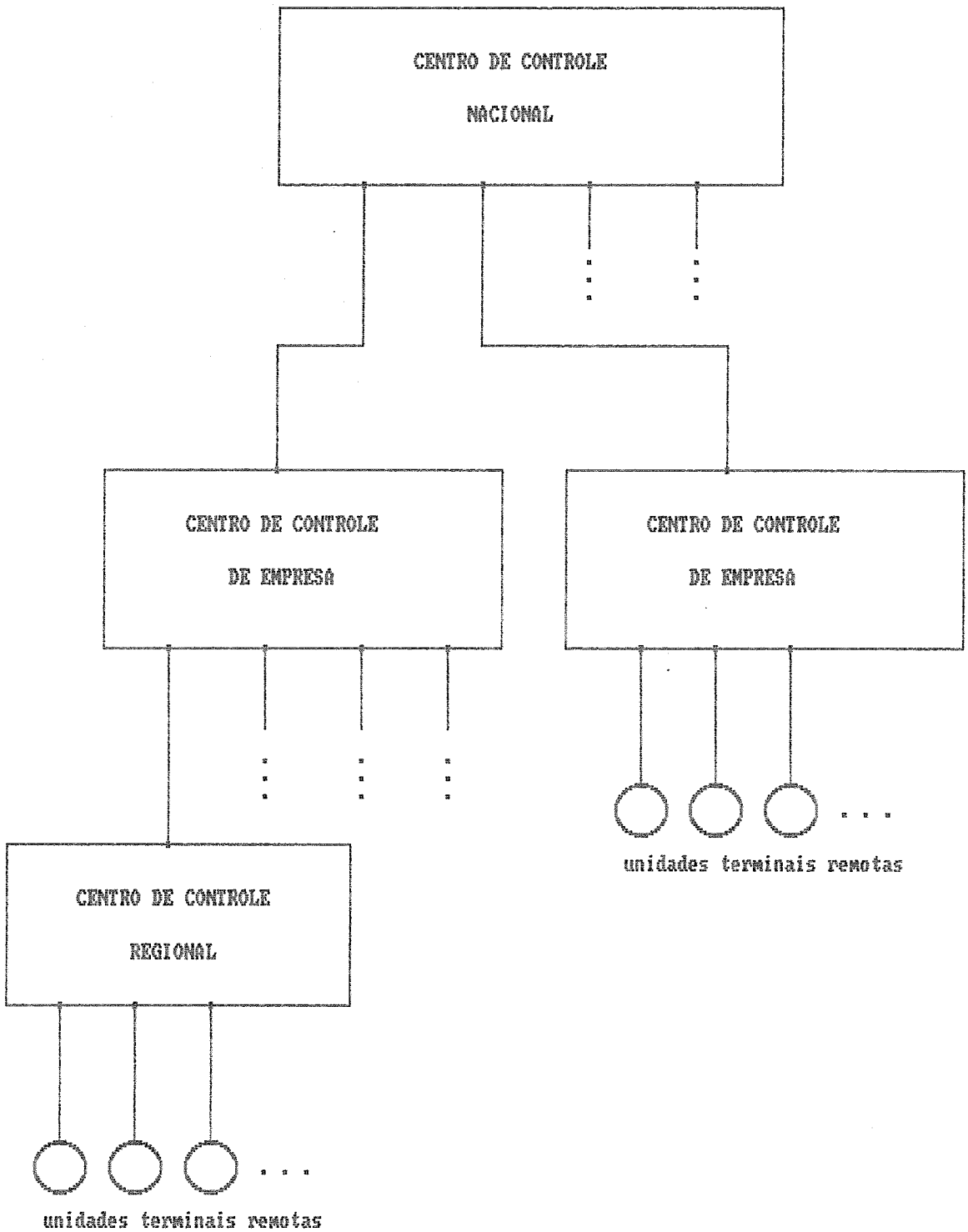

fig. II.1 - diagrama esquemático da hierarquia dos centros de controle

Cada RP, que pode ser humano ou computacional, é em geral distinto e de certo modo independente dos outros. Assim, o fluxo de dados e informações no centro é um fluxo entre RPs. Isto forma o que se pode chamar de um Sistema Distribuído de Resolução de Problemas [3].

Para exemplificar podemos citar os seguintes RPs:

- o sistema de comunicações, usualmente denominado de SCADA ("Supervisory Control and Data Acquisition"), que realiza a varredura em tempo-real de valores de grandezas elétricas de interesse e o Estimador de Estado, que realiza uma filtragem dos erros existentes nesses valores, são RPs que colaboram para a solução do problema da qualidade do fornecimento.
- a função de Análise de Segurança, que permite avaliar o grau de segurança corrente de operação, contribui para a solução do problema da confiabilidade do suprimento.
- o Despacho Econômico, que permite controlar o custo da operação em tempo-real, soluciona o problema econômico.

Os RPs computacionais da atualidade são algorítmicos e produzem resultados e indicações em grande quantidade e dos mais variados tipos. É tarefa do sistema computacional dispor de tudo isto de forma adequada para pronto uso pelo operador, sem necessidade de uma elaboração mental adicional por parte deste.

O Sistema de Gerência de Energia, conforme concebido na atualidade, não se manifesta mais para o operador como um conjunto de algoritmos sofisticados, mas sim como um sistema com uma interface flexível.

Os objetivos primários desta interface devem ser aliviar o operador nas tarefas rotineiras e dar-lhe mais segurança e agilidade no processo de tomada de decisão [4], tendo em vista que o operador, que pode também ser visto como um RP humano, funciona como elemento integrador do fluxo de informações no centro.

Além dos aspectos da variedade de tipos de informação e da grande quantidade de fontes de informação no centro, outro que deve ser considerado diz respeito à adequação da informação às variações nas necessidades de monitoração e controle do sistema elétrico. Diferentes estados de atividade podem ocorrer no centro de controle, dependendo da condição do sistema elétrico [6]. A operação em tempo-real envolve normalmente longos períodos de atividade rotineira mesclados com momentos ocasionais de crise. Há uma necessidade de diferenciação de informações disponíveis entre os diferentes níveis de atividade.

Três modos de operação, ou "Estados Operativos", podem ser identificados, segundo dy Liacco [8]: o Estado Normal, o Estado de Emergência e o Estado Restaurativo. A figura II.2 apresenta um diagrama das transições mais importantes que podem ocorrer entre esses três estados. As transições representadas por linhas cheias correspondem àquelas originadas de eventos no sistema elétrico e as representadas por linhas pontilhadas, às transições decorrentes de ações corretivas tomadas pelo operador.

Os sistemas de potência operam em Estado Normal a maior parte do tempo [7]. Esses sistemas são capazes de se manter

operando normalmente, com pouca ou nenhuma intervenção do operador, quando sujeitos a alterações programadas nas grandezas elétricas de controle, comportamento das cargas dentro do previsto e manobras planejadas de equipamentos.

Durante a operação normal do sistema de potência os procedimentos são bem conhecidos e a necessidade de atenção do operador não é intensa, de modo que o desgaste do operador é apenas o usual.

As dificuldades reais começam quando o sistema elétrico se coloca ou mostra tendência a entrar em um dos seguintes estados:

- Estado de Emergência - no qual a capacidade de algum equipamento é excedida ou alguma grandeza assume valor fora de limites razoáveis.
- Estado Restaurativo - quando uma ou mais cargas são desligadas, sendo cortado o suprimento para um grupo de consumidores.

Quando uma crise ocorre, há um modo acelerado de operação com uma urgência de tempo para a tomada de decisões que depende da gravidade da situação. O operador tem muitas variáveis e um número maior de graus de liberdade para enfrentar o processo de decisão [7].

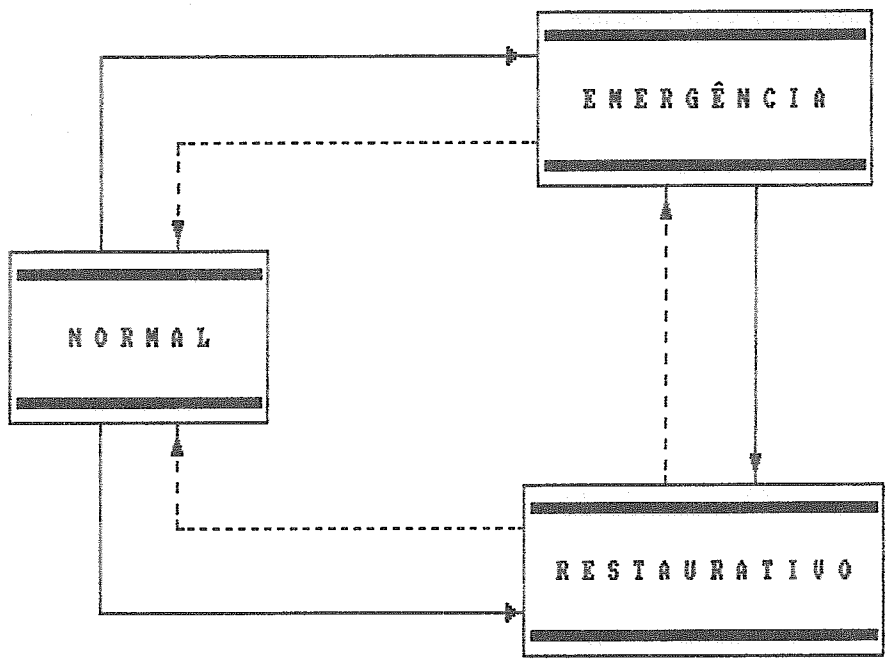

fig. II.2 - Transições entre os Estados Operativos do sistema elétrico

Deve-se considerar que o operador possui uma capacidade limitada de resposta a estímulos dessa natureza, só conseguindo absorver e processar uma certa quantidade de informação em um determinado intervalo de tempo, e que numa crise ele enfrenta muitas vezes situações novas que dependem do seu julgamento baseado em informações adequadas.

Um outro aspecto refere-se à natureza do que é apresentado ao operador. Muito do que é apresentado são dados puros e não a informação que esses dados podem conter. As ferramentas computacionais devem tratar os dados, enfatizando para o operador as informações obtidas tais como: taxas de variação, proximidade de limites, razões entre valores de grandezas, indicação da ultrapassagem de limites, etc. e apresentando-as em uma forma adequada para o processo de decisão.

É portanto tarefa do sistema computacional traduzir o comportamento da rede sob todas as condições [9], devendo no entanto reconhecer em cada momento a situação operativa atual de modo a não sobrecarregar o operador com informações inúteis para o instante em questão [10].

III - PROCESSAMENTO DE ALARMES EM CENTROS DE CONTROLE

Tendo sido abordado no capítulo anterior o ambiente operacional do centro de controle do ponto de vista dos estados operativos e da informação presente, enfocaremos mais especificamente neste capítulo o Processamento de Alarmes, uma das funções básicas de qualquer centro de controle.

III.1 CONCEITOS

Alarmes são informações destinadas a alertar o operador sobre transições de estado detetadas em tempo-real.

Essas transições podem significar anormalidades no sistema elétrico como ultrapassagem de limites de valores de grandezas, tendência de alteração do estado do sistema elétrico ou alteração na conexão de equipamentos elétricos ou de manobra. Por outro lado, um alarme pode ser usado para sinalizar uma condição normal, como por exemplo, retorno à operação normal de um equipamento que se encontrava sobrecarregado.

O operador toma ciência dos alarmes e atua sobre estes por meio dos dispositivos do subsistema de Interface Homem-Máquina (IHM). As mensagens de alarme são normalmente visualizadas através de telas de terminais coloridos semi-gráficos. Os alarmes são grupados em "listas de alarmes", cada lista dando origem a uma tela de "display".

A anunciação da ocorrência de um alarme faz uso, em alguns casos, de dispositivos sonoros capazes de emitir um certo número de sinais que identificam o tipo de alarme. Além disso o painel mímico é comumente utilizado para indicar, através de lâmpadas, a localização geográfica do alarme (qual subestação ou usina) [16].

É comum haver uma ou mais linhas dedicadas nas telas de "display" para apresentação das mensagens mais recentes de alarme.

Ao visualizar uma lista de alarmes via IHM, o operador pode tipicamente realizar as seguintes operações:

- reconhecer um determinado alarme.
- reconhecer todos os alarmes de uma página de "display" (reconhecimento em bloco).
- eliminar um determinado alarme.
- eliminar todos os alarmes de uma página de "display" (eliminação em bloco).

Esses procedimentos de eliminação e reconhecimento em bloco normalmente implicam em alguma perda de informação e somente são justificados porque os sistemas atuais permitem a geração excessiva de alarmes.

O conceito de alarme procura distinguir dois grupos dentre todas as indicações de transição recebidas [11]:

- indicações resultantes de ações específicas do operador, com resultados previsíveis, ou ainda de atividades periódicas do centro de controle.
- indicações resultantes de eventos inesperados, não-autorizados ou não-desejados.

Embora para alguns sistemas não se faça distinção entre esses dois grupos, o mais correto em termos de qualificação da informação no centro de controle é considerar apenas o segundo grupo como definindo as condições de alarme.

A título de ilustração podemos apresentar alguns tipos de grandezas ou situações que podem dar origem a indicações de alarme:

- medidas elétricas - fluxo ativo, fluxo reativo, corrente, tensão, frequência.
- estados digitais - posição de disjuntores e seccionadoras.
- operação de proteção - vários tipos.
- valores de intercâmbio com outras empresas.
- outras - nível d'água, velocidade do vento, temperatura ambiente, umidade ambiente, temperatura de mancal de máquinas girantes, temperatura de óleo de transformadores, nível de óleo de transformador, vibração de mancal, vazamento de elemento refrigerante, estado de unidade terminal remota, temperatura de enrolamento de máquinas ou transformadores, etc.

As origens de alarme apresentadas acima são as usuais em um centro de controle clássico e se referem apenas às grandezas cobertas por um sistema SCADA ("Supervisory Control and Data Acquisition"). Uma abordagem mais completa sobre as fontes de alarme pode ser encontrada em Denzel [12].

Conforme descrito no capítulo II a evolução dos centros de controle tende ao desenvolvimento de um número crescente de Resolvedores de Problemas (RP). A função de alarme passa então

a ser utilizada também por esses RPs, para comunicar ao operador as suas conclusões (resultados) ou informar sobre anomalias que podem ocorrer durante a sua execução (condições anormais devidas aos dados de entrada ou ao comportamento dos algoritmos), configurando deste modo novas fontes de alarme no sistema.

As indicações de alarme, além de servir para informar ao operador sobre situações problemáticas, podem ainda ser utilizadas para iniciar a execução de algum RP designado para tratar o problema operativo trazido pela condição de alarme sinalizada. Este tipo de utilização das mensagens de alarme permite substituir de maneira automática um processo mental de raciocínio e ação do operador, do tipo: SE tal condição se faz presente ENTÃO selecionar um método de solução adequado.

III.2 FATORES HUMANOS

Em função do papel do operador no centro de controle, toda a funcionalidade definida para o centro, ao ser especificada, leva obrigatoriamente em consideração os requisitos próprios deste RP humano. Isto é particularmente verdadeiro no caso de sistemas que apresentam o interfaceamento com o operador como uma característica predominante, como é o caso do processamento de alarmes.

Deve-se observar que certos problemas que podem surgir na utilização de um sistema de alarmes, são na verdade afetos à estruturação funcional e organização da informação no centro de controle. Sob este prisma a melhor utilização de um sistema de alarmes é dependente de:

- distribuição e integração das atividades nas diversas fases do processo operativo (acompanhamento da operação em tempo-real, normalização após perturbações, análise pós-operativa, etc) seguida da classificação da informação necessária para cada atividade.
- definição de atividades e atribuições entre os despachantes (p.ex. supervisão, monitoração de geração, monitoração de transmissão) e da informação de interesse para cada atividade.
- procedimentos para troca de turno de operadores, notadamente com relação à transferência de informações no caso de normalização apos distúrbio ocorrido em turno anterior e ainda com relação a alterações nas atividades dos despachantes de um turno para outro.

Neste trabalho não nos aprofundaremos nesses aspectos organizacionais, já que são problemas locais de cada empresa. Interessa-nos estudar os problemas que podem ocorrer na interface do sistema com o operador e quais características deve possuir o sistema de alarmes para melhorar o desempenho do operador.

Quando ocorre um alarme, o operador executa normalmente uma sequência de ações em resposta a esta ocorrência [13]:

- o operador percebe a anunciação do alarme;
- entende o seu conteúdo;
- determina o evento que causou o alarme, localizando-o;
- analisa as consequências do evento;
- procura determinar a sequência de ocorrências que levou à condição de alarme;

- determina uma ação a ser efetuada para normalizar a situação, se necessário.

Na prática o operador segue esses passos não para cada alarme isolado, mas fazendo uma correlação entre os alarmes já existentes, os que estão ocorrendo, o estado recente do sistema elétrico, regras pré-estabelecidas e ainda a sua experiência própria. Na realização deste processo, no entanto, o operador se defronta com alguns problemas que dificultam o seu trabalho, como:

- quantidade excessiva de alarmes durante perturbações no sistema elétrico.
- significância inconsistente - alarmes cuja informação não tem valor em uma determinada situação.
- alarmes múltiplos para o mesmo evento - como no caso de abertura de uma linha de transmissão, no qual mais de uma dezena de alarmes podem ser gerados, correspondendo à atuação da proteção e operação dos disjuntores em ambos os extremos da linha.
- alarmes sendo apresentados muito rapidamente nas telas para poderem ser lidos - esta é uma decorrência direta da quantidade excessiva de alarmes.
- alarmes sem uma ordem de prioridade adequada - sugerindo que a prioridade de cada alarme seja uma característica dinâmica.
- falsos alarmes e alarmes ocorrendo repetidamente - devido ao mau-funcionamento de algum equipamento.
- falta de temporização dos eventos - isto é, o sistema não tem capacidade de informar a sequência exata em que os eventos ocorrem [14].

Outros problemas são apontados por Wollenberg [13], mas como visto anteriormente, pertencem aos aspectos de organização:

- falta de alarmes de parâmetros chave para a operação.
- alarmes com detalhe insuficiente.
- alarmes específicos demais.

Quando ocorrem perturbações no sistema elétrico, um aumento dramático no número de pedidos de geração de alarmes é a consequência natural. No entanto, como visto anteriormente, é precisamente nesta hora que o sistema computacional deve suprir os operadores com as informações mais importantes e significativas e somente com estas [17].

É evidente que se torna inconveniente colocar para os operadores uma tal quantidade de alarmes que são sinalizados pelos subsistemas, sem nenhum pré-processamento, já que seria impossível a sua assimilação em tempo hábil.

Como exemplo a referência [5] apresenta um caso real de distúrbio no qual 1500 alarmes foram gerados em 53 segundos. O autor faz então um cálculo, considerando que 25 mensagens de alarme são mostradas em cada página de "display", perfazendo um total de 60 páginas. Supondo que o operador leve 30 segundos para ler e entender cada página, seriam necessários 30 minutos somente nesta tarefa. Numa situação destas, o operador tende a ignorar as lista de alarme, procurando descobrir evidências por outros meios, como por exemplo através do painel mímico. Isto configura claramente um caso em que uma superabundância de informação que não pode ser absorvida acarreta finalmente um perda de informação.

Pelo exposto acima pode-se avaliar a dificuldade com que se depara o operador ao tentar concluir com segurança um diagnóstico de uma situação operativa, a partir de informações algumas vezes incompletas, outras vezes redundantes, sujeitas a falha, sem ordem cronológica, em grande quantidade e misturadas em termos de significância.

Podemos concluir que uma análise exata das indicações de alarme necessárias quando de ocasiões de distúrbio é essencial [5], como também o é que o sistema de alarmes possua a capacidade de perceber essas situações e proceder a uma redução dos alarmes gerados somente ao estritamente necessário.

III.3 ASPECTOS SISTÊMICOS

Como já visto anteriormente, o ambiente operacional dos centros de controle tem se modificado, novas funções tem sido incorporadas para fazer par com o aumento da complexidade operativa e os requisitos de uso tem também se modificado paralelamente, a seu modo. Neste ítem interessa-nos estudar os aspectos básicos do processamento de alarmes tendo em vista os requisitos advindos da sua aplicação em tempo-real no centro de controle e ainda deste ambiente em evolução.

Em decorrência das considerações feitas até o momento podemos identificar a Adaptabilidade como uma característica básica que os sistemas de alarme devem apresentar. Estes sistemas devem ter capacidade de se ajustar aos diferentes ambientes de utilização, decorrentes de mudanças nas condições da rede elétrica e tarefas executadas no centro, e ainda das observações obtidas da experiência de uso do próprio sistema.

A característica de adaptabilidade reflete-se tanto no curto prazo, com o intuito de promover um ajuste imediato do processamento à situação operativa do momento, como também no longo prazo, de modo a acompanhar as alterações nos requisitos e a experiência de uso do sistema.

III.3.1 Adaptabilidade a Curto Prazo

Com relação à adaptabilidade a curto prazo dois aspectos podem ser ressaltados: o aspecto <u>estático</u> e o aspecto <u>dinâmico</u> com que o sistema se adapta aos requisitos.

III.3.1.1 Adaptabilidade Estática

A adaptabilidade estática inclui aquelas características que se aplicam independentemente do estado do sistema, podendo ser alteradas apenas por intervenção humana. Refere-se a como as mensagens de alarme são manipuladas e a quem devem ser informadas. Baseia-se nos conceitos de <u>direcionamento</u> e <u>priorização</u>, que são discutidos a seguir.

Devido ao grande número de funções do centro de controle e ao número reduzido de consoles de operação (tipicamente de duas a três consoles na sala de controle) é usual distribuirem-se as atividades de modo a que cada console corresponda a uma determinada responsabilidade, englobando um número de funções a serem desempenhadas. Por exemplo, poderíamos ter uma console para monitoração da transmissão, outra para monitoração de geração e uma terceira para o supervisor. Cada console teria então interesses distintos com relação a alarmes específicos.

Outros alarmes poderiam ser de interesse comum a duas ou mais consoles.

O direcionamento da informação de alarme pode ser implementado através do conceito de classes de alarmes. Cada mensagem de alarme pertence a uma classe determinada e para esta é definido em quais consoles será visualizada. No caso-exemplo de três consoles teríamos 7 possibilidades para as classes de alarmes [17]:

1- somente console de transmissão

2- somente console de geração

3- somente console de supervisão

4- consoles de transmissão e geração

5- consoles de transmissão e supervisão

6- consoles de geração e supervisão

7- todas as consoles

Há ainda a possibilidade de que uma classe de alarmes não envolva nenhuma console, sendo as mensagens de alarme direcionadas para armazenamento ou impressão, quando o interesse é apenas para registro ou análise posterior "off-line" (p.ex. para análise de atuação da proteção em caso de distúrbio).

Através do conceito de classes teremos um grupo de classes atribuído a cada console. Essa necessidade advém de dois fatores: falhas de equipamento e troca de turno. No primeiro caso, se uma console falha, suas atribuições seriam

distribuídas pelas remanescentes. No segundo, observa-se que para muitos sistemas há diferenciação nas atividades dos despachantes de acordo com o período do dia, ocorrendo mesmo em muitos casos uma redução de pessoal no período noturno. Com isto algumas ou todas as classes atribuídas a uma console seriam deslocadas para as demais. A implementação desta facilidade poderia se dar através de um "display" com uma matriz de atribuições do tipo Consoles "versus" Classes [18] atualizável pelo operador.

O conceito de prioritização baseia-se no fato de que frequentemente não é desejável que a prioridade de cada alarme seja a mesma para consoles distintas. Um exemplo disto seria um alarme de tensão, que pode ter muita importância para um supervisor de turno e pouca para um despachante que esteja monitorando os intercâmbios com empresas vizinhas. A prioridade de cada alarme deve refletir a sua importância relativa para o despachante que o observa e o tipo de ação e grau de urgência com que o operador deve responder quando da sua ocorrência. Assim por exemplo, poderíamos ter as seguintes classes de prioridades:

- alarmes de urgência
- alarmes de advertência
- alarmes de informação
- alarmes de normalização (retorno à condição normal)

As diferentes prioridades com que um alarme pode se apresentar podem ser refletidas na ênfase visual ou audível da anunciação, diferenciando-se as prioridades através de atributos de vídeo como cor de contorno, cor de fundo e

piscamento do texto da mensagem de alarme ou ainda tom sonoro da anunciação.

Em resumo, a característica de adaptabilidade estática implica que a atribuição de classes de mensagens às consoles (direcionamento) e a atribuição de prioridades distintas das mensagens a cada console (prioritização) possa ser feita facilmente e em tempo-real.

III.3.1.2 Adaptabilidade Dinâmica

A característica de adaptabilidade dinâmica refere-se à capacidade de ajuste no tratamento dos alarmes de modo a acompanhar mudanças no estado do sistema elétrico.

Esta necessidade vem do fato de que a importância relativa de cada alarme e a consequente resposta do operador dependem da condição operativa do momento. Assim, a informação trazida por um alarme em uma determinada situação pode ser de muita valia enquanto que em uma situação diversa teria importância reduzida ou seria mesmo inconveniente.

A adaptabilidade dinâmica visa a aliviar a sobrecarga sobre o despachante, notadamente em situações de emergência, quando o tempo que o operador tem para reagir é curto, de modo a apresentar somente o que é relevante para o momento e contribuir no processo de decisão. Como regra básica estabelece que as prioridades das mensagens de alarme sejam alteráveis dinâmicamente de acordo com a situação [19]. Isto permite que certos alarmes sejam enfatizados e muitos outros não sejam

sequer apresentados para o operador, por serem irrelevantes ou por só contribuírem para perturbar o trabalho deste.

Os critérios para se proceder dinâmicamente a uma alteração no tratamento dos alarmes podem ter várias origens. Seriam, por exemplo, critérios válidos:

- a ocorrência de um determinado alarme;
- a existência de uma combinação de alarmes;
- a ocorrência de um determinado número de alarmes em um determinado intervalo de tempo;
- a declaração explícita da condição operativa feita pelo operador ou como resultado de análise de algum módulo de software.

Uma característica que pode ser vista como dinâmicamente adaptativa refere-se ao Reconhecimento Automático de Alarmes. Como é usual, determinados tipos de alarme exigem "Reconhecimento" por parte do operador, após o qual podem ser eliminados das listas de alarme. Em períodos de alta atividade, estes alarmes tendem a manter cheias as listas de alarme. O reconhecimento automático poderia então ser usado para reduzir as listas, substituindo o operador nesta tarefa. Um critério válido seria o de reconhecer todos os alarmes que existam há mais do que um certo tempo (definido para cada tipo de alarme). Alguns desse alarmes poderiam ainda ser automaticamente eliminados. Os critérios para se proceder ao reconhecimento automático por tempo poderiam ser os mesmos apontados acima ou ainda, por exemplo, a existência de uma certa quantidade de alarmes na listas.

III.3.2 Adaptabilidade a longo prazo

No longo prazo, a adaptabilidade refere-se a aspectos de projeto de software. Não somente a experiência de uso, mas a própria evolução do conceito do centro de controle, trazem naturalmente a necessidade de alterações funcionais dos sistemas envolvidos, o que implica em manutenção de software. As dificuldades envolvidas neste tipo de atividade devem ser de certo modo previstas na fase de projeto e desenvolvimento, de modo a orientar a escolha de linguagens, algoritmos e estrutura do sistema.

III.3.3 Processamento Combinatório

Outras técnicas, não ligadas propriamente às características de adaptabilidade, mas destinadas a reduzir o número de itens de informação no centro, podem ser desenvolvidas. A referência [19] apresenta um estudo sobre a aplicabilidade do "Processamento Combinatório de Alarmes".

Esta técnica basela-se no fato de que a ocorrência de uma combinação de alarmes pode trazer mais informação do que a contida em cada um dos alarmes individuais.

A função do sistema computacional seria, no caso, a de detetar a existência dessas combinações e emitir alarmes que as sintetizem, suprimindo a anunciação dos alarmes componentes. Isto produz um alívio de carga sobre o despachante sem reduzir o nível de informação disponível e

ainda absorve parte do seu trabalho mental de correlacionar eventos individuais.

Na mesma referência o conceito de processamento combinatório é estendido para incluir a consideração do tempo (instante) de ocorrência e duração de eventos. Sugere-se que há eventos esperados, dentro de certos intervalos de tempo, para os quais os alarmes seriam suprimidos (por serem esperados). Por outro lado, seriam gerados alarmes especiais sempre que eventos esperados não ocorram dentro dos intervalos previstos.

III.4 CARACTERÍSTICAS DESEJÁVEIS DE UM SISTEMA DE ALARMES

Ao longo deste capítulo têm sido apresentados aspectos gerais do processamento de alarmes no ambiente do centro de controle. Neste ítem procuramos identificar as características mais importantes que um sistema de alarmes deve apresentar de modo a integrar-se na rotina de operação como uma ferramenta plenamente utilizável e de grande efeito no desempenho do centro de controle. São elas:

- Distinção entre Anunciação e Registro.

O sistema deve permitir a existência de alarmes somente para fins de registro, sem anunciação para o despachante. O Registro seria realizado através de armazenamento em memória de massa ou por meio de impressão, e seria destinado a tarefas posteriores de análise de ocorrências.

Um exemplo corresponde à indicação de alteração de posição de chaves seccionadoras. Esses elementos normalmente não podem ser operados sob carga, o que implica que a sua manobra, por

ser um evento sem consequências no estado do sistema elétrico, não daria origem a situações de alarme para o operador. Poderia no entanto ser implementado um alarme somente para fins de registro, com vistas a verificações posteriores de sequências de manobras de equipamentos.

- Capacidade de identificar situações que estabelecem alterações na condição operativa do sistema.

  Essas condições poderiam ser, por exemplo:

  - declaração explícita feita pelo operador;
  - ocorrência de um determinado alarme;
  - existência de uma combinação de alarmes;
  - sinalização feita por algum módulo de software, com base em valores de parâmetros-chave do sistema;
  - ocorrência de um determinado número de alarmes em um determinado intervalo de tempo.

- Capacidade de alterar dinâmicamente a prioridade dos alarmes.

  Este ajuste seria realizado como resposta à identificação da condição operativa.

- Capacidade de reconhecer e tratar propriamente os eventos esperados.

  Os eventos esperados não dariam origem à anunciação de alarmes, podendo no entanto gerar mensagens para fins de registro. Este tipo de tratamento envolve a consideração do tempo, ao estabelecer um período (instantes inicial e final) para a ocorrência do evento. Por outro lado, se esta

ocorrência se da' fora do intervalo previsto, um alarme próprio deve ser gerado.

- Capacidade de suprimir a anunciação de determinados alarmes, com base na existência de outros alarmes.

Um exemplo da aplicação desta característica seria a supressão de um alarme de tensão quando ja' existir outro alarme de tensão para a mesma estação do sistema elétrico. Isto ocorre por ser usual nos sistemas SCADA a existência de múltiplos pontos de medição de tensão em uma mesma estação.

- Capacidade de Reconhecimento Automático de alarmes existentes.

  Alguns critérios que poderiam ser usados para determinar que um alarme deve ser reconhecido automaticamente, seriam:

  - a ocorrência de um outro alarme, correlato e de maior importância (por exemplo, um elemento que passa de uma condição atual de alerta para uma condição de emergência);
  - o tempo de existência de um alarme;
  - a cessação da condição de alarme, verificável através da ocorrência de um outro alarme informando a normalização.

- Capacidade de Eliminação Automática de alarmes existentes.

  Critérios similares ao do Reconhecimento Automático poderiam ser empregados.

- Tempo de Resposta

Um valor máximo para o tempo de resposta (desde que é pedido um alarme até que é anunciado) deve ser garantido, tendo em vista a operação em tempo-real. Independente do tempo de acesso do operador às listas de alarme próprias para a ocasião, o tempo de resposta se reflete primeiramente no painel mímico do centro de controle, onde é representado um esquemático do sistema elétrico completo da companhia. Este deve ser atualizado o mais imediatamente possível, para permitir uma pronta localização dos eventos.

- Facilidade de Tradução de Procedimentos Heurísticos da operação em tempo-real.

  O sistema deve ser capaz de internalizar as heurísticas reconhecidas pelo pessoal de operação com respeito ao tratamento da informação. Esses procedimentos, aplicados ao tratamento de alarmes, implementam as características anteriormente apresentadas como as combinações de alarmes, as condições de supressão e reconhecimento de alarmes, etc. Esta introdução de informação no sistema deve apresentar uma interface adequada para utilização por não-programadores.

Outras características importantes correlacionadas ao processamento de alarmes podem ser identificadas, mas não seriam implementadas propriamente pelo sistema de alarmes:

- Identificação de Situações Conflitantes

Requer estruturas de dados próprias relacionando os elementos para os quais se verificaria a inconsistência e utilizaria módulos específicos para esta tarefa.

Como exemplos desse tipo de situação podemos apresentar:

- operação do sistema de proteção para um determinado equipamento elétrico, mas ausência de indicação de abertura do disjuntor correspondente;
- indicação de operação de disjuntor inconsistente com os fluxos atuais nos circuitos.

Nos casos acima haveria necessidade de estruturas de dados no banco de dados contendo a descrição do relacionamento entre disjuntores e proteções e disjuntores e equipamentos elétricos, respectivamente.

- Capacidade de Diagnóstico.

Característica de certo modo relacionada com o processamento de alarmes, principalmente pela aplicação que tem sido proposta na literatura, qual seja o diagnóstico de "faltas" no sistema elétrico que levam à operação do esquema de proteção [38].

Os vários tipos de relés de proteção, bem como a descrição da lógica existente em cada local individual e do alcance de cada relé tem que ser incluídos no banco de dados e são melhor tratados por módulos dedicados de software.

- Consideração do Tempo de Continuidade de uma situação de alarme.

A informação do tempo de duração de eventos no centro de controle é uma característica reclamada pelos operadores [15] mas não implementada nos centros atuais. Deriva do fato de cada equipamento elétrico possuir um tempo característico durante o qual suporta condições anormais de operação (como sobrecarga, por exemplo) a partir do qual há prejuízo físico para o equipamento ou este é retirado automáticamente de serviço pela atuação do sistema de proteção.

Esta característica seria implementada a partir dos módulos de software que testam contra limites os valores obtidos em tempo-real no centro. Esses módulos detetariam a transição do equipamento para a condição anormal de operação e realizariam periodicamente a integração do tempo em que cada equipamento permanece nesta condição, informando este tempo ao operador ou talvez o tempo que resta para o operador para agir. A correlação com o processamento de alarmes existe no fato de que as transições para condições anormais são situações de alarme.

IV. INTELIGÊNCIA ARTIFICIAL E SISTEMAS ESPECIALISTAS

IV.1 EVOLUÇÃO DOS CONCEITOS

A ciência da computação tem-se voltado cada vez mais para o aperfeiçoamento da comunicação dos seres humanos com os computadores [20]. O que foi anteriormente buscado através de novas gerações de linguagens computacionais de nível cada vez mais alto tende a se generalizar no desenvolvimento de pesquisa de conceitos de sistemas que tratem problemas a partir de abordagens declarativas ou prescritivas em lugar do modo imperativo dos sistemas convencionais.

Este é o campo de atuação da Inteligência Artificial (IA), cujo início se deu conjuntamente com o advento do computador na década de 50. Segundo o professor Marvin Minsky, um dos pioneiros neste campo, a IA é uma tentativa de fazer com que o computador realize tarefas que, se realizadas por seres humanos, seriam consideradas como inteligentes.

O conceito de algo "inteligente" é no entanto relativo, desde que certas atividades consideradas difíceis em uma determinada época podem-se tornar atividades rotineiras e independentes de habilidade em uma época posterior. Um exemplo é a extração de logaritmos, que hoje é executada por simples calculadoras de bolso.

Apesar desta evolução dos conceitos, a finalidade básica da IA permanece a mesma - aumentar o escopo do emprego dos computadores e aplicá-los à resolução de novos problemas.

O enfoque das pesquisas na area de IA tem também sofrido alterações ao longo do tempo.

No início os pesquisadores admitiam como princípio que seria possível desenvolver "Resolvedores Gerais de Problemas" baseados em métodos formais de raciocínio. Esses métodos, utilizando princípios bastante gerais, seriam implementados nos computadores, que os utilizariam para resolver quaisquer problemas que se lhes apresentassem.

A par de alguns bons resultados nas áreas de jogos e prova automática de teoremas, a aplicação prática generalizada não ocorreu.

A idéia básica utilizada, adequada para um certo número de problemas, é a chamada "busca no espaço de estados", de formulação bastante simples [21]:

- define-se um estado inicial;
- define-se um teste para detetar estados finais, que são as soluções do problema;
- define-se um conjunto de operações que são aplicadas para se mudar o estado do sistema.

O espaço de estados pode ser visualizado como um grafo no qual os nós são os estados e os arcos são as operações. A busca no espaço de estados se traduz numa busca em grafo, o qual pode ter que ser construído passo a passo, conforme o procedimento

de solução se desenvolve. Um algoritmo simples, ou de "força bruta" pode ser especificado da seguinte forma:

1- gere uma tentativa de solução, selecionando uma das operações possíveis a partir do estado atual;

2- teste se o estado atual é uma solução;

3- se o estado atual é uma solução, termine, caso contrário volte ao passo 1.

Para a seleção definida no passo 1 pode-se escolher uma das técnicas usuais de busca em grafos, dependendo do tipo de problema tratado.

O problema com este tipo de técnica é que o número de passos da busca pode crescer exponencialmente com o tamanho do problema, fenômeno este denominado de "explosão combinatória". Um programa para jogar xadrez, por exemplo, usando força bruta e considerando apenas 4 jogadas à frente para cada jogador exigiria a análise de alguns bilhões de posições [22].

Como resultado da impraticabilidade desse tipo de busca exaustiva, observou-se que a busca teria que ser guiada a partir de características do tipo de problema em questão. Em outras palavras, seria necessária a inclusão de heurísticas no processo de busca. Neste caso, entendia-se por "heurística" uma "função de avaliação" especialmente construída e que, calculada a cada passo da busca no grafo, produzia como resultado um número que representava uma estimativa do "custo" relativo de cada caminho possível de ser seguido.

Assim este tipo de estratégia seria aplicada a certos tipos de problemas formalizáveis através do conceito de espaço

de estado. As estratégias de Resolução Geral de Problemas mostraram então a sua inabilidade no tratamento da maioria dos problemas complexos. Esta inabilidade decorre do fato de que muitos dos processos de decisão encontrados na prática são mal-formalizados, raramente possuindo especificações e procedimentos claros e bem definidos desde o início, sendo as suas características e peculiaridades absorvidas pouco a pouco com a sua aplicação. A tarefa para muitos problemas não é, portanto, a de codificar um algoritmo de solução, mas de explorar o problema e possíveis soluções.

A compreensão da necessidade de desenvolvimento de sistemas para esse tipo de problema deu origem aos Sistemas Baseados em Conhecimento. A capacidade de um sistema inteligente como um Resolvedor de Problemas estava agora colocada na representação explícita do conhecimento específico sobre um tipo de problema, e não em mecanismos sofisticados de inferência. O conhecimento representado inclui o uso generalizado de heurísticas na modelagem do problema.

Os sistemas baseados em conhecimento diferem dos usuais na sua organização, no tipo de modelagem do problema tratado e no modo de busca de soluções, além do aspecto externo do ponto de vista de uso.

O termo Engenharia do Conhecimento foi cunhado para se referir à extração e formulação do conhecimento humano, projeto e construção de Sistemas Baseados em Conhecimento.

As principais fontes de conhecimento humano, transponíveis para esse tipo de sistema são os especialistas humanos, livros, bancos de dados, artigos técnicos e experiência pessoal [23].

Quando o conhecimento é estável e formalizado, programas algorítmicos são mais adequados para a obtenção de soluções. Se o conhecimento é no entanto subjetivo, fragmentado, mal-codificado, resultante de julgamento humano ou heurístico, os sistemas baseados em conhecimento são melhor aplicados.

A Engenharia do Conhecimento (EC) visa a construir sistemas que [24]:

- incorporem grandes quantidades de conhecimento não formalizado;
- sigam adaptativamente as linhas de raciocínio mais adequadas ao problema particular em questão;
- atualizem os seus conhecimentos conforme a compreensão do problema evolui;
- permitam ao usuário monitorar a linha de "raciocínio" seguida pelo sistema na obtenção de uma solução particular.

Os problemas fundamentais da EC são [20]:

- Representação do Conhecimento - como representar o conhecimento humano através de estruturas adequadas à máquina.

  Atualmente um dos aspectos de destaque da IA é a observação de que um comportamento "inteligente" pode ser simulado por meio de manipulação de símbolos [26]. Os símbolos podem representar

objetos, situações e estratégias arbitrariamente complexas, descrevendo o problema em questão e os procedimentos heurísticos para a sua solução. Desta forma, um programa simbólico incorpora conjuntamente os papéis de código e dados.

- Geração de Inferências - como utilizar as estruturas simbólicas que representam o conhecimento para produzir soluções para um determinado problema.

  A geração de inferências se traduz em esquemas de controle, independentes dos dados do problema particular, baseados em formalismos que em última análise recaem em uma busca guiada destinada a explorar as possíveis soluções.

  Cada tipo de mecanismo de inferência requer formas adequadas de representação do conhecimento, de modo que esse dois aspectos são muito ligados.

- Aquisição do Conhecimento - como transferir o conhecimento humano das suas fontes para um programa, incluindo-se aqui os aspectos de modificação dos conhecimentos adquiridos e incorporação de novos conhecimentos advindos da experiência.

IV.2 SISTEMAS ESPECIALISTAS

Os sistemas baseados em conhecimento são usualmente conhecidos como "Sistemas Especialistas". Na verdade, muitos dos sistemas convencionais tambem são sistemas especialistas, já que se aplicam a resolver um determinado tipo especializado de problema. Os assim chamados "Sistemas Especialistas" no

entanto se distinguem por tentar replicar o comportamento de especialistas humanos e por apresentarem a abordagem da Engenharia do Conhecimento.

Um especialista humano é alguém que já adquiriu conhecimento bastante em um determinado assunto e o usa com grande eficácia.

Por seu lado, a força de um Sistema Especialista reside no conhecimento que engloba, sendo os esquemas de inferência e representação do conhecimento apenas mecanismos para o seu uso.

Nem todos os sistemas especialistas são necessariamente construídos tendo em vista igualar ou superar um especialista humano. Um sistema especialista pode também servir como um assistente inteligente em um processo de decisão. Sua função seria a de enumerar alternativas promissoras, possivelmente interagindo com o usuário, mas deixando o julgamento final para este.

Embora um sistema especialista possa, em tese, chegar mesmo a superar o seu correspondente humano em um determinado assunto, algumas deficiências intrínsecas normalmente impedem que se alcance tal desempenho [25]:

- inabilidade para aprender por si só com a experiência;
- incapacidade de tratar com analogias;
- falta de intuição;
- os especialistas humanos sabem avaliar em que ocasiões as regras devem ser quebradas (possivelmente nas fronteiras do conhecimento específico, onde o humano pode utilizar a sua cultura geral).

Um especialista possui usualmente Regras de Julgamento próprio ou empíricas, as quais permitem determinar quais Conclusões ou hipóteses são suportadas pelas Evidências. A organização de um sistema especialista segue de certo modo este modelo de associação estímulo-resposta. Um determinado padrão de resposta é obtido quando se faz presente um determinado padrão de estímulo. O conhecimento é representado em módulos e o "raciocínio" efetuado pelo casamento de padrões entre:

- dados que entram no sistema e módulos de conhecimento armazenados;
- módulos de conhecimento que se tornam ativos (se tornam de interesse), acompanhando o estado atual de resolução do problema e outros módulos de conhecimento correlacionados.

Os componentes elementares de um sistema cujo mecanismo de inferência é dirigido por padrões seriam [21]:

- uma coleção de módulos representativos do conhecimento, com padrões definidos;
- estruturas de dados dinâmicas, utilizadas pelos módulos ativos para representar o estado atual do processo de solução;
- um controlador que realize o casamento de padrões, selecionando e ativando os módulos pertinentes, repetidamente.

O comportamento do controlador pode ser sumarizado pelos seguintes passos:

1- descubra quais módulos de conhecimento podem ser ativados devido ao seu padrão;

2- se mais de um módulo pode ser ativado, escolha um e o aplique (isto recebe o nome de "Resolução de Conflitos");

3- aplique o conhecimento do módulo, alterando o estado do sistema através das estruturas dinâmicas. Volte ao passo 1.

Os programas convencionais consistem de duas partes distintas: algoritmos e dados. Os algoritmos determinam como resolver tipos específicos de problemas e os dados caracterizam os parâmetros de um problema particular.

Os Sistemas Especialistas se distinguem dos convencionais pela clara separação que se faz entre [26]:

- o conhecimento geral do problema (módulos de conhecimento);
- informação sobre o problema específico em questão (dados de entrada);
- métodos de aplicação do conhecimento para se obter uma solução (mecanismo de inferência).

Esta separação resulta em características muito interessantes para os problemas em que são aplicados:

- facilidade de construção de protótipos - que são muito úteis no tratamento de problemas mal-formulados;

- desenvolvimento incremental do sistema - se o conhecimento é organizado de maneira adequada, torna-se fácil corrigir definições e acrescentar conhecimentos sem necessidade de mexer no "código" do programa. Então esses ajustes poderiam ser efetuados pelo próprio usuário, conforme a experiência de uso e a compreensão do problema aumentam.

A arquitetura de um sistema especialista, com os seus componentes básicos é apresentada na figura IV.2-1.

A Base de Conhecimento é o repositório do conhecimento necessário para o problema. Os módulos de conhecimento são sentenças em uma linguagem simbólica destinada a representar o conhecimento tanto para o usuário quanto para o sistema e cujos componentes elementares incluem implicitamente regras de interpretação (semântica). Esta linguagem apresenta um aspecto descritivo do conhecimento e um aspecto procedimental, que especifica como o conhecimento deve ser tratado.

Como os módulos de conhecimento não fazem parte do "código" do programa, o Motor de Inferência é apenas um interpretador, possuindo um mecanismo geral de raciocínio adequado ao tipo de representação adotado.

O Motor de Inferência realiza o casamento de padrões entre elementos da Memória de Trabalho e da Base de Conhecimento, através do Seletor, que escolhe um módulo entre os possíveis de serem ativados, passando-os para o Aplicador. Este último aplica o conhecimento tornado ativo, efetuando alterações nas estruturas da Memória de Trabalho ou informando o resultado.

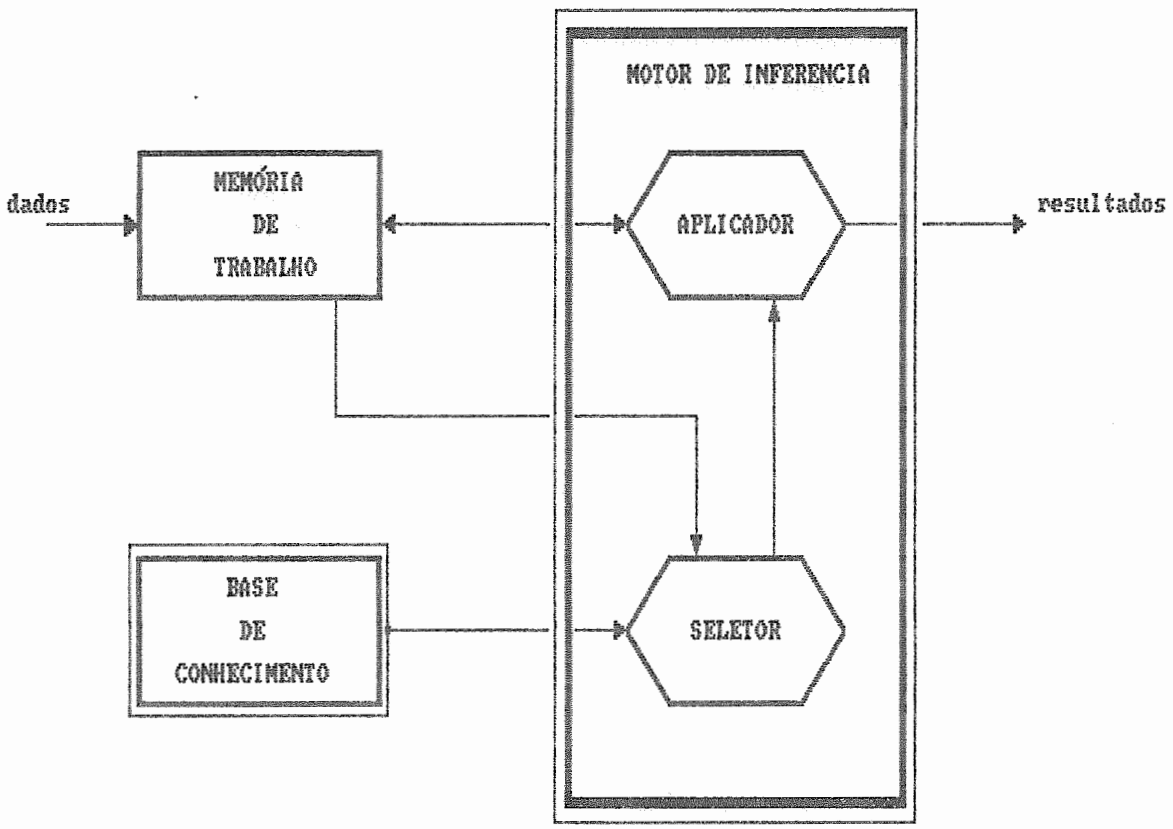

fig. IV.2-1 - Componentes básicos de um Sistema Especialista

O processamento do Motor de Inferência é cíclico, alternando as duas fases de Seleção e Aplicação.

A Memória de Trabalho armazena os dados de entrada, a representação do estado do algoritmo de busca, bem como as inferências intermediárias que são geradas no processo de solução.

IV.3 SISTEMAS BASEADOS EM REGRAS

Vários tipos de formalismos podem ser utilizados para a representação do conhecimento contido na Base de Conhecimento. Um esquema simples e intuitivo, e por isso mesmo muito utilizado, emprega os conceitos de Fatos e Regras.

Os Fatos são estruturas estáticas que representam um conhecimento que se auto-define, como as descrições de objetos individuais ou classes de objetos.

As regras são associações entre padrões de dados ou objetos que se fazem presentes e Ações que o sistema efetua como consequência. Essas Ações podem ser modificações em estruturas na memória de trabalho, ou mesmo novas conclusões lógicas que representam padrões que se tornam ativos.

Uma forma comum para as Regras é a seguinte:

SE <condição_1>
E <condição_2>
...
E <condição_n>
ENTÃO <conclusão>

A expressividade das regras advém da similaridade com os modelos humanos do tipo estímulo-resposta, causa-efeito ou ainda premissa-conclusão. Além disso, pela sua forma, são extremamente adequadas à abordagem por meio de casamento de padrões.

Alguns tipos de informação que podem ser representados através de regras são, por exemplo [27]:

- relacionamentos entre classes de objetos;
- generalizações e categorizações de dados;
- conclusões específicas verificadas para casos específicos;
- condições para se alcançar um objetivo;
- consequências de situações hipotéticas;
- causas prováveis de sintomas.

Cada regra individual representa um módulo de conhecimento. Os módulos de conhecimento devem se ligar, formando uma rede, de modo a descrever o problema completo. Assim a CONCLUSÃO de uma regra é normalmente uma das CONDIÇÕES de outras regras definidas. Esta ligação estática entre as regras da base de conhecimento pode ser representada como um grafo, como na figura IV.3-1, onde as linhas horizontais indicam as regras, e as setas, as condições e conclusões.

Neste exemplo a CONCLUSÃO da regra R1 funciona como CONDIÇÃO para a regra R2. Por outro lado, a CONCLUSÃO da regra R2 serve como CONDIÇÃO para duas regras, R3 e R4.

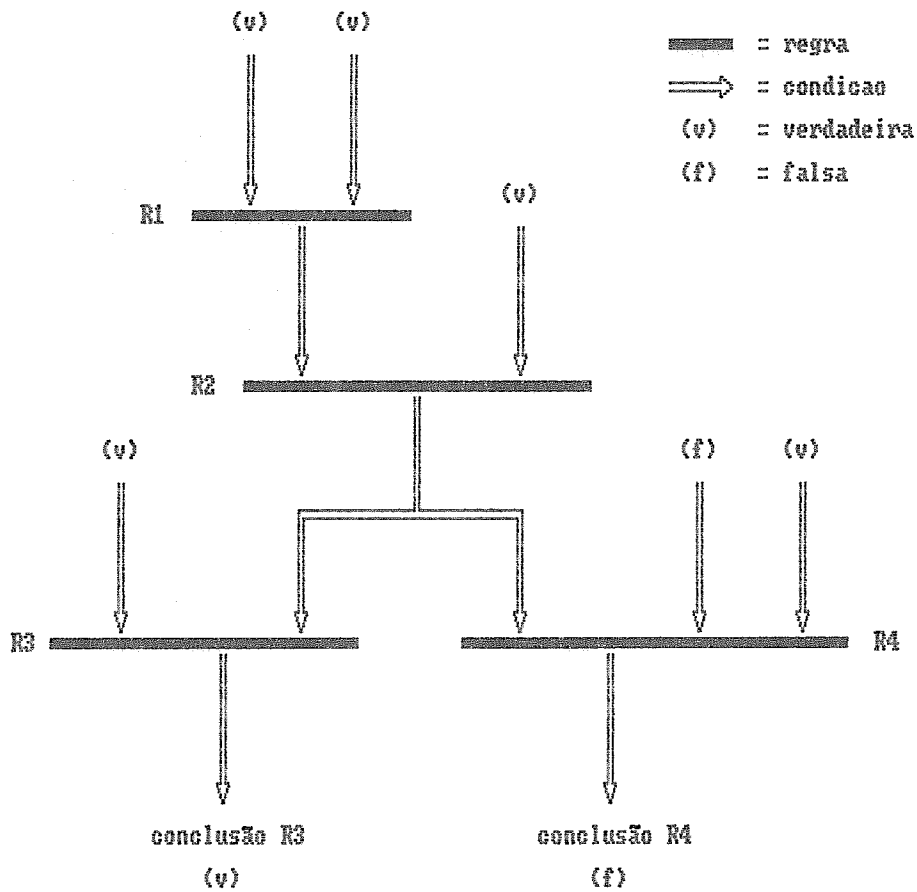

fig. IV.3-1 - Exemplo de ligação entre as regras da Base de Conhecimento

O mecanismo de inferência pode identificar a cada passo quais regras têm todas as suas condições satisfeitas. A aplicação de uma regra adiciona sua conclusão à memória de trabalho, o que é um padrão novo que é adicionado ao sistema. Esse novo padrão é comparado com o das CONDIÇÕES das regras existentes e assim por diante.

O encadeamento das regras na busca de uma solução é conhecido como Estratégia de Controle. Como as regras tem dois lados, o lado das CONDIÇÕES e o lado das CONCLUSÕES, há também duas possibilidades básicas para as estratégias de controle: o Encadeamento Progressivo e o Encadeamento Regressivo.

O Encadeamento Progressivo ("forward chaining"), que funciona como descrito acima, recebe esta denominação porque as CONDIÇÕES antecedem logicamente a CONCLUSÃO, e como partimos das CONDIÇÕES para as CONCLUSÕES, o caminhamento é para a frente [28].

No Encadeamento Regressivo ("backward chaining") um resultado já ocorreu e o objetivo é descobrir por quê. Portanto estamos voltando às origens do problema. Neste tipo de estratégia de controle, partimos de uma CONCLUSÃO que pretendemos estabelecer, ou um resultado que sabemos existir e buscamos as CONDIÇÕES (causas) que os originaram, procurando verificar quais dentre as CONDIÇÕES possíveis são suportadas por fatos. Neste caso então são ativadas a cada passo do mecanismo de inferência aquelas regras cuja parte da CONCLUSÃO tem o padrão casado com uma das CONDIÇÕES da regra anterior.

Um dos aspectos mais característicos dos sistemas baseados em conhecimento, segundo vários autores, é a habilidade desses sistemas de fornecer explicações para o usuário sobre as razões que permitiram ao programa chegar às conclusões ou resultados finais. Esta "Capacidade de Explanação", como é denominada, tem sido no entanto sobre-enfatizada [29]. O que se tem na prática é a apresentação pura e simples de frases associadas aos módulos de conhecimento que contribuíram para a solução. Nos sistemas baseados em regras, isto é obtido através da memorização do "encadeamento" das regras consideradas válidas e apresentação de uma mensagem pré-definida para cada uma dessas regras, na mesma ordem do "encadeamento". Esta sequência de frases permitiria seguir-se a linha de "raciocínio" do programa, mas na verdade se assemelha muito mais a um "trace" utilizado comumente nos programas convencionais, do que a uma explicação acerca de um resultado.

A Explanação pode no entanto ser útil para usuários que precisam adquirir confiança nas conclusões ou recomendações fornecidas pelo sistema, como também para auxiliar na criação/manutenção da Base de Conhecimento, quando, por meio de testes, deve-se verificar como o conhecimento adicionado ou alterado se incorpora ao sistema.

Os Sistemas Baseados em Regras, a par de suas grandes vantagens para determinados tipos de aplicações, apresentam na atualidade alguns pontos a serem desenvolvidos, de modo a permitir que este tipo de tecnologia tenha um emprego mais amplo:

- técnicas de verificação da consistência e completude do conjunto de regras;
- técnicas ou teorias de organização do conhecimento que permitam a expansão do sistema sem perda de inteligibilidade e sem introdução de redundâncias;
- compiladores especializados para a formatação de regras da Base de Conhecimento;
- bibliotecas de regras úteis para operações corriqueiras;
- integração desse tipo de sistema com sistemas convencionais;
- arquiteturas e linguagens adequadas ao processamento paralelo, com vistas à redução do tempo de processamento.

IV.4 A LINGUAGEM PROLOG

A lógica matemática pode ser utilizada como fundamento para a construção de sistemas baseados em conhecimentos. Aplicando-se os princípios da lógica na definição de uma linguagem computacional, os conhecimentos sobre um problema podem ser representados por meio de sentenças lógicas e, por meio da aplicação de leis básicas, conclusões podem ser obtidas, simulando-se desse modo o raciocínio necessário para se resolver o problema.

Como os princípios da lógica são independentes do problema tratado, uma linguagem lógica de programação permite construir-se um programa de maneira descritiva. O conhecimento representado inclui "premissas" ou "crenças" (afirmações que

consideramos verdadeiras) acerca dos objetos escolhidos para modelar o problema e ainda relacionamentos entre essas crenças e as conclusões que podem ser obtidas. A obtenção de conclusões a partir de hipóteses que se desejam testar é realizada de maneira sistemática com a utilização de Regras de Inferência, que são procedimentos de aplicação de princípios lógicos. Esses procedimentos, por independerem das características do problema tratado podem ser deixados implícitos, embutidos na própria linguagem de programação, do que resulta o aspecto descritivo de um programa desse tipo.

Esta forma descritiva de construção de programas permite que novas informações sobre um problema sejam adicionadas sem que haja necessidade de revisão de algoritmos e alteração de código de programas.

A linguagem PROLOG (de "PROgramming In LOgic") é uma linguagem lógica de programação, cuja primeira versão oficial foi desenvolvida na França no início da década de 70, tendo se tornado desde então popular entre os pesquisadores europeus da área de Inteligência Artificial. Nos Estados Unidos o PROLOG foi por muitos anos preterido em favor da linguagem LISP, por ser esta considerada mais poderosa, embora de aprendizado e uso mais difíceis do que o PROLOG. O maior interesse recente pelo PROLOG deveu-se ao surgimento de versões com bom desempenho para micro-computadores e ainda à decisão dos japoneses de utilizar esta linguagem como linguagem de sistema para a máquina de 5a geração, cujo projeto se encontra em andamento.

Um aspecto bastante característico da linguagem PROLOG refere-se à ênfase que se coloca, ao escrever um programa, na especificação ou descrição da natureza do poblema, em lugar de se ditarem os passos a serem seguidos pelo programa, como nas liguagens convencionais. Devido ao seu fundamento na lógica matemática, o PROLOG é único na sua habilidade de inferir (obter a partir de princípios formais de "raciocínio") conclusões a partir de hipóteses.

Um programa em PROLOG é escrito com informações sobre os objetos que escolhemos para modelar um problema, objetos esses que podem corresponder a entidades físicas ou a conceitos abstratos. Essas informações formam um coleção de dados ou fatos que declaramos para os objetos e ainda relacionamentos entre esses dados ou fatos. O programa pode ser visto, portanto, como um banco de dados.

Feita uma descrição do problema, de maneira lógica e formal, o usuário define uma "meta" ("goal") a ser alcançada, que também pode ser considerada como uma hipótese a ser testada. O programa, tenta provar se a meta ou hipótese é Válida (Verdadeira) ou Inválida (Falsa), utilizando-se para isto de procedimentos embutidos na própria linguagem. Se a meta é provada válida, resultados podem ser produzidos.

A representação do conhecimento em PROLOG é realizada tendo-se como base o conceito de PREDICADO. Um predicado pode ser visto como uma função que, quando ativada, retorna um valor "Verdadeiro" ou "Falso". Predicados expressam uma relação ou uma propriedade, sendo representados da seguinte forma:

nome_do_predicado(elem_1,elem_2, ... ,elem_n)

onde a palavra fora dos parênteses corresponde ao nome da relação ou da propriedade e os elementos entre parênteses podem ser nomes de objetos individuais ou nomes de variáveis. Em PROLOG, por convenção, escrevem-se os nomes de objetos com a inicial minúscula e as variáveis com a inicial maiúscula.

Um programa em PROLOG é constituído por um conjunto de "cláusulas" que podem ser FATOS ou REGRAS, construídas com o auxílio de predicados.

Um FATO declara a existência de uma propriedade de um objeto ou de uma relação entre objetos. Um FATO portanto é uma cláusula sempre Verdadeira, sendo representado como um predicado seguido de um ponto à direita. Por exemplo:

aresta(b,c).

é um FATO que pode indicar em um determinado problema a existência, em um grafo orientado, de uma aresta com origem no vértice "a" e extremidade no vértice "b". Deve-se notar que o significado de cada elemento dentro do predicado é arbitrário e definido a gosto do programador, mas uma vez definido, tem que ser seguido um padrão consistente de ordem dos elementos ao longo de todo o programa.

Uma REGRA é uma expressão que afirma que a verdade de um fato particular depende de outros fatos, segundo uma determinada lei. Em PROLOG as REGRAS podem ser representadas das seguinte forma:

P1 if (P2 and P3 and ... and Pn).

ou, alternativamente:

P1 :- P2 , P3 , ... , Pn.

Nas formas acima, P1 é um predicado que representa a CONCLUSÃO da regra. P2 a Pn são predicados que representam as PREMISSAS da regra, podendo corresponder a FATOS ou a CONCLUSÕES de outras regras, que terão que ser testadas para se verificar se estas premissas são válidas. Somente se todas as PREMISSAS são válidas (verdadeiras), a CONCLUSÃO da regra é também válida.

Além dos relacionamentos do tipo "AND" (indicados por ","), relacionamentos "OR" podem também ser representados (utiliza-se o símbolo ";") em uma regra, porém é mais usual, por ser mais legível, utilizarem-se regras múltiplas, que produzem o mesmo efeito. Por exemplo:

P1 :- (P2 , P3) ; (P4 , P5).

é equivalente a:

P1 :- P2 , P3.

P1 :- P4 , P5.

Neste caso diz-se que a regra P1 apresenta duas "instâncias". Ao tentar provar a regra, o PROLOG tenta fazê-lo inicialmente com a primeira instância. Se esta é válida, a regra é imediatamente considerada válida e a segunda instância

não é testada, caso contrário, o PROLOG testa então a segunda para determinar o resultado final.

Quando o usuário especifica uma meta, ou hipótese a ser provada, esta é colocada na seguinte forma (segundo a sintaxe do TURBO-PROLOG):

goal P1 , P2 , ... , Pn.

onde os Pn são predicados a serem testados. Cada predicado compõe uma sub_meta do problema.

O algoritmo de controle de inferência, embutido na linguagem, tenta provar as sub_metas uma de cada vez, sequencialmente da esquerda para a direita. Para cada sub-meta analisada são procuradas todas as ocorrências de fatos e regras no banco de dados do programa que possam ter os seus padrões casados com a sub_meta. Este procedimento, chamado de Unificação, realiza ainda a atribuição de valores às variáveis eventualmente utilizadas nos predicados.

Se uma regra tem a sua CONCLUSÃO casada com a sub-meta a ser testada, passa então a ser considerada como a nova meta a verificar e as suas PREMISSAS, as novas sub-metas. Este procedimento é aplicado recursivamente até que se consiga provar como válida uma sub_meta. Neste caso passa-se para a sub-meta imediatamente à direita da atual e continua-se o processo. Ao contrário, quando se consegue provar que uma sub-meta é inválida, o PROLOG retorna um passo no procedimento recursivo e tenta descobrir uma nova instância a ser testada

para a sub-meta imediatamente à esquerda da atual, processo este chamado de "Backtracking".

A figura abaixo apresenta um grafo orientado, constituído por 4 vértices e 5 arestas, com relação ao qual serão apresentados alguns exemplos e regras em PROLOG.

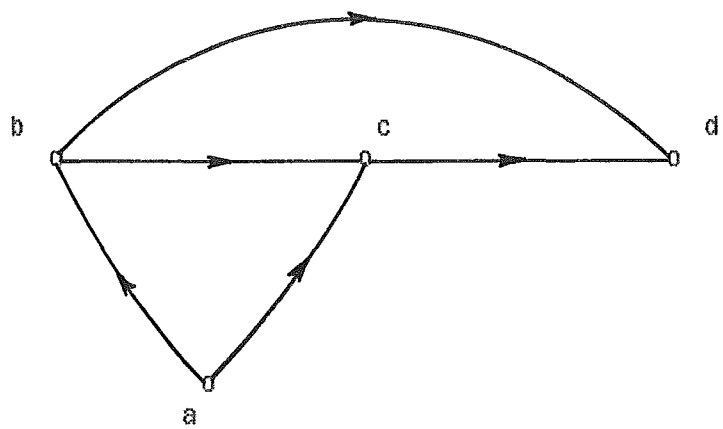

fig. IV.4-1 - grafo-exemplo

Este grafo pode ser descrito através dos seguintes FATOS:

```
aresta(a,b).
aresta(a,c).
aresta(b,c).
aresta(b,d).
aresta(c,d).
```

Suponhamos agora que o problema seja descobrir se um determinado vértice é um Sumidouro. Para que um vértice seja um Sumidouro somente podem existir arestas convergindo para ele, não podendo existir nenhuma aresta que o tenha como origem.

Um regra para determinar se um vértice é um Sumidouro poderia ter a seguinte formulação:

```
sumidouro(X) :- not( aresta(X,_) ).
```

onde o "not" é um operador que nega o resultado de um predicado e o símbolo "_" indica que o elemento nesta posição pode assumir qualquer valor dentre os possíveis, sem com isso afetar a regra.

Colocando esta definição de regra e mais os fatos acima que definem o grafo, poderíamos realizar algumas consultas para o programa, cada consulta correspondendo a uma hipótese a ser testada. Por exemplo, para a consulta:

```
sumidouro(a).
```

o PROLOG responderá: "False".

Neste caso, ao tentar provar o primeiro predicado da regra, "aresta(X,_)", a primeira instância do fato "aresta" no banco de dados que satisfaz o padrão do predicado é "aresta(a,b)" que é um fato que torna o predicado Verdadeiro. Devido ao operador NOT este resultado é negado e a meta tem o resultado final "False".

É interessante notar que, segundo a regra acima, se fizéssemos a consulta para um vértice inexistente, como por exemplo:

```
sumidouro(z).
```

o PROLOG responderia "True", o que é absurdo.

Esta aparente anomalia advém do fato de que o PROLOG trata de um "mundo fechado" de conhecimentos. Qualquer conhecimento não representado neste mundo é considerado Falso. Assim, ao procurar uma aresta do tipo "aresta(z,_)" e não a tendo encontrado, o predicado foi considerado Falso, mas devido à presença do "not", o resultado final foi "True".

O problema acima poderia ser corrigido melhorando-se a definição da regra, da seguinte forma:

```
sumidouro(X) :- not( aresta(X,_) ) , aresta(_,X).
```

onde o primeiro predicado exige que o vértice não seja origem de nenhuma aresta e o segundo, que o vértice apresente pelo menos uma aresta incidente.

A regra acima poderia ser reescrita em uma outra forma logicamente equivalente:

```
sumidouro(X) :- aresta(_,X) , not( aresta(X,_) ) .
```

onde simplesmente trocamos a ordem dos predicados na regra.

Esta segunda forma, no entanto, é bem menos eficiente do que a primeira, em termos de processamento. Para verificar este fato tomemos como exemplo a seguinte consulta:

```
sumidouro(c).
```

Para provar esta hipótese o PROLOG percorrerá os seguintes passos:

- tenta-se provar o predicado "aresta(X,_)", o qual é Verdadeiro, devido à instância "aresta(a,c)" existente no banco de dados.

- passa-se a tentar provar o segundo predicado, "not(aresta(X,_))", o qual, devido ao "not", é Falso.

- como o segundo predicado é Falso o PROLOG realiza o "backtracking", voltando a testar o primeiro predicado com uma nova instância de "aresta(_,X)". Isto é obtido com o fato "aresta(b,c)" e o predicado considerado novamente Verdadeiro.

- pela segunda vez o PROLOG tenta provar o segundo predicado, o qual é novamente "Falso".

- como não há novas instâncias para o primeiro predicado, o processo de "backtracking" não é continuado e o resultado final é "Falso".

Deve-se notar que antes do "backtracking" ser iniciado já se podia concluir ser a regra Falsa, devido à condição expressa pelo segundo predicado.

Este problema pode ser resolvido com a aplicação de um operador CUT (simbolizado por "!"), que permite um controle explícito do processo de inferência. O CUT é uma cláusula pré-definida do PROLOG que retorna o valor "Verdadeiro" apenas na primeira vez em que é ativado em uma regra (ou seja quando se está caminhando da esquerda para a direita). Quando um processo de "backtracking" é disparado, e o PROLOG retorna a um predicado à esquerda do atual, se este é um CUT, o processo de

"backtracking" na regra atual é interrompido e a regra considerada inválida.

A regra anterior seria escrita da seguinte forma, com o emprego do "CUT":

```
sumidouro(X) :- aresta(_,X) , ! ,  not( aresta(X,_) ).
```

sendo que agora a regra é considerada inválida logo da primeira vez que atinge o predicado "not(aresta(X,_))", para a mesma consulta anterior.

Outro predicado pré-definido da linguagem PROLOG que permite o controle explícito do processo de inferência é o predicado "FAIL". Este, quando ativado, retorna sempre o valor "Falso", de modo que é usado quando se quer forçar o "backtracking".

Um exemplo de uso de CUT e FAIL em conjunto pode ser visto abaixo, no qual a mesma regra anterior foi reescrita na forma de uma regra múltipla:

```
sumidouro(X) :- aresta(X,_) , ! , fail.

sumidouro(X) :- aresta(_,X).
```

A primeira instância da regra especifica que se um vértice é origem de alguma aresta, já se pode concluir que a regra é falsa, sem necessidade de se examinar a segunda instância. Isto ocorre porque: o CUT é alcançado e é válido desta vez; a seguir o "fail" é ativado e o "backtracking" iniciado; com o retorno ao CUT, este termina o "backtracking", impedindo que a segunda

instância da regra seja examinada; o resultado "Falso" é obtido.

Ao contrário, caso o vértice não seja origem de nenhuma aresta, o primeiro predicado da primeira instância da regra falha, e portanto o CUT e o FAIL não são ativados. Como a primeira instância da regra falhou, o PROLOG passa então a tentar provar a segunda instância.

A utilização correta dos predicados CUT e FAIL requer extremo cuidado por parte do programador, não apenas devido à eficiência computacional, mas principalmente devido aos efeitos inesperados que podem ser obtidos por uma localização errônea desses predicados. Exemplos de aplicações de CUT e FAIL para diversas finalidades podem ser encontrados em Rogers [43].

A existência de predicados que interferem explicitamente, a comando do programador, no processo de inferência realizado pelo PROLOG corrompe de certo modo um dos aspectos mais interessantes da linguagem, que corresponde à abordagem declarativa dos problemas. Para a correta aplicação destes predicados o programador tem que ter em mente os procedimentos embutidos na linguagem. Nos exemplos anteriores pudemos constatar que não somente o uso desse tipo de predicados, mas a própria ordem de colocação dos predicados nas regras exige uma análise meticulosa. Com tudo isso, não somente a construção de programas torna-se complexa mas também a legibilidade final fica prejudicada.

Como conclusão podemos observar que a linguagem PROLOG é bastante avançada nos seus conceitos, na sua maneira de tratar

os problemas e na sua capacidade lógica. Ao mesmo tempo apresenta-se como uma linguagem que requer cuidados especiais e muita habilidade do programador, principalmente quando da sua aplicação a problemas práticos.

O texto clássico sobre PROLOG é o de Clocksin e Mellish [42]. Bons textos introdutórios são os de Rogers [43] e Bharath [20]. A descrição da versão TURBO-PROLOG para IBM-PC e compatíveis pode ser encontrada em Borland [40] e Townsend [41].

V. SISTEMA EXEMPLO

V.1 OBJETIVO

Nos capítulos anteriores estudamos os sistemas de alarme, na sua integração com o ambiente operacional do centro de controle. Foram observados os conceitos importantes para o desenvolvimento de um sistema desse tipo visando à sua aplicação em um centro de controle moderno. O aspecto da flexibilidade de adaptação às variações nas condições e necessidades operativas é considerado fundamental, tanto no dia a dia como para acompanhar a evolução do centro de controle ao longo do tempo.

Algumas sugestões de características desejáveis foram colocadas. Na prática observa-se que os sistemas existentes na atualidade carecem dessas características, por serem as técnicas convencionais de desenvolvimento de sistemas inadequadas para a construção desse tipo de software. Alguns novos desenvolvimentos de sistemas convencionais tem sido reportados [18], mas representam muito pouca evolução em relação aos sistemas anteriores.

Pela sua estrutura, o processamento de alarmes é um problema que pode, a princípio, ser adequadamente tratado com a aplicação da tecnologia de Sistemas Especialistas. Um relatório recente do EPRI [30],no qual são estudados conceitos avançados na operação de sistemas, indica o processamento de alarmes como

uma das áreas mais promissoras para a aplicação deste tipo de técnica.

Por outro lado, não se tem notícia do emprego atual desse tipo de ferramenta na rotina de operação de sistemas elétricos. Alguns desenvolvimentos tem sido reportados [31] ou se apresentam ainda em fase de estudo de viabilidade [13].

Outros trabalhos relatam propostas ou idéias para o desenvolvimento de Sistemas Especialistas para outras funções no centro de controle, diferentes do processamento de alarmes, como por exemplo, Diagnóstico de Faltas [32,33,34], Recomposição do Sistema [35], Controle de Reativo/Tensão [36] e Assistência ao Operador [37].

Este capítulo apresenta um sistema-exemplo, desenvolvido como parte das atividades desta pesquisa de tese, com o intuito de avaliar a viabilidade de construção de um sistema especialista para processamento de alarmes.

Este sistema-exemplo não se destina à aplicação direta em um centro de controle, sendo mais um simulador "off-line", que inclui, além do tratamento de alarmes, a simulação das interfaces com outros subsistemas e da interação com o operador (Interface Homem-Máquina). Permite estudar os vários aspectos do processamento de alarmes e verificar como as características apontadas podem ser implementadas.

É importante notar que o sistema-exemplo pode ser considerado como um protótipo de um sistema a ser desenvolvido para aplicação "on-line". O emprego de protótipos é um recurso

altamente recomendável na área de sistemas especialistas [39]. Um protótipo permite o desenvolvimento incremental do sistema ao mesmo tempo em que se adquire experiência e aumenta a compreensão do problema, complementando-se e verificando-se o conhecimento adquirido, antes da sua aplicação no ambiente do centro de controle.

Um grande esforço foi colocado na definição do formato das Regras de processamento, com o intuito de se oferecer um meio de introdução e alteração de conhecimento, intuitivo e altamente legível, de modo a não se exigir pessoal altamente especializado para esses fins.

O sistema foi desenvolvido na linguagem TURBO-PROLOG versão 1.0 [40,41], para um micro-computador compatível com IBM PC/XT, com a seguinte configuração:

- memoria principal com 640 kbytes;
- 2 diskettes de 5 1/4 ";
- impressora de 80 colunas;
- monitor de vídeo monocromático.

As seções a seguir descrevem a arquitetura do sistema-exemplo, a compilação de alarmes e a implementação das Regras de processamento de alarmes. Um caso-teste é apresentado.

V.2 ARQUITETURA

O sistema-exemplo é formado por dois programas, o Compilador de Definições de Alarme (CDA) e o Processador de

Alarmes (PA). Neste item detalharemos somente a arquitetura do Processador de Alarmes, sendo o Compilador apresentado em um ítem proprio.

O PA é dividido em três módulos principais que se interligam por meio da Base de Conhecimento (BC), conforme representado na figura V.2-1. O Núcleo do sistema é formado pelo módulo Filtro e pela Base de Conhecimento. Os demais módulos destinam-se a simular as interfaces com outros subsistemas ou Resolvedores de Problemas encontrados em um centro de controle.

Na Base de Conhecimento, onde são armazenados os módulos de conhecimento do sistema, podem-se distinguir Fatos e Regras específicos para cada módulo individual e os Fatos e Regras gerais que modelam o problema em questão. Cada módulo possui o seu mecanismo de inferência próprio para a finalidade a que se destina e é relativamente independente dos outros em termos de processamento. Assim, cada um dos módulos pode ser visto como um sistema especialista que trata uma parte do problema e o sistema completo como um Sistema de Especialistas Cooperantes, que se integram através da Base de Conhecimento.

Todos os módulos são escritos em PROLOG. A Memória de Trabalho não se encontra explicitada na arquitetura por ser embutida na linguagem e portanto transparente para o programador. Na memória de trabalho o Prolog armazena estruturas que controlam os processos de Unificação e "Backtracking", característicos da linguagem [42,43].

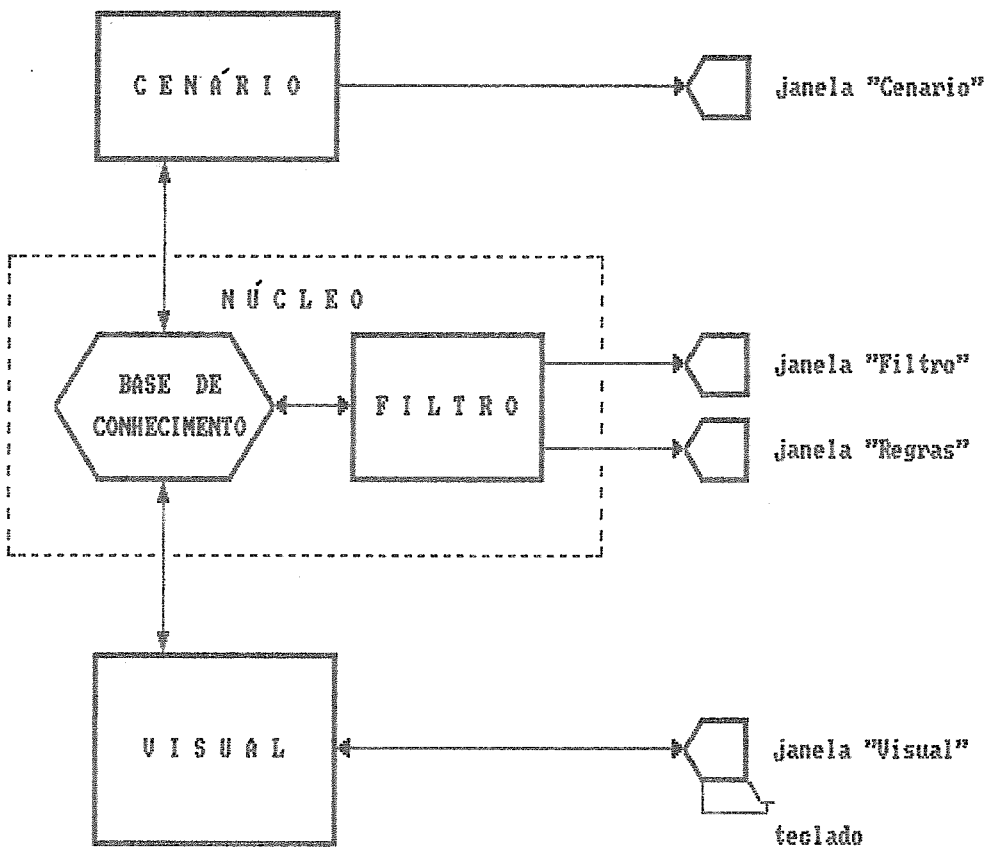

fig. V.2-1 - Componentes do Processador de Alarmes

O sistema é provido de várias janelas de vídeo dedicadas (figura V.2-2). Estas janelas permitem o acompanhamento dos eventos no sistema, a observação dos resultados parciais e ainda a interação com o usuário. Os módulos Cenário e Visual possuem cada um a sua janela própria e o módulo Filtro gerencia duas janelas, conforme será visto adiante.

O sistema realiza uma varredura cíclica dos 3 módulos, determinando a cada momento quais módulos devem ser executados. O módulo Cenário é processado sempre que há um evento programado para o instante atual. O módulo Visual é executado a intervalos regulares (2 segundos) para atualização de uma lista de alarme na tela ou ainda sob demanda, sempre que há uma interação com o usuário via teclado. O módulo Filtro pode ser ativado por exceção, sempre que um Pedido de tratamento de alarme é depositado na Base de Conhecimento, ou ainda por tempo marcado, quando se trata de eventos esperados.

Os pedidos de tratamento de alarme são inseridos na Base Conhecimento pelo módulo Cenário.

Os Pedidos de tratamento de alarme reconhecidos pelo sistema são os seguintes:

- Gerar um alarme - dado que foi detetada uma condição de alarme, pede-se ao processador de alarme que anuncie a mensagem correspondente.

```
( visual )                                                    ( cenario )
 HH:MM:SS                                   Lista: id_lista
rHH:MM:SS texto_da_mensagem_de_alarme. chave1.. chave2.. urg      cenas
            .                                                   executadas
            .
            .
            .
            . (demais mensagens da mesma lista)                   hora
            .                                                      da
            .                                                    proxima
            .                                                     cena
            .

( filtro )                                  ( regras )

identificacao do pedido                     nome e narrativa sucinta
      recebido.

                                               das Regras Externas
 resultado final do
tratamento do pedido                           consideradas validas
```

fig. V.2-2 Formato das janelas de vídeo do Processador de Alarmes

- Inibir um alarme - uma determinada mensagem de alarme para um elemento específico não deve ser apresentada.
- Habilitar um alarme - um determinado alarme, que estava inibido, deve retornar à condição de poder ser apresentado, quando ocorrer.
- Eliminar um alarme - uma determinada mensagem de alarme deve ser eliminada de todas as listas de alarme onde se apresenta.
- Reconhecer um alarme - informar ao sistema que o despachante, ou algum módulo de software, tomou conhecimento da anunciação de um determinado alarme.
- Inserir um alarme na lista de esperados - informar ao sistema que é prevista a ocorrência de um determinado alarme para um elemento específico, dentro de um intervalo de tempo fornecido; portanto este alarme não deve ser anunciado.
- Remover um alarme da lista de esperados - informar ao sistema que a ocorrência prevista de um alarme dentro de um intervalo de tempo anteriormente fornecido não é mais válida.

Módulo CENÁRIO

O módulo Cenário funciona como um simulador, substituindo as fontes de alarme encontradas em um sistema real. Sua tarefa é controlar o ambiente de simulação, ativando em horas pré-determinadas os Pedidos de tratamento de alarme. Esses pedidos incluem aqueles advindos das fontes de alarme bem como

das atuações do despachante nas consoles de operação (reconhecimento, eliminação, inibição, habilitação).

Os pedidos produzidos pelo Cenário são introduzidos na Base de Conhecimento na hora prevista e disparam a execução do módulo Filtro.

Para o processamento do sistema é necessária a elaboração de um cenário de simulação composto de "Cenas" que correspondem a eventos com hora marcada. Essas cenas são colocadas em um arquivo com o uso de um editor de textos comum. O formato deste arquivo é apresentado na figura V.2-3.

Cada cena dá origem a um registro do arquivo de cenário. O primeiro registro é utilizado apenas como demarcador dos campos dos registros posteriores. O segundo registro define a hora inicial de simulação, a qual será estabelecida no início do processamento como hora do sistema. Deve-se observar que apenas a hora inicial é arbitrada; daí por diante o sistema trabalha em tempo-real.

Os registros seguintes definem as várias cenas do cenário de simulação. Comentários podem ser utilizados para melhorar a legibilidade do arquivo, bastando introduzir-se um ponto-e-virgula (";") na coluna 1.

```
|--------|------------|------------|--------|--------|--------|--------|
 HH:MM:SS
 HH:MM:SS gerar_alarme id_alarme... chave1.. chave2..|
 HH:MM:SS reconhecer   id_alarme... chave1.. chave2..|
 HH:MM:SS eliminar     id_alarme... chave1.. chave2..|
 HH:MM:SS inibir       id_alarme... chave1.. chave2..|
 HH:MM:SS habilitar    id_alarme... chave1.. chave2..|
 HH:MM:SS esperado     id_alarme... chave1.. chave2.. horainic hora_fim|
 HH:MM:SS nao_esperado id_alarme... chave1.. chave2.. horainic hora_fim|
 HH:MM:SS dummy        |
; ... comentario ...
- ... cena qualquer desativada ...
```

fig. V.2-3 - Formato do arquivo de cenário

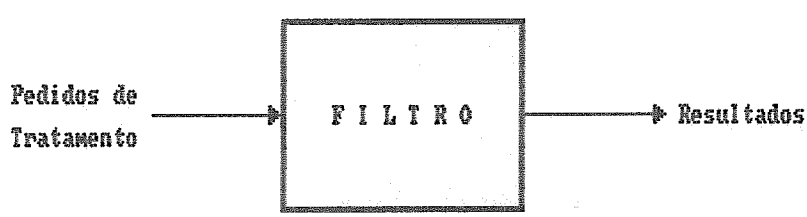

fig. V.2-4 - A função do módulo "Filtro" no sistema

Para cada cena define-se a sua hora de ativação, a identificação do pedido que será gerado e os parâmetros próprios para cada tipo pedido. Os registros de cena são colocados sequencialmente no arquivo na ordem crescente da hora de ativação. Várias cenas podem ser definidas para ocorrer no mesmo instante. A coluna 1 dos registros pode ser utilizada para a desativação de cenas. Caso seja colocado nesta coluna o sinal ("-") a cena será desconsiderada durante o processamento. Isto permite algumas variações na simulação a partir do mesmo cenário básico. No campo "Id" deve ser preenchida a identificação do alarme que será tratado (as Id's de alarme são definidas com o uso do CDA). Os campos denominados "chave1" e "chave2" correspondem a duas chaves do banco de dados que identificam univocamente o elemento a que se refere o alarme. Por exemplo Id=KVURG, chave1=B1_KV, chave2=REM33 poderiam indicar um alarme de urgência de tensão detetado no ponto de medição B1_KV da remota REM33. Os registros correspondentes ao tratamento de alarmes esperados informam o intervalo (hora inicial e hora final) de ocorrência prevista do alarme.

Cada um dos tipos de pedido anteriormente apresentados possui uma cena correspondente no arquivo de cenário. A cena do tipo "dummy" no entanto não corresponde a nenhum pedido, tendo sido incluída para melhorar o controle do tempo de simulação, conforme será visto adiante.

O sistema permite que se observe o processamento em tempo-real. Desse modo os eventos (cenas) são executados quando o tempo do sistema alcança o instante programado para uma cena do arquivo de cenário. É possível no entanto, a qualquer

momento, executar-se um salto no tempo do sistema para o instante da próxima cena. Este salto é executado sob controle do usuário, com intermediação do módulo Visual. Neste ponto é que as cenas "dummy" mostram a sua utilidade. Através destas pode-se executar um salto não para o momento exato de um evento, mas para um momento próximo, de modo a permitir uma melhor observação da ocorrência em tempo-real.

Módulo VISUAL

O módulo Visual tem a função de apresentar uma réplica aproximada dos recursos de interface homem-máquina (IHM) que o despachante tem à disposição nas consoles de operação do centro. É atraves da IHM que o operador interage com o sistema. No caso do processamento de alarmes, a IHM é responsável pela anunciação dos alarmes, através das listas de alarmes que são apresentadas na tela ou por outros meios. O despachante pode usualmente visualizar as listas de alarme, inibir, habilitar, reconhecer ou eliminar alarmes individuais, ou em bloco.

O módulo Visual é uma IHM simplificada. Das funções acima descritas apenas a visualização é implementada. As demais funções são executadas a partir do arquivo de cenário, gerando pedidos a serem tratados pelo módulo Filtro. Isto permite um melhor controle do ambiente de simulação.

O conteúdo da janela "Visual" é mostrado na figura V.2-2, juntamente com o formato das mensagens de alarme . Deve-se notar que cada lista de alarmes existente no sistema dá origem

a uma tela própria onde são apresentadas as mensagens atuais de alarme que formam a lista. Na coluna 1 das mensagens a letra "r" indica um alarme cuja existência já terá sido reconhecida explicitamente pelo operador. O campo "texto" corresponde ao texto da mensagem de alarme, definido em tempo de compilação de alarmes, e apresentado como uma narrativa sucinta do evento considerado como condição de alarme. O campo "URG" apresenta um mnemônico da urgência do alarme, definida por meio do GDA.

O módulo Visual realiza o tratamento do teclado, através do qual o usuário pode interagir com o sistema. As funções do teclado são apresentadas na tabela V.2-1. Algumas teclas são pré-definidas para a apresentação de determinadas listas de alarme mais importantes, de modo a agilizar o acesso a estas.

Uma função de muita utilidade é a apresentação da definição das Regras Externas na sua forma em linguagem natural, o que permite ao usuário o acompanhamento detalhado das regras que são executadas (a informação dos nomes das regras consideradas válidas é dada a cada momento pelo módulo Filtro, através da janela "Regras").

Duas funções muito importantes são a Ativação e a Desativação de Regras Externas. Através destas funções pode-se interferir no tratamento de alarmes em tempo-real e observar-se os efeitos que são produzidos. Quando uma regra é desativada, passa a não ser considerada no tratamento de alarmes executado pelo módulo Filtro. Para se ativar ou desativar um regra, esta deve primeiramente ser apresentada na janela "Visual" na sua forma de linguagem natural, devendo-se em seguida pressionar a

tecla programada própria para a ativação/desativação da referida regra (vide tabela V.2-1).

O usuário pode, a qualquer momento, suspender a simulação por um tempo indeterminado, durante o qual o tempo de simulação é congelado, e retornar posteriormente à simulação, quando então o sistema restaura o último instante anotado. Esta característica permite ao usuário a consulta demorada às listas de alarme ou às definições de regras sem a preocupação de que a próxima cena possa ser disparada enquanto a consulta é feita. O usuário pode ainda reinicializar a simulação, fazendo voltar o sistema ao instante inicial, definido no arquivo de cenário.

Módulo FILTRO

O módulo Filtro e a Base de Conhecimento formam o núcleo, que é a parte mais importante do sistema. Este módulo é responsável pelo processamento dos pedidos de tratamento de alarme e gerenciamento das diversas listas de alarme, procurando implementar algumas características adaptativas.

O nome "Filtro" advém da sua função de passar os pedidos feitos por outros agentes através de um processo que pode determinar efeitos diversos daqueles previstos nos pedidos originais. Assim, por exemplo, um pedido de geração de um alarme pode levar à eliminação de outros alarmes, pode gerar simplesmente o alarme pedido, ou ainda ser completamente

ignorado, dependendo da situação atual. É este processo de filtragem que permite adicionar-se com grande flexibilidade as características de adaptabilidade requeridas.

De maneira esquemática a função do Filtro pode ser observada na figura V.2-4, anteriormente apresentada.

O módulo Filtro é dividido em três partes principais: o Escalonador (ESC), o Processador Básico de Alarmes (PBA) e o Processador de Regras Externas (REX). Os dois primeiros compartilham a Janela "Filtro" e o último possui a sua Janela dedicada ("Regras"), como pode ser observado na figura V.2-5.

A sequência de processamento dentro do módulo Filtro é a seguinte: primeiramente cada pedido recebido é passado ao Escalonador o qual determina quais regras externas se encontram ativas; cada regra ativa é processada pelo REX, resultando em regras válidas e não-válidas; as regras consideradas válidas executam procedimentos próprios de tratamento de alarmes; como consequência as listas de alarme são alteradas e o pedido pode ser mantido ou ser eventualmente suprimido por uma das regras válidas; finalmente, após o processamento das regras externas e se o pedido não tiver sido suprimido por uma delas, este é então passado para o PBA, que executa as suas ações básicas pré-definidas.

TECLA	F U N C A O
+	avanca o tempo para o instante da proxima cena.
F1	apresenta uma Regra Externa em linguagem natural.
F2	apresenta uma Lista de Alarmes.
F3	torna ativa uma Regra Externa.
F4	torna inativa uma Regra Externa.
?	'help' - lista as funcoes do teclado.
l	apresenta o nome das listas de alarme definidas.
F9	suspende/retoma a simulacao.
F10	reinicializa a simulacao.
ESC	fim de programa.

tabela V.2-1 - Funções do teclado para o modulo 'Visual'.

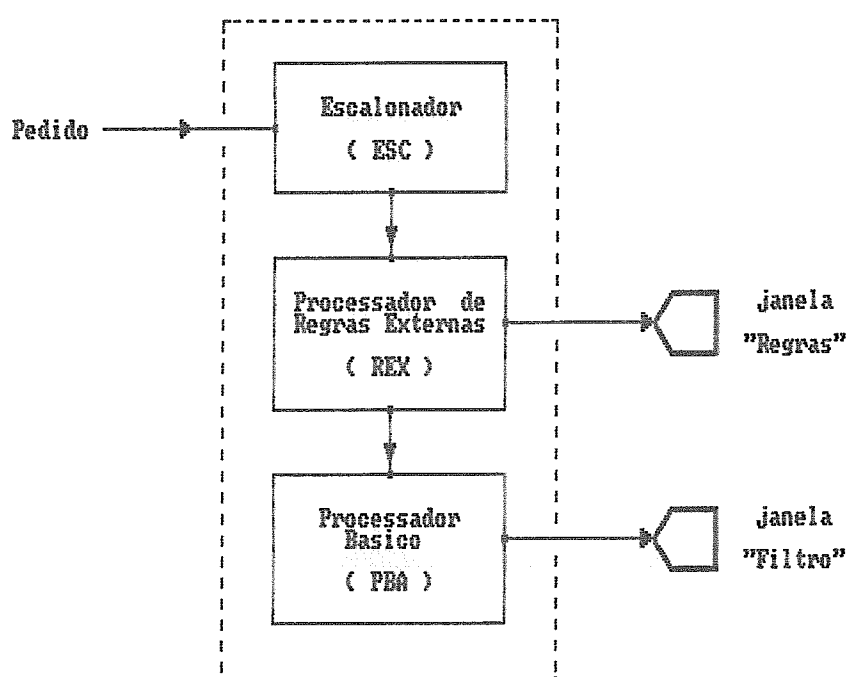

fig. V.2-5 - Organização do módulo "Filtro"

O ESC é responsável por identificar as Regras Externas que se encontrem ativas no momento, e que portanto podem ser aplicadas (a ativação e a desativação das Regras Externas são realizadas pelo módulo Visual). Além disso, informa através da sua janela de vídeo o tipo de pedido a ser tratado, no momento da sua recepção.

O REX realiza o processamento das Regras Externas, que são regras modificáveis diretamente pelo usuário. Essas regras tratam principalmente das associações de alarmes e do seu tratamento quando ocorrem em conjunto. Podem ainda fazer referência a identificações específicas de alarme, fontes de alarme ou localizações, de modo a identificar situações operativas, e produzem muitas vezes efeitos diversos daqueles originalmente especificados pelos pedidos individuais.

O REX utiliza a janela "Regras" para informar os nomes e fornecer uma explanação sucinta das regras consideradas válidas no tratamento do pedido atual. Caso nenhuma regra tenha-se mostrado válida, este fato é informado.

O PBA é constituído por regras escritas em PROLOG, não diretamente modificáveis pelo usuário, que realizam o tratamento básico dos pedidos feitos. Implementa o gerenciamento das diversas listas de alarme existentes, aceitando um pedido por vez e produzindo quando possível o efeito previsto pelo pedido, como: geração (inserção nas listas de alarmes), eliminação (remoção de todas as listas de alarme onde aparece), habilitação, inibição, reconhecimento,

inserção/remoção da lista de esperados. Segue uma lógica básica pré-definida para o tratamento de alarmes, como por exemplo:

- pedidos de eliminação de alarmes não são aceitos se o alarme é do tipo que necessita reconhecimento.
- pedidos de geração não são aceitos se o alarme se encontra inibido.
- pedidos de reconhecimento de alarme provocam a eliminação do alarme se este é do tipo "eliminar no reconhecimento".
- pedidos aceitos de geração de alarme devem ser inseridos não somente nas listas correspondentes pré-definidas como também na lista do "macro-alarme" e na lista do "local" informados pelo pedido particular em análise.
- pedidos de geração de alarme não são aceitos se ocorrerem dentro do intervalo esperado.
- um alarme especial é gerado se um alarme esperado não ocorrer dentro do intervalo previsto.

As regras de que se compõe o PBA podem ser consideradas Regras Internas, ou Fixas, e necessitam pessoal especializado para a sua alteração. Essas regras atuam segundo os atributos de alarme presentes nas "definições de alarme" produzidas pelo Compilador de Definições de Alarme. Não fazem referência a identificações específicas de alarme ou de fontes de alarme.

O PBA utiliza a Janela dedicada "Filtro" para informar o resultado das suas ações básicas, normalmente confirmando o efeito esperado do pedido, outras vezes informando porque o pedido não pode ser executado, como por exemplo um pedido de geração de alarme que era esperado.

V.3 COMPILADOR DE ALARMES

O Compilador de Definições de Alarme (CDA) é um programa independente processado em modo "off-line" e que permite definir e atualizar as diversas mensagens de alarme a serem suportadas pelo Processador de Alarmes.

Este programa é escrito em PROLOG, sendo processado em modo interativo via teclado e janelas de vídeo. As definições são entradas pelo usuário interativamente e após verificada a sua validade são armazenadas em arquivo próprio. O conteúdo deste arquivo é carregado na Base de Conhecimento, quando da execução do Processador de Alarmes.

São três os modos de operação do programa : Alarme, Lista, e Severidade. Cada modo permite um certo número de operações-padrão sobre as entidades correspondentes, como: Definir, Alterar, Eliminar, Listar e Imprimir. Assim, no modo Lista, por exemplo, essas operações são efetuadas sobre Listas de Alarme somente.

Alguns comandos gerais, independentes de modo, são:

- Consultar - carrega as definições contidas em um arquivo já existente.
- Salvar - salva no arquivo de definições as modificações efetuadas no processamento corrente.
- Limpar - apaga todas as definições carregadas na memória, mantendo o conteúdo do arquivo.
- Fim - termina a execução do programa.

Os atributos de Alarmes, Listas e Severidades são pré-definidos. O CDA apenas preenche valores para as várias ocorrências dessas entidades. Os atributos definidos para o caso-exemplo podem ser observados no Apêndice I, devendo-se notar que foram incluídos apenas os necessários para a demonstração do sistema-exemplo, não tendo havido a preocupação de incluir todos os que seriam necessários para o processamento em um centro de controle. Os atributos das definições de alarme para o caso-exemplo são os seguintes:

Id - identificação do alarme - mnemônico utilizado para informar o tipo específico de alarme que será processado quando de um pedido de tratamento de alarme. As Id's de alarme são unívocas em tempo de compilação de alarmes, isto é, não podem ser definidos dois alarmes com a mesma identificação. As Id's de alarme, recebidas nos pedidos de tratamento, são armazenadas internamente pelo PA, não sendo incluídas nas listas de alarme apresentadas na janela "Visual"

Texto - texto ou narrativa da mensagem de alarme, apresentada como uma descrição sucinta da condição de alarme.

Severidade - mnemônico do grau de importância relativa que o alarme representa para a operação. Assim, por exemplo, há alarmes de informação, de advertência, de urgência, etc.

Listas - sequência de listas de alarme às quais este alarme pertencerá quando for anunciado. Cada lista de alarme dá origem a uma tela individual de "display", apresentada pelo módulo "Visual".

Tipo - um qualificador para o tipo genérico do alarme definido. Vários alarmes com Id's diferentes podem possuir o mesmo Tipo de alarme. A definição dos "Tipos" de alarme permite a construção de Regras Externas mais genéricas do que seria permitido com a referência a Id's individuais.

V.4 FORMAÇÃO DA BASE DE CONHECIMENTO

A Base de Conhecimento armazena todos os módulos de conhecimento necessários para o processamento de alarmes. Esses conhecimentos descrevem não somente as estruturas que suportam os alarmes, suas características, formação e organização, mas ainda prescrevem os diferentes tipos de tratamento para as diversas situações e eventos operativos previsíveis.

O conhecimento internalizado pelo sistema é representado por meio de Fatos e Regras, cujo conceito foi apresentado no capítulo IV e segue os padrões definidos pela linguagem PROLOG.

A Base de Conhecimento apresenta duas partições principais, a Partição Estática e a Partição Dinâmica, que diferenciam os módulos de conhecimento pelo MODO de internalização dos mesmos no sistema.

Os módulos da Partição Estática representam conhecimentos que são incluídos no sistema em tempo de compilação do Processador de Alarmes (PA). Como consequência, qualquer alteração ou acréscimo efetuado nesta partição implica na necessidade de recompilação do programa.

Formam a Partição Estática as Regras, que podem ser Internas e Externas.

As Regras Internas fazem parte do código do programa, exigindo para a sua manutenção, pessoal especializado em programação PROLOG. Correspondem às regras do PBA que realizam o tratamento básico de alarmes (geração, inibição, eliminação, etc) e ainda as regras que definem a lógica dos módulos Cenário e Visual. Também são consideradas como regras internas as "Primitivas" formadoras de Regras Externas, a serem detalhadas posteriormente.

As Regras Externas, por outro lado, podem ser construídas por não-programadores, normalmente os próprios usuários do sistema (operadores ou pessoal de estudo), escritas em uma

meta-linguagem derivada do PROLOG e armazenadas em arquivo próprio, antes de carregadas na Base de Conhecimento.

A necessidade de recompilação do PA para se implementar qualquer alteração nas Regras Externas resulta de uma restrição da linguagem utilizada e não da arquitetura do sistema. O TURBO-PROLOG não permite a inclusão de Regras em tempo de execução, somente admitindo a inclusão de Fatos. Deve-se notar no entanto que o sistema permite ao usuário incluir ou excluir de consideração regras individuais pertencentes ao conjunto de regras pré-definidas, em tempo de execução, por intermédio da função Ativar/Desativar Regras, provida pelo módulo Visual.

A Partição Dinâmica é composta exclusivamente por módulos de conhecimento que representam Fatos, os quais são incorporados ao sistema em tempo de execução. Não é necessária a recompilação do PA quando esses módulos são alterados, bastando que o sistema seja reinicializado.

Os Fatos armazenados na Partição Dinâmica podem ser divididos em duas classes de módulos de conhecimento: os fixos e os variáveis.

Os módulos fixos correspondem às definições carregadas na Base de Conhecimento com o uso do predicado "Consult" do PROLOG. São módulos fixos as definições de alarme, lista, severidade, etc, preparadas com o auxílio do CDA e armazenadas em arquivo próprio e ainda as definições de teclas de função, contendo as funções do teclado e armazenadas da mesma forma em arquivo dedicado.

Os módulos variáveis correspondem às estruturas que se alteram ao longo do processamento do programa. Essas estruturas são incorporadas ou retiradas da Base de Conhecimento dinâmicamente, com o uso de predicados do tipo "Assert" e "Retract" do PROLOG. São exemplos de módulos dinâmicos:

- os pedidos de tratamento de alarme;
- as mensagens atuais de alarme;
- os alarmes que se encontram inibidos;
- os alarmes que se encontram reconhecidos;
- os eventos esperados;
- a lista de Regras Externas que se encontram ativas;
- sublistas temporárias de alarme, existentes durante o tempo de execução de uma Regra Externa;
- variáveis e estruturas locais para cada um dos módulos de software do PA;

A figura V.4-1 apresenta esquematicamente a formação da Base de Conhecimento e os tipos de interação utilizados para criá-la e atualizá-la.

A figura V.4-2 apresenta um diagrama do conteúdo da Base de Conhecimento e seu particionamento.

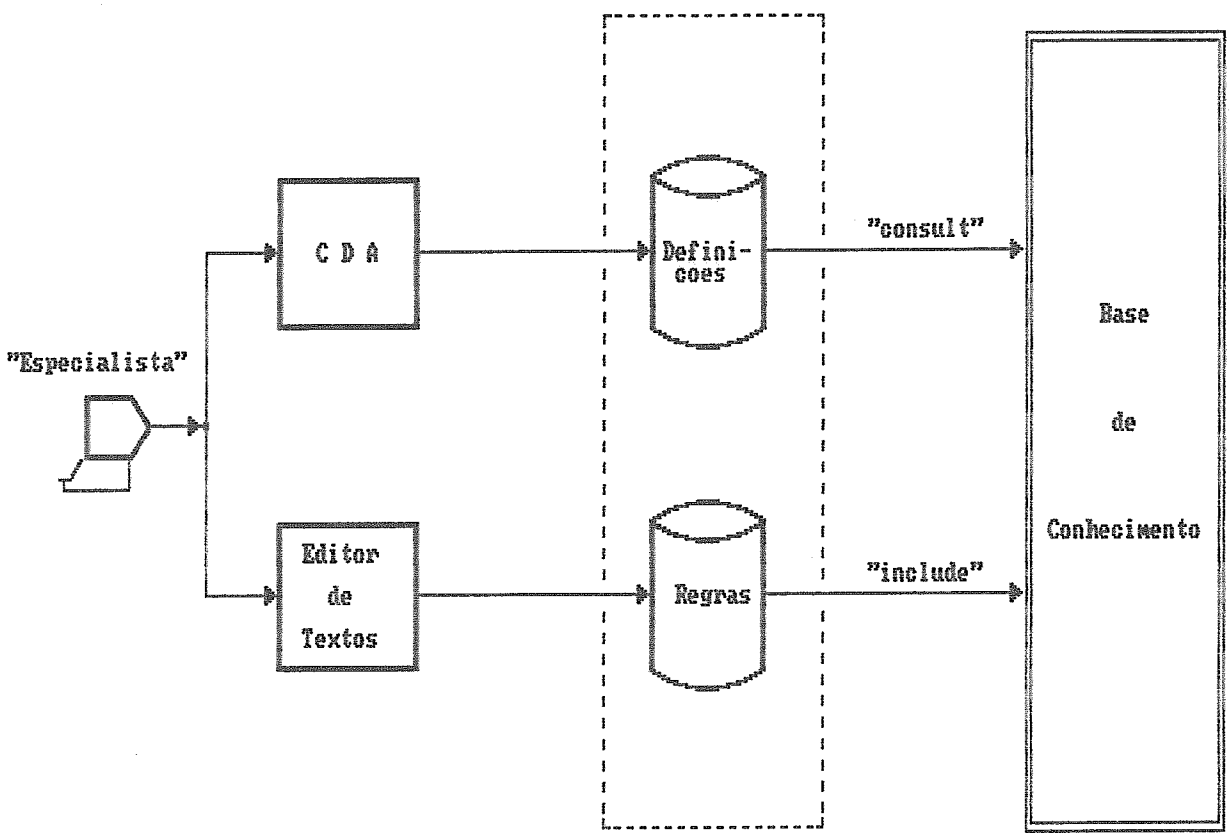

fig. V.4-1 - Formação da Base de Conhecimento

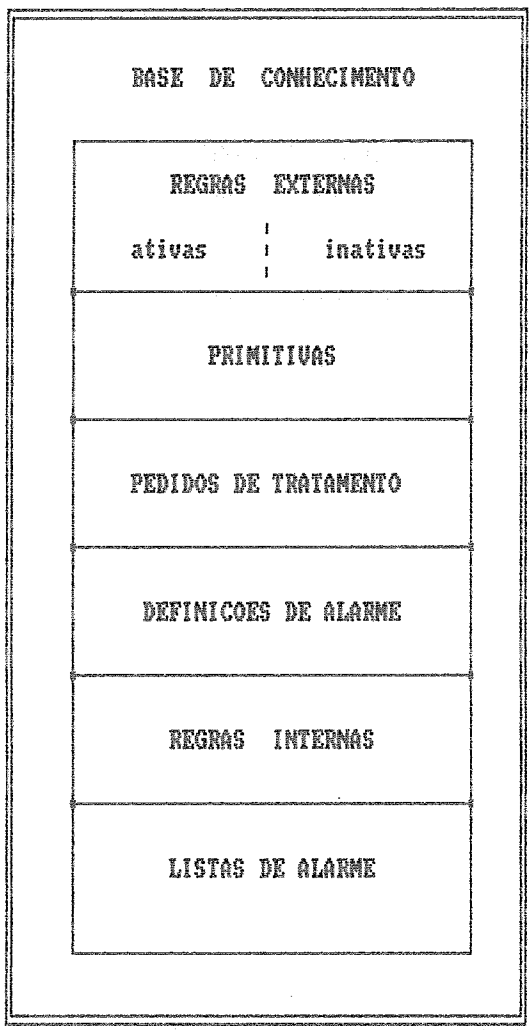

fig. V.4-2 - Conteúdo da Base de Conhecimento

V.5 FORMAÇÃO DAS REGRAS EXTERNAS

As Regras Externas são o meio pelo qual se pode implementar as heurísticas operativas, que permitem ajustar o tratamento dos alarmes às necessidades da operação em tempo-real.

As Regras Externas se destinam a reconhecer situações operativas e agir correspondentemente. O reconhecimento das situações específicas que devam ser tratadas pode ser obtido através de um processo de refinamento progressivo das características particulares dos pedidos de alarme ou dos alarmes existentes. A combinação das várias características que são identificadas configura a situação que a regra deve tratar.

Cada passo do refinamento é realizado por meio de uma operação que podemos chamar de "Seleção". Cada Seleção testa uma condição acerca dos pedidos ou das listas de alarme existentes e se esta condição é provada Verdadeira, pode produzir como resultado uma Sublista de alarmes que satisfazem à condição. Se a Seleção não pode ser satisfeita, é considerada Falsa e nenhuma Sublista é produzida; neste caso a Regra Externa da qual a Seleção faz parte é imediatamente considerada inválida.

As Sublistas são estruturas temporárias cujo tempo de vida corresponde ao tempo de execução da Regra Externa correspondente. As Sublistas que vão sendo formadas podem ser passadas de uma seleção para outra dentro da mesma regra, de modo a permitir a continuação do processo de refinamento. São exemplos de algumas seleções que podem ser úteis:

- qual o tipo do pedido que está sendo tratado;
- a qual tipo de alarme se refere o pedido;
- a qual estação ou equipamento se refere o pedido;
- obter um conjunto de alarmes (sublista) de um determinado tipo;
- obter uma sublista de alarmes que ocorreram em uma determinada estação ou equipamento.
- testar a severidade dos alarmes de uma sublista;
- testar o instante de ocorrência dos alarmes de uma sublista;

Somente se todas as Seleções componentes de uma Regra são provadas verdadeiras, a Regra correspondente é considerada Válida e neste caso cada uma das Ações especificadas é então executada.

As Ações executadas pelas Regras consideradas válidas efetuam modificações dinâmicas na Base de Conhecimento. Exemplos de ações que podem ser executadas pelas regras são:

- suprimir o pedido original;
- eliminar ou reconhecer todos os alarmes existentes com um determinado tipo, ou ainda alarmes individuais.
- habilitar ou inibir alarmes de um determinado tipo, ou alarmes individuais.
- determinar a anunciação de um alarme novo, correspondente à situação identificada;
- substituir vários alarmes existentes por um único;
- executar um procedimento ou ativar outro módulo de software;

- emitir uma mensagem de aviso para outros módulos de software;
- estabelecer índices que identificam o estado operativo do momento;

Caso o pedido de tratamento não seja suprimido por alguma Regra provada válida, ou então não se tenha configurado nenhuma situação prevista nas Regras, o pedido original é passado para o PBA e processado normalmente.

As Regras Externas são formatadas com o auxílio de um editor de textos comum e armazenadas em arquivo próprio, cujo lay-out é mostrado na figura V.5-1. Este arquivo possui duas divisões principais. Na primeira, identificada pela palavra reservada "clauses" do PROLOG, são simplesmente declarados os nomes de todas as regras cuja definição virá a seguir. Na segunda divisão são então registradas as regras no seu formato próprio.

Na definição das regras, os trechos do arquivo entre as palavras-chave "enunciado" e "formato" tem apenas o efeito de comentários (observar que esses trechos se colocam entre os delimitadores "/*" e "*/" indicativos de comentários no PROLOG), sendo no entanto utilizados pelo módulo Visual para a apresentação da definição das Regras Externas na sua forma de linguagem natural através do comando "Listar_Regra".

```
/*---- Lista de nomes de regras que serão definidas --------------------*/

clauses

    nome_de_regra(nome_1).
    nome_de_regra(nome_2).
          .
          .
          .
    nome_de_regra(nome_n).

/*---- Definições das Regras Externas ----------------------------------*/

  REGRA(nome_n)

/*
  enunciado:
      " ... enunciado da Regra nome_n, em linguagem natural ...
        ...                                                  ...
        ...                                                  ..."

  formato:
*/

  IF
       seleção_1,
       seleção_2,
           .
           .
           .
       seleção_j,

  THEN,
       ação_1,
       ação_2,
           .
           .
           .
       ação_k,

  END_IF.

/*-----------------------------------------------------------------------*/
```

fig. V.5-1 - Formato do arquivo de Regras Externas

É importante observar que as Regras Externas são, em última análise, escritas em PROLOG. No entanto, devido ao emprego dos construtos "Regras()", "IF", "THEN," e "END_IF", da ordem em que as cláusulas são organizadas e da existência de várias conclusões para uma mesma regra (várias ações), o aspecto externo difere bastante do utilizado pelo PROLOG, configurando assim uma meta-linguagem específica para este propósito.

As seleções e ações devem ser escolhidas a partir de um conjunto de Primitivas pré-programadas e que formam uma Biblioteca de Primitivas. O usuário não precisa conhecer como as Primitivas, que são regras escritas em PROLOG, são codificadas, mas apenas como combiná-las para obter o resultado final desejado. O emprego de uma meta-linguagem, com a sua forma própria de definição de regras, construtos e primitivas pré-definidas, permite isolar o usuário das complexidades do PROLOG. Detalhes como o emprego de estruturas de controle da inferência e a estrutura interna das definições de alarmes, ficam embutidas nas Primitivas e portanto transparentes para o usuário. Com isto o usuário não precisa ser um especialista em programação para efetuar a formação e a manutenção da Base de Conhecimento, nem conhecer detalhes de implementação do sistema.

O apêndice III apresenta as primitivas definidas e construídas para o sistema-exemplo, com uma descrição funcional de cada uma. As Regras Externas para o caso-exemplo são apresentadas no Apêndice IV.

V.6 CASO-EXEMPLO

Um caso-exemplo foi elaborado com o intuito de demonstrar a aplicação do sistema especialista desenvolvido. Este caso-exemplo é caracterizado por:

- um conjunto de definições de Alarmes, Listas de Alarme e Severidades, criadas por meio do Compilador de Alarmes. Essas definições são apresentadas no Apêndice I.
- um cenário de simulação composto por uma sequência de pedidos de tratamento de alarme a serem efetuados em instantes determinados. O Apêndice II apresenta o conteúdo do arquivo de cenário criado para o caso-exemplo.
- um conjunto de Primitivas, escritas em PROLOG e apresentadas resumidamente no Apêndice III.
- um conjunto de Regras, formadas segundo a metodologia proposta. O Apêndice IV apresenta a definição dessas Regras.

A figura V.6-1 apresenta um diagrama unifilar de um sistema elétrico hipotético sobre o qual o cenário de simulação se desenvolve. Este sistema elétrico é composto por 6 estações (A,B,C,D,E,F), um transformador (TR1), uma carga (CG1), um banco de reatores (RT1) e uma linha de interligação (INT1). Em cada estação é suposto haver dois barramentos (de nomes barra_A

e barra_B), não mostrados no diagrama unifilar. Os nomes das estações e equipamentos são utilizados para identificar univocamente os alarmes gerados, correspondendo aos campos "chave1" e "chave2" das mensagens de alarme.

Alguns eventos no sistema elétrico são simulados através dos alarmes que seriam sinalizados como consequência desses eventos. Com o arquivo de cenário definido, o programa pode ser processado e pode-se acompanhar a sequência de disparo dos pedidos de alarme, a ativação das Regras pertinentes e os efeitos da aplicação dessas Regras sobre os Pedidos de Alarme e sobre as Listas de Alarme. O processamento dos vários módulos componentes do sistema pode ser visualizado por meio das janelas de vídeo dedicadas.

A figura V.6-2 apresenta uma aspecto das janelas de vídeo do sistema logo após à inicialização do sistema, antes da execução da primeira cena programada. Como se pode observar a regra r3 está sendo apresentada na janela "Visual".

A seguir são descritos os eventos que compõem o cenário de simulação para o caso-exemplo e as suas consequências, na ordem de ocorrência no tempo.

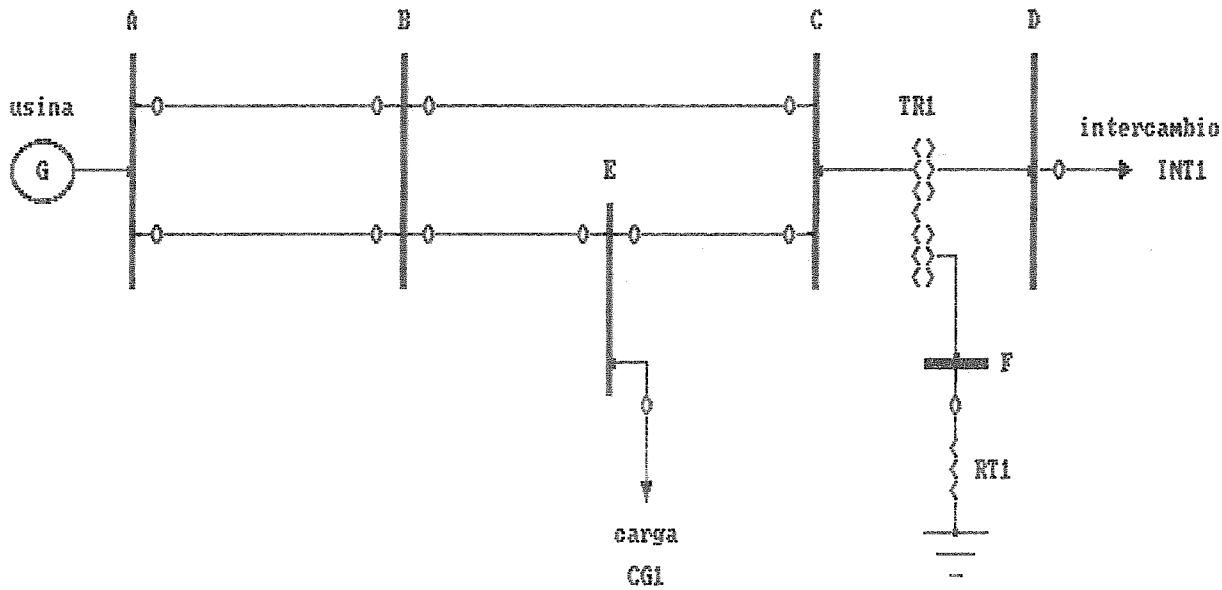

fig. V.6-1 - Diagrama unifilar do sistema elétrico para o caso-teste

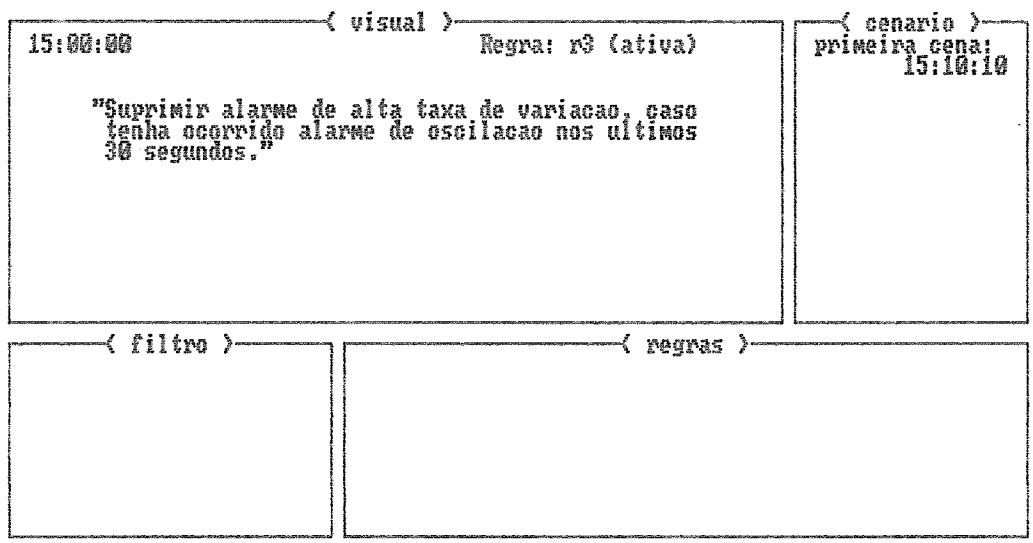

fig. V.6-2 - Aspecto das janelas de vídeo na inicialização do sistema

- evento E1: tensão na estação D desce a nível de alerta.

  São sinalizados dois alarmes às 15:15:10, um para cada barramento da estação. O primeiro alarme é anunciado normalmente mas o segundo é suprimido devido à aplicação da Regra r1 (figura V.6-3).

- evento E2: ocorre um defeito na linha de transmissão BE e esta é desligada automaticamente por atuação do sistema de proteção.

  Dois alarmes de desconexão, um para cada extremo da linha, são recebidos com um intervalo de 10 segundos. O primeiro, às 15:15:00, é anunciado normalmente (figura V.6-4) mas o segundo, às 15:15:10 é suprimido devido à aplicação da Regra r4. A mesma regra elimina então o alarme gerado 10 segundos antes e anuncia um alarme de "Linha Desenergizada" que resume com uma só mensagem a mesma condição sinalizada pela combinação dos dois alarmes de desconexão (figura V.6-5).

- evento E3: como consequência da abertura da linha BE, a tensão na estação D reduz-se ainda mais passando para nível de urgência.

  Dois alarmes indicativos de sub-tensão de urgência na estação D, um para cada barramento, são sinalizados às 15:17:20. O primeiro é anunciado normalmente, mas, como consequência da Regra r2, provoca a eliminação do alarme de

alerta de tensão que havia sido gerado às 15:10:10. O segundo alarme é suprimido devido à Regra r1 (figura V.6-6).

- evento E4: ainda como consequência da abertura da linha BE, o carregamento da linha de transmissão BC passa para nível de alerta.

  São sinalizados dois alarmes indicativos de sobrecorrente de alerta, com 10 segundos de diferença. O primeiro às 15:19:00 e proveniente da estação B é anunciado normalmente (figura V.6-7). O segundo, às 15:19:10 e proveniente da estação C é suprimido, como consequência da Regra r5 (figura V.6-8).

- evento E5: para corrigir a tensão na estação D o operador comanda remotamente a retirada do banco de reatores RT1 por meio da abertura do disjuntor disj_RT1.

  Como se trata de um evento iniciado pelo operador, com consequências previsíveis, o software que implementa o comando remoto de disjuntores sinaliza às 15:25:00 para o processador de alarmes um pedido de inserção de um alarme de abertura de disjuntor como sendo "esperado" para os próximos 2 minutos, que seria o tempo que o despachante teria para completar a sequência de comandos de abertura do disjuntor (figura V.6-9). O alarme de abertura do disjuntor

do reator é recebido às 15:26:00 e suprimido, por ter ocorrido dentro do intervalo previsto (figura V.6-10).

- evento E6: a tensão na estação D volta ao normal.

  Dois alarmes de normalização de tensão são recebidos às 15:26:20, um para cada barramento da estação D. O primeiro é anunciado normalmente e o segundo suprimido como consequência da aplicação da Regra r1 (figura V.6-11).

- evento E7: com a evolução natural da carga do sistema o carregamento da linha BC, que se encontrava em nível de alerta passa para nível de urgência.

  São sinalizados dois alarmes, um para cada extremidade da linha. O alarme das 15:26:50 é anunciado normalmente, mas provoca a eliminação do alarme de alerta anteriormente gerado para a mesma linha (15:19:00), conforme prescrito pela Regra r6 (figura V.6-12). O alarme das 15:27:00 é suprimido devido à aplicação da Regra r5 (figura V.6-13).

- evento E8: por atuação do sistema de proteção, a carga CG1 é desligada.

  Um alarme de abertura do disjuntor associado à carga é recebido e anunciado normalmente (figura V.6-14).

- evento E9: como consequência do desligamento da carga CG1, o carregamento da linha BC volta ao normal.

  Dois alarmes de normalização são recebidos às 15:30:30 e às 15:30:40. O primeiro, correspondente à extremidade da linha BC na estação B é anunciado normalmente (figura V.6-15). O segundo, correspondente à extremidade da linha na estação C é suprimido devido à Regra r5 (figura V.6-16).

- evento E10: ocorre uma oscilação de frequência no sistema.

  É sinalizado um alarme de oscilação de frequência, detetado na estação E, o qual é anunciado normalmente (figura V.6-17).

- evento E11: como consequência da oscilação de frequência ocorrida, algumas grandezas elétricas variam muito rapidamente.

  São recebidos 3 pedidos de alarme de "alta taxa de variação" de corrente em equipamentos. O primeiro, às 15:33:20, detetado no gerador ger_1 da estação est_A, é suprimido por estar dentro do período de 30 segundos prescrito pela Regra r3 (figura V.6-18). O segundo, às 15:33:30, detetado no transformador tr_1, estação est_C, é suprimido pela mesma razão (figura V.6-19). O último, às 15:33:42 é anunciado normalmente, por ocorrer fora do escopo da regra r3 (figura V.6-20).

```
( visual )
15:10:12                                  Lista: geral
15:10:10 tensao baixa ............... barra_A  estac_D  ALE

( cenario )
primeira cena:
        15:10:10
gerar_alarme
        15:10:10
gerar_alarme
----------------
segue: 15:15:00

( filtro )
gerar_alarme subtenal
  - alarme gerado -
gerar_alarme subtenal
  - alarme gerado -

( regras )
--- nenhuma regra aplicavel ---
-----------------------------------------------------
r1: suprime tensao severidade igual na estacao
-----------------------------------------------------
```

fig. V.6-3 - Situação após a execução das cenas das 15:10:10

```
( visual )
15:15:02                                  Lista: geral
15:10:10 tensao baixa ............... barra_A  estac_D  ALE
15:15:01 terminal desconectado ..... lt_BE    estac_B  ALE

( cenario )
primeira cena:
        15:10:10
gerar_alarme
        15:10:10
gerar_alarme
----------------
        15:15:00
gerar_alarme
----------------
segue: 15:15:10

( filtro )
gerar_alarme ltdescon
  - alarme gerado -

( regras )
--- nenhuma regra aplicavel ---
-----------------------------------------------------
```

fig. V.6-4 - Situação após a execução da cena das 15:15:00

```
( visual )
15:15:10                                  Lista: geral
15:10:10 tensao baixa ............... barra_A  estac_D  ALE
15:15:10 linha desenergizada ....... lt_BE               ALE

( cenario )
primeira cena:
        15:10:10
gerar_alarme
        15:10:10
gerar_alarme
----------------
        15:15:00
gerar_alarme
----------------
        15:15:10
gerar_alarme
----------------
segue: 15:17:20

( filtro )
gerar_alarme ltdescon
 - pedido suprimido -

( regras )
r4: Abertura lt vazio deriva desenergizacao
-----------------------------------------------------
```

fig. V.6-5 - Situação após a execução da cena das 15:15:10

```
( visual )                                                    ( cenario )
15:17:22                                Lista: geral          ----------------
15:15:10 linha desenergizada ...... lt_BE                ALE        15:15:00
15:17:21 tensao baixa ............. barra_A  estac_D  URG   gerar_alarme
                                                              ----------------
                                                                   15:15:10
                                                              gerar_alarme
                                                              ----------------
                                                                   15:17:20
                                                              gerar_alarme
                                                                   15:17:20
                                                              gerar_alarme
                                                              ----------------
                                                              segue  15:19:00

( filtro )                                  ( regras )
gerar_alarme subtenur                       r2: Substitui tensao severidade < na estacao
  - alarme gerado -                         ----------------------------------------------
gerar_alarme subtenur                       r1: Suprime tensao severidade = na estacao
  - pedido suprimido -                      ----------------------------------------------
```

fig. V.6-6 - Situação após a execução das cenas das 15:17:20

```
( visual )                                                    ( cenario )
15:19:02                                Lista: geral          ----------------
15:15:10 linha desenergizada ...... lt_BE                ALE        15:15:10
15:17:21 tensao baixa ............. barra_A  estac_D  URG   gerar_alarme
15:19:01 sobrecorrente ............ lt_BC    estac_B  ALE   ----------------
                                                                   15:17:20
                                                              gerar_alarme
                                                                   15:17:20
                                                              gerar_alarme
                                                              ----------------
                                                                   15:19:00
                                                              gerar_alarme
                                                              ----------------
                                                              segue: 15:19:10

( filtro )                                  ( regras )
gerar_alarme sobcoral                       --- nenhuma regra aplicavel ---
  - alarme gerado -                         ----------------------------------------------
```

fig. V.6-7 - Situação após a execução da cena das 15:19:00

```
( visual )                                                    ( cenario )
15:19:10                                Lista: geral          ----------------
15:15:10 linha desenergizada ...... lt_BE                ALE        15:17:20
15:17:21 tensao baixa ............. barra_A  estac_D  URG   gerar_alarme
15:19:01 sobrecorrente ............ lt_BC    estac_B  ALE        15:17:20
                                                              gerar_alarme
                                                              ----------------
                                                                   15:19:00
                                                              gerar_alarme
                                                              ----------------
                                                                   15:19:10
                                                              gerar_alarme
                                                              ----------------
                                                              segue: 15:25:00

( filtro )                                  ( regras )
gerar_alarme sobcoral                       r5: Suprime carregamento severid = mesma linha
  - pedido suprimido -                      ----------------------------------------------
```

fig. V.6-8 - Situação após a execução da cena das 15:19:10

```
( visual )                                                    ( cenario )
15:25:02                                Lista: geral          15:17:20
15:15:10 linha desenergizada ...... lt_BE              ALE    gerar_alarme
15:17:21 tensao baixa ............. barra_A  estac_D  URG    ----------------
15:19:01 sobrecorrente ............ lt_BC    estac_B  ALE    15:19:00
                                                              gerar_alarme
                                                              ----------------
                                                              15:19:10
                                                              gerar_alarme
                                                              ----------------
                                                              15:25:00
                                                              esperado
                                                              ----------------
                                                              segue: 15:26:00

( filtro )                                  ( regras )
esperado     disjabrt                       --- nenhuma regra aplicavel ---
 - alarme inserido -                        ----------------------------------------
```

fig. V.6-9 - Situação após a execução da cena das 15:25:00

```
( visual )                                                    ( cenario )
15:26:02                                Lista: geral          15:19:00
15:15:10 linha desenergizada ...... lt_BE              ALE    gerar_alarme
15:17:21 tensao baixa ............. barra_A  estac_D  URG    ----------------
15:19:01 sobrecorrente ............ lt_BC    estac_B  ALE    15:19:10
                                                              gerar_alarme
                                                              ----------------
                                                              15:25:00
                                                              esperado
                                                              ----------------
                                                              15:26:00
                                                              gerar_alarme
                                                              ----------------
                                                              segue: 15:26:20

( filtro )                                  ( regras )
gerar_alarme disjabrt                       --- nenhuma regra aplicavel ---
  * era esperado *                          ----------------------------------------
```

fig. V.6-10 - Situação após a execução da cena das 15:26:00

```
( visual )                                                    ( cenario )
15:26:22                                Lista: geral          ----------------
15:15:10 linha desenergizada ...... lt_BE              ALE    15:25:00
15:17:21 tensao baixa ............. barra_A  estac_D  URG    esperado
15:19:01 sobrecorrente ............ lt_BC    estac_B  ALE    ----------------
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE    15:26:00
                                                              gerar_alarme
                                                              ----------------
                                                              15:26:20
                                                              gerar_alarme
                                                              15:26:20
                                                              gerar_alarme
                                                              ----------------
                                                              segue: 15:26:50

( filtro )                                  ( regras )
gerar_alarme normtens                       --- nenhuma regra aplicavel ---
  - alarme gerado -                         ----------------------------------------
gerar_alarme normtens                       r1: Suprime tensao severidade = na estacao
 - pedido suprimido -                       ----------------------------------------
```

fig. V.6-11 - Situação após a execução das cenas das 15:26:20

```
( visual )                                                    ( cenario )
15:26:52                                Lista: geral          ----------
15:15:10 linha desenergizada ...... lt_BE                ALE        15:26:00
15:17:21 tensao baixa ............. barra_A  estac_D  URG   gerar_alarme
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE   ----------
15:26:51 sobrecorrente ............ lt_BC    estac_B  URG        15:26:20
                                                             gerar_alarme
                                                                 15:26:20
                                                             gerar_alarme
                                                             ----------
                                                                 15:26:50
                                                             gerar_alarme
                                                             ----------
                                                             segue: 15:27:00

( filtro )                                  ( regras )
gerar_alarme sobcorur                       r6: Substitui carregamento sever < na linha
  - alarme gerado -                         ----------------------------------------
```

fig. V.6-12 - Situação após a execução da cena das 15:26:50

```
( visual )                                                    ( cenario )
15:27:02                                Lista: geral          ----------
15:15:10 linha desenergizada ...... lt_BE                ALE        15:26:20
15:17:21 tensao baixa ............. barra_A  estac_D  URG   gerar_alarme
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE       15:26:20
15:26:51 sobrecorrente ............ lt_BC    estac_B  URG   gerar_alarme
                                                             ----------
                                                                 15:26:50
                                                             gerar_alarme
                                                             ----------
                                                                 15:27:00
                                                             gerar_alarme
                                                             ----------
                                                             segue: 15:30:00

( filtro )                                  ( regras )
gerar_alarme sobcorur                       r5: Suprime carregamento severid = na linha
 - pedido suprimido --                      ----------------------------------------
```

fig. V.6-13 - Situação após a execução da cena das 15:27:00

```
( visual )                                                    ( cenario )
15:30:02                                Lista: geral          15:26:20
15:15:10 linha desenergizada ...... lt_BE                ALE   gerar_alarme
15:17:21 tensao baixa ............. barra_A  estac_D  URG   ----------
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE       15:26:50
15:26:51 sobrecorrente ............ lt_BC    estac_B  URG   gerar_alarme
15:30:01 disjuntor aberto ......... disj_CG1 estac_E  ALE   ----------
                                                                 15:27:00
                                                             gerar_alarme
                                                             ----------
                                                                 15:30:00
                                                             gerar_alarme
                                                             ----------
                                                             segue: 15:30:30

( filtro )                                  ( regras )
gerar_alarme disjabrt                       --- nenhuma regra aplicavel ---
  - alarme gerado -                         ----------------------------------------
```

fig. V.6-14 - Situação após a execução da cena das 15:30:00

```
( visual )                                                    ( cenario )
15:30:32                          Lista: geral                    15:26:50
15:15:10 linha desenergizada ...... lt_BE                 ALE   gerar_alarme
15:17:21 tensao baixa ............. barra_A  estac_D  URG   ----------------
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE          15:27:00
15:26:51 sobrecorrente ............ lt_BC   estac_B  URG   gerar_alarme
15:30:01 disjuntor aberto ......... disj_CG1 estac_E  ALE   ----------------
15:30:31 corrente,retorno ao normal lt_BC    estac_B  ALE          15:30:00
                                                            gerar_alarme
                                                            ----------------
                                                                  15:30:30
                                                            gerar_alarme
                                                            ----------------
                                                            segue: 15:30:40

( filtro )                                  ( regras )
gerar_alarme normcorr                       --- nenhuma regra aplicavel ---
  - alarme gerado -                         ----------------------------------------
```

fig. V.6-15 - Situação após a execução da cena das 15:30:30

```
( visual )                                                    ( cenario )
15:30:40                          Lista: geral                    15:27:00
15:15:10 linha desenergizada ...... lt_BE                 ALE   gerar_alarme
15:17:21 tensao baixa ............. barra_A  estac_D  URG   ----------------
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE          15:30:00
15:26:51 sobrecorrente ............ lt_BC   estac_B  URG   gerar_alarme
15:30:01 disjuntor aberto ......... disj_CG1 estac_E  ALE   ----------------
15:30:31 corrente,retorno ao normal lt_BC    estac_B  ALE          15:30:30
                                                            gerar_alarme
                                                            ----------------
                                                                  15:30:40
                                                            gerar_alarme
                                                            ----------------
                                                            segue: 15:33:10

( filtro )                                  ( regras )
gerar_alarme normcorr                       r5: Suprime carregamento severid = na linha
  - pedido suprimido -                      ----------------------------------------
```

fig. V.6-16 - Situação após a execução da cena das 15:30:40

```
( visual )                                                    ( cenario )
15:33:12                          Lista: geral                    15:30:00
15:15:10 linha desenergizada ...... lt_BE                 ALE   gerar_alarme
15:17:21 tensao baixa ............. barra_A  estac_D  URG   ----------------
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE          15:30:30
15:26:51 sobrecorrente ............ lt_BC   estac_B  URG   gerar_alarme
15:30:01 disjuntor aberto ......... disj_CG1 estac_E  ALE   ----------------
15:30:31 corrente,retorno ao normal lt_BC    estac_B  ALE          15:30:40
15:33:11 oscilacao de tensao ......          estac_E  URG   gerar_alarme
                                                            ----------------
                                                                  15:33:10
                                                            gerar_alarme
                                                            ----------------
                                                            segue: 15:33:20

( filtro )                                  ( regras )
gerar_alarme oscilac                        --- nenhuma regra aplicavel ---
  - alarme gerado -                         ----------------------------------------
```

fig. V.6-17 - Situação após a execução da cena das 15:33:10

```
( visual )                                                    ( cenario )
15:33:22                          Lista: geral                  15:30:30
15:15:10 linha desenergizada ...... lt_BE              ALE    gerar_alarme
15:17:21 tensao baixa ............. barra_A  estac_D  URG    ---------------
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE          15:30:40
15:26:51 sobrecorrente ............ lt_BC   estac_B  URG    gerar_alarme
15:30:01 disjuntor aberto ......... disj_CG1 estac_E  ALE    ---------------
15:30:31 corrente,retorno ao normal lt_BC    estac_B  ALE          15:33:10
15:33:11 oscilacao de tensao ......          estac_E  URG    gerar_alarme
                                                             ---------------
                                                                   15:33:20
                                                             gerar_alarme
                                                             ---------------
                                                             segue: 15:33:30

( filtro )                                  ( regras )
gerar_alarme altaxcor                       r3: Suprime taxa de variacao apos oscilacao
 - pedido suprimido -                       ---------------------------------------------
```

fig. V.6-18 - Situação após a execução da cena das 15:33:20

```
( visual )                                                    ( cenario )
15:33:30                          Lista: geral                  15:30:40
15:15:10 linha desenergizada ...... lt_BE              ALE    gerar_alarme
15:17:21 tensao baixa ............. barra_A  estac_D  URG    ---------------
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE          15:33:10
15:26:51 sobrecorrente ............ lt_BC   estac_B  URG    gerar_alarme
15:30:01 disjuntor aberto ......... disj_CG1 estac_E  ALE    ---------------
15:30:31 corrente,retorno ao normal lt_BC    estac_B  ALE          15:33:20
15:33:11 oscilacao de tensao ......          estac_E  URG    gerar_alarme
                                                             ---------------
                                                                   15:33:30
                                                             gerar_alarme
                                                             ---------------
                                                             segue: 15:33:42

( filtro )                                  ( regras )
gerar_alarme altaxcor                       r3: Suprime taxa de variacao apos oscilacao
 - pedido suprimido -                       ---------------------------------------------
```

fig. V.6-19 - Situação após a execução da cena das 15:33:20

```
( visual )                                                    ( cenario )
15:33:43                          Lista: geral                  15:33:10
15:15:10 linha desenergizada ...... lt_BE              ALE    gerar_alarme
15:17:21 tensao baixa ............. barra_A  estac_D  URG    ---------------
15:26:21 tensao - retorno ao normal barra_A  estac_D  ALE          15:33:20
15:26:51 sobrecorrente ............ lt_BC   estac_B  URG    gerar_alarme
15:30:01 disjuntor aberto ......... disj_CG1 estac_E  ALE    ---------------
15:30:31 corrente,retorno ao normal lt_BC    estac_B  ALE          15:33:30
15:33:11 oscilacao de tensao ......          estac_E  URG    gerar_alarme
15:33:43 alta taxa variac corrente  int_1    estac_D  URG    ---------------
                                                                   15:33:42
                                                             gerar_alarme
                                                              Fim de Cenario

( filtro )                                  ( regras )
gerar_alarme altaxcor                       --- nenhuma regra aplicavel ---
 - alarme gerado -                          ---------------------------------------------
```

fig. V.6-20 - Situação após a execução da cena das 15:33:42

VI - CONCLUSÃO

Este trabalho analisou o processamento de alarmes em centros de controle na sua integração com a rotina de operação, identificando as características desejáveis que um sistema de alarmes deve possuir de modo a representar um auxílio efetivo no processo operativo. Tais características são de difícil implementação com o emprego das técnicas convencionais de programação e talvez por essa razão os sistemas de alarme não têm acompanhado a evolução dos centros de controle nas últimas décadas.

Os Sistemas Especialistas formam um ramo da pesquisa na área de Inteligência Artificial e apresentam aspectos que a priori são bastante adequados para solucionar o problema dos alarmes. Este trabalho apontou esta possibilidade e apresentou um sistema especialista especialmente desenvolvido para demonstrar a viabilidade de construção de um sistema flexível o bastante para acompanhar as variações nas necessidades operativas do despachante.

Este sistema-exemplo, construído na forma de um simulador "off-line", com a utilização da linguagem PROLOG, apresenta como ponto de destaque o emprego de Regras do tipo condição-ação para a definição da lógica do processamento de alarmes, regras estas definidas e construídas pelo próprio usuário do sistema. A sintaxe proposta para as regras permite remover os detalhes das estruturas da linguagem PROLOG,

constituindo uma meta-linguagem que permite a definição de regras bastante legíveis, facilitando a sua construção por não-programadores. Por outro lado, como as Regras são formadas pela simples combinação de Primitivas pré-programadas armazenadas em uma biblioteca, toda a complexidade inerente ao tratamento computacional do problema permanece embutida nas Primitivas escolhidas para formar as Regras. Ao contrário das Regras, as Primitivas exigem pessoal especializado em programação para eventuais alterações ou expansões na biblioteca de primitivas.

O sistema-exemplo mostrou ser viável o objetivo pretendido, principalmente do ponto de vista da expressividade e legibilidade alcançadas pelas Regras, conforme ficou demonstrado no caso-exemplo apresentado.

Durante o desenvolvimento do sistema-exemplo observou-se que a linguagem PROLOG cria um ambiente propício para o desenvolvimento incremental de um sistema, pela característica modular da adição de conhecimento ao sistema. Do ponto de vista da adequação da linguagem ao tipo de problema tratado, observou-se que para a parte central do sistema, que realiza o tratamento dos alarmes, a linguagem mostrou toda a sua expressividade e relativa simplicidade de construção da lógica do programa, embora a legibilidade final e a consequente facilidade de manutenção tenham deixado a desejar. Para os demais módulos, que realizam a simulação das interfaces que seriam encontradas no centro de controle e são portanto acessórios em relação ao problema tratado, o programa escrito em PROLOG revelou-se de construção bem mais complexa, depuração

mais difícil e menos legível se comparado ao uso de linguagens convencionais.

Para uma aplicação prática em um centro de controle, o núcleo do sistema desenvolvido poderia ser tomado como base. Naturalmente as definições de alarme deverão ser elaboradas com maior detalhe, as interfaces com os outros sistemas terão que ser redefinidas e deverá ser desenvolvido um conjunto mais completo de Regras e Primitivas.

O emprego das técnicas apresentadas neste trabalho na construção de um sistema para processamento de alarmes em um centro de controle fornecerá condições para que se obtenha um sistema adaptável ao ambiente operativo, flexível o bastante para ser atualizado pelo próprio pessoal de operação e que possa acompanhar a evolução dos requisitos do centro de controle.

APÊNDICE I - DEFINIÇÕES DE ALARME PARA O CASO-EXEMPLO

Id:sobtenal
Texto:tensao alta .....
Severidade:ALE
Listas:(geral,transmissao,alerta)
Tipo:tensao

Id:sobtenur
Texto:tensao alta .....
Severidade:URG
Listas:(geral,transmissao,urgente)
Tipo:tensao

Id:subtenal
Texto:tensao baixa .....
Severidade:ALE
Listas:(geral,transmissao,alerta)
Tipo:tensao

Id:subtenur
Texto:tensao baixa .....
Severidade:URG
Listas:(geral,transmissao,urgente)
Tipo:tensao

Id:sobcoral
Texto:sobrecorrente .....
Severidade:ALE
Listas:(geral,transmissao,alerta)
Tipo:carregamento

Id:sobcorur
Texto:sobrecorrente .....
Severidade:URG
Listas:(geral,transmissao,urgente)
Tipo:carregamento

Id:normtens
Texto:tensao - retorno ao normal
Severidade:ALE
Listas:(geral,transmissao,normal,alerta)
Tipo:tensao

Id:normcorr
Texto:corrente,retorno ao normal
Severidade:ALE
Listas:(geral,transmissao,normal,alerta)
Tipo:carregamento

Id:ltdescon
Texto:terminal desconectado ....
Severidade:ALE
Listas:(geral,transmissao,alerta)
Tipo:terminal_aberto

Id:ltdesen
Texto:linha desenergizada .....
Severidade:ALE
Listas:(geral,transmissao,alerta)
Tipo:desenergizacao

Id:disjabrt
Texto:disjuntor aberto .....
Severidade:ALE
Listas:(geral,alerta)
Tipo:abertura_de_disjuntor

Id:disjfech
Texto:disjuntor fechado .....
Severidade:ALE
Listas:(geral,alerta)
Tipo:fechamento_de_disjuntor

Id:oscilac
Texto:oscilacao de tensao .....
Severidade:URG
Listas:(geral,transmissao,urgente)
Tipo:oscilacao

Id:altaxcor
Texto:alta taxa variac corrente
Severidade:URG
Listas:(geral,transmissao,urgente)
Tipo:taxa_de_variacao

Id:remfora
Texto:remota fora de servico ...
Severidade:INF
Listas:(geral)
Tipo:comunicacoes

Id:remserv
Texto:remota em servico .....
Severidade:INF
Listas:(geral)
Tipo:comunicacoes

APÊNDICE II - CENARIO DE SIMULAÇÃO PARA O CASO-EXEMPLO

```
|--hora--|---pedido---|---ident----|-equip--|--est---|--inic--|---fim--|
 15:00:00
;
; subtensao na estacao D
;
 15:10:10 gerar_alarme subtenal      barra_A  estac_D
 15:10:10 gerar_alarme subtenal      barra_B  estac_D
;
; abertura da linha BE
;
 15:15:00 gerar_alarme ltdescon      lt_BE    estac_B
 15:15:10 gerar_alarme ltdescon      lt_BE    estac_E
 15:17:20 gerar_alarme subtenur      barra_A  estac_D
 15:17:20 gerar_alarme subtenur      barra_B  estac_D
 15:19:00 gerar_alarme sobcoral      lt_BC    estac_B
 15:19:10 gerar_alarme sobcoral      lt_BC    estac_C
;
; retirada do banco de reatores RT1
;
 15:25:00 esperado     disjabrt      disj_RT1 estac_F  15:25:00 15:27:00
 15:26:00 gerar_alarme disjabrt      disj_RT1 estac_F
 15:26:20 gerar_alarme normtens      barra_A  estac_D
 15:26:20 gerar_alarme normtens      barra_B  estac_D
;
; variacao natural da carga
;
 15:26:50 gerar_alarme sobcorur      lt_BC    estac_B
 15:27:00 gerar_alarme sobcorur      lt_BC    estac_C
;
; desligamento da carga CG1
;
 15:30:00 gerar_alarme disjabrt      disj_CG1 estac_E
 15:30:30 gerar_alarme normcorr      lt_BC    estac_B
 15:30:40 gerar_alarme normcorr      lt_BC    estac_C
;
; oscilacao de tensao no sistema
;
 15:33:10 gerar_alarme oscilac                estac_E
 15:33:20 gerar_alarme altaxcor      ger_1    estac_A
 15:33:30 gerar_alarme altaxcor      tr_1     estac_C
 15:33:42 gerar_alarme altaxcor      int_1    estac_D
```

APÊNDICE III - PRIMITIVAS CONSTRUÍDAS PARA O CASO-EXEMPLO

Primitiva: O_Pedido_e_Para(Tipo_do_Pedido)

Função: Recebe como entrada um "Tipo_do_Pedido" e compara com o tipo do pedido corrente, retornando o valor "True" caso sejam iguais.

Primitiva: Alarme_Pedido_e_do_Tipo(Tipo_do_Alarme)

Função: Recebe como entrada um "Tipo_do_Alarme" e compara com o tipo do alarme pedido corrente, retornando o valor "True" caso sejam iguais.

Primitiva: Existem_Alarmes_do_Tipo(Tipo_do_Alarme,Subl_Out)

Função: Retorna na sublista Subl_Out todos os alarmes existentes com o tipo "Tipo_do_Alarme", caso exista algum. Caso contrário retorna o valor "False".

Primitiva: Estacao_do_Pedido(Estacao)

Função: Retorna o nome da estação do sistema elétrico onde se localiza o alarme que está sendo pedido.

Primitiva: Equipamento_do_Pedido(Equipamento)

Função: Retorna o nome do equipamento do sistema elétrico onde se localiza o alarme que está sendo pedido.

Primitiva: Existem_na_Estacao(Estacao,Subl_In,Subl_Out)

Função: Retorna na sublista Subl_Out todos os alarmes recebidos na sublista Subl_In que se localizem na estação "Estação" passada como parâmetro. Caso nenhum alarme satisfaça à condição, retorna o valor "False".

Primitiva: Existem_no_Equipamento(Equipamento,Subl_In,Subl_Out)

Função: Retorna na sublista Subl_Out todos os alarmes recebidos na sublista Subl_In que se localizem no equipamento "Equipamento" passado como parâmetro. Caso nenhum alarme satisfaça à condição, retorna o valor "False".

Primitiva: Comparando_severidade_pedido(Operador,Sb_In,Sb_Out)

Função: Compara a severidade do alarme pedido com a de cada alarme da sublista Subl_In de acordo com o Operador passado como parâmetro. Todos os alarmes que satisfaçam a comparação são retornados na sublista Subl_Out. Os operadores válidos são: <, <=, =, >, >= e <>.

Primitiva: Tempo_de_Existencia(Oper,TmpSeg,Subl_In,Subl_Out)

Função: Compara o tempo de existência de cada alarme da sublista Subl_In de acordo com o operador (Oper) e o tempo em segundos (TmpSeg) passados como parâmetros. Todos os alarmes que satisfaçam a comparação são retornados na sublista Subl_Out. Os operadores válidos são: <, <=, =, > e >=.

Primitiva: Suprimir_Pedido

Função: Anula o pedido corrente de alarme.

Primitiva: Eliminar_os_Alarmes(Sublista)

Função: Causa a eliminação de todos os alarmes pertencentes à sublista passada como parâmetro.

Primitiva: Inibir_os_Alarmes(Sublista)

Função: Causa a eliminação de todos os alarmes pertencentes à sublista passada como parâmetro.

Primitiva: Gerar_o_Alarme(Id,Key1,Key2)

Função: Efetua a anunciação e inclusão do alarme cujos atributos são passados como parâmetros em todas as listas de alarme pertinentes.

Primitiva: Reporte(String)

Função: Escreve o "string" de caracteres passado como parâmetro na janela "Regras". Utilizada normalmente para informar uma narrativa de uma Regra provada válida.

APÊNDICE IV - REGRAS DEFINIDAS PARA O CASO-EXEMPLO

```
/*_____________________________________________________________*/

clauses

    nome_de_regra(r1).
    nome_de_regra(r2).
    nome_de_regra(r3).
    nome_de_regra(r4).
    nome_de_regra(r5).
    nome_de_regra(r6).


/*-------------------------------------------------------------*/

  REGRA(r1)

/*...

  enunciado:

      "Suprimir alarme de tensao, caso ja' exista
       outro alarme de tensao na mesma estacao,
       com mesma severidade".

  formato:

...*/

  IF
       O_Pedido_e_Para(gerar),

       Alarme_Pedido_e_do_Tipo(tensao),

       Estacao_do_Pedido(Estacao),

       Existem_Alarmes_do_Tipo(tensao,"Alarmes_similares"),

       Existem_na_Estacao(Estacao,"Alarmes_similares",
                                "Similares_da_estacao"),

       Comparando_severidade_pedido("=","Similares_da_estacao",
                                        "Resultam_os_seguintes"),

  THEN,
       Reporte("r1: Suprime tensao severidade = na estacao"),

       Suprimir_Pedido,

  END_IF.
```

```
/*-------------------------------------------------------------*/

  REGRA(r2)

/*...

  enunciado:

      "Substituir pelo alarme pedido os alarmes de
       tensao de severidade menor ocorridos na
       mesma estacao."

  formato:

...*/

  IF
       O_Pedido_e_Para(gerar),

       Alarme_Pedido_e_do_Tipo(tensao),

       Estacao_do_Pedido(Estacao),

       Existem_Alarmes_do_Tipo(tensao,"Alarmes_similares"),

       Existem_na_Estacao(Estacao,"Alarmes_similares",
                                "Similares_da_estacao"),

       Comparando_severidade_pedido(">","Similares_da_estacao",
                                       "Resultam_os_seguintes"),

  THEN,

       Reporte("r2: Substitui tensao severidade < na estacao"),

       Eliminar_os_Alarmes("Resultam_os_seguintes"),

  END_IF.
```

```
/*-----------------------------------------------------------------*/

  REGRA(r3)

/*...

  enunciado:

      "Suprimir alarme de alta taxa de variacao, caso
       tenha ocorrido alarme de oscilacao nos ultimos
       30 segundos.

  formato:

...*/

  IF
       O_Pedido_e_Para(gerar),

       Alarme_Pedido_e_do_Tipo(taxa_de_variacao),

       Existem_Alarmes_do_Tipo(oscilacao,"Todas_Oscilacoes"),

       Tempo_de_Existencia("<=",30,"Todas_Oscilacoes",
                                "Oscilacoes_Recentes"),

  THEN,
       Reporte("r3: Suprime taxa de variacao apos oscilacao"),

       Suprimir_Pedido,

  END_IF.
```

```
/*-----------------------------------------------------------------*/

  REGRA(r4)

/*...

  enunciado:

      "Derivar alarme de linha desenergizada caso
       ocorra alarme de desconexao para linha que
       ja' se encontra em vazio.  O alarme pedido
       deve ser suprimido e o existente eliminado"

  formato:

...*/

  IF
       O_Pedido_e_Para(gerar),

       Alarme_Pedido_e_do_Tipo(terminal_aberto),

       Equipamento_do_Pedido(Linha),

       Existem_Alarmes_do_Tipo(terminal_aberto,
                               "Linhas_em_vazio"),

       Existem_no_Equipamento(Linha,"Linhas_em_vazio",
                                    "Mesma_Linha"),

  THEN,

       Reporte("r4: Abert lt vazio: deriva desenergizacao"),

       Gerar_o_Alarme(ltdesen,Linha,""),

       Eliminar_os_Alarmes("Mesma_Linha"),

       Suprimir_Pedido,

  END_IF.
```

```
/*-----------------------------------------------------------------*/

  REGRA(r5)

/*...

  enunciado:

      "Suprimir alarme de carregamento caso ja' exista
       outro alarme de carregamento para a mesma linha
       com mesma severidade."

  formato:

...*/

  IF
       O_Pedido_e_Para(gerar),

       Alarme_Pedido_e_do_Tipo(carregamento),

       Equipamento_do_Pedido(Linha),

       Existem_Alarmes_do_Tipo(carregamento,
                               "Alarmes_similares"),

       Existem_no_Equipamento(Linha,"Alarmes_similares",
                                   "Similares_da_Linha"),

       Comparando_severidade_pedido("=","Similares_da_Linha",
                                    "Resultam_os_seguintes"),

  THEN,

      Reporte("r5: Suprime carregamento severid = na linha"),

      Suprimir_Pedido,

  END_IF.
```

```
/*-----------------------------------------------------------------*/

  REGRA(r6)

/*...

  enunciado:

      "Substituir pelo alarme pedido os alarmes de
       carregamento ocorridos na mesma linha do
       pedido e com severidade menor."

  formato:

...*/

  IF
       O_Pedido_e_Para(gerar),

       Alarme_Pedido_e_do_Tipo(carregamento),

       Equipamento_do_Pedido(Linha),

       Existem_Alarmes_do_Tipo(carregamento,
                               "Alarmes_similares"),

       Existem_no_Equipamento(Linha,"Alarmes_similares",
                                  "Similares_da_Linha"),

       Comparando_severidade_pedido(">","Similares_da_Linha",
                                    "Resultam_os_seguintes"),

  THEN,

      Reporte("r6: Substitui carregamento sever < na linha"),

      Eliminar_os_Alarmes("Resultam_os_seguintes"),

  END_IF.

/*_________________________________________________________________*/
```

REFERÊNCIAS

1- Russel,J.C., Masiello,R.D., Bose,A. - "Power System Control Center Concepts" - IEEE 1979 Power Industry Computer Applications Conference.

2- Cegrell,T., Dahlfors,F. - "125 Computerized Power System Control Centres - an Experience Base for Future Concepts" - International Conference on Large High Voltage Electric Systems - 1984 Session - CIGRE.

3- Talukdar,S.N., Cardozo,E. - " Artificial Intelligence Technologies for Power System Operations", Report EL_4323, Electric Power Research Institute, California, Jan/1986.

4- Kenealy,T.P. - "Man-machine Interface Subsystem" - IEEE Tutorial Course - Energy Control Center Design, pp 72-77, 1977.

5- Schaffer,G. - "User oriented Power System Control" - IFAC Symposium on Planning and Operation of Electric Energy Systems - Beijing, China, 1986.

6- Fink,L.H., Carlsen,K. - "Operating under Stress and Strain" - IEEE Spectrum, mar/1978, pp 48-53.

7- Eletrocon Intl, Inc. - "Considerations in Developing and Utilizing System Operator Training Simulators", Draft Final Report RP1915-1, Electric Power Research Institute, California.

8- Dy Liacco,T.E. - "The Adaptive Reliability Control System" - IEEE Trans. on PAS vol.PAS-86, may/1967.

9- Hanson,K. - "Some Aspects of Computer Loading Problems in Modern Control Centres" - CIGRE Study Committee 39 Meeting in Tokyo, Japan, oct/1987.

10- Schulz,R.P., Price,W.W. - "Classification and Identification of Power System Emergencies" - IEEE/PES 1984 Winter Meeting, Dallas, Texas, Jan/1984.

11- Schaffer,G., Davis,M., Sandmayr,H. - "Supervisory Network Control Systems Reflect User Requirements" - Brown-Bovery Review, vol.71,sep/oct/1984.

12- Denzel,D., Mantynen,R. - "Some Aspects of Data Acquisition for Alarm Display and Recording of Disturbances - part 1" - International Conference on Large High Voltage Electric Systems - 1984 Session - CIGRE.

13- Wollenberg,B. - "Feasibility Study for an Energy Management System Intelligent Alarm Processor" - IEEE Trans. on Power Systems, vol.PWRS-1,no.2,may/1986, pp 241-247.

14- Consórcio HIDROSERVICE-SCI - "Análise das Funções de Operação e Controle de FURNAS" - Eletrobrás, Jan/1977.

15- EPRI - Human Factors Review of Electric Power Dispatch Control Centers", Report EL-1960, Project 1354-1, Electric Power Research Institute, California, 1981.

16- Winter,W.H. - "Alarm Handling in Control Centers" - CIGRE SC39 Workshop on Alarm Handling in Control Centers, sep/1986.

17- Amelink,H., Forte,A.M.,Goberman,R.P. - "Dispatcher Alarm and Message Processing" - IEEE Trans. on Power Systems, vol.PWRS-1, no.3, aug/1986,pp 188-194.

18- Conradie,F. - "Alarm Handling - the ESCOM View and Experience" - CIGRE SC39 Workshop on Alarm Handling, sep/1986.

19- EPRI - "Advanced CRT Concepts for Power System Dispatch Centers", Report EL-81-5-LD, Electric Power Research Institute, California, apr/1981.

20- Bharath, R. - "An Introduction to PROLOG" - Tab Books Inc., Blue Ridge, 1986.

21- Jackson,P. - "Introduction to Expert Systems" - Addison-Wesley, 1986.

22- Corlett,R.A. - "Artificial Intelligence Languages and their Environments" - GEC Journal of Research, vol.5, no.1, 1987, pp 21-30.

23- Buchanan,B.G., Barstow,D., Bechtal,R., Bennett,J., Clancey,W., Kulikowsky,C., Mitchell,T., Waterman,A. - "Constructing an Expert System" - in "Building Expert Systems", edited by Hayes-Roth,F., Waterman,D.A., Lenat,D.B. , pp 127-167, Addison-Wesley, 1983.

24- Hayes-Roth,F. - "The Knowledge-Based Expert System: a Tutorial" - IEEE Computer, sep/1984, pp 11-14.

25- Williamson,M. - "Artificial Intelligence for Micro-computers" - Brady Communications, New York, 1985.

26- Duda,R., Gasching,J.G. - "Knowledge-Based Expert Systems Coming of Age" - Byte, vol.6, no.9, sep/1981.

27- Hayes-Roth,F. - "Rule-Based Systems" - Communications of the ACM - vol.28, no.9, sep/1985, pp 921-932.

28- Levine,R.I., Drang,D.E., Edelson,B. - "A Comprehensive guide to AI and Expert Systems" - McGraw-Hill, 1986.

29- Hwa,L.K. - "Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica Usando Técnicas de Sistemas Baseados em Conhecimentos" - tese de mestrado - COPPE-UFRJ, jul/1987.

30- EPRI - "Workshop on Advanced Concepts in System Operations", Proceedings, EL-81-11-LD, Electric Power Research Institute, California, jul/1981.

31- Montravel,G. - "A real time Expert System for Alarm Handling in the Future EDF Regional Control Centres" - CIGRE SC39 Workshop on Alarm Handling in Control Centers, sep/1986.

32- Hein,F. - "Expert System diagnosing Network Incidents - Design and Experiencies" - CIGRE SC39 Workshop on Alarm Handling in Control Centers, sep/1986.

33- Komai,K., Sakaguchi,T. - "Application of Artificial Intelligence Method to Power System Fault Diagnosis" - Study Committee 39 Meeting in Tokyo,Japan - oct/1987.

34- Fukui,C., Kawakami,J. - "An Expert System for Fault Section Estimation using Information from Protective Relays and Circuit Breakers" - IEEE Trans. on Power Delivery, vol.PWRD-1, no.4, oct/1986.

35- Sakaguchi,T., Matsumoto,K. - "Development of a Knowledge Based System for Power System Restoration" - IEEE Trans. on PAS, vol.102, no.2, Feb/1983, pp 320-329.

36- Liu,C.C., Tomsovic,K., - "An Expert System Assisting Decision-Making of Reactive Power/Voltage Control" - IEEE Trans. on Power Systems, vol PWRS-1, no.3, aug/1986, pp 195-201.

37- Talukdar,S.N., Cardozo, E., Perry,T. - "The Operator's Assistant - An Intelligent, Expandable Program for Power System Trouble Analysis" - IEEE Trans. on Power Systems, vol.PWRS-1, no.3, aug/1986, pp 182-187.

38- Cardozo,E., Talukdar,S.N. - "A Distributed Expert System for Fault Diagnosis", IEEE Power Industry Computer Applications Conference, Montreal, Canada, 1987.

39- Arruda,D.M. - "Sistemas Especialistas - Idéias Básicas" - COPPE/UFRJ, PDD-02/86, 1986.

40- Borland Intl. Inc. - "Turbo-PROLOG Owner's Handbook", California, 1986.

41- Townsend,C. - "Introduction to PROLOG" - Sybex, Berkeley, 1987.

42- Clocksin,W.F., Mellish,C.S. - "Programming in PROLOG" - Springer-Verlag, New York, 1981.

43- Rogers,J.B. - "A PROLOG Primer" - Addison-Wesley, 1986.