LIANE MARGARIDA ROCKENBACH TAROUCO

Lic. em Física, Universidade Federal do Rio Grande do Sul, 1970

INTELIGÊNCIA ARTIFICIAL
APLICADA AO
GERENCIAMENTO DE REDES DE COMPUTADORES

Tese apresentada à
Escola Politécnica da USP
para obtenção do título
de Doutor em Engenharia

Orientador: Prof. Dr. Lucas Antonio Moscato, Depto. Eng. Mecânica

São Paulo, 1990

A meu marido e filhos:

Luiz Carlos, Larson e

Lauren.

AGRADECIMENTOS

Agradeço,

Ao Prof. Lucas A. Moscato pela orientação e pelo incentivo desde o início do curso, sem cujo apoio não teria sido possível a sua realização.

A Profa. Stefânia Stiubiener pelo apoio, amizade e pelo conhecimento que transmitiu durante o curso.

A Universidade Federal do Rio Grande do Sul onde pude encontrar apoio e suporte para a realização do trabalho ora concluído.

A Dra. Elizabeth Feinler, Diretora do Network Information Center do SRI-Stanford Research Institute em Menlo Park, California, pelo apoio oferecido durante a realização de estágio naquele centro de pesquisa, onde foi desenvolvida uma parte importante deste trabalho, relativa à Inteligência Artificial e sua aplicação.

A meu marido Luiz Carlos e a meus filhos Larson e Lauren que, com afeto e carinho, me apoiaram e incentivaram a prosseguir, especialmente nos momentos em que os acidentes da vida me fizeram pensar em parar e sentar à margem do caminho.

A minha mãe pela lição de serenidade que transmite com seu exemplo de vida.

A Jeane de Lucas pelo auxílio na digitação deste trabalho.

RESUMO

O aumento em quantidade e diversidade dos equipamentos integrantes de uma rede de computadores atual tende a ocasionar maiores dificuldades na determinação de problemas, bem como na busca da solução. Diferentes equipamentos, fornecidos por diferentes fabricantes têm maneiras diferentes de sinalizar a ocorrência de eventos derivados da ocorrência de problemas.

Uma solução com grau de aceitação crescente para o problema de heterogeneidade dos integrantes de uma rede de computadores implica usar o modelo de referência OSI da ISO como paradigma de organização da rede. Contudo, esta nova forma de organização requer mecanismos de gerência adequados que estão ainda sendo definidos pelos órgãos de padronização. Em consequência, os responsáveis pela operação da rede devem receber um volume de informações sobre a operação da rede em quantidade e variedade tal que torna difícil a análise para derivar conclusões sobre os problemas eventualmente ocorrendo na rede. Adicionalmente, a forma destas informações pode sofrer alterações com o passar do tempo, na medida em que a padronização em desenvolvimento evoluir.

Este trabalho apresenta a caracterização do problema da gerência de redes de computadores, alguns resultados parciais para sua solução e proposta padrão em elaboração pela ISO. Com base na análise realizada, é proposta uma solução que utiliza inteligência artificial no projeto de um sistema de apoio ao operador de uma rede OSI analisando os eventos sinalizados e provendo recomendações para tratamento dos problemas.

ABSTRACT

The nowadays increase in number and variety of equipments integrating computer networks causes more difficulties on problem determination and search for solutions. Dissimilar equipment, provided by dissimilar manufacturers has dissimilar ways to signalize event occurrence derived from problems.

A solution which has increasingly been accepted to the problem of computer networks components heterogeneity imply on the use of ISO OSI reference model as network organization paradigm. However, this new organization form requires adequate management mechanisms that are still being defined by standardization bodies. Consequently, network operators and managers are expected to receive information concerning network operation in amount and variety that make the analysis to derive conclusions on problems eventually happening in the network difficult. Additionally, the information formats and meaning may change in time as the developing standardization progresses.

The work being presented shows the problem of computer network management characterization, some partial results for its solution, the standard proposal being developed by ISO. Based on the analysis realized it is proposed a solution using artificial intelligence on the design of a system to support an ISO network operator on analyzing signalized events and providing recommendations for its handle.

SIGLAS E ABREVIAÇÕES

ACE - Automated Cable Expertise

ACF/TCAM - Advanced Communication Facility/ Telecommunications Access Method

ACF/VCM - Advanced Communication Facility/ Virtual Telecommunications Access Method

ACSE - Association Control Service Element

AE - Application Entity

AI - Artificial Inteligence

ASN.1 - Abstract Syntax Notation One

CASE - Common Application Service Element

CCITT - Comite Consultivo Internacional de Telefonia e Telegrafia

CMIP - Common Management Information Protocol

CMISE - Common Management Application Service Element

CPqD - Centro de Pesquisas e Desenvolvimento

CSC - Centro de Supervisão e Controle

DCA - Digital Communications Associates

DEC - Digital Equipment Corporation

DR-TPDU - Disconnect Request Transport Protocol Data Unit

DT-TPDU - Data Transport Protocol Data Unit

RDSI - Redes Digitais de Servicos Integrada

KBES - Knowledge-Based Expert System

OSI - Open Systems Interconnection

ISO - International Organizations for Standard

ETD - Equipamento de Transmissão de Dados

LE - Layer Entity

LED - Light Emissor Diode

LISP - List Processing

LM - Layer Manager

LME - Layer Management Entity

LU - Logical Unit

M-TRACE-PDU - Management Trace Protocol Data Unit

MIB - Management Information Base

MISE - Management Information Service Element

MAPDU - Management Application Protocol Data Unit

n-SAP- Service Access Point de um nível n

NAU - Network Addressable Units

NCC - Network Control Center

NCCF - Network Communications Control Facility

Netview/PC - Netview /Personal Computer

NLDM - Network Logical Data Manager

NCMS - Network Connetion Management Subprotocol

NPDA - Network Problem Determination Application

NSAP - Network Service Access Point

OAP - Ordinary Application Process

PDU - Protocol Data Unit

PE - Ponto de Entrada

PF - Ponto Focal

PROLOG - Programming in Logic

PS - Ponto de Servico

ROSE - Remote Operation Service Element

QOS - Quality of Service

SASE - Specific Application Service Element

SCF - Sistema de Comunicacão Financeira

SMAE - System Management Application Entity

SMAP - System Management Application Process

SMISE - Specific Management Information Service Element

SNA - Systems Network Architecture

SSCP - System Services Control Point

TAF - Terminal Access Facility

TARA - Threshold Analysis and Remote Access

T-CONNECT - Transport Connect

T-DATA - Transport-Data

T-DISCONNECT - Transport Disconnects

TPDU - Transport Protocol Data Unit

TPDU-DT - Transport Protocol Data Unit de DADOS

TPDU-ER - Transport Protocol Data Unit de ERRO

TSAP- Transport Service Access Point

TSDU - Transport Service Data Unit

XTEL - Expert on Telecommunications

ÍNDICE GERAL

INDICE DE FIGURAS

INDICE DE TABELAS

1. O PROBLEMA DO GERENCIAMENTO DE REDES DE COMPUTADORES

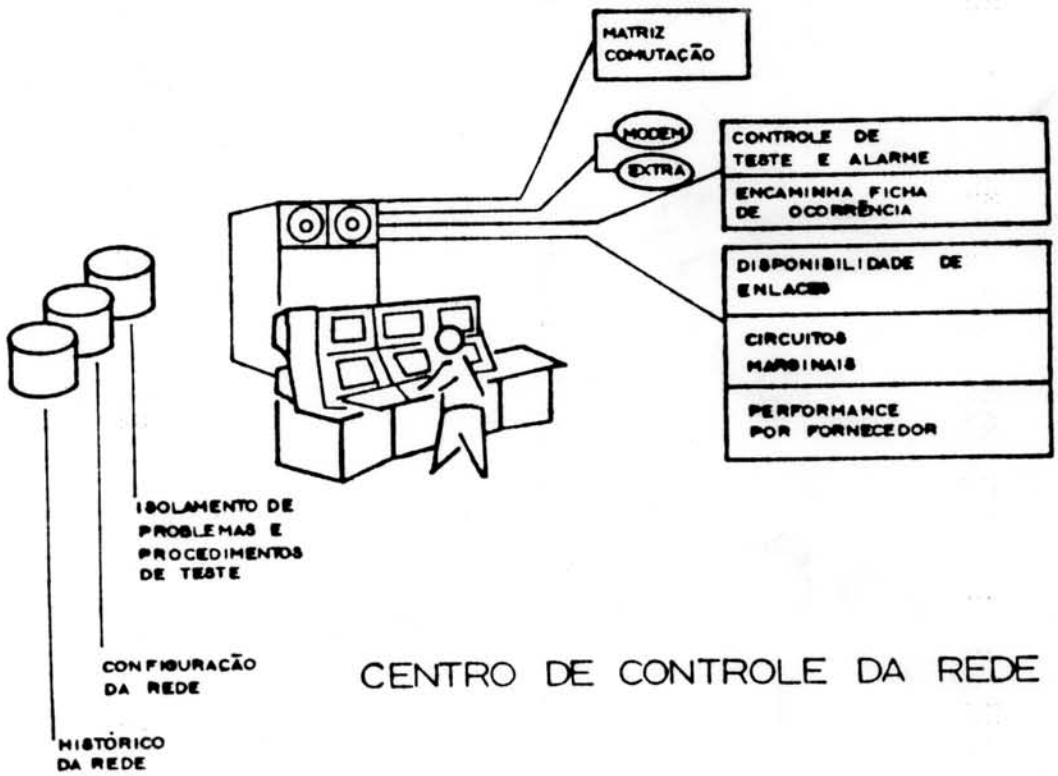
O contínuo crescimento em número e diversidade dos componentes das redes de computadores tornou a atividade de gerenciamento de rede muito mais complexa. Isto se agrava quando estão envolvidos muitos fornecedores. O isolamento e teste dos problemas das redes tornou-se muito difícil devido a duas causas principais:

- Muitos níveis de pessoal envolvido: técnicos de manutenção, operadores controladores de rede, gerentes de sistemas de informações e gerente de comunicações [101].

- Diversidade de formas de controle e monitoração: embora os produtos envolvidos na rede tornem-se gradativamente mais complexos, cada fornecedor oferece ferramentas de controle de redes próprias, para monitorar seus produtos.

Dentre a gama de soluções possíveis, a mais indicada, segundo [27], é utilizar um computador, programado para manipular uma grande variedade de redes, tal como ilustrado a seguir:

Figura 1.1: Centro de controle da rede



Pode ser montado um banco de dados, neste computador gerente da rede, que contenha informações adequadas, tais como as necessárias para o gerenciamento de problemas. O gerenciamento dos problemas é o esforço para identificar, rastrear e resolver situações envolvendo falha física, mau funcionamento de software ou aspectos de treinamento que afetam o funcionamento adequado da rede. Como o tempo decorrido, enquanto o usuário aguardar a restauração do serviço, deve ser menor do que nunca, tudo isso deve ser feito eficientemente.

Normalmente, tem-se um operador controlador de rede, que de uma console, de onde monitora a rede, mantem-se ao par das suas condições, trabalhando com vistas a efetuar o diagnóstico sobre os pro blemas da rede. O diagnóstico de um problema pode ser efetuado em três níveis, tal como sugerido por [58]. O primeiro nível de diagnóstico, ocorre quando chega o relato do problema. Notificações de problemas podem chegar, por telefone, telex ou alarme do sistema. O operador interage então com o computador, executando uma série de testes visando a determinação da causa do problema. Informa ao sistema os dados de identificação do equipamento envolvido, abrindo uma ficha de ocorrência. É recuperada toda a informação existente no banco de dados, pertinente ao problema, tais como, modificações na topologia de rede ou problemas anteriores afetanto aquele dispositivo.

Se a pessoa responsável pela rede pode corrigir a situação, o evento é classificado como um incidente operacional. Senão, é classificado como um problema e encaminhado para uma área especializada. Neste caso, o operador envia a ficha de ocorrência a um técnico encarregado de reparar o problema (integrante dos quadros da própria empresa ou do fornecedor do equipamento envolvido). Costuma ser assinalado na ficha de ocorrência, um código indicati-

vo do impacto de problema sobre o usuário. Isto permite à área especializada determinar prioridades de atendimento, se vários problemas carecem de atenção simultaneamente. Na área especializada ocorre um segundo nível de diagnóstico. São feitos testes sobre os circuitos e outros equipamentos envolvidos (modems, multiplexadores, etc.) sendo acionado o pessoal de manutenção apropriado.

O terceiro nível de diagnóstico envolve a atuação de programadores e analistas, sendo concernente ao funcionamento dos programas de aplicação e software básico. Na prática, somente um pequeno número de problemas chega até o terceiro nível.

Quando o equipamento estiver funcionando corretamente, o operador encerra a ficha de ocorrência.

A interação do operador com o sistema de apoio à atividade de controle e manutenção deve poder ser efetuada de uma maneira fácil [55]. Todo o jargão e nomenclatura técnica complexa, que só é familiar aos técnicos que efetuam os consertos na rede, deve ser atenuado, sendo propiciada possibilidade de interação usando linguagem conversacional. É o computador e não o operador que deve interpretar comandos; o operador deve somente enviar comandos de alto nível, tais como TESTE, para que o computador acione as sequências apropriadas de ações, para obedecer ao comando.

Enquanto o problema que impede o funcionamento da rede (ou de parte dela) não for solucionado, existem três alternativas a seguir, para assegurar a continuidade do serviço [101]:

- uso de equipamento redundante;

- encaminhamento a outra fonte de atendimento (outro terminal, outro concentrador, outra forma de comunicação, outro computador, etc.);

- reconfiguração

Tanto os custos derivados da perda da capacidade de atendimento

podem ser altos, quanto os custos inerentes à manutenção de facilidades adicionais redundantes, para uso em caso de falha. Por outro lado, diagnóstico incorreto ou tentativa de retomar o uso do componente com problemas, antes que os consertos adequados sejam efetivados, somente aumentam os custos, aumentando o tempo de paralização e a frustração dos técnicos e dos usuários envolvidos.

Em vista de tudo isso, deseja-se evidenciar a necessidade de um esquema compreensivo de gerenciamento e uma arquitetura projetada para reduzir os custos diretos do processo de restauração da operacionalidade da rede, bem como para prover a informação e flexibilidade necessária.

O mecanismo de apoio à operação de gerenciamento da rede constitui a ferramenta mais importante para rastrear a resolver problemas [55]. Um esquema deste tipo deverá incorporar as seguintes características:

- O gerenciamento da rede deverá ser parte integral da mesma.

- Deverão ser permitidos múltiplos pontos de acesso ao gerenciamento da rede (estações de controle).

-A informação sobre a rede, bem como as estatísticas de sua atividade (normal ou dos problemas), incluindo os dados de eventos, deverão passíveis de obtenção centralizadamente e apresentados em forma facilmente compreensível, possivelmente usando gráficos.

-Um esquema de tratamento prioritário deve permitir que as mensagens de controle da rede precedam outros tipos de tráfego.

- Devem existir mecanismos de segurança que limitem o acesso à rede e detectem uso não autorizado.

- As funções de gerenciamento de rede devem operar independentemente do meio de transmissão.

- Deverá existir um banco de dados contendo informaçoes sobre

todos os componentes da rede e de seus usuários.

- Alterações na rede e redirecionamento deverão poder ser efetivados de forma flexível e simples.

A integração das funções de gerenciamento na rede é importante e não pode ser acomodada pela simples adição de equipamento extra. Nas atuais arquiteturas de redes, em que é usada uma abordagem de estratificação de funções em níveis, o gerenciamento deve também ser parte das funções inerentes a cada nível [86]. Por exemplo, testes que envolvem o interface físico do equipamento (tal como o comando de teste por loop-back do conector modem-equipamento de processamento de dados) devem estar embutitos no nível físico, devendo ser possível seu acionamento, de alguma forma, controlado pelo sistema gerenciador da rede.

Por outro lado, para que isto seja implementado, devem ser evitadas soluções particulares que impliquem no uso exclusivo de modems com características não padronizadas. É preciso buscar uma solução que permita a gerência de redes mesmo num ambiente heterogêneo.

Em linhas digitais, num ambiente RDSI - Redes Digitais de Serviços Integrados existe a previsão do uso de uma parcela de capacidade do canal para sinalização de controle [69]. Contudo, nem sempre serão utilizados canais digitais numa rede.

Uma importante parte do processo baseia-se na apropriação de informações sobre a rede, sendo a mais importante aquelas relativas a erros, falhas e outras condições problemáticas. Tais dados devem ser armazenados em forma bruta mas também é importante ter valores aceitáveis, como limiares de tolerância que, quando ultrapassados, determinam uma sinalização ao operador ou início de uma ação corretiva. Tais limites não são necessariamente absolutos, tal como o número de erros num circuito por unidade de tempo, sen-

do necessário dispor de estatísticas de erros em função do tráfego existente. Um determinado limiar pode ser aceitável numa situação de carga leve na rede mas intolerável numa outra situação, de carga mais intensa, onde o número de retransmissões fa-ria com que o tráfego total excedesse a capacidade do enlace, afetando seriamente o tempo de resposta.

Quando uma rede encontra-se em situação de operação de forma degradada, pode ser necessário tomar decisões quanto à suspensão de certas atividades, tendo em vista a capacidade diminuída da rede e a importância e grau de urgência dos serviços que devem continuar a ser prestados. Esta importância e grau de prioridade pode variar, segundo a hora do dia ou o local.

O gerenciamento da rede implica na existência de um banco de dados contendo informações completas sobre todos os elementos da mesma (linhas, modems, processadores de rede, terminais, computadores, software, etc.). Para cada um dos itens o operador da rede deve ser capaz de acessar informações tal como: proprietário, localização, custo operacional, arrendatário, número serial, identificação de circuito, uso normal, etc. O acesso deve ser facilitado por meio de linguagens de alto nível para adicionar e manipular dados nele, bem como acessá-los para o cumprimento de funções de gerenciamento de rede. Os dados devem incluir informações sobre os fornecedores, o grau de confiabilidade de seus produtos, com vistas a orientar futuras aquisições ou mesmo substituição de equipamento da rede.

Na medida em que a complexidade da rede aumenta, um vídeo gráfico (preferencialmente colorido) deve oferecer possibilidade de mostrar a rede toda, indicando a natureza e o local das falhas e problemas [11]. Condições críticas devem ser assinaladas por meio de sinais de alerta. Na medida em que forem sendo efetivadas re-

configurações, a rede resultante também deve ser visualizada graficamente. Uma facilidade de aproximação ("zoom") deve permitir ao operador visualizar algum ponto, com grau de detalhamento maior.

Um componente também importante no esquema de apoio ao gerenciamento de rede é a orientação no que tange aos passos ou etapas que devem ser cumpridos. Cada nível de diagnóstico tem um roteiro próprio. O funcionamento de alguns roteiros implica no uso de temporizadores internos que emitem um alerta se certos limiares de tempo são atingidos sem que o problema seja solucionado.

A maioria dos esquemas de gerenciamento de rede existentes atualmente são deficientes, no que tange às necessidades aqui discutidas, especialmente quando é utilizado equipamento de múltiplos fornecedores. É surpreendente constatar que redes de grande porte têm, as vezes, uma estrutura de gerenciamento de problemas manual, baseada em papel. Isto funciona quando a rede é pequena mas, a medida que ela cresce, o processo torna-se incapaz de sequer registrar o universo dos incidentes, nesta forma manual [27].

Alguns usuários têm tentado uma solução paliativa, adquirindo produtos de gerenciamento de rede, produzidos independentemente, constituídos de hardware e/ou software especialmente orientado à combinação de produtos existentes na sua rede [83]. Outra abordagem poderia ser o confinamento a um único fornecedor, quando possível, adotando suas convenções e metodologias para montagem e gerenciamento da rede.

O melhor caminho, porém, seria um gerenciamento de rede coerentemente estruturado, como um elemento de controle integral, que permitisse a conexão de equipamentos, compatíveis ou não, conforme defendido em [101, 104 e 113].

Uma solução assim genérica e integrada auxiliaria os usuários a evitar os altos custos de um pacote gerenciador "customizado" e

facilitaria uma manutenção econômica, o futuro crescimento e evolução da rede, bem como o suporte técnico para sua operação. Conforme destacado por [75] o gerenciamento da rede deve possibilitar uma atuação preventiva em relação aos problemas e não meramente reativa.

1.1 O que deve ser feito

Não basta que um sistema de gerência de rede seja capaz de apropriar dados sobre o funcionamento da mesma. É preciso que alguma forma de análise seja efetuada sobre os mesmos visando determinar tendências de comportamento da rede para antecipar medidas que evitem tais problemas [65]. As projeções futuras ou indicação de tendências visam permitir que, ao invés de deixar a extrapolação puramente à intuição humana, os dados passados sejam usados para gerar projeções estatísticas de eventos críticos. Isto é conseguido correlacionando padrões de dados observados com eventos subsequentes. O padrão dos dados consiste de segmentos de dados correntes e os eventos são ocorrências de falhas catastróficas. Um evento pode consistir de uma sequência de ocorrências e pode incluir informação de várias fontes, tais como monitoração de performance e atividades de manutenção.

O processamento das monitorações de performance, alarmes e atividades de manutenção pode ser visualizado como um processo de dois estágios: filtrar os dados brutos e efetuar sobre eles algum tipo de análise.

É importante verificar efeitos espúrios nos dados para evitar uma interpretação errada dos relatos. Uma lista parcial dos itens a considerar seria:

-identificar condições anormais manualmente causadas, mediante correlação de eventos percebidos nos dados monitorados, que tenham atividade de manutenção relatada;

- uso dos dados de atividade de manutenção para distinguir entre atividades de conserto e problemas que são solucionados sozinhos;

- relacionar problemas que podem ser causados por outros fatores estranhos (perturbações atmosféricas, linha cruzada, etc.).

A partir da análise estatística dos dados resultantes, é possível detectar indícios de eventos críticos nos dados de performance, usando algum método de projeção, tal como relatado em [82].

A idéia básica do método de projeção é que um evento de interesse é precedido por uma sequência de ocorrências que, pelo menos posteriormente, mostram ter a ver com aquele evento. A projeção é determinada em termos de probabilidades condicionais empiricamente determinadas. As sequências de ocorrências foram denominadas **padrão** e o evento de interesse foi denominado **resultado** em [7]. As projeções levam à: "PROBABILIDADE DE UM PARTICULAR RESULTADO, DADO QUE TENHA SIDO OBSERVADO COM CERTO PADRÃO DE COMPORTAMENTO EM ALGUM DOS COMPONENTES DA REDE".

Como se deduz, um sistema adequado para apoio à gerência de uma rede, implica não apenas na obtenção de dados sobre sua atividade, mas também, prove dados sobre as ações corretivas e, adicionalmente, provê subsídios para apoiar a análise de tendências do comportamento dos componentes da rede. Tem-se aqui, um sistema que "aprende" com o passar do tempo e com a ocorrência de eventos.

1.2 O que pode ser feito

Existe uma tendência em gerência de redes heterogêneas, que implica no uso de um sistema independente, que recebe, por monitoração e/ou mediante solicitação, dados sobre a atividade normal e/ou exceção, dos componentes da rede. O que começa a ser considerado de ora em diante é a adição de inteligência a tais sistemas transformando-os em sistemas especialistas de apoio a gerência das redes, conforme discutido em [89, 8 e 62].

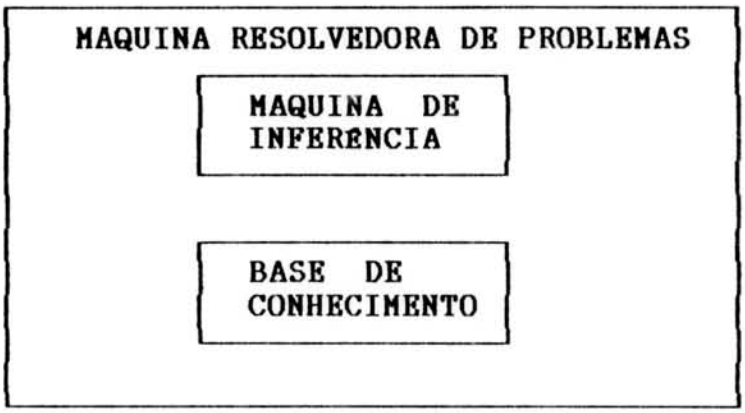
A quantidade de dados decorrentes da atividade de monitoração de uma rede é muito alta. O tipo de processamento clássico, determinando médias, pode eclipsar mudanças de estado de algum componente que, se analisadas num contexto mais "inteligente" teriam condições de fornecer indicios de tendências de comportamento, tal como no sistema da Bell descrito em [69, 82 e 8].

Sistemas capazes de prover soluções para problemas complexos, podendo substituir especialistas humanos, começam a ser projetados e implantados em algumas áreas específicas [102]. Tais sistemas são denominados especialistas ou "Knowledge-Based Expert System". Em geral, sistemas especialistas são programas resolvedores de problemas que aplicam as técnicas de Inteligência Artificial. O cerne de tais sistemas é uma base de conhecimento, que consiste numa coleção de fatos, definições, procedimentos e regras heurísticas, adquiridas diretamente do especialista humano. A base do conhecimento representa os conhecimentos da área de aplicação.

O grande volume de dados que precisa ser analisado, no caso de redes, demanda uma forma de estruturação mais complexa, que permita buscas mais eficientes [13, 30, 56 e 88]. Por isso, um banco de dados pode ser usado, adicionalmente, para guardar os dados de forma organizada.

Os sistemas especialistas são construídos com dois módulos básicos que, juntos formam a máquina resolvedora de problemas:

Figura 1.2 : Sistemas especialistas



A máquina de inferência é o componente do sistema que controla o processo dedutivo.

A representação de tudo o que vai na base do conhecimento deve ser feita através da escolha de um conjunto adequado de símbolos e através da definição das convenções.

Assim, uma tarefa básica para o projeto de um sistema especialista consiste na construção de base de conhecimento. Existem algumas questões básicas que devem ser respondidas previamente. São elas:

a) Que tipo de conhecimento é envolvido?

b) Como pode o conhecimento ser representado?

c) Quanto conhecimento é necessário?

d) Qual é exatamente o conhecimento necessário?

A busca de respostas para a pergunta a, pode ser encontrada a partir de um estudo exaustivo sobre o comportamento de redes típicas.

Para a decisão sobre a forma de representação do conhecimento é preciso selecionar uma forma dentre as várias possíveis (regras de produção, lógica formal, regras heurísticas). Também deverá existir uma linguagem para representar o conhecimento. Os critê-

rios básicos para uma linguagem de representação de conhecimento, segundo [17] são:

- Poder de expressão: podem os especialistas comunicarem seu conhecimento efetivamente ao sistema?;

- Compreensibilidade: podem os especialistas compreenderem o que o sistema sabe?;

- Acessibilidade: pode o sistema usar a informação fornecida?

O uso de sistemas especialistas para apoiar a gerência de redes tem sido alvo de um crescente número de pesquisas e prevê-se que no futuro seu uso será cada vez mais intenso.

Incorporando o conhecimento adquirido de uma forma incremental, os sistemas especialistas como um todo abrem a porta para aplicações de computadores em áreas onde existe muita carência de especialistas humanos [15, 67 e 87]. Este é o caso de gerência de redes [113].

1.3 O que este trabalho propõe

Em vista do exposto, pretende-se ter apontado a necessidade de um sistema independente de gerência de redes, capaz de interagir com os demais componentes da rede para coletar dados sobre seu comportamento e capaz de interagir com os responsáveis pela rede para apoiar sua tomada de decisão no que concerne à manutenção, reconfiguração ou expansão da rede. Também pretende-se apresentar neste trabalho a tendência existente em utilizar sistemas especialistas para apoiar a área de gerência de redes [5, 65, 75, 7 e 56].

Neste sentido, foi realizado um estudo que iniciou por uma análise de soluções típicas para os contextos de rede mais comuns, existentes no mercado atualmente ou previsíveis num futuro

próximo. A seguir foi elaborada uma proposta de uma estratégia para gerenciar redes OSI apoiada em Inteligência Artificial. Os conceitos relevantes da Inteligência Artificial foram estudados com vistas à determinação dos que fossem apropriados ao uso na estratégia proposta.

As informações necessárias à implantação da estratégia de gerência de rede OSI usando Inteligência Artifical foram relacionadas e estruturadas, usando-se uma metodologia orientada a objetos. Finalmente, com base neste trabalho foi projetado um sistema especialista que constitui um protótipo da estratégia definida. O trabalho conclui com uma avaliação do que foi realizado e uma indicação da continuidade do trabalho.

O trabalho está estruturado em capítulos adicionais a seguir resumidos:

O capítulo 2 apresenta alguns exemplos de soluções para gerência de redes em contextos diferentes: usando multiplexadores estatísticos para montar uma sub-rede e embutindo nestes as facilidades de gerência; usando uma rede homogênea, constituida de equipamentos computacionais de um único fornecedor com os mecanismos de gerência embutidos nos diversos componentes da rede; usando uma sub-rede de comutação de pacotes que também tem em cada nó o suporte para a gerência da rede e de seus acessos. Por último, é apresentado neste capítulo um resumo da proposta ISO para gerenciamento de uma rede de computadores que use a arquitetura OSI.

O capítulo 3 contém o resultado da análise realizada no âmbito deste trabalho, concernente aos objetos gerenciáveis num contexto OSI, às modalidades para apropriar os dados relevantes e úteis para a gerência da rede e as bases para seu processamento com vistas a prover suporte à gerência operacional da rede.

O capítulo 4 mostra os conceitos da Inteligência Artificial

aplicáveis ao projeto de um sistema especialista para gerência de redes e a metodologia de trabalho.

O capítulo 5 descreve as informações necessárias para a gerência de rede e sua estruturação.

O capítulo 6 descreve como foi estruturado o protótipo do sistema especialista para gerência de redes OSI implementado para testar a idéia proposta neste trabalho.

O capítulo 7 apresenta comentários sobre a implementação feita bem como uma análise de sua viabilidade. Contém também uma análise do trabalho efetuado como um todo, as conclusões resultantes e as perspectivas de continuidade para o mesmo.

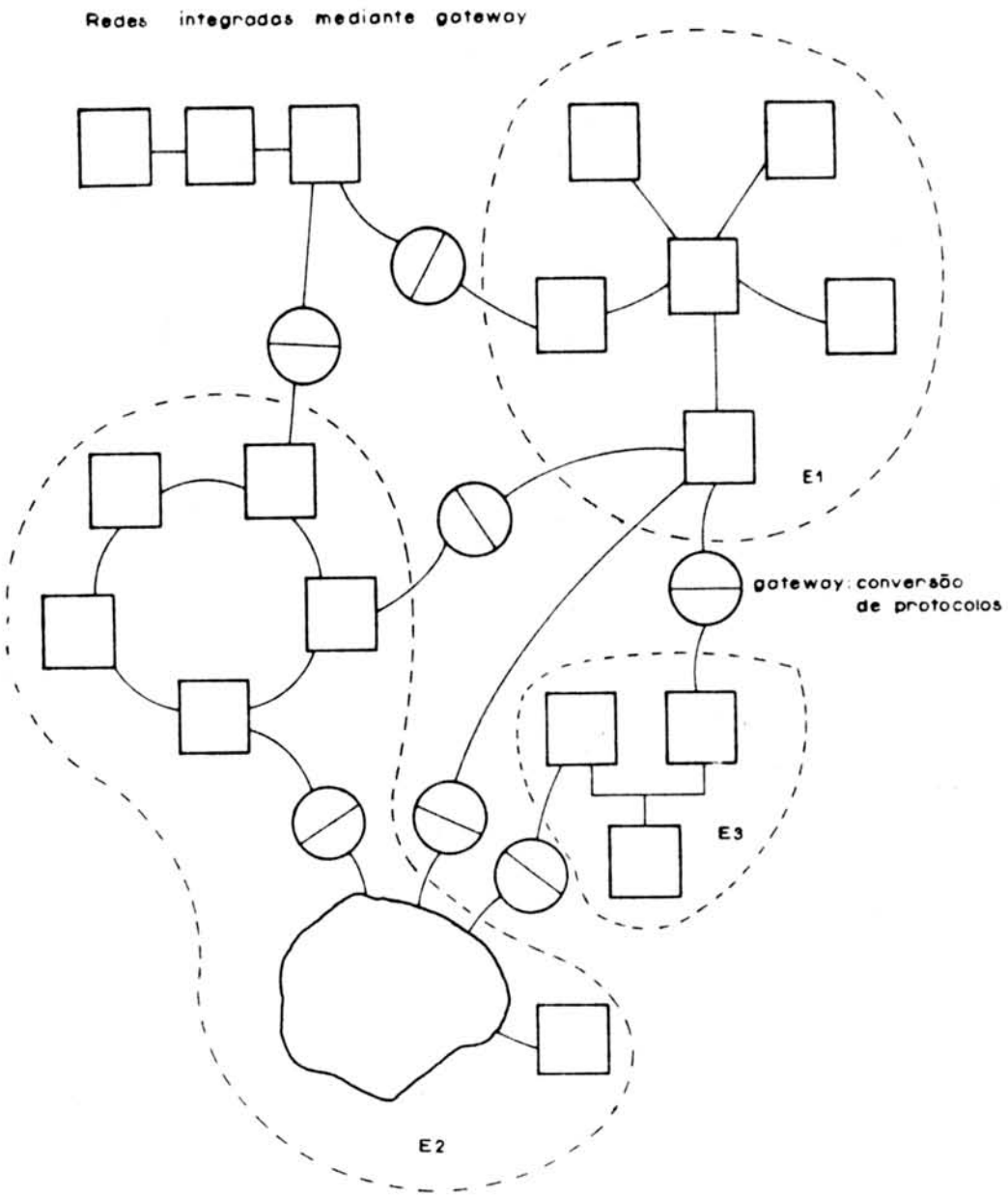
2. PANORAMA ATUAL DE GERÊNCIA DE REDES DE COMPUTADORES

A gerência de redes de computadores tornou-se mais complexa na medida em que as próprias redes cresceram em tamanho e em grau de diversidade. Conforme descrito por [83], diversos fornecedores estão provendo produtos para gerência de rede, que variam de microcomputadores a mainframes. As facilidades para gerenciamento de rede providas nestes produtos variam muito, existindo inclusive os que nada oferecem neste sentido [99 e 11].

Os responsáveis pela gerência da rede estão convivendo atualmente com ilhas de comunicação de dados, nas quais estão integrados equipamentos com características comuns (integram uma mesma arquitetura de rede e são gerenciáveis por um mesmo sistema). Em cada uma destas ilhas pode existir ou não facilidades para gerenciamento da rede as quais são, na maioria dos casos, não integráveis umas às outras, o que leva à necessidade da implantação de inúmeros gateways para conversão dos relatos de eventos e comandos usados nos diferentes contextos (figura 2.1).

Assim como a solução para o problema das ilhas de comunicação é uma arquitetura de comunicação comum (modelo OSI), a solução para o gerenciamento de ambientes heterogeneos também é uma arquitetura de gerenciamento comum [86]. Existindo uma arquitetura de gerenciamento padrão os diversos fornecedores serão capazes de prover gateways entre suas arquiteturas de gerenciamento de redes a arquitetura padrão, tornando possível a intercomunicação de dados e comandos de gerência de rede entre elas [104].

Figura 2.1: Redes heterogêneas x gerenciamento



E1, E2, E3: Esquemas de gerenciamento diferentes

Figura 2.1 - Redes heterogeneas x gerenciamento

Neste capítulo serão descritas brevemente algumas soluções para gerenciamento de rede usadas em três ambientes de redes: redes baseadas em multiplexadores estatísticos, uma rede com arquitetura proprietária e uma rede pública de comutação de pacotes. Posteriormente será descrito, com maior grau de detalhamento, a arquitetura padrão de gerenciamento de redes sendo elaborada pela ISO.

2.1 Gerenciamento de redes com multiplexadores estatísticos

Muitas redes no Brasil são baseadas em multiplexadores estatísticos que, além das funções de transporte eficiente de dados, oferecem, adicionalmente, algumas funções de gerenciamento da rede de transporte, especialmente no que concerne aos enlaces de alta velocidade entre os multiplexadores.

Tais redes sinalisam, por meio de alarmes, algumas condições de exceção que podem detectar tal como mostrado na tabela 2.1. Observa-se nos sistemas de gerenciamento de rede implementados nestes multiplexadores estatísticos uma preocupação com o processamento dos dados provenientes da monitoração da rede. São usados microcomputadores externos, para os quais passam os dados que serão manuseados pelo software apropriado, de modo a fornecer ao gerente da rede uma boa parte dos serviços previstos. A CASE usa um equipamento da Convergent Technologies, a DCA usa um micro tipo IBM PC/AT e a RACAL-MILGO usa um micro PDP da DEC [83]. A apresentação dos dados de forma a serem facilmente compreendidos pelos usuários é outra preocupação presente em tais sistemas sendo usados gráficos e/ou sistemas de cores para para as mensagens de alerta e diagramas da rede.

Os dados derivados de monitoração remota são passados através da rede entremeadamente com o tráfego normal (DCA e RACAL-MILGO)

ou via sinalização fora de banda (CASE e RACAL-MILGO).

Pode-se observar que a ênfase de tais sistemas é voltada aos componentes físicos diretamente ligados aos multiplexadores e aos enlaces entre os mesmos conforme destacado por [83].

Tabela 2.1 : Sistemas baseados em multiplexadores estatísticos

FORNECEDOR/EQUIPAMENTO	CASE/DCX	DCA/NPS PC	RACAL-MILGO/CMS
Alarmes gerados			
Falha de equipamen		x	x
Falha de ETD	x		
Modem inoperante	x		x
Qualidade do sinal	x		
Potência do sinal na linha	x		
Linha inoperante		x	x
Tremulação de fase	x		
Degradação do serviço		x	x
Ausência de polls		x	
Congestionamento		x	
Tipos de alarmes			
Audível	x	x	x
Mensagem de console/impr.	x	x	x
Gráficos	x	x	
LED	x		
Dados registrados			
Configuração	x	x	x
Atividade	x(linha)	x	x
Hardware	x	x	x
Alarmes	x	x	x
Problemas de linha	x		
Tipo de modem		x	x
Operações efetuadas/Histórico	x	x	x

(Cont. Tabela 2.1)

FORNECEDOR/EQUIPAMENTO	CASE/DCX	DCA/NPS PC	RACAL-MILGO/CMS
Relatórios gerados			
Esquema/configuração da rede	x	x	x
Lista da rede	x		
Componentes		x	x
Aviso de problemas	x	x	x
Rastreamento p/auditoria		x	
Resumo p/contabilização	x	x	x
Histórico de atividades	x	x	x
Tendência de performance			x
Outros serviços			
Análise de performance		x	x
Monitoração	x	x	x
Diagnóstico	x	x	x
Configuração		x	x
Análise da qualidade de linha			x
"Strapeamento" por software	x		
Discagem para backup	x		
Gerenciamento de falhas		x	

2.2 Gerenciamento de rede de comunicações no ambiente SNA

A SNA-Systems Network Architecture da IBM pode ser visualizada como sendo constituida de duas redes, uma física e uma lógica [105].

A rede física é constituida de uma conjunto de nodos (componentes de hardware) interconectados por enlaces. Atualmente existem quatro classes de tais componentes de hardware: computador host, controladoras de comunicação, controladoras de terminais e terminais. Estes componentes são reunidos em domínios, sob o controle de um host, no qual o SSCP-System Services Control Point é executado e tem a função de prover interface com o operador da rede, interface com o gerenciamento da rede, controle da configuração, iniciação da rede e reinício de operações, mesmo em caso de parada anormal. O SSCP também participa do estabelecimento de sessões. Sessões são estabelecidas entre as unidades lógicas da rede para fins de intercâmbio de dados e controle.

A rede lógica consiste de entidades denominadas NAU-Network Addressable Units, que interagem, através da rede física, estabelecendo sessões, mediante o uso de protocolos, com o fim de intercambiar os dados do usuário e outras informações de controle da rede. Existem três tipos de NAUs na rede: o SSCP em si, as unidades físicas e as unidades lógicas. A unidade lógica (LU-Logical Unit) é um conjunto de serviços de gerenciamento de funções que apoiam um programa de aplicação ou um usuário de um terminal (ambos são denominados usuário final de uma sessão). Inerente ao fucionamento de uma unidade lógica existe um conjunto de protocolos que são usados para que um usuário possa comunicar-se com outro usuário, o que ocorre mediante o estabelecimento de uma sessão. Diferentes tipos de unidades lógicas usam diferentes tipos de pro-

tocolos para estabelecer as características das sessões estabelecidas, os métodos de controle de fluxo e outras propriedades da sessão. O estabelecimento de uma sessão também implica no uso de certas entidades físicas (nodos e enlaces) ao longo da tragetória entre as duas unidades lógicas, bem como no uso de recursos tais como buffers. O iniciador de uma sessão usa nomes lógicos, denominados nomes de rede para identificar os usuários finais de uma sessão. Tais nomes de rede são mapeados pelo SSCP em endereços de rede.

2.2.1 Gerenciamento de recursos físicos

No passado, a determinação de problemas era orientada principalmente à rede física. O relato de problemas e os mecanismos de registro de erros visavam a realização on-line de diagnóstico de nodos locais e remotos. Produtos tais como o NPDA-Network Problem Determination Application foram desenvolvidos para permitir a um usuário da rede SNA receber alertas dos nodos da rede (a respeito de pendências ou de condições de falha) ou solicitar dados de status a nodos da rede e a modems inteligentes. Este software usa os nomes de rede na comunicação com os operadores da rede. O relacionamento entre os nomes de rede e os endereços físicos é efetivado mediante o uso do SSCP. Por exemplo, a notificação de uma falha física de algum nodo, que é comunicada internamente na rede com o endereço físico na rede, é transladado para os nomes lógicos, antes de ser enviada ao operador da rede ou a aplicações de gerenciamento.

Na estrutura de rede em árvore, inicial da SNA, um SSCP controla todo o domínio. Mas houve a adição de sub-redes que tem, por sua vez, recursos gerenciados por si mesmo, tal como o Sistema de

Comunicação Financeira IBM 3600 que continha estações de trabalho interligadas num loop que não eram de conhecimento do SSCP. Isto limitava a flexibilidade do gerenciamento da rede, conforme [90].

Foi então desenvolvido um programa, TARA-Threshold Analysis and Remote Access, como uma função adicional do NPDA, o qual, rodando nos dois sistemas, o SSCP e sistema financeiro, permite que o primeiro solicite dados ao segundo. Os dados podem ser solicitados pelo operador da rede ou ser encaminhados por comandos iniciados em função de períodos de tempo decorrido. Os dados incluem parâmetros operacionais, status dos loops ligados ao 3600, contadores básicos e extendidos, contadores de tempo de resposta na controladora e para as estações de trabalho.

Uma vez que os dados tenham sido coletado, a aplicação no host (TARA) pode prover uma análise contínua dos dados, comparando-os com limiares de erros ou de performance, alertado o operador quando tais limiares são excedidos.

2.2.2 Gerenciamento de recursos lógicos

Com o surgimento do produto NLDM-Network Logical Data Manager foi possível prover suporte on-line que possibilita ao usuário (operador da rede ou técnico efetuando diagnóstico) obter interativamente dados sobre a rede lógica para fins de determinação de problemas. Tais dados são coletados continuamente sobre as sessões estabelecidas entre duas unidades lógicas na rede [90].

Os problemas com a rede lógica são tipicamente atribuíveis a erros de software e podem ser do tipo detectável ou não detectável. Um erro detectável manifesta-se de várias maneiras:

- uma mensagem de erro, a qual pode ser registrada em arquivos sequenciais de log ou apresentada ao operador;

- notificação de falha da sessão, notificada ao usuário final da sessão;

- armazenamento de "dump" causado pelo término anormal de um produto de software.

Um erro não detectável é evidenciado aos usuários das sessões pela ausência de reação da rede, não havendo outros sinais ou notificações de problemas. A causa de tais erros varia de violações de protocolo de sessão até congestionamento de rede. No que concerne a erros de protocolo, dois tipos podem ocorrer:

- Erro ou mal-entendimento dos protocolos requeridos para iniciação da sessão. Tais tipos de erro podem ocorrer devido a erro de programa no estabelecimento da sessão ou engano devido ao uso de diferentes níveis de software de suporte usados nas sessões.

- Incorreto estabelecimento de estados do protocolo pela unidade lógica que está enviando ou pela que está recebendo.

A perda de uma mensagem é outro tipo de erro não detectável. Contudo, mesmo os erros não detectáveis são tratados pelo NLDM.

As primeiras abordagens ao problema, usadas na arquitetura SNA incluiam:

- Ativação de rastreamento, o que requeria o processamento em batch para a impressão dos dados, bem como a transmissão de grandes quantidades de dados através da rede. Isto degradava a performance da rede e gerava grandes quantidades de dados a serem examinados manualmente. Alem do mais, para que um erro não detectado pudesse ser rastreado, ele deveria tornar a acontecer durante o período em que o rastreamento estava ativado.

-Realizando "dumps" de todos os componentes de software suspeitos, envolvidos com uma sessão que evidenciou um erro não detectado (ausência de resposta)

- Colocando comandos no software suspeito para causar um térmi-

no anormal quando ocorre o erro, efetuando então o "dump" e manipulando o erro como uma falha detectável (para a qual existe notificação de ocorrência). Isto pode afetar o próprio funcionamento da rede, na medida em que um componente dela é desativado.

A partir de um estudo, realizado pela IBM para revisar os problemas de erros lógicos relatados pelos clientes, foram incluidas no NLDM mais funções visando:

- Prover facilidade para apoiar o usuário na determinação de problemas com a rede lógica;

- Registrar dados relevantes sobre a sessão e atividades em andamento, antes e durante a ocorrência da falha.

- Registrar os protocolos associados com uma sessão, no momento de sua iniciação;

- Registrar dados relevantes sobre os nodos físicos nos pontos terminais de uma sessão para apoiar o isolamento de problemas num nodo específico;

- Permitir que estas informações pudessem ser acessadas on-line a partir de um ponto centralizado ou de múltiplos terminais de operação numa rede distribuida.

A partir de então o NLDM passou a coletar, armazenar e monitorar dados para determinação de problemas lógicos. São coletados dois tipos de dados relacionados com uma sessão: estado e dados de rastreamento. O estado é provido pelos métodos de acesso usados na arquitetura SNA (ACF/TCAM e ACF/VTAM) informando ao NLDM que uma sessão foi realizada com sucesso. Os dados providos envolvem: indicação de início e fim da sessão, nomes de redes dos usuários da sessão, tipo de sessão, e informação de configuração sobre os pontos onde a sessão inicia e termina. Os dados de rastreamento incluem partes do cabeçalho das mensagens: cabeçalho de transmissão (transmission header) o cabeçalho de solicitação e resposta

(request/reponse header) e os primeiros 11 bytes da unidade de solicitação (request unit) ou dados do usuário. Tais dados somente são coletados para sessões envolvendo um recurso para o qual o rastreamento foi ativado. O NLDM coleta estes dados, mantendo-os no armazenamento provisório, enquanto as sessões estão ativas e colocando-os em arquivos, após o término das sessões, onde permanecem, por tempo limitado, para fins de visualização. Outras informações de rastreamento, retrospectivas, concernentes às funções de apoio à sessão são também passíveis de solicitação e são gravadas quando a sessão é encerrada.

O NLDM usa os serviços do NCCF- Network Communications Control Facility para solicitar, obter e apresentar dados concernentes a entidades remotas, para fins de determinação de problemas. Encaminhamento de solicitação de ativamento do rastreamento em algum ponto remoto, bem como a obtenção dos dados gerados por rastreamento são exemplos de uso do NCCF pelo NLDM.

Uma sessão própria para intercâmbio de tais dados é usada pelo NLDM para obter os dados remotos. É preciso existir uma cópia do NLDM em cada nodo host contendo um SSCP controlando uma sessão.

Com a possibilidade de estabelecimento de múltiplas rotas para interconexão numa rede SNA, que foi uma facilidade adicionada posteriormente, uma versão nova de NLDM passou a coletar também dados sobre os nodos intermediários, por onde os dados da sessão seriam direcionados. Outra facilidade incluida nesta versão 2 do NLDM foi uma facilidade de teste de eco, pela qual uma mensagem é enviada a um específico nodo físico da rede e "ecoada", para verificar a conectividade física entre ambos. Se o teste falha, uma notificação sobre a localização da falha é enviada ao originador do teste e ao gerenciador do ponto falho.

Outra facilidade incluida no software de gerenciamento de rede

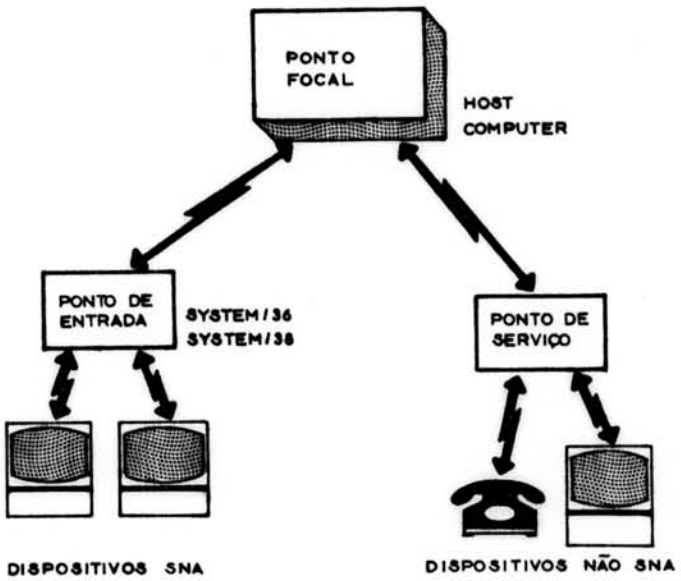
da SNA é TAF-Terminal Access Facility, uma parte do NCCF. Esta facilidade permite que sejam usadas uma variedade de terminais de video e ferramentas de gerenciamento orientadas à visualização dos dados. TAF permite que as mensagens não sejam perdidas enquanto o video estiver dedicado a uma outra aplicação.

Em resumo, no contexto da SNA, os dois softwares, NPDA e NLDM atuam para prover o gerencimento dos recursos físicos e lógicos da rede, respectivamente. Em ambos, alertas podem ser gerados em pontos remotos da rede e encaminhados a um SSCP onde serão manipulados e onde um operador da rede pode visualisa-los, de forma online. Estes produtos foram integrados e agregados de algumas facilidades adicionais, passando a constituir um pacote para gerenciamento de rede SNA denominado Netview.

A proposição da IBM para um gerenciamento de redes teve um grande impacto no mercado porque, como acontece com quase tudo o que a companhia anuncia, começa a tornar-se um padrão de fato, a despeito de alguns problemas que apresenta. Conforme destacado por [83] alguns analistas do produto apontam problemas tais como a quantidade de recursos consumidos (cerca de 5 a 15% dos circuitos em período de carga normal) e a falta de capacidade de comunicação entre as ferramentas que compõem o Netview e com sistemas não IBM.

O lançamento do Netview/PC abriu uma porta para o intercambio de informações de gerenciamento a partir de equipamentos não-IBM. Um conjunto de protocolos foi definido para permitir o envio de informação sobre alertas dos equipamentos não-IBM em seu formato nativo. Foi criado o conceito de **PONTO FOCAL**, que residente num host, tal como mostrado na figura 2.2, prove gerenciamento centralizado no ambiente SNA.

Figura 2.2: Ponto de acesso ao gerenciamento SNA



Também passaram a existir **PONTOS DE ENTRADA** que, baseados em minicomputadores constituem dispositivos endereçáveis dentre da arquitetura SNA e que concentram dados de gerenciamento de rede provenientes dos dispositivos ligados a eles e passam tais dados para o PONTO FOCAL. Os **PONTOS DE SERVIÇO** são equipamentos IBM ou não-IBM tal como microcomputadores que convertem os formatos de mensagem não IBM para os formatos de mensagens de gerencia de rede da SNA e envia-os para o PONTO FOCAL para processamento. Além do PONTO DE SERVIÇO não é necessária uma conexão SNA.

Inicialmente os fornecedores dos equipamentos atuantes como PONTO DE SERVIÇO proviam apenas um resumo do status de seu equipamento de rede e alertas. Durante o ano de 1987 foi definido um formato de relato de alertas padronizado que eliminou a necessidade de mapeamentos.

Pode-se notar que mesmo num contexto de arquitetura proprietá-

ria como é o caso da SNA existe a necessidade de prover "portas" para que informação de gerenciamento proveniente de equipamentos não integrantes daquela arquitetura possam também ser interconectado e gerenciados.

Mais recentemente, a IBM passou a investir no desenvolvimento de ferramentas para extender as funcionalidades existentes no NetView, usando sistemas especialistas [16]. Está sendo desenvolvido, por exemplo, um assistente inteligente para o operador da rede. Este sistema especialista recebe mensagens de erros do NetView e interativamente guia o operador na solução do mesmo, formulando perguntas e sugerindo ações. Outros projetos, relatados em [16] contemplam o desenvolvimento de sistemas especialistas para auxiliar a função de help-desk, para determinação de problemas com impressoras e modems.

2.3 Um centro de supervisão e controle de uma rede de pacotes

No projeto de um centro de supervisão e controle para gerência de uma rede de comutação de pacotes, desenvolvido no CPqD da TELEBRAS [64], o Centro de Supervisão e Controle (NCC) é visto pela rede como se fosse um assinante comum. Em cada nó da rede também existe um assinante interno com o qual o NCC se comunica. Este conjunto de assinantes formam um grupo fechado de assinantes.

Informações descrevendo os elementos de comunicação (linhas, enlaces, circuitos virtuais e assinantes) estão distribuidas nos nós da rede de pacotes e também estão no NCC. Protocolos de aplicação especificamente desenvolvidos para este fim são usados para o intercâmbio de informações entre o NCC e os assinantes internos nos nós.

Uma linguagem de operação da rede, baseada em recomendações do

CCITT (Z.311 e Z.341) é usada para que os operadores possam inserir comandos, a partir de terminais de operação. Para cada comando existe uma resposta imediata mas em alguns comandos, de execução longa podem vir respostas posteriores, dando informações sobre a execução do comando. Outras mensagens, de emissão expontânea também são enviadas para informar os operadores sobre ocorrências diversas durante o funcionamento da rede, tais como falhas de equipamento, problemas de software, alterações de estado dos elementos etc.

Um NCC é responsável pela supervisão e controle de um conjunto de nós de comutação e concentradores de acesso à rede que compõem o que é denominado de região de atuação do NCC. Sobre uma mesma região de atuação é possível existir um outro NCC, formando uma configuração duplex em que ambos os NCCs desempenham conjuntamente a função de supervisão e controle.

Os dados intercambiados pelas entidades que cooperam na gerência da rede ocupam uma parcela da disponibilidade do canal. Para evitar que isto ocorra, em detrimento do atendimento das atividades normais da rede, o envio das informações de gerência pode ser postergado até situações em que não exista tráfego normal de dados. Contudo, nem sempre isto é possível, devido à urgência com que um comando ou resposta tenha que ser propagado. Em tais situações, o tráfego derivado da gerência da rede pode impactar o seu desempenho. Se não houver uma outra forma de transmiti-los, tal como usado em certos modems (uso de canal secundário), é preciso atentar para a necessidade de reduzir ao máximo o intercâmbio de dados de gerência de rede.

2.4 A padronização em gerência de redes

Atualmente, a necessidade por facilidades de gerenciamento de redes está evoluindo em duas áreas específicas. De um lado, para o provedor dos serviços de rede, existe a necessidade de poder, a partir de um único ponto, ter uma visão geral da rede com possibilidade de atuar sobre cada um dos componentes visando corrigir possíveis problemas existentes. De outro lado, é desejável que se possa interconectar na rede diferentes tipos de equipamentos e ainda assim reter a capacidade de gerenciar o todo.

O advento da padronização de protocolos para redes, possibilitado com a adoção do modelo OSI (figura 2.3) irá cada vez mais permitir que redes heterogeneas sejam implantadas e, para poder gerenciar este ambiente, uma arquitetura de gerenciamento de redes compostas de sistemas abertos também está sendo definida pela ISO há bastante tempo [35 e 39].

Figura 2.3: Modelo OSI da ISO

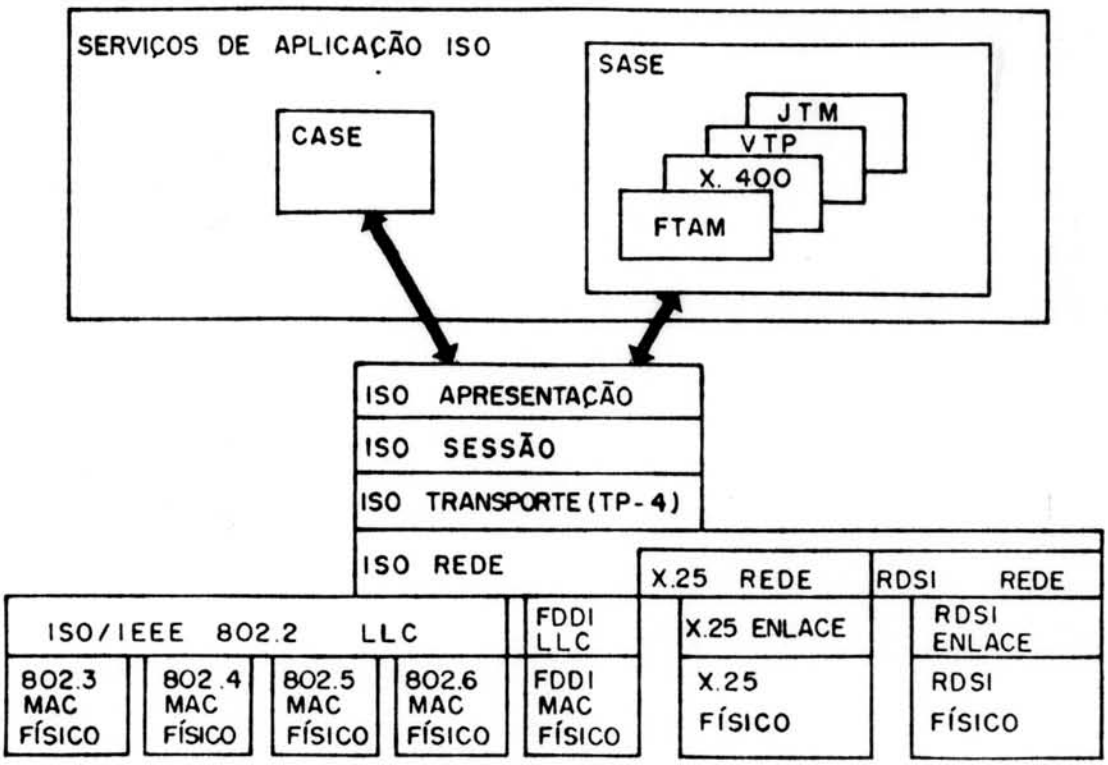


Figura 2.3 - Modelo OSI

As atividades de padronização sobre Gerenciamento de Sistemas Abertos interconectados estão sendo desenvolvidas no âmbito da ISO. O grupo de trabalho 4 do subcomitê TC97/SC21 está trabalhando no sentido de desenvolver padrões para gerenciamento do ambiente OSI. A tarefa foi dividida em duas partes: uma estrutura de gerenciamento [37 e 39] e um serviço de informações de gerenciamento. As discussões sobre o tema tem sido intensas e as propostas evoluem evidenciando mudanças de ano para ano, como mostram os documentos intermediários [42, 43 e 47] .

A estrutura de gerenciamento constitui uma recomendação básica [38], para a área toda e já se tornou um padrão internacional oficial. O objetivo deste padrão é descrever um esquema básico para as atividades de gerenciamento pertinentes ao contexto OSI e para identificar os serviços de gerenciamento que serão suportados pelos protocolos de gerenciamento OSI. O serviço de informações de gerenciamento é concernente à definição de um conjunto de elementos de serviço que possam ser utilizados para apoiar o gerenciamento de aspectos OSI de sistemas abertos reais. Esses elementos de serviço permitirão que :

- sistemas abertos remotos relatem eventos não solicitados a um sistema aberto monitor;

- um sistema aberto monitor solicite dados OSI a um sistema aberto remoto;

- um sistema aberto tenha a possibilidade de causar ações ou alterar o status de recursos OSI um sistema aberto remoto.

Todos estes serviços visam atender às necessidades de usuários de sistemas abertos, tais como:

- Desenvolver atividades que permitam aos gerentes planejar, organizar, supervisionar, controlar e contabilizar o uso de serviços de interconexão;

- Habilidade de reagir a requisitos de mudança;

- Facilidade para assegurar um comportamento previsível para as comunicações;

- Facilidades que propiciem proteção da informação e autenticação da fonte e destino dos dados transmitidos.

As ferramentas de gerenciamento que proveem este suporte podem variar em complexidade, dependendo dos requisitos dos usuários. Tais ferramentas podem operar localmente ou cooperar através de um certo número de sistemas abertos.

Estes requisitos são satisfeitos através de uma série de facilidades que podem ser acionadas pela operação local e/ou pela comunicação de informações entre sistemas abertos. As facilidades definidas são:

a) Gerenciamento de problemas.

b) Gerenciamento de contabilização.

c) Gerenciamento de configuração e nomes.

d) Gerenciamento de performance.

e) Gerenciamento de segurança.

Cada uma destas facilidades será descrita a seguir:

Gerenciamento de problemas é o conjunto de facilidades que habilita a detecção, isolamento e correção de condições anormais de operação do ambiente OSI. Tais condições podem ser persistentes ou transientes e os problemas são manifestados como erros na operação de um sistema aberto. Visando sua detecção deverão existir facilidades para: manter e examinar logs de erros, aceitar notificações de detecção de erros e atuar em função das mesmas, rastrear problemas, executar sequências de testes de diagnóstico e corrigir os problemas.

Gerenciamento de contabilização é o conjunto de facilidades que permite a cobrança pelo uso dos objetos gerenciados. Isto implica

em: informar os usuários sobre os custos incorridos ou recursos consumidos, estabelecer limites para o uso dos objetos gerenciados, combinar os custos quando são usados vários objetos gerenciados para atender um determinado objetivo de comunicação.

Gerenciamento de configuração e nomes é o conjunto de facilidades que exerce o controle sobre os objetos gerenciados; identifica-os, coleta e provê dados aos mesmos para apoiar o provimento de operação contínua de serviços de interconexão. Isto implica em: setar os parâmetros do sistema aberto; inicializar e encerrar a operação dos objetos gerenciados, coletar dados sobre o status do sistema aberto de forma rotineira e em função de alguma mudança significativa de estado; associar nomes com conjuntos de objetos gerenciados.

Gerenciamento de Performance é o conjunto de facilidades necessárias para avaliar o comportamento de objetos gerenciados e a efetividade das atividades de comunicação. Isto implica em: apropriar dados estatísticos, manter e examinar logs com o histórico dos estados do sistema para fins de planejamento e análise.

Gerenciamento de segurança relaciona-se com os aspectos de segurança num contexto OSI, essenciais para operar o Gerenciamento OSI corretamente e proteger os objetos gerenciados. Isto implica em: apoiar autenticação; controlar e manter facilidades para autorização; controlar e manter controles de acesso; apoiar gerenciamento de chaves para criptografia; manter e examinar logs de segurança.

2.4.1 O modelo do gerenciamento OSI

O gerenciamento no ambiente OSI engloba as atividades necessárias para controlar, coordenar e monitorar os recursos que permi-

tem a comunicação no ambiente OSI. Estas atividades relacionam-se aos meios pelos quais um sistema aberto real obtem dados para possibilitá-lo supervisionar e controlar seus recursos de comunicação e coopera com outros sistemas abertos reais para supervisionar e controlar o ambiente OSI. A figura 2.4 apresenta a estrutura básica do modelo de gerenciamento OSI.

No modelo OSI, aplicações (A) comunicam-se com outras aplicações, em outros sistemas remotos, usando os serviços providos pelos níveis inferiores. O interface com o nível imediatamente inferior é uma entidade de aplicação (AE-Application Entity).

Na arquitetura OSI, a operação de cada nível visa facilitar a operação do nível superior a ele. Cada nível é responsável pelo gerenciamento de erros operacionais que ocorram na sua atuação. Para isso, em cada nível existe uma entidade de gerenciamento de nível (**LME**-Layer Management Entity), responsável pela coleta de informações de gerenciamento do nível, pela alteração de parâmetros operacionais, pelo relato de eventos extemporâneos e pela mudança de estado da operação do nível. Uma entidade de gerenciamento de nível "N" gerencia todas as conexões ativas entre a entidade de nível **N** e outras entidades pares.

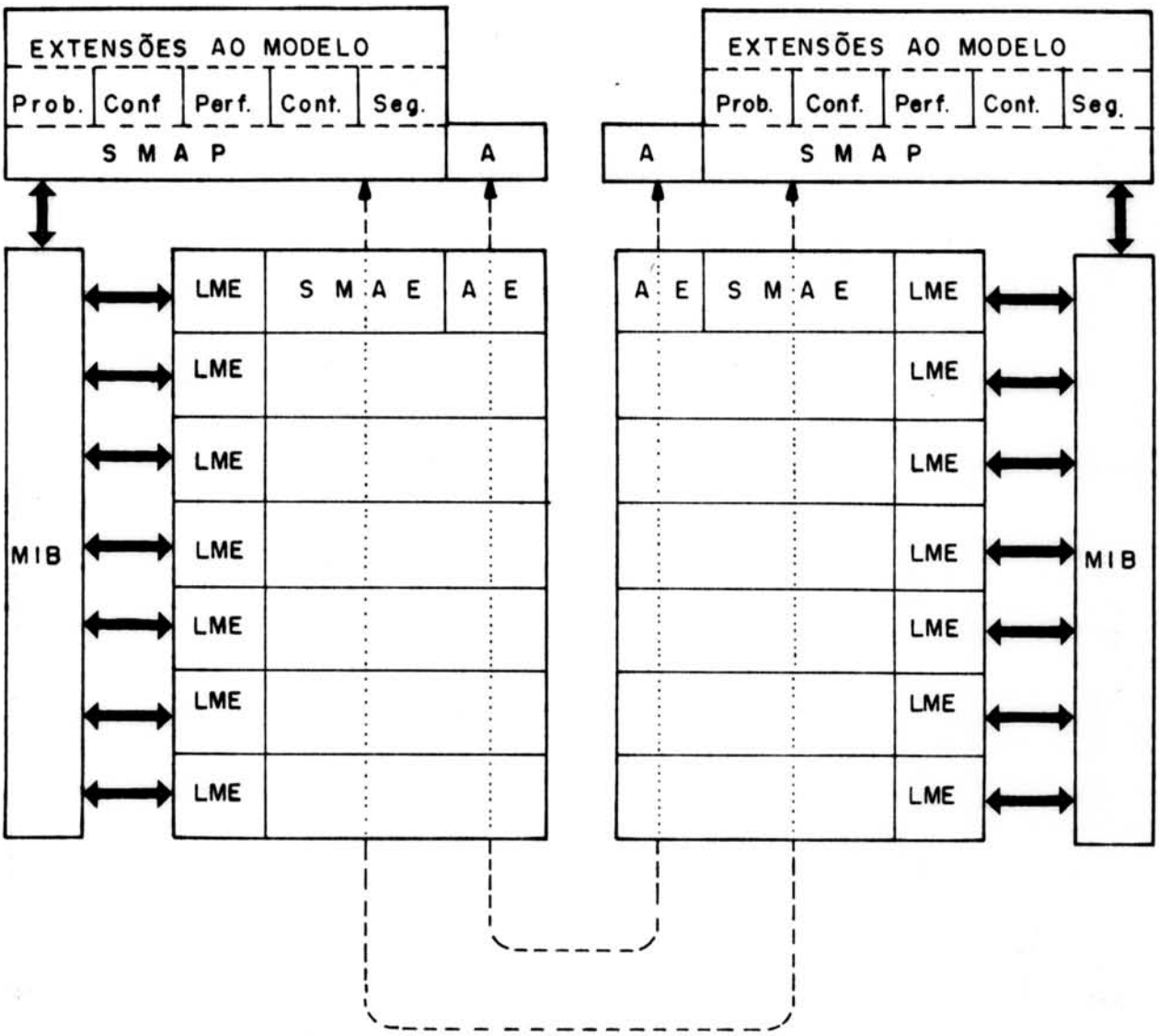


Figura 2.4 - Ambiente de gerenciamento OSI

A base de informação de gerenciamento (**MIB**-Management Information Base) é o repositório lógico para toda a informação pertinente ao gerenciamento OSI. Exemplos de informações que integram a MIB podem ser: informação sobre o estado das entidades gerenciadas, contadores, parâmetros operacinais, limiares, etc. As informações da MIB podem estar dispersas por áreas de memória, registros, discos, etc. Os dados da MIB são estruturados de acordo com as exigências do processo gerenciador que necessita acessá-los, conforme ilustrado na figura 2.5.

Figura 2.5 : Formas de acesso à MIB

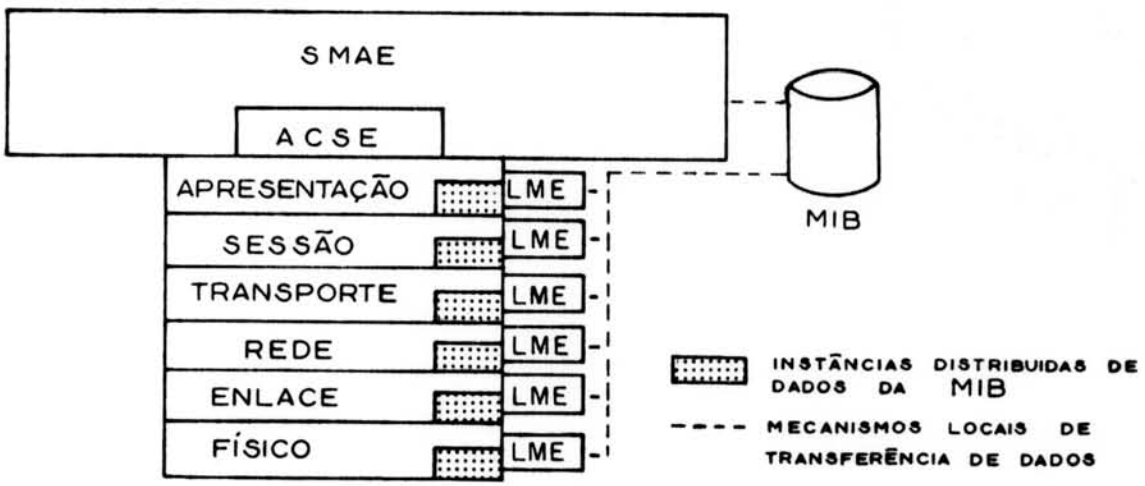


Figura 2.5 - Formas de acesso à MIB

O gerenciamento do sistema aberto é efetivado por um processo de aplicação denominado SMAP-System Management Application Process. Este processo comunica-se com outros SMAPs, em outros sistemas abertos, obtem informação, deles ou da MIB, comunica-se com as LMEs (de forma não definida pelo padrão OSI) e provê interface aos programas ou usuários humanos.

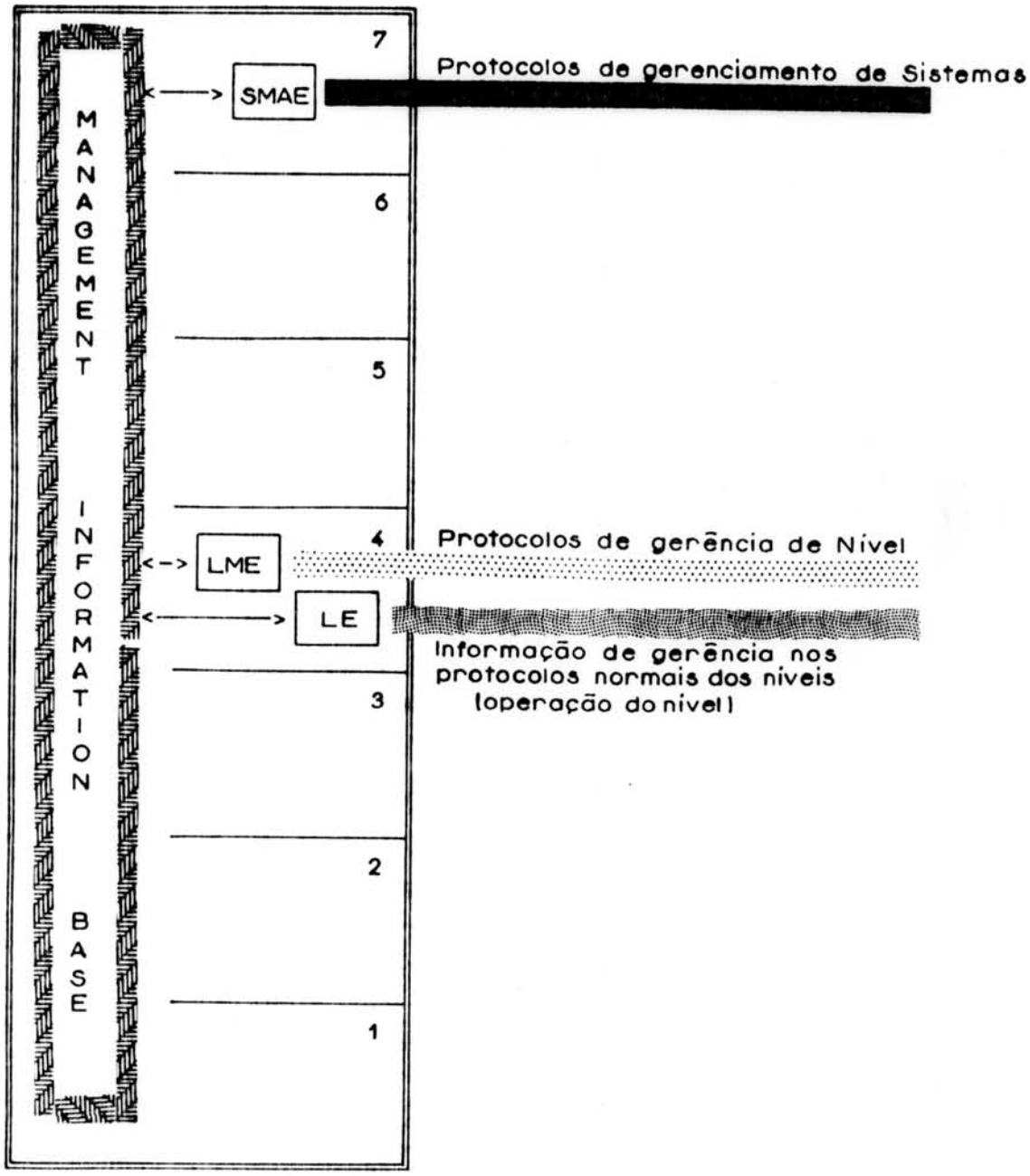
A **SMAE**-System Management Application Entity é a porção OSI do processo SMAP. Esta entidade provê ao SMAP o serviço de apoio na interação com outro SMAP. A SMAE prove ao SMAP os serviços definidos pela ISO para o nível de aplicação (CASE-Common Application Service Elements) [34]. Uma SMAE prove elementos de serviço comuns de aplicação (ACSE-Application Service Common Application) e elementos de serviço específicos de gerenciamento (MISE-Management Information Service Elements) [37 e 42]. Um processo de aplicação comum também poderia usar os serviços providos pela SMAE para o intercâmbio de informações de gerenciamento. Os elementos de serviço específicos de gerenciamento (MISE) são de dois tipos: os comuns (CMISE-Common Management Application Service Elements) [36] e os específicos (de problemas, de performance, de configuração, de segurança e de contabilização) [38].

Acima de todos estes serviços pode-se ter software ou usuários humanos que, interagindo com estas entidades, através dos serviços por elas providos, recebem informações sobre a rede e exercem controle sobre a mesma. Os processos de gerenciamento, que coletivamente proveem o gerenciamento OSI, recebem informação de controle não apenas de pessoas e/ou software atuantes com agentes administrativos locais. Eles também recebem informação de controle de seus SMAPs, das entidades de gerenciamento de nível (LME) e das entidades de nível (LE-Layer Entity). Os processos de gerenciamento exercem controle:

a) diretamente sobre objetos no mesmo sistema aberto, através de mecanismos locais, ilustrados na figura 2.5;

b) sobre objetos em outros sistemas abertos, mediante intercâmbio de PDUs (Protocol Data Units). As entidades responsáveis pelo intercâmbio de PDUs com outros sistemas abertos são as mostradas na figura 2.6: gerenciamento de sistema (SMAE), gerenciamento do nível "N" (LME) e entidade de nível "N" (LE).

Figura 2.6: Intercâmbio de informações no gerenciamento OSI



O Gerenciamento do sistema provê mecanismos para monitorar, controlar e coordenar todos os objetos gerenciados dentro dos sistemas abertos, em um ou mais níveis. O Gerenciamento de Sistemas é realizado mediante o uso de protocolos de nível de aplicação, que são usados para o intercâmbio de informação relativa à monitoração, controle e coordenação dos recursos. O Gerenciamento de sistemas tem vários pontos de acesso através dos quais interage com outros processos. Um deles é com as pessoas e o software que requisitam o serviço do processo de gerenciamento do sistema. Pedidos e respostas de serviço passam através deste ponto de acesso para invocar uma ou mais facilidades do sistema de gerenciamento. Este ponto de acesso está dentro do ambiente local e não é sujeito a padronização. Outro ponto de acesso, entre o gerenciamento de sistemas e o gerenciamento do nível "N" provê facilidades para o sistema de gerenciamento:

- Ler, setar e efetuar ações com respeito aos vários valores, contadores e status dentro de um certo nível.
- Responder a consultas feitas por uma entidade de gerenciamento do nível "N" concernente a valores e funções.
- Ela provê facilidades para o gerenciamento do nível "N":
- Responder a ações feitas pelo gerenciamento de sistema.
- Requisitar informações da base de informações de gerenciamento.
- Solicitar que alguma informação seja colocada na base de informações de gerenciamento.

O ponto de acesso entre o gerenciamento de sistemas (SMAP) e a entidade de nível 7 (SMAE) é o principal meio de comunicação com os outros gerenciamentos de sistemas (SMAPs). Esse ponto de acesso é usado pelos serviços de informação de gerenciamento.

O **Gerenciamento do nível "N"** (LME) provê mecanismos para moni-

toração, controle e coordenação dos recursos do nível "N" usados para efetuar as atividades de comunicação dentro de um nível "N". As entidades de gerenciamento de nível "N" comunicam-se entre si para apoiar o controle de gerenciamento de recursos usados para atividades de comunicação entre seus respectivos sistemas abertos. A comunicação de gerenciamento de nível "N" é efetuada através de protocolos do gerenciamento de sistemas ou através do protocolo de gerenciamento do nível "N" ou ambos.

A **entidade de nível N** provê o conjunto de facilidades que controla e gerencia as comunicações com as entidades pares. Estas comunicações correm de acordo com os protocolos inerentes a cada nível. Alguns protocolos tem mecanismos para conduzir informações de gerenciamento em certos n-PDUs, tal como o protocolo X.25, nível 3, que pode ter no pacote de encerramento de conexão, informação sobre tarifação.

Figura 2.7 Protocolos usados no gerenciamento

2.4.2 Serviços de informação de gerenciamento

O serviço de informação de gerenciamento é o serviço de aplicação disponível ao processo de gerenciamento de sistema (SMAP) para permitir que ele consiga entrar em contato com processos de gerenciamento de outros sistemas abertos. Esse serviço apoia o SMAP provendo uma série de elementos de serviços a serem usados na comunicação com a sua entidade par (outro SMAP). Conforme referido na seção 2.4, além dos elementos de serviço comuns a qualquer aplicação, que são denominados CASE (Common Application Service Element), a SMAE prove elementos de serviço específicos da aplicação de gerência de rede, que constituem uma classe de SASE (Specific Application Service Element), denominada MISE-Management Information Service Element. Estes elementos de serviços também são, por sua vez sub-divididos em comuns e específicos. Os primeiros são referidos como CMISE-Common Management Informations Service e os segundos como SMISE-Specific Management Information Service Element. O CMISE prove meios para que os SMAPs possam enviar e receber informações concernentes à gerência da rede. Os seis elementos de serviço comuns (CMISE) [36], (figura 2.8) são classificados em 3 categorias:

I. Notificação de eventos

I.1 **Relato de evento (EVENT-REPORT)**: usado pela SMAE para reportar algum evento não solicitado, ocorrido com um recurso, para outra SMAE em outro sistema aberto

I.2 **Relato de evento confirmado (EVENT-REPORT-CONFIRMED)**: idêntico ao anterior mas aguardando uma resposta

II. Transferência de informação

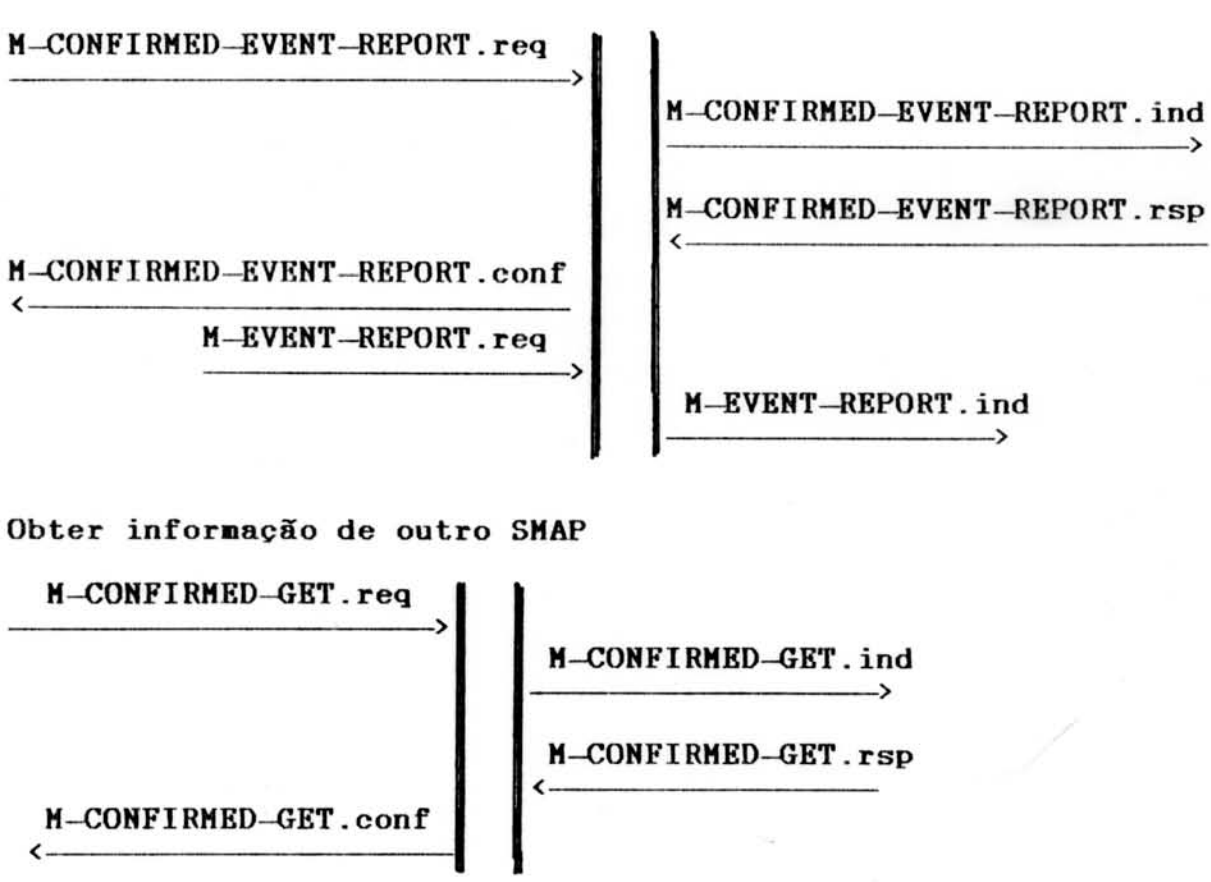
II.1 **Obter (GET)**: usado por um SMAP para requisitar transferência de informação de gerenciamento de outra SMAE em outro sistema aberto

III.Controle

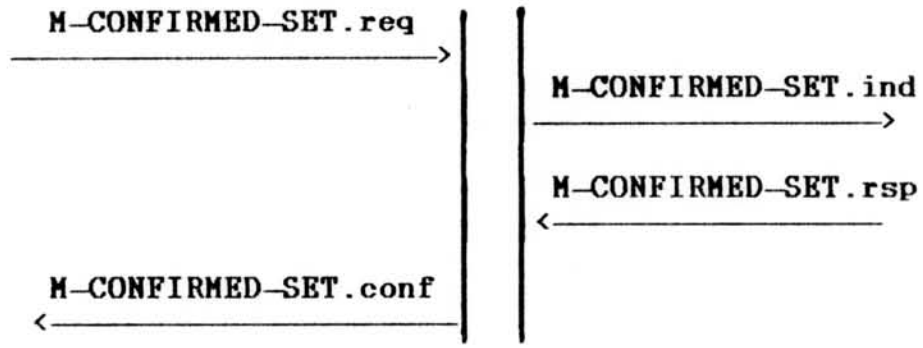
III.1 **Setar (SET)**:provê a capacidade de requisitar à outra SMAE, em outro sistema aberto, que sejam alterados os valores de atributos naquele sistema. Este é o mecanismo básico para exercer controle sobre os recursos.

III.2 **Ação (ACTION)**: prove a capacidade de requerer a uma SMAE em outro sistema aberto a execução de alguma operação. Deve ser usado somente quando não for possível conseguir o resultado desejado somente com a manipulação de atributos definida em III.1.

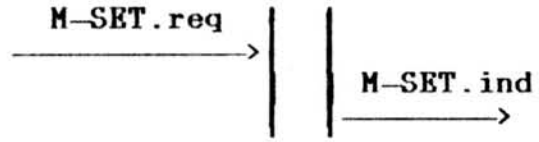
Figura 2.8 Elementos de serviço comum



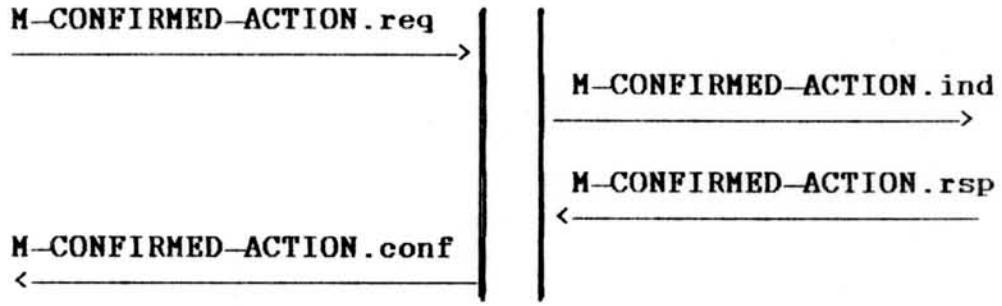
Setar informação em outro sistema aberto de forma confirmada



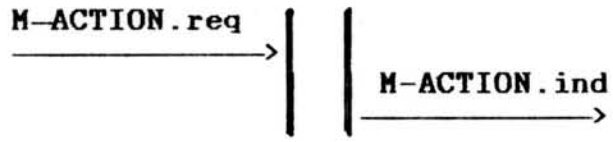
Setar informação em outro sistema aberto de forma não confirmada



Agir sobre outro sistema de forma confirmada



Agir sobre outro sistema de forma não confirmada



2.4.3 Elementos de serviço específicos

Os elementos de serviço específicos foram agregados nas áreas anteriormente referidas:

gerenciamento de problemas

gerenciamento de contabilização

gerenciamento de configuração

gerenciamento de performance

gerenciamento de segurança

Os elementos de serviço de **gerenciamento de problemas** são usados pelo administrador do sistema (pessoa ou software) para assisti-lo na tomada de decisões concernentes à operação anormal de um sistema aberto e para encaminhar a ação corretiva sobre o recurso com problemas. A informação intercambiada mediante o uso destes elementos de serviço é usada para três atividades primárias:

Detecção de problemas

Diagnóstico de problemas

Correção de problemas

A fim de prover informação para essas atividades gerais são definidas certas facilidades para o serviço de gerenciamento de problemas que são:

-Relato espontâneo de erros: com este elemento de serviço o SMAP pode enviar relatos de erros tão logo o mecanismo de comunicação permita.

-Obtenção de cumulativa de erros: com esta facilidade o SMAP pode, periodicamente, solicitar informação de um contador de erros de um gerenciador de sistemas par a fim de determinar o histórico recente de eventos. Ele pode então tomar decisões a respeito de que ação posterior é necessária desencadear.

-Relatório de notificação de limiar de erros: essa facilidade

permite que o SMAP possa enviar relatórios sobre ultrapassagem de limiares. O receptor de um relatório deste tipo pode tomar decisões a respeito de que ações são necessárias.

-Relato de eventos: essa facilidade permite ao gerenciador de sistemas continuamente monitorar eventos ocorridos num sistema par.

-Monitoração contínua: prevê mecanismos para que um SMAP envie a outro todos os relatos de eventos ocorridos com algum recurso.

- Teste de confiabilidade: essa facilidade permite que um SMAP direcione outro SMAP no sentido de efetuar um teste sobre algum recurso para determinar se ele é capaz de efetuar seu serviço.

-Teste diagnóstico: essa facilidade permite que um SMAP direcione outro SMAP para efetuar um teste sobre um recurso para auxiliá-lo no diagnóstico sobre um problema.

-Rastreamento de tragetoria de comunicação: essa facilidade provê um mecanismo para um SMAP iniciar um rastreamento de tragetória para uma série de outros SMAPs para testar as facilidades de comunicação disponíveis. Cada SMAP, após o M-TRACE-PDU (Protocol Data Unit) encaminha-lo-á ao próximo SMAP indicado e retornará uma confirmação de recebimento ao originador. Assim, o originador pode acompanhar e registrar de quem recebe as confirmações para auxiliar a determinar onde podem existir problemas nos serviços de comunicação sendo usados.

-Registro de Problemas: provê um mecanismo para um SMAP reportar atividades sobre um particular problema, para outros SMAPs, para que possam cooperar no gerenciamento total de problemas nos recursos OSI.

-Reinicializador de recursos: essa facilidade permite a um SMAP direcionar outro SMAP a reinicializar um recurso fazendo-o retornar a um estado conhecido. Essa facilidade é usada quando um recurso entra num estado de problema irrecuperável e o único meio de

reparà-lo é recolocá-lo num estado inicial conhecido.

O objetivo do serviço de **gerenciamento de contabilização** é permitir ao usuario gerenciar os custos encorridos e decorrentes de tarifas pelo uso dos recursos de comunicação. Geralmente existem dois tipos de tarifação. Tarifas fixas e tarifas por uso. As tarifas fixas normalmente nada tem a ver com número de unidades de dados transmitidos ou com a duração da conexão enquanto que as tarifas por uso, além de dependerem do volume de dados transmitidos, podem ser influenciadas por:

- Hora do dia
- Duração de uma conexão
- Número de unidades de dados contabilizáveis transmitidos
- Qualidade do serviço requerido
- Uso de serviço de valor adicionado

Até o momento existe somente duas facilidades definidas para gerenciamento de contabilização: relato de informação de contabilização e leitura de informação de contabilização. Com a primeira um SMAP efetua um relatório não solicitado de informação de contabilização para um outro SMAP. Com o segundo serviço o SMAP solicita informação de contabilização de uma entidade par. Para ambas as facilidades a informação de contabilização é classificada em dois níveis:

- Informação de contabilização ao nível de rede e
- Informação de contabilização a nível de aplicação.

A primeira é concernente com os custos incididos pelo uso dos recursos do nível de rede e inferiores. Essa informação é subdividida nos seguintes elementos: número de pacotes recebidos e enviados, número de pacotes de controle recebidos e enviados, número de caracteres recebidos ou enviados, número de unidades de contabilização (segmentos) enviados ou recebidos, hora de início e fim.

O serviço de **gerenciamento de configuração** é a definição, coleta, monitoração, gerenciamento e uso de dados de configuração. Os dados de configuração são definidos como qualquer informação sobre os recursos OSI que são necessários para gerenciar o sistema aberto ou a rede de sistemas abertos. Os dados que as funções de gerenciamento de configuração acessam, através de mensagens e parâmetros, consistem de dados administrativos e dados de tempo real. Alguns exemplos de dados de tempo real são: status dos nodos e lista de parâmetros. Dados administrativos são: número da versão de hardware e software, localização física de um nó, tipo de um nó, conjunto de protocolos suportados por cada nível.

O serviço de **gerenciamento de performance** define facilidades necessárias que podem ser subdivididas em:

-Coleta de estatísticas do sistema

-Controle da coleta de estatísticas do sistema

-Armazenamento das estatísticas do sistema e histórico de estatística

-Análise das estatísticas do sistema

-Apresentação das estatísticas do sistema

Os tipos de informação usados são essencialmente contadores. Por exemplo, a nível de rede, deve ser mantida informação tal como valores de contadores que indicam quão frequentemente o sistema envia/recebe solicitações de/para as redes. Para o nível de rede, o número de retransmissões, o número de conexões etc provê informações para determinar o throughput ou gargalos existentes na rede.

O serviço de **gerenciamento de segurança** define o conjunto de regras práticas e procedimentos necessários suficientes para provisão de serviços seguros. Isto implica em prover continuamente proteção ao sistema contra acesso não autorizado ou modificações não autorizadas. Os mecanismos envolvidos são:

- gerenciamento de autenticação,

- gerenciamento de controle de acesso,

- gerenciamento de chaves, contabilização e segurança.

A fim de prover estes mecanismos estão sendo consideradas uma série de facilidades tais como:

- distribuição de passwords: facilidade que permite a um SMAP transmitir ao outro dados de forma não especificada que devem ser usados para autenticação entre os pares.

- início, coleta e término de informação de rastreamento de auditoria.

- relato de eventos de segurança; O SMAP relata de tentativas aparentes de violar o sistema de segurança do sistema aberto.

- controle de segurança: o SMAP fica capacitado a alterar ou deletar informações de segurança tal como dados de controle de acesso e dados de capacidades de outro sistema aberto.

- verificação com terceiros: essa facilidade permite que um SMAP solicite a outro que verifique se a possibilidade de comunicação com um terceiro é possível.

– distribuição de chaves: facilidade que permite a transmissão de uma chave para ser usada para as operações de criptografia e decodificação.

3. ASPECTOS GERENCIAVEIS NA ARQUITETURA OSI

Pretendendo orientar o estudo ora realizado à gerência de problemas, o primeiro passo empreendido, consistiu em pesquisar mecanismos já disponíveis na arquitetura OSI, capazes de prover informações sobre eventos anormais. Neste sentido, foi realizado um estudo exaustivo sobre toda a documentação existente sobre cada camada da arquitetura OSI. Buscou-se levantar os mecanismos previstos para detectar problemas.

Isto implicou em uma análise dos protocolos padronizados para cada nível, inspecionando cada tipo de nPDU (Protocol Data Unit) previsto no que tange a campos e códigos definidos para relatar erros ou problemas. Os diagramas de transição de estado dos protocolos também foram analisados com este mesmo objetivo: buscar situações anormais previstas na definição do modelo OSI e seus protocolos.

O resultado desta análise preliminar, apresentado em [91], permitiu a elaboração da tabela 3.1. Posteriormente, um estágio mais elaborado da pesquisa foi levado a discussão no grupo de trabalho WG 6.6-Gerência de Rede do IFIP-TC6 (Comite Técnico de Comissão de Dados da International Federation for Information Processing" e apresentado num congresso organizado por este grupo [92]. A partir deste estudo ficou evidente que a camada de transporte é quem oferece maior quantidade de mecanismos para detecção de erros e, em consequência, foi decidido que o projeto de um sistema inteligente para apoio à gerência de rede poderia começar com esta camada.

Tabela 3.1: PROBLEMAS DETECTAVEIS NAS CAMADAS OSI

PROBLEMAS	Níveis Físico	Enlace	Rede	Transporte	Sessão	Apresentação	Aplicação
Rejeição sem explicação		X	X	X	X	X	
Desconexão normal	X	X	X	X	X		X
Endereço desconhecido			X	X	X		X
Endereço inatingível	X	X	X	X	X		
Fora de sequência		X	X	X	X		
Congestionamento			X	X	X		
Versão do protocolo inaceitável		X		X	X	X	
Classe de serviço não disponivel			X	X	X	X	
Capacidade não disponivel		X	X	X	X	X	X
Formato inválido		X	X	X	X	X	X
Mensagem muito longa		X		X		X	
Erro de transmissão		X		X			
Erro de protocolo		X	X	X	X		
Acesso barrado (password)			X	X	X		X

Esta escolha pode ser justificada não só pelo fato de ser a camada de transporte a que já tem previsão do maior número de mecanismos de detecção de problemas mas também por ser a que tem a incumbência de assegurar o transporte confiável dos dados, isolando as aplicações distribuidas das idiossincrasias possivelmente existentes na operação das camadas inferiores da arquitetura OSI (níveis 1 a 3) que estejam sendo usadas. Tambem é verdade que a camada de transporte é a primeira camada a implementar um controle fim-a-fim, em que os sistemas finais envolvidos na comunicação tem controle total sobre sua operação. Nas camadas inferiores existe informação de gerenciamento, emitida pelos sistema intermediários, usados na comunicação, a qual não chega ao conhecimento dos siste-

mas fim que os utilizam, uma vez que tais informações podem ficar confinadas ao sistema de gerência da própria sub-rede. Um exemplo de gerência em separado de sub-rede é descrito na seção 2.3 e tal situação é bastante comum nas redes de computadores heterogêneas em que é utilizado um sub-sistema de comunição provido por fornecedor diferente daquele que produz os computadores usuários da sub-rede.

Todos estes argumentos reforçam a propriedade da seleção inicial do nível 4 como o mais adequado para apoiar a implantação de um sistema inteligente de gerência de rede orientado à gerência de problemas.

3.1 Detecção de problemas na entidade de transporte

O serviço de transporte provê transferência de dados de forma transparente entre seus usuários, os quais, não se preocupam com a forma pela qual são usados os meios de comunicação que propiciarão o transporte da informação. Cabe ao serviço de transporte selecionar e otimizar o emprego dos recursos de comunicação disponíveis, para proporcionar a qualidade de serviço exigida [33].

A qualidade de serviço é especificada por meio de seleção de valores para os parâmetros de qualidade de serviço, que representam características tais como vazão, atraso de trânsito, taxa de erro residual e probabilidade de falhas. O serviço de transporte deve prover a qualidade de serviço, independentemente da qualidade do serviço de rede disponível, isto é, deve complementar o serviço de rede de molde a garantir os parâmetros de qualidade requeridos. Assim, por exemplo, se é solicitado um atraso de trânsito muito baixo mas o nível de rede tarda muito a autorizar o envio de novos pacotes (mantendo a janela fechada durante muito tempo), a entida-

de de transporte não terá condições de oferecer o serviço com aquela qualidade, a menos que use de artifícios. Um destes artifícios poderia ser dispersar a conexão de transporte em várias conexões de rede e enviar os DT-TPDUs em paralelo, por diferentes conexões de rede. Contudo, se a entidade de transporte não toma esta iniciativa, tudo o que poderá fazer, quando receber a solicitação de um serviço com atraso de trânsito baixo, é solicitar ao nível de rede que garanta um certo *throughput*. Isto é uma possibilidade técnicamente viável, pois está previsto na recomendação X.25 que um ETD solicite, na abertura de um circuito virtual, esta facilidade. Porém, certas redes não aceitam tais solicitações, e este é o caso da rede brasileira, a RENPAC.

3.1.1 Parâmetros concernentes à qualidade do serviço prestado pela entidade de transporte

Conforme referido na seção anterior, a entidade provedora do serviço de transporte tem certos parâmetros que devem pautar sua atuação. A QS-Qualidade do serviço de transporte a ser prestado pode ter determinados limiares exigidos pela entidade usuária do serviço de transporte, concernente à uma conexão de transporte. Os requisitos de qualidade de serviço exigido para a conexão de transporte sendo solicitada são apresentados como um conjunto de parâmetros na primitiva de serviço de pedido T-CONNECT. Tais parâmetros podem ser classificados segundo o momento ou fase em que são verificados (fase de estabelecimento de conexão, fase de transferência de dados e fase de liberação de conexão), conforme [97 e 98], sendo adicionalmente sub-divididos em duas categorias: os que se relacionam à velocidade da transmissão e os que se relacionam à precisão e integridade com que a mesma ocorre, conforme

apresentado na tabela 3.2.

Tabela 3.2: CRITÉRIO DE PERFORMANCE

FASE	VELOCIDADE	PRECISÃO/INTEGRIDADE
Estabelec.	retardo p/ estabelecimento	probabilidade de falha no estabelecimento
Transfer.	throughput	taxa de erros residual (alteração,duplicação,perda)
	retardo de trânsito	resistência da conexão de transporte
		probabilidade de falha na transferência
Liberação	retardo	probabilidade de falha na liberação

O **retardo no estabelecimento da conexão** é o máximo retardo aceitável entre um pedido T-CONNECT e a confirmação T-CONNECT correspondente. Destaque-se que este retardo inclui um componente dependente do usuário do serviço de transporte respondedor. Portanto, se este tempo for excedido, pode não ser culpa das entidades provedoras do serviço de transporte (iniciadora e respondedora). Em vista disso, se a entidade provedora do serviço de transporte iniciadora detectar um retardo inaceitável e sinalizar isto para a entidade de gerenciamento do nível de transporte, haverá ainda a necessidade de buscar, junto à entidade de transporte respondedora, informação sobre o tempo decorrido entre a indicação T-CONECT e a resposta T-CONNECT. De posse desta informação será possível deduzir se a culpa pelo não estabelecimento da conexão de transporte está com alguma das entidades provedoras do serviço de transporte ou com a entidade usuário do serviço de transporte respondedor.

A **probabilidade de falha no estabelecimento** da conexão de transporte é a razão entre o total de falhas no estabelecimento de conexões de transporte e o total de tentativas, no intervalo de tempo medido. Uma falha no estabelecimento de uma conexão de transporte ocorre quando a conexão de transporte não é estabeleci-

da no período de tempo máximo aceitável, quando ela foi recusada ou foi estabelecida de forma errada. Se o não estabelecimento ocorre por culpa do usuário do serviço de transporte a tentativa é excluida do cálculo de probabilidade de falha no estabelecimento.

O **throughput** é definido em termos de uma sequência de ao menos 2 TSDUs (Transporte Service Data Units) transferidos com sucesso. Isto implica em transferir os TSDUs ao destinatário correto, sem erros e em ordem. Computando o throughput para uma sequência de n TSDUs (onde n é maior ou igual a 2), o throughput é então definido em termos do número de octetos de dados contidos nos últimos n-1 TSDUs dividido pelo tempo entre o primeiro e o último pedido ou indicação T-DATA (dependendo de quem está medindo). O throughput é especificado separadamente para cada direção de transferência numa conexão de transporte. Existe um valor médio e um valor máximo para cada uma delas. O valor máximo representa a máxima taxa em que o provedor do serviço de transporte pode aceitar TSDUs, sem retardos ocasionados pelo usuário emissor ou retardos ocasionados pelo controle de fluxo aplicado pelo usuário receptor.

O **retardo de trânsito** é o tempo decorrido entre um pedido de transmissão de dados (T-DATA) e a indicação T-DATA correspondente, quando a transferência for bem sucedida (sem erros e em sequência correta). Quando este retardo é aumentado em decorrência do uso de controle de fluxo,tais ocorrências devem ser excluidas do cálculo.

A **taxa de erros residual** é a razão entre o total de TSDUs incorretos, perdidos ou duplicados e o total de TSDUs transferidos, durante um período de medição.

A **probabilidade de falha na transferência** é a razão entre o total de falhas na transferência e o total de transferências observadas num intervalo de tempo de medição. Este tempo de medição normalmente coincidirá com o tempo de duração da conexão.

da no período de tempo máximo aceitável, quando ela foi recusada ou foi estabelecida de forma errada. Se o não estabelecimento ocorre por culpa do usuário do serviço de transporte a tentativa é excluida do cálculo de probabilidade de falha no estabelecimento.

O **throughput** é definido em termos de uma sequência de ao menos 2 TSDUs (Transporte Service Data Units) transferidos com sucesso. Isto implica em transferir os TSDUs ao destinatário correto, sem erros e em ordem. Computando o throughput para uma sequência de n TSDUs (onde n é maior ou igual a 2), o throughput é então definido em termos do número de octetos de dados contidos nos últimos n-1 TSDUs dividido pelo tempo entre o primeiro e o último pedido ou indicação T-DATA (dependendo de quem está medindo). O throughput é especificado separadamente para cada direção de transferência numa conexão de transporte. Existe um valor médio e um valor máximo para cada uma delas. O valor máximo representa a máxima taxa em que o provedor do serviço de transporte pode aceitar TSDUs, sem retardos ocasionados pelo usuário emissor ou retardos ocasionados pelo controle de fluxo aplicado pelo usuário receptor.

O **retardo de trânsito** é o tempo decorrido entre um pedido de conexão (T-CONNECT) e a indicação T-DATA correspondente, quando a transferência for bem sucedida (sem erros e em sequência correta). Quando este retardo é aumentado em decorrência do uso de controle de fluxo, tais ocorrências devem ser excluidas do cálculo.

A **taxa de erros residual** é a razão entre o total de TSDUs incorretos, perdidos ou duplicados e o total de TSDUs transferidos, durante um período de medição.

A **probabilidade de falha na transferência** é a razão entre o total de falhas na transferência e o total de transferências observadas num intervalo de tempo de medição. Este tempo de medição normalmente coincidirá com o tempo de duração da conexão.

O **retardo de liberação** da conexão de transporte é o máximo retardo aceitável entre um pedido T-DISCONNECT iniciado pelo usuário e a liberação da conexão no usuário par (sinalizada por uma indicação T-DISCONNECT). Esta medida não se aplica quando a liberação é iniciada pelo provedor do serviço de transporte.

A **probabilidade de falha na liberação** é a razão entre o total de pedidos de liberação que resultaram em falha e o total de pedidos de falha ocorridos no período de tempo de medição. Este indicativo é especificado independentemente para cada usuário. A falha ocorre quando o usuário que solicitou a liberação não recebe a resposta num período de tempo aceitável.

A **resistência** da conexão de transporte é a probabilidade de que um provedor do serviço de transporte inicie uma liberação da conexão de transporte (acionando uma indicação T-DISCONNECT sem um pedido T-DISCONNECT previo), num dado intervalo de tempo.

3.1.2 Erros detectados pela entidade de transporte

Os erros detectados na fase de transferência ou liberação da conexão são sinalizados ao usuário do serviço de transporte, por uma indicação T-DISCONNECT. Quando ocorre uma indicação de liberação de conexão de transporte é obrigatório o uso do parâmetro **razão** que fornece indicação sobre a causa da liberação, a qual pode ser, segundo [98].

a) liberação solicitada pelo usuário remoto
b) liberação invocada pelo provedor do serviço de transporte:
    b.1 falta de recursos locais ou remotos
    b.2 qualidade de serviço inferior ao requerido
    b.3 comportamento errôneo do provedor do serviço de transporte

b.4 usuário do serviço de transporte chamado desconhecido

b.5 usuário do serviço de transporte chamado não disponível

b.6 razão desconhecida

Neste momento, a entidade de transporte também deverá passar à entidade de gerenciamento de transporte as informações necessárias para a atuação desta.

Por outro lado, quando é recebido um TPDU do tipo DR (Disconnect Request), o parâmetro RAZÃO define a causa para a desconexão da conexão de transporte e pode assumir um dos seguintes valores:

CODIGO EXPLICAÇÃO

usados nas classes 1 a 4

128+0 desconexão normal iniciada pela entidade de sessão

128+1 congestionamento na entidade de transporte remota no momento da solicitação de conexão

128+2 negociação da conexão falhou (isto é, classe proposta não aceita ou suportada)

128+3 detectada referência de origem duplicada para o mesmo par de NSAP-Network Service Access Point

128+4 referências descombinadas

128+5 erro de protocolo

128+6 não usado

128+7 overflow de referência

128+8 pedido de conexão recusado nesta conexão de rede

128+9 não usado

128+10 comprimento de cabeçalho ou parâmetro inválido

usados em todas as classes

0 razão não especificada

1 congestionamento no TSAP-Transport Service Access Point

2 entidade de sessão não conectada ao TSAP

3 endereço desconhecido

Também pode ocorrer um erro de protocolo durante qualquer das fases. Neste caso será enviado um TPDU-ER, que conterá um campo indicando o tipo de erro detectado. As causas previstas são:

CODIGO	EXPLICAÇÃO
0000 0000	razão não especificada
0000 0001	código de parâmetro inválido
0000 0010	tipo de TPDU inválido
0000 0011	tipo de parâmetro inválido

Este TPDU também contem os bits do TPDU rejeitado até o octeto que causou a rejeição (este parâmetro é obrigatório na classe 0).

As interações da entidade de transporte com os níveis inferiores e superiores, para as classes 0 até 4 são descritas formalmente nas tabelas, contidas nas normas [98].

Nestas tabelas são indicadas todas as transições normais e, em caso de qualquer evento anormal, a entidade gerenciadora do nível de transporte seria informada sobre o ocorrido e isto seria processado localmente (simples tabulação, comparação com limiares, etc...). Em certos casos o SMAP-System Management Application Processo seria também notificado. Na seção seguinte são apresentados os principais problemas que uma análise do funcionamento da entidade de transporte, permitiu antecipar.

3.1.3 Problemas na entidade de transporte

Os problemas na atuação na entidade de transporte podem ser subdivididos como se segue:

congestionamento

- A entidade de transporte não pode abrir mais conexão de transporte e vai reagir com uma indicação T-DISCONNECT

pedido T-CONNECT que receber enquanto estiver nesta condição. Também vai responder ao cada CR (Connect Request) que receber com um DR (Disconnect Request)

- A entidade de transporte recebe um TSDU e não consegue enviar porque a janela está fechada e está expirado o temporizador de envio de dados.

- A entidade de transporte deseja abrir mais conexões de rede e não consegue porque existe um limite no número de conexões de rede que podem ser abertas

bug de software

- São sinalizados erros de protocolos (TPDU-ER)

- Ocorre um número excessivo de retransmissões

- O usuário não é contactável (problema com o endereço fornecido pelo usuário iniciador da conexão)

rede não confiável

- Número excessivo de perdas de TPDUs (detectados pelo controle de sequência)

- Conexão de rede fechada ou "resetada" intempestivamente sendo isto sinalizado à entidade de transporte (N-RESET, N-DISCONNECT, N-RESTART)

outros

- Desconexão por "inactivity timer"; significa que a entidade de sessão possivelmente tenha caído sem desconectar a conexão de transporte.

3.2 Ações corretivas recomendáveis

Quando são detectados problemas pode ser requerida a atenção imediata do gerente da rede ou pode haver uma simples anotação. O

lização de funcionamento de algum componente crucial. Assim, ser ou não crucial é um atributo que deve fazer parte da definição dos objetos gerenciados. O segundo tipo de problemas provoca apenas uma anotação e os objetos gerenciados anotados neste sentido serão relacionados ao gerente da rede apenas mediante expressa solicitação sua neste sentido.

Quando o gerente da rede solicita a indicação dos componentes em estado semi-crítico, sua apresentação deve poder ser feita por ordem de importância (em função de quantos outros componentes dependem daquele para operar), por ordem alfabética do nome do componente ou ainda por localização física (para otimizar o deslocamento das pessoas encarregadas do teste ou manutenção.

Problemas que requerem a atenção imediata do gerente da rede seriam:

-Quando for detectada paralização de funcionamento de algum componente crucial (tal como a entidade de transporte inteira); portanto, na definição dos aspectos a gerenciar é importante inserir algum indicação que auxilie a determinar o grau de importância de cada componente;

-Quando for detectada uma degradação de performance com tendência a aumentar em um dos componentes. Neste caso, deve ser indicado, por extrapolação da evolução do MTBF (Mean Time Between Failures), o instante provável em que a degradação se tornar intolerável. Para isto, deve haver uma maneira de calcular esta probabilidade a partir de uma série histórica de problemas.

Problemas que devem apenas ser anotados seriam:

-os que não impedem o funcionamento de algum segmento crítico da rede mas apenas atrapalham seu funcionamento (como uma taxa de retransmissões acima do especificado para uma da-

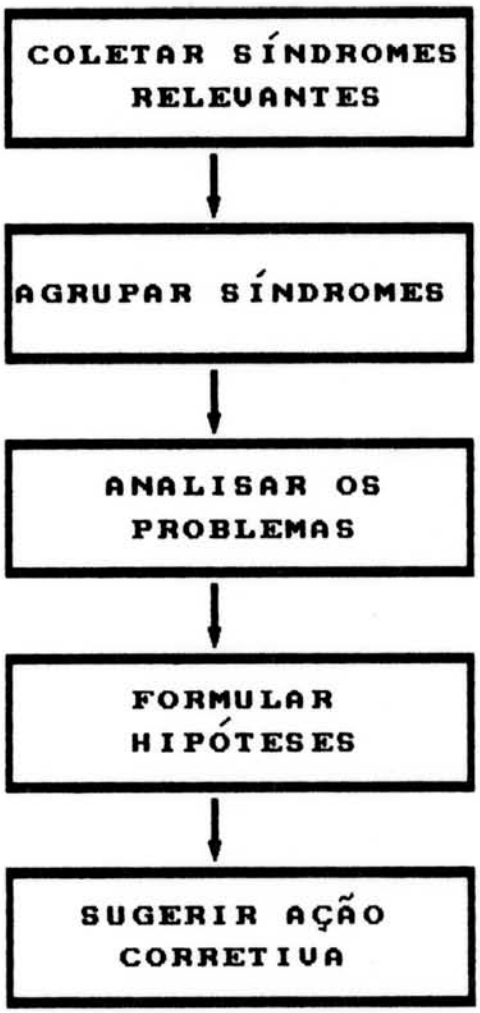
da conexão de transporte e que leva ao seu encerramento);

-problemas intermitentes (tal como recusa no estabelecimento de uma conexão por congestionamento) que não tendem à degradação crítica num futuro próximo.

O processo de análise dos problemas, passou pelos passos indicados na figura 3.1. Os problemas relacionados na seção anterior foram agrupados e analisados. Diversas hipóteses sobre suas causas foram relacionadas e avaliadas, mediante cuidadosa inspeção nos padrões que definem o comportamento da entidade de transporte; procurava-se determinar a ação cabível em cada caso.

O resultado foi um conjunto de recomendações que estão apresentadas a seguir.

Figura 3.1: Os passos do estudo do problema



PROBLEMA --> AÇAO RECOMENDAVEL E SEUS CONDICIONANTES

congestionamento

--> Se a taxa de rejeição de pedidos de estabelecimento excede o limiar significa que esta entidade de transporte está sub-dimensionada para a quantidade de serviço que está sendo solicitada a prover. O administrador do sistema deve ser avisado para redimensionar a entidade de transporte, permitindo um maior número de conexões simultaneas. Contudo, talves ele queira saber se a causa deste tipo de congestionamento deriva de demanda genérica que cresceu vegetativamente ou se algum serviço em particular está com uma demanda muito alta. Neste último caso ele deveria considerar a hipótese de mover o serviço para outra máquina para solucionar o problema da entidade de transporte em estudo. É importante destacar que, embora ciente do problema, o gerente da rede pode não ter condições imediatas de solucioná-lo. Neste caso, uma solução paliativa cabível seria a alteração do limiar da taxa de rejeição de pedidos de conexões.

--> Se a taxa de rejeição de pedidos de estabelecimento evidencia um crescimento, sendo previsível que o limiar vai ser ultrapassado a curto prazo, pode-se concluir que a demanda está em crescimento vegetativo. O gerente deve ser avisado a respeito, bem como sobre o prazo em que isto provavelmente acontecerá. O mesmo estudo indicado no caso anterior deverá ser realizado pelo gerente da rede neste caso mas com menos urgência, uma vez que o problema real ainda está por acontecer.

-->Se a entidade de transporte está encontrando, numa dada conexão, a janela fechada com muita frequência e por tempo excessivo, isto indica que a entidade par não está conseguindo receber os dados na velocidade adequada. Se o problema for localizado (ocorrer numa só conexão de transporte) a culpa será da entidade par e um

alerta não crítico poderá ser gerado para o gerente da rede para notificá-lo a respeito. Se o sistema no qual se encontra a entidade par pertencer ao seu domínio administrativo ele poderá tomar as providências cabíveis (aumentar a capacidade daquele sistema) se for o caso (se o problema não for esporádico). Se o problema ocorrer em todas as conexões de transporte a culpa provavelmente será do parâmetro **Número Máximo de Retransmissões** que deverá ser alterado para um valor maior, pois o estabelecido não está sendo suficiente para dar tempo às entidades pares com quem se comunica reagirem.

-->Se o gargalo está sendo na rede que está rejeitando novos pedidos de abertura de conexões a solução indicada consiste em aumentar o número de canais lógicos passíveis de serem abertos simultaneamente. Isto pode implicar em alterar contratos de prestação de serviços de rede, quando o serviço é externo à organização, ou pode determinar a necessidade de aumentar também o número de conexões físicas com a rede (ou sua velocidade) para não saturar o enlace utilizado com um número excessivo de conexões de rede.

bug de software

--> Os TPDU-ER vão provavelmente ocorrer entre duas entidades de transporte que estiverem interoperando pela primeira vez ou há pouco tempo. Uma delas poderá usar alguma funcionalidade que a outra não tem implementada e não está preparada para reagir àquele estímulo. Assim, será um problema que terá que ser resolvido com manutenção de software básico em uma das entidades de transporte e provavelmente não será resolvido logo. Portanto, uma vez que o problema tenha sido sinalizado e o gerente da rede notificado, não deverá haver mais notificações de cada ocorrência, se o caso for com a mesma entidade par. Isto implica em dispor de informa-

ções sobre ocorrências anteriores deste tipo de anormalidade.

--> Um número excessivo de retransmissões pode também ocorrer em decorrência de erros de software, especialmente em implementações novas. Os cálculos de tempo de espera podem não estar sendo bem feitos pela entidade de transporte e causar pedido de retransmissão prematuro. Esta hipótese teria que ser confirmada pelo recebimento sistemático da resposta esperada num tempo curto após a sinalização do pedido de retransmissão e pela vinda de uma duplicata desta resposta algum tempo depois. Em se configurando esta situação, bastaria aumentar o time-out que determinou o pedido de retransmissão que está ocorrendo excessivamente.

-->Se uma conexão não pode ser estabelecida devido a problemas de endereçamento, isto decorre de erro na especificação dos endereços o que, via de regra, tem que ser corrigido com alteração de parâmetros e/ou endereços contidos em tabelas ou diretórios na entidade de transporte ou na aplicação que está usando a conexão de transporte em que o erro foi evidenciado. Por isso, o gerente da rede ao receber notificação sobre o problema deve também receber informação adicional, informando as entidades de mais alto nível envolvidas na tentativa de contacto.

serviço de rede não confiável

-->Se TPDUs de dados estão sendo perdidos pelo serviço de rede, isto implica em que a rede sendo usada é do tipo C, conforme classificação contida na especificação do serviço de transporte, o que quer dizer que a rede tem um número excessivo de erros não detectados. Neste caso, em, primeiro lugar há que investigar a causa do número excessivo de erros, analisando os níveis inferiores. Se não for possível melhorar ou trocar o serviço de rede e a aplicação requerer um serviço de transporte confiável, uma solução consiste em usar a classe 4 de transporte com checksum para aumentar o grau

de confiabilidade do serviço de transporte.

-->Se os problemas do serviço de rede estiverem ligados a erros sinalizados (mediante o uso dos pacotes RESET e RESTART num serviço X.25, por exemplo) a ação requerida também implica em analisar o serviço de rede para tentar descobrir a causa. Se não houver solução para a precariedade do serviço de rede deve ser recomendado o uso das unidades funcionais do serviço de sessão que permitam a sincronização e ressincronização, com marcação de pontos de sincronização maior (os que requerem confirmação) para que os dados transmitidos e confirmados não precisem ser retransmitidos devido a queda na conexão de rede. Numa situação como esta, ao ser reiniciada a transmissão teriam que ser retransmitidos apenas os dados que tivessem sido transmitidos após o último ponto de sincronização. Em função do que foi aqui exposto, percebe-se que para resolver este problema seria preciso dispor de informações sobre os níveis vizinhos aos de transporte.

outros

-->Se a conexão de transporte ficou aberta sem uso até o "inactivity timer" ocorrer, implica em que a sessão que tinha sido estabelecida sobre aquela conexão de transporte provavelmente já não está sendo usada porque caiu ou o parâmetro que define este timeout está muito baixo, para os tempos de reação da aplicação usando o serviço. Cabe, neste último caso, sugerir o aumento do valor deste parâmetro para evitar a ocorrência deste problema uma vez que causará incômodo ao processo usuário que terá que reiniciar a comunicação.

3.3 Necessidade de automatizar as reacões

As ações corretivas recomendadas na sessão anterior implicam frequentemente em análise do contexto requerendo um bom conhecimento dos protocolos usados nas diversas camadas da arquitetura OSI. É pouco provável que se consiga ter pessoal com suficiente formação para ficar atendendo todas as notificações que tendem a ocorrer com maior frequência quando há alterações na rede ou quando a mesma se encontra sob tráfego mais intenso. A manutenção da operação eficiente e confiável da rede, inclusive nestes momentos críticos, torna-se difícil se for apoiada apenas pelas ferramentas tradicionais que se resumem a apresentar a informação concernente aos problemas para que o operador da rede tome as decisões apropriadas.

A complexidade e diversidade dos alertas que podem ser canalizados para a console do operador da rede e, num escalonamento do problema, para o gerente da rede, fez com que tivesem que ser buscadas soluções que permitissem multiplicar e disseminar o conhecimento de uns poucos especialistas, capazes de resolver problemas específicos nas diversas áreas envolvidas com a operação da rede. Tal conhecimento precisa ser tornado acessível em tempo hábil e nos locais onde é requerido, independente da disponibilidade das pessoas que o detém. Como a camada de transporte integra aquela parcela de software conhecida como software básico, somente projetistas de software de rede terão o conhecimento necessário para apoiar o processo de tomada de decisão concernente à maior parte dos problemas inerentes a este contexto. Tais especialistas talvez sejam raros até mesmo em países desenvolvidos mas seu conhecimento pode ser transferido, ao menos em parte, para sistemas especialistas para apoiar a equipe local de uma instalação a inferir pro-

blemas e suas causas, a partir de monitoração da rede, ou a partir da análise das notificações sobre eventos anormais que a própria rede pode gerar, conforme comentado na seção 1.2.

Conforme visto na seção anterior, em que foram relacionadas as ações recomendadas, muitas delas não implicam em resposta em tempo real pois há mecanismos previstos na própria definição da entidade de transporte, e nas demais entidades da arquitetura OSI, capazes de ocasionar a reação automática da entidade aos efeitos dos problemas detectados. O que se necessita é uma orientação para correção dos problemas no que concerne à sua causa ou real origem para que não fiquem ocorrendo reiteradamente, prejudicando o funcionamento da rede.

Inteligência Artificial e, mais especificamente, Sistemas Especialistas tem sido aplicados em alguns casos específicos de gerência de rede com algum sucesso, conforme relatado em [59], mas poucos trabalhos estão sendo realizados no gerenciamento de redes OSI. Assim, no presente trabalho, buscou-se a definição de uma proposta para solução do problema capaz de funcionar com o grau de generalidade necessário para poder ser utilizável em qualquer contexto.

A partir de experiência de utilização com alguns sistemas OSI projetados e implantados como parte da pesquisa realizada em apoio ao desenvolvimento do trabalho ora relatado [93] ou com a observação de sistemas OSI recebidos de outras fontes, como o sistema EAN desenvolvido na University of British Columbia [74], verificou-se que um sistema pode interoperar com outro nos moldes previstos na arquitetura OSI sem que qualquer mecanismo de gerência de rede esteja disponível. O problema daí decorrente é que quando ocorre um problema maior ou reincidente, o sistema simplesmente para de funcionar e é preciso reativa-lo mediante intervenção do operador do

sistema. Em função desta experiência, ficou patente que qualquer solução de gerência de rede que venha a ser implantada num sistema real deve conviver com a possibilidade de que alguns sistemas abertos não sejam capazes de emitir qualquer notificação de eventos anormais.

Para incluir tais sistemas no esquema de gerenciamento será necessário dispor de monitoração que capture e registre o tráfego entrante ou gerado em tais sistemas. Neste sentido, está sendo iniciado um projeto de monitor orientado ao contexto OSI [96] destinado a apoiar a observação do tráfego OSI, o qual poderá servir de base para o desenvolvimento de um processo que derive inferências sobre o comportamento de um sistema aberto, a partir da observação do tráfego, e sua análise por um sistema especialista orientado a este fim. O tráfego será registrado de varias maneiras: por auto-monitoração no próprio sistema, invocando facilidades de "trace" disponíveis; pelo uso de funções existentes nas placas de redes locais que permitem a operação em modo promíscuo (capturando todos os quadros transferidos na rede) ou mediante o uso de equipamentos de monitoração externos (analisadores de dados). Em qualquer dos casos, o resultado da monitoração é filtrado e transferido para um sistema independente onde será analisado pelo sistema especialista.

Para projetar um paradigma para um sistema de gerência de rede OSI apoiado por inteligência artificial, foi efetuada uma análise dos conceitos fundamentais desta área da ciência da computação, com vistas a determinar a metodologia mais apropriada para ser usada em sistemas como o que foi objeto da pesquisa ora sendo relatada. O resultado desta análise é apresentado a seguir, no capítulo 4.

4. A UTILIZAÇÃO DE INTELIGÊNCIA ARTIFICIAL NO SISTEMA DE GERENCIAMENTO DE REDES

Um dos maiores problemas no gerenciamento de redes de computadores é a falta de mão de obra qualificada para lidar com os eventos decorrentes da atividade normal de uma rede de computadores, especialmente aqueles decorrentes de problemas não triviais, isto é, aqueles relativos as camadas superiores (3 a 7 no modelo OSI). Os problemas inerentes aos niveis 1 e 2 são mais facilmente rastreáveis e suas causas elicitadas por equipamentos de teste usualmente disponiveis nos centros de operação das redes.

Tais equipamentos tem evoluido consideravelmente operando com graus crescentes de inteligência. De simples monitores de linha, capazes apenas de apresentar os caracteres que passam na linha em hexadecimal, evoluiram para sistemas capazes de reconhecer frames, isolar campos, identificar tipos de mensagens, propiciando ao operador informações mais elaboradas. Contudo, tais informações são, na maioria dos casos limitadas a eventos relacionados com entidades físicas, tais como um modem, um concentrador uma linha etc..., não provendo informações sobre eventos internos concernentes a entidades lógicas, tais como um programa de aplicação ou uma entidade de nível de sessão. Mesmo provendo tal tipo de informação, o volume de dados decorrentes torna-se tão elevado que métodos de processamento usuais, aplicados a tais dados, tais como tabulações ou medias, não são suficientes para apoiar adequadamente o processo de tomada de decisão inerente a atividade de gerência da rede.

Recentemente, o campo da Inteligência Artificial avançou a ponto de derivar aplicações práticas de seus resultados, tal como descrito em [24, 31, 51, 52, 61 e 73]. Na maior parte dos casos, isto envolve sistemas especialistas [7, 59, 65 e 66]. Trata-se de

sistemas que usam conhecimento para solução de problemas específicos a uma area, alcançando um alto nível de performance numa atividade que normalmente considerar-se-ia restrita a um especialista humano.

4.1 Conceitos relevantes da Inteligência Artificial

A Inteligência Artificial (IA) dá ao computador uma capacidade adicional de computação, propiciando condições para que este possa exibir um comportamento mais inteligente no manuseio de dados.

Um elemento chave em qualquer aplicação de inteligência artificial é o conhecimento. O conhecimento consiste de fatos, conceitos, teorias, procedimentos e relacionamento. Conhecimento é também informação que foi organizada e analisada de modo a torná-la compreensível e aplicável à solução de problemas e tomada de decisão.

Inteligência artificial é o software que permite à qualquer computador digital duplicar algumas funções do cérebro humano de uma forma limitada. Embora hardware especial esteja sendo construido [72 e 115], a maior parte do software de inteligência artificial roda em computadores de uso genérico, de grande porte, em minicomputadores [6] e até em computadores pessoais [85]. Os programas de inteligência artificial podem ser escritos em qualquer linguagem e tem sido escritos em assembler, BASIC, FORTRAN, PASCAL, C e FORTH. Contudo, tem havido desenvolvimento de linguagens de programação especiais para inteligência artificial, sendo as mais populares LISP e PROLOG [9, 23, 25, 63].

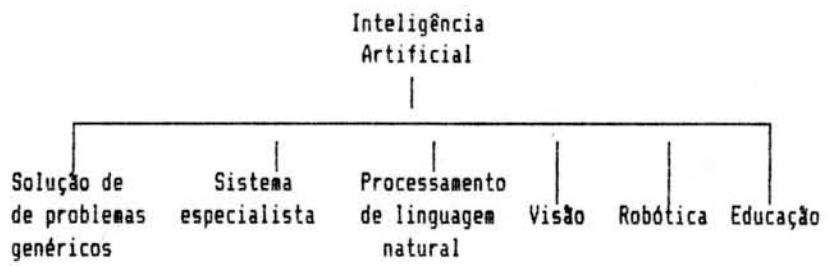
O software de IA não é baseado em um processo algorítmico mas sim em representação e manipulação simbólica. Em IA um símbolo é uma letra, palavra ou número que é usado para representar objetos,

processos e seus relacionamentos. Objetos podem ser pessoas, coisas, ideias, conceitos, eventos ou estabelecimento de fatos. Usando símbolos é possível criar uma base de conhecimento que estabelece fatos, conceitos e os relacionamentos entre eles. O processo é qualitativo e não quantitativo como num algoritmo computacional convencional típico. O conjunto de conhecimento disponível para o software de IA é armazenado numa base de conhecimento.

Uma vez que a base de conhecimento e suas associações lógicas tenha sido construida, é preciso desenvolver meios para usa-la e resolver problemas. O software de IA raciocina ou infere a partir desta base de conhecimento. As técnicas básicas usadas são busca e reconhecimento de padrões. Dada alguma informação inicial, o software de AI pesquisa a base de conhecimento procurando condições específicas ou padrões, buscando encontrar algo que satisfaça o critério para resolver o problema [50, 53].

Devido à grande flexibilidade do processo de inteligência artificial, o software de IA pode ser adequado a qualquer problema requerendo tais capacidade. A figura 4.1 mostra as maiores áreas de aplicação da inteligência artificial.

Figura 4.1 : Aplicações da inteligência artificial



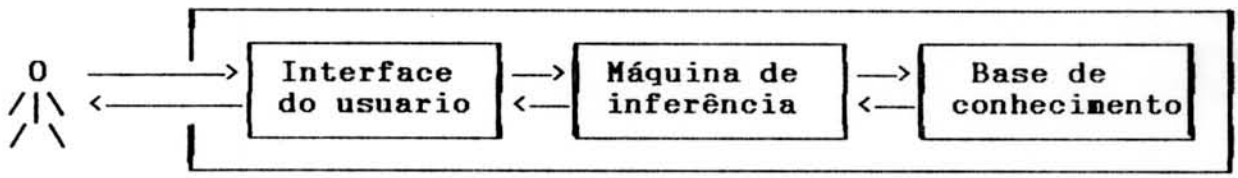
Solucionar problemas genéricos é uma aplicação muito propria da IA e pode ser aplicada em varios contextos. Houve algumas tentativas de criar um programa capaz de resolver problemas genéricos mas embora um limitado sucesso tenha sido obtido, a abordagem não era adequada para todos os tipos e soluções de problemas [5, 6,

103]. Em decorrência, praticamente todo o software de IA é orientado para uma area problema especificca, tal como prova de teoremas ou redução de fórmulas. Outra aplicação de solução de problemas é planejamento (estabelecer passo-a-passo, uma sequência de etapas para atingir algum resultado). Passando os passos ao programa de IA, juntamente com algum critério para julgá-los, um plano otimizado pode ser gerado. O programa tenta combinar os passos de varias maneiras, tentando otimizar o processo, conforme ilustrado em [1, 19 e 84].

O maior uso de inteligência artificial atualmente é em sistemas especialistas. São programas de IA que atuam como consultores inteligentes. Um sistema especialista permite que o conhecimento e a experiência de um ou mais especialistas sejam capturados e armazenados num computador, ficando disponíveis para qualquer um.

Um sistema especialista consiste de três componentes principais: uma base de conhecimento, uma máquina de inferência e um interface para o usuário (figura 4.2).

Figura 4.2 : Esquema geral de um sistema especialista

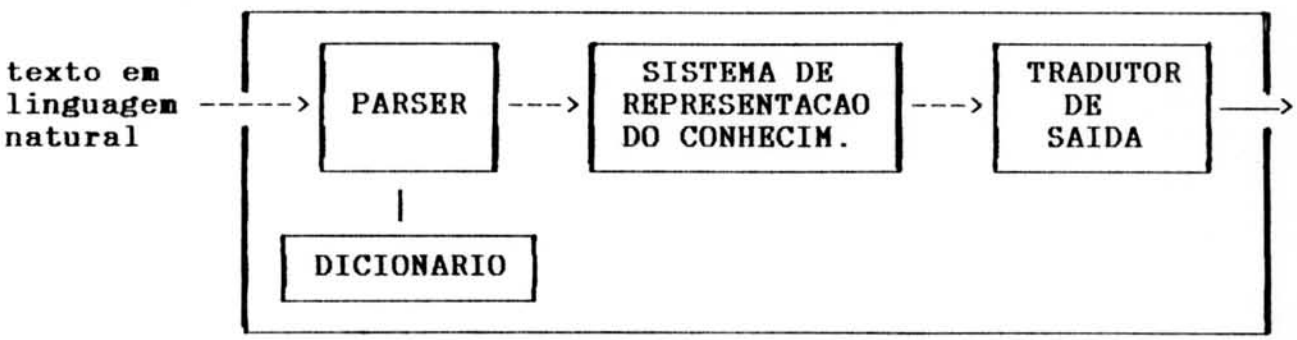


A base de conhecimento contém todos os fatos, ideias, relacionamentos e interações de um dominio limitado. A máquina de inferência analisa o conhecimento e deriva conclusões. O interface com o usuário permite que novo conhecimento seja apropriado e implementa a comunicação com o usuário. O usuário pode também requerer que o sistema especialista explique o processo pelo qual a conclusão foi alcançada permitindo ao usuário acompanhar a lógica envolvida. Sistemas especialistas tem sido criados para auxiliar na identificação e conserto de equipamentos complexos, como auxílio

no diagnóstico médico [26] e para apoiar análise financeira ou prospecção geológica. A variedade de aplicações é muito grande [4,18 e 54].

A segunda maior aplicação de IA é o processamento de linguagem natural. Programas de processamento de linguagem natural usam técnicas de inteligência artificial para permitir a um computador compreender e gerar linguagem natural para usar como interface para outros sistemas aplicativo [12, 77, 29]. Programas de processamento de linguagem natural usam técnicas de IA para analisar entradas expressas em linguagem natural, digitadas a partir de um terminal. Estes programas tentam identificar a sintaxe, semântica e contexto contido numa sentença para extrair seu significado. Se o programa pode compreender o que é dito, ele pode responder, por exemplo obedecendo aos comandos especificados pelo usuario. Uma aplicação típica de programas de compreensão de linguagem natural é em interfaces para outros softwares, tais como um sistema operacional, um sistema de gerenciamento de banco de dados, uma planilha eletrônica, um editor de texto etc... A figura 4.3 mostra os componentes básicos de um sistema de compreensão de linguagem natural: o "parser" (que funciona apoiado por um dicionário), um sistema de representação de conhecimento e um tradutor de saídas.

Figura 4.3: Componentes de um sistema de compreensão de linguagem natural



O parser desmembra a sentença em linguagem natural em componen

tes gramaticais (nomes, verbos, adjetivos, preposições etc...). Esta análise usualmente resulta na montagem de um grafo em forma de árvore que organiza as palavras de acordo com seu significado e fim. Um dicionário, associado com o parser auxilia a determinar o significado das palavras mas é limitado. Em consequência, os sistemas de compreensão de linguagem natural também são limitados ou dedicados a uma area ou aplicação específica. O sistema de representação do conhecimento, analisa a saída resultante do parser e baseado em seu conhecimento, tenta compreender o que veio do usuario. O tradutor de saída, toma a interpretação do sistema de representação de conhecimento e inicia alguma ação. Pode responder em linguagem natural ou criar saídas especiais para outros programas.

Reconhecimento de imagens é outra aplicação para inteligência artificial, cujas técnicas são usadas para analisar e avaliar informação visual, permitindo ao computador examinar uma figura ou cena real identificando objetos em particular, características ou padroes. A análise de fotografias aéreas é uma das aplicações desta area.

Inteligência artificial também é usada no campo da robótica. Robôs são máquinas ou manipuladores capazes de efetuar funções físicas limitadas. Um robô com inteligência pode responder a condições de contexto variantes e em função disto re-ordenar os passos do processo que executa, eliminando alguns ou de alguma outra maneira modificando sua operação para se ajustar à situação. Para usar inteligência artificial o robô necessita receber *inputs* de seu ambiente, usando sensores para detectar a posição de suas partes e outras condições ambientais, tais como pressão, temperatura e luz.

Inteligência artificial também pode ser usada em educação e

treinamento. Usando IA, novas formas de software para treinamento assistido por computador podem ser criados, permitindo ao computador atuar como um tutor. Usando instrução assistida por computador, é possível ajustar o treinamento ao treinando, de acordo com seu conhecimento, experiência, dificuldades etc... Também pode ser usado um interface em linguagem natural para a interação do treinando com o software.

Outras aplicaçãoes para a inteligência artificial começam a despontar. Programação automática é um exemplo. Neste caso, as técnicas da IA são usadas para auxiliar um programador a desenvolver e codificar programas para problemas especiais. Idealmente, o programa seria especificado em linguagem natural e o resultado, seria o programa. Outra importante aplicação é CAD-Computer-Aided-Design. Neste caso, técnicas de IA propiciam ao projetista conhecimento em projeto e capacidade de solucionar problemas de projeto.

4.2 Sistemas especialistas

Os sistemas especialistas podem ser classificados de acordo com a maneira como ele se interagem com o mundo [20]. Eles podem ser **dirigidos por diálogos**, quando eles apropriam suas entradas de um ser humano, ou por **sensores** quando eles apropriam as tais entradas diretamente do processo sensoriado. O sensoreamento pode provir através de comunicação de outro computador segundo um protocolo. Além disso os sistemas podem ser classificados como **orientadores** se eles oferecem suas saída a uma audiência humana ou **controladores** se eles dirigem suas saídas diretamente sobre o mundo físico na forma de sinais de controle a outros dispositivos. Essas duas dimensões: fonte de entrada e destino da saída definem um espaço de quatro tipos de sistemas: consultores (orientadores baseados em

como controladores. Mais recentemente houve interesse em agentes, sistemas especialistas que apropriam a informação diretamente do sensores sobre o mundo físico exercem algumas funções de controle, também diretamente sobre ele. Como a maioria dos sistemas especialistas são experimentais, são projetados de modo que muitas ou todas as suas funções de controle sejam exercidas através de instruções explícitas a um ser humano.

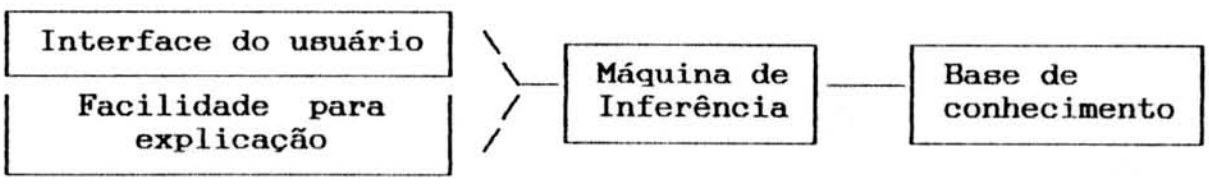
Um sistema especialista pode ser denominado **ativo** quando recebe entrada dos sensores no ambiente exerce funções de controle também diretamente sobre aquele ambiente. O controle pode ser exercido por conexões diretas no ambiente ou através de um intermediário humano que executa a orientação do sistema especialistas sem modifição.

4.2.1 PROJETO DE SISTEMAS ESPECIALISTAS

Uma abordagem diferente é requerida para projetar e programar este tipo de sistema. Ao invés de construir uma conjunto de procedimentos que o sistema deve executar, projeta-se um corpo de conhecimento, a partir do qual o sistema pode derivar conclusões. O projeto do sistema consiste em suprí-lo com este conhecimento e com métodos para solução de problemas.

De acordo com [62], os sistemas especialistas são compostos dos módulos mostrados na figura 4.5.

Figura 4. 5 :Componentes de um sistema especialista



A base de conhecimento detem a descrição do conhecimento. Existem muitos métodos para descrever o conhecimento e espelham os

diálogos), monitores (orientadores,baseados em sensoriamento), servidores (controladores baseados em diálogos) e agentes (controladores baseados em sensores) tal como ilustrado na figura 4.4.

Figura 4.4 : Classificação dos sistemas especialistas

Na prática, um dado sistema pode englobar elementos de mais de um tipo. A maior parte das aplicações dos sistemas especialistas até o momento se caracterizam como consultores que operam tipicamente com uma descrição estática do mundo e apresentam orientações ou conclusões através de um interface amigável. Essa descrição engloba sistemas especialistas clássicos como MYCIN, PROSPECTOR e CASNET bem como a maior parte dos sistemas mais recentes, conforme referido em [20].

Na década de 80 foram desenvolvidos um certo número de sistemas especialistas que se encaixam na categoria direcionado por sensores. Tais sistemas tipicamente recebem informação do ambiente em tempo real e são projetados para operar continuamente. A maioria destes sistemas pode ser classificada mais como monitores do que

diferentes modos de abordagem [3, 21, 22].

Redes semânticas descrevem relacionamentos entre objetos. A relação mais simples é do tipo "pxyz é modem" [21].

Regras de produção descrevem o conhecimento em termos de regras do tipo "SE-então" [28, 79]. Boa parte do conhecimento necessario para reagir aos eventos detectados na rede pode ser expressa desse modo. Por exemplo, se uma linha esta em manutenção, então relatos de eventos do tipo "queda de portadora" não devem gerar alarmes para a operação.

Segundo [28], sistemas baseados em regras constituem um meio bastante usual para codificar o "Know-how" dos especialistas humanos em resolver problemas. Os especialistas tendem a expressar a maioria de suas técnicas de resolver problemas em termos de um conjunto de regras situação-ação, ou SE-ENTÃO. E isto sugere que sistemas baseados em regras deveriam ser o método escolhido. O sistema especialista ACE, descrito em [89] usou regras de produção para representar a base de conhecimento usada para prover relatórios de problemas para apoiar a análise gerencial, orientada à manutenção de cabos telefônicos.

Os sistemas baseados em regras apresentam certas propriedades básicas:

- incorporam o conhecimento humano prático na forma de regras condicionais do tipo SE-ENTÃO;
- sua capacidade aumenta numa taxa proporcional ao aumento de sua base de conhecimento;
- eles podem resolver uma grande variedade de problemas possivelmente complexos, selecionando regras e combinando resultados de forma apropriada;
- eles determinam, de forma adaptativa, a melhor sequência de regras a executar;

- eles podem explicar suas conclusões relatando suas linhas de raciocínio e transladando a lógica de cada regra empregada em linguagem natural.

Sistemas baseados em regras atendem a necessidade de capturar, representar, armazenar, distribuir, argumentar sobre e aplicar conhecimento humano, eletronicamente. Isto provê um meio prático para construir especialista automatizados. Pode-se definir sistemas baseados em regras como sistemas modularizados de "know-how" relativo ao modo de resolver problemas. Este conhecimento consiste de vários tipos de informação, incluindo, segundo [21]:

- inferências específicas derivadas de observações específicas;
- abstrações, generalizações e categorizações de certos dados;
- condições necessárias e suficientes para alcançar algum objetivo;
- locais preferenciais para eliminar incerteza ou minimizar outros riscos;
- consequências prováveis de situações hipotéticas;
- causas prováveis de sintomas.

"Frames" são úteis para o tipo de problema que requer conjuntos de informacões a serem coletadas, para que uma solução possa ser encontrada [17, 88].

Taxionomias podem descrever conjuntos de informação, como nos frames e podem descrever como a informação é interrelacionada, como nas redes semânticas [30]. Por exemplo um frame pode descrever as características de modems e outro frame pode descrever as características de um particular tipo de modem e pode-se definir um relacionamento entre os dois frames como no exemplo seguinte:

```
modem-d([tipo:digital,sincronismo:assincrono,velocidade:1200]).
  pxyz([teste:loop]).
  ser(pxyz,modem_d).
```

O frame **modem-d** descreve as características de um modem como sendo do tipo digital, assíncrono e de velocidade 1200. O frame **pzyx** diz somente que o tipo de teste é loop, mas a relação "**ser**" estabelece que **pzyx** é um modem.

A informação numa base de conhecimento é estática até que alguma força externa analisa a rede semântica, põe as regras de produção em ação ou começa a preencher e analisar os slots de um frame. Tais tarefas são executadas pelo que é denominado uma "máquina de inferência". As máquinas de inferência imitam o tipo de pensamento que um especialista humano emprega quando tenta solucionar um problema. Isto é, começa com uma hipótese e tenta encontrar evidência que apoiem a hipótese, ou pode começar com as evidências disponiveis e tentar determinar que conclusões são deriváveis. Em sistemas especialistas estes dois métodos são denominados "encadeamento reverso" (backward chaining) e "encadeamento para adiante" (forward chaining) respectivamente [103].

Sistemas especialistas apropriam informação do usuario formulando questões, tal como um especialista o faria. Este módulo do sistema é frequentemente denominado "facilidade de consulta" [111]. Adicionalmente, os usuarios podem formular questões sobre como o sistema que chegou a uma particular solução ou porque o sistema formulou uma dada questão. Esta parte do sistema é denominada "facilidade de explanação".

Na vida real, os especialistas não podem sempre tirar conclusões com absoluta certeza. O mesmo vale para sistemas especialistas que devem ser capazes de informar o grau de confiabilidade das respostas que fornece [4]. Existem três abordagens básicas para lidar com isto:

-O grau de confiança pode designar a força relativa de uma conclusão, medida em termos relativos, como um número entre -1 e

+1. Esta abordagem é denominada de **abordagem standard.**

-Alguns pesquisadores consideram mais natural expressar a confiabilidade usando qualificadores tais como "alguns", "a maioria", em lugar de números. Esta abordagem é baseada em lógica inexata ou **lógica vaga** ("fuzzy logic"). Internamente, estes qualificadores são mapeados em números, manipulados pelo sistema e então transladados de volta em qualificadores para reportar o resultado, tal como descrito em [4, 26].

-Um grau de confiabilidade pode definir a probabilidade estatística de que uma certa conclusão está correta. Esta abordagem é baseada em **probabilidade Bayesiana**. Assim, como o método standard, este método expressa o grau de confiabilidade mediante um número entre 0 e 1. Contudo, o número indica a percentagem de exatidão e não a força relativa da conclusão.

Graus de confiabilidade podem ser aplicáveis em dois pontos. Primeiramente, podem ser aplicados à informação que o sistema necessita para formar uma conclusão. Em segundo lugar, eles podem ser aplicados às conclusões. O sistema especialista deve combinar os graus de confiabilidade em diferentes estágios. Dependendo do método usado para apoiar o raciocínio inexato, eles são combinados conforme mostra a tabela 4.1 abaixo.

Tabela 4.1 : Cálculo do grau de confiabilidade

Método	Fórmula
Standard	Grau de confiabilidade da conclusão é o grau de confiabilidade mínima entre os antecedentes
Lógica vaga	Grau de confiabilidade da conclusão é o grau de confiabilidade mínimo dos antecedentes
Bayesiano	Grau de confiabilidade da conclusão é o produto dos graus de confiança dos antecedentes.

4.2.2 Ciclo de vida num projeto de sistema especialista

O desenvolvimento de um sistema especialista requer uma abordagem diferente da que é usada em sistemas convencionais. As tradicionais metodologias, estruturadas, que requerem que as especificações sejam concluidas quando começa o projeto e que este esteja concluido quando começa a implementação, são inadequadas [103].

Como o conhecimento almejado é de natureza heurística, é impossível saber antecipadamente qual é este conhecimento. Na verdade, parte do conhecimento pode permanecer inconsciente ate que o especialista seja defrontado com uma situação que requeira tal conhecimento.

Uma abordagem para este problema pode ser a usada pela GTE no sistema NEMESYS referido em [62]. Ela começa limitando o problema a uma area de conhecimento fechada, pequena. A seguir prototipa rapidamente as técnicas de solução de problemas e finalmente, trabalha com o especialista para expandir interativamente o conhecimento. Outros autores sugerem que os passos a serem seguidos no projeto de um sistema especialista sejam os seguintes [103]:

IDENTIFICAÇAO: Trata-se de um problema cuja solução requeira técnicas de sistemas especialistas?

CONCEITUALIZAÇAO:Que informações e técnicas o especialista usa?

FORMALIZAÇAO: Como o conhecimento pode ser formalmente representado?

IMPLEMENTAÇAO: Como estas técnicas se ajustam nas estruturas e estratégias de sistemas especialistas?

TESTE: Teste de aceitação final

Cada um destes passos será comentado a seguir.

4.2.2.1 Identificacão

Durante a identificação o engenheiro do conhecimento e o especialista determinam as características do problema. Isto inclui a identificação do problema em si (tipo e escopo), os participantes do processo de desenvolvimento (outros especialistas), os recursos exigidos (tempo e facilidades computacionais) e as metas ou objetivos da construção do sistema especialista (por exemplo melhorar a performance ou distribuir conhecimento escasso).

A tecnologia de sistemas especialistas foi criada para ser usada em certos tipos de atividades de programação que não eram bem atendidas por outras tecnologias de programação. É essencial que o problema a ser solucionado com o uso de sistemas especialistas seja analizado para determinar-se se a melhor solução para ele realmente implica no uso de tal tecnologia. A seguir é apresentada uma lista de características comumente associadas com tarefas para as quais o uso de sistemas especialistas é adequado.

-Problemas tipicamente executados por especialistas

Um especialista usa embasamento de conhecimento e experiência numa área específica para resolver problemas ou chegar a conclusões. A diferença entre especialistas e as pessoas em geral reside no fato de que estes adquiriram um nível de conhecimento que lhes permite chegar a conclusões ou resolver problemas em sua especialidade, de uma forma consistente e fidedigna. A tecnologia de sistemas especialistas, que visa manipular uma grande base de conhecimento, é apropriada para as atividades tipicamente realizadas por especialistas. No presente caso, para solucionar um problema relativo a uma rede OSI, é preciso levar em conta um número elevado de condições que podem causar interferências mutuamente. Em certos casos, os indícios existentes nao sao suficientes para le-

var a uma conclusão segura, sendo necessário serem efetuados testes adicionais para verificar as condições de alguns outros elementos da rede. Não é razoável esperar que especialistas que detém o conhecimento para isto fiquem eternamente presos à atividade de gerenciamento direto da rede. Ao contrário, é desejável que o sistema especialista passe gradativamente a fazer o que eles fariam, sendo solicitada sua intervenção somente quando surge uma situação ainda não prevista. A partir do momento em que mais este novo curso de ação tenha sido definido, o mesmo pode também ser incorporado à base de conhecimento, dispensando a intervenção do especialista humano na próxima vez que uma situação similar a esta ocorrer.

-Tarefas que envolvem conhecimento que pode ser expresso como "regras práticas"

A maioria das tarefas executadas por especialistas pode ser descrita em termos de "regras práticas". Isto significa que é possível descrever, em termos de um conjunto finito de regras, o processo de raciocínio pelo qual o especialista chega à solução de um problema ou a uma conclusão. No caso do sistema de apoio à gerência de uma rede OSI, no início, tais regras serão derivadas da análise da definição dos serviços e protocolos inerentes a cada nível. Contudo, à medida em que redes OSI passarem a serem operadas, será acumulado um conhecimento maior sobre seu funcionamento sendo possível derivar regras práticas que constituam atalhos no processo de solução dos problemas mais comuns. Por exemplo, para o nível físico, existe uma série de testes, previstos nas recomendações do CCITT. Contudo, conversando com os especilaistas que atuam neste contexto, verifica-se que eles sempre começam a bateria de testes com um teste de atenuação ou continuidade, verificando a seguir o nível total de ruído. Dizem que não adianta fazer os de-

mais testes se nestes testes não encontrarem um limiar mínimo de desempenho que a prática evidenciou indispensável para o funcionamento adequado da linha. Estas regras não existem para os níveis OSI mais altos porque não existem muitas redes OSI. Contudo um sistema especialista pode ir incorporando tais regras à medida em que forem sendo derivadas.

-Tarefas que não são baseadas apenas em bom senso

Quando os especialistas usam bom senso na solução de problemas ou para chegar a certas conclusões, na verdade estão baseando-se num estado de conhecimento que na maior parte das vezes não pode ser definido usando um conjunto de regras razoável. A aplicação de bom senso a uma situação requer um profundo e amplo conhecimento dos fatores associados com a situação, mesmo que tal conhecimento esteja presente de forma subliminar na mente do especialista. É melhor evitar a tentativa de imitar tal processo de raciocínio envolvendo apenas limitada quantidade de conhecimento pois pode-se chegar a conclusões falsas.

-Tarefas que envolvam conhecimento que muda rapidamente

Sistemas especialistas são projetados de modo que o conhecimento possa ser facilmente adicionado ou alterado. Em vista disso, tarefas que envolvam conhecimento que esteja rapidamente sendo alterado ou atualizado são muito adequadas ao uso de sistemas especialistas. Diferentemente da programação convencional, fazer alterações ou adições a um sistema especialista não requer grandes restruturações do programa. No contexto de gerenciamento de rede OSI, em que nem mesmo o conhecimento formal está consolidado (muitas normas estão sofrendo revisões constantemente) a tecnologia de sistemas especialistas se apresenta como ferramenta apropriada por esta capacidade de fácil ajuste.

-Tarefas que podem ser representadas como um conjunto de ações ou condições independentes

Sistemas especialistas são idealmente orientados a tarefas que podem ser representadas como um conjunto de módulos independentes. Uma vez que a adição ou deleção de conhecimento é fácil num sistema especialista, os módulos individuais podem ser desenvolvidos incrementalmente e adicionados ao sistema quando se tornarem estáveis. Assim, no sistema de gerenciamento da rede OSI, cada nível pode constituir um módulo separadamente desenvolvido e adicionado ao sistema pois a própria arquitetura OSI é orientada à separação das funções em níveis.

-Tarefas que envolvem raciocínio inexato

O raciocínio inexato envolve escolher a melhor alternativa numa situação em que não podem ser tomadas conclusões definitivas. Por exemplo, diagnóstico de problemas de rede envolve tirar conclusões de um conjunto de sintomas. Tais conclusões são baseadas numa combinação de pesquisa e experiência mas a complexidade do problema às vezes torna impossível chegar a uma conclusão exata resultando num diagnóstico é a melhor conclusão que pode ser determinada com o conhecimento disponível. Sistema especialistas podem lidar com raciocínio inexato ou vago, associando coeficientes de certeza tanto a premissas nas quais se apoiam certas conclusões como às próprias conclusões. Este fator é tão importante que até mesmo circuitos integrados tem sido desenvolvido para apoiar a construção de computadores que funcionem de acordo com lógica vaga (fuzzi logic) [115].

Considerações adicionais para determinar se uma tarefa é apropriada para solução mediante o uso de um sistema especialista incluem o tempo de desenvolvimento necessário para criar o sistema especialista e os custos associados com o desenvolvimento. Embora

existam ambientes de desenvolvimento de sistemas especialistas que reduzem o tempo de desenvolvimento, uma tarefa ambiciosa e complexa pode requerer muito tempo de desenvolvimento. A complexidade da tarefa a ser implementada como um sistema especialista deve implicar em recursos apropriados alocados à tarefa. Uma abordagem de desenvolvimento incremental, começando com uma parcela apenas é sempre preferível a tentar desenvolver o sistema inteiro de uma só vez [10 e 103].

4.2.2.2 Conceitualização

Durante a conceitualização o engenheiro do conhecimento e o especialista decidem que conceitos, relações e mecanismos de controle são necessários para descrever a solução do problema.

No caso do sistema especialista para lidar com a gerência de uma rede de computadores baseada na arquitetura OSI, a etapa de CONCEITUALIZAÇÃO consistiria em elaborar uma relação dos alarmes e outras indicações de problemas que podem ocorrer em cada nível e planejar a estratégia para lidar com cada um. Contudo, a experiência dos técnicos em redes com ambientes OSI é limitadada atualmente. Em vista disso o sistema deve ser projetado para crescer incrementalmente, isto é, a cada problema sem solução prevista, o técnico será informado e terá que prever uma solução. Esta solução será incorporada à base de conhecimento do sistema especialista.

Deverão ser previstas abordagens diferenciadas para erros permanentes e erros transientes. Os erros transientes serão provavelmente relatados posteriormente, quando o sistema aberto que os detectou ja solucionou o problema, provavelmente mediante retentativas de realizar a tarefa pretendida. É importante, todavia,

registrar tais erros porque podem indicar uma deficiência do sistema. Por exemplo, erros tipo congestionamento na entidade de transporte destinataria são transientes mas, se frequentes devem ser tratados de alguma forma pelo sistema, pois em caso contrario, a alta incidência de tentativas não concluidas de estabelecimento de conexão de transporte ocasionarão desempenho ruim para aquele usuario e, em consequência do tráfego artificial gerado,na rede queda de desempenho para os demais usuários que estiverem compartilhando os mesmos recursos de rede.

Muitos problemas não são detectados imediatamente. Erros intermitentes podem ser consequência de deterioração gradativa de algum componente físico ou bug em algum componente lógico, que sob determinadas condições perde algum dado ou informação de status. Tais problemas são detectáveis pelo estudo dos sintomas ao longo do tempo. Tais problemas não persistem o suficiente para serem detectados por diagnósticos e testes. A tarefa de identificar a causa de tais problemas é usualmente dificultada pela multiplicidade de alarmas originados pelo mesmo problema. Peritos, ao analisar tais sintomas usam seu conhecimento da arquitetura da rede, experiência em analisar padrões de sintomas e conhecimento da probabilidade de falha em certos pontos da rede. Depois, analisam tudo e listam por ordem os componentes suspeitos.

A determinação de um modelo conceitual para o problema da gerência de redes de computadores implica no colecionamento de síndromes de rede relevantes: muitos sintomas ocorrem e poucos são relevantes na identificação de um dado problema. Síndromes devem ser agrupadas em rajadas: rajadas são grupos de sintomas que são separados por uma quantidade suficiente de tempo para tornar aparente quando os dados de um problema começam e os de outro terminam.

Um modelo conceitual diferente deve ser usado para analisar cada tipo de problema. Cada análise requer um conhecimento acurado da arquitetura da rede .

O sistema deve ser projetado para ser capaz de formular hipóteses: determinar testes a serem feitos, receber resultados, reavaliar o problema e repetir este procedimento tantas vezes quanto necessário.

A seguir o sistema especialista deve sugerir uma ação corretiva, tal como: trocar linhas, modems, configuração da rede etc...

4.2.2.3 Formalização

A fase de formalização envolve expressar os conceitos chaves definidos na fase de conceitualização nas representações formais de engenharia do conhecimento. Esta fase consiste na organização do conhecimento e é uma das mais difíceis do processo, conforme [48, 80, 112, 111].

Existe uma variedade de esquemas de representação do conhecimento que são classificados em declarativos ou procedurais. Um esquema declarativo é usado para representar fatos e asserções. Um esquema procedural lida com ações ou procedimentos. Métodos de representação declarativos incluem lógica, redes semânticas, frames e roteiros. Esquemas de representação do conhecimento procedural incluem procedimentos ou subrotinas e regras de produção.

A forma mais antiga de representação do conhecimento talvez seja a lógica [71]. A forma geral de qualquer processo lógico é baseada em que primeiro é apropriada informação, são feitas afirmativas e observações; isto é a entrada para o processo lógico e são denominadas **premissas**. As premissas são usadas pelos processo lógico para criar uma saída que consiste de conclusões, denominadas

inferências. Com este processo, fatos conhecidos como verdadeiros podem ser usados para derivar novos fatos que também devem ser verdadeiros. Existe dois tipos básicos de racionício: **dedutivo** e **indutivo.** Ambos são usados na lógica para gerar inferências a partir de premissas.

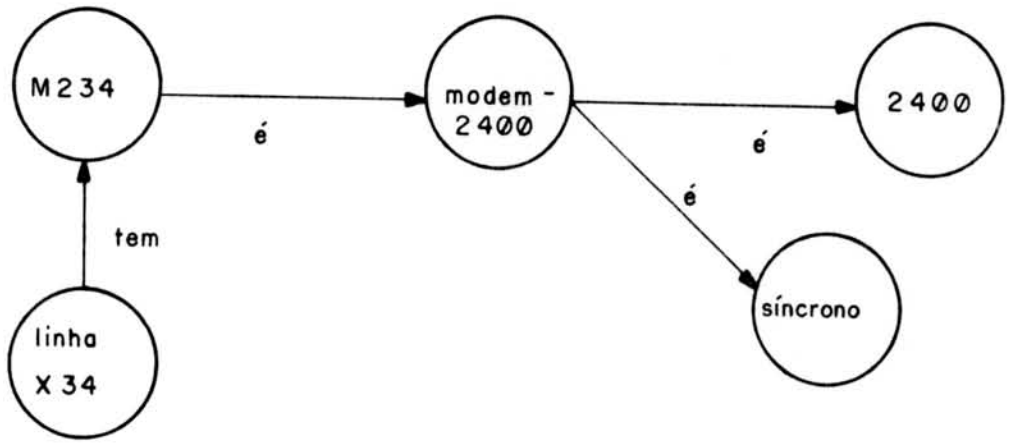
Quando premissas gerais são usadas para obter uma inferência específica, o processo é denominado **raciocínio dedutivo** ou dedução. **Raciocínio indutivo** usa um certo número de fatos estabelecidos ou premissas para derivar alguma conclusão genérica. Neste último caso, as conclusões nunca são definitivas ou absolutas. Conclusões podem mudar, se novos fatos são descobertos. Existirá sempre algum grau de incerteza nas conclusões até que todos os fatos sejam descobertos e incluidos nas premissas o que, às vezes, é impossivel. Em decorrência, o resultado do processo de raciocícnio indutivo sempre contém alguma medida de incerteza.

Um dos mais antigos esquemas de representação de conhecimento é a **rede semântica** [103]. Redes semânticas são descrições gráficas do conhecimento que mostram relacionamentos hierárquicos entre objetos. Elas são compostas de um número de círculos ou nós que representam objetos e descrevem a informação sobre estes objetos. Objetos podem ser qualquer item físico tal como um modem, um concentrador, uma linha, ou um técnico. Nós podem ser também conceitos, eventos ou ações. Atributos (tamanho, classe, origem ou outras carcterísticas) de um objeto também podem ser usados como um nó. Desta maneira, informações detalhadas sobre objetos podem ser apresentadas.

Os nós numa rede semântica são também interconetados por laços ou arcos que mostram os relacionamentos entre os vários objetos e fatores descritivos. Alguns dos mais comuns arcos são do tipo ***tem*** e ***é***. Um dos fatos mais interessantes e úteis sobre uma rede semân-

tica é que ela pode representar herança. Uma vez que a rede semântica é basicamente uma hierarquia as várias características de alguns nós podem herdar as características de outros.

Figura 4.6: Rede semântica



Embora a rede semântica seja de natureza gráfica, ela não aparece deste modo num computador. Os vários objetos e seus relacionamentos são declarados em termos verbais, e estes programados no computador usando vários tipos de linguagens.

Frames são blocos de conhecimento sobre um particular objeto, evento, local, situação ou outro elemento. Frames descrevem aquele objeto em grande detalhe. O detalhe é dado da forma de *slots* que descrevem os vários atributos e características do objeto ou situação. Frames são normalmente usados para representar conhecimento baseado em características e experiências bem conhecidas. O processo de raciocínio ocorre com frames é essencialmente a procura de confirmação de várias expectativas. Um frame provê o meio de organizar o conhecimento em slots que contém características e atributos. Alguns *slots* podem conter valores default. Outro tipo de *slot* é a ligação procedural. Este é um slot que permite que informação nova seja adicionada na medida do necessário. Outro tipo de slot é um que referencia outro frame, o qual pode prover maior detalhe sobre o atributo em particular daquele *slot*.

A maioria dos sistemas de inteligência artificial usam frames que são constituídos de coleções de outros frames. Juntos eles formam uma hierarquia que pode ser usada para fins de raciocínio.

Roteiro é um esquema de representação do conhecimento similar ao frame mas invés de descrever um objeto o roteiro descreve uma sequência de eventos. Alguns dos elementos de um roteiro típico são: condições de entrada, suportes, regras e cenários. As condições de entrada descrevem situações que devem ser satisfeitos antes que os eventos no roteiro possam ocorrer ou ser válidos. Suportes referem-se a objetos que são usados na sequência de eventos que ocorrem. Regras referem-se as pessoas envolvidas no roteiro. Cenário descreve a real sequência de eventos ocorrem. Um roteiro é útil para predizer o que vai acontecer numa certa situação. Mesmo que certos eventos não tenham sido observados, o roteiro permite ao computador predizer o que acontece, a quem e quando. Se o computador dispara um roteiro, perguntas podem ser formuladas e respostas precisas derivadas com pouco ou com nenhum conhecimento prévio.

Regras de produção são um dos mais populares esquemas de representação de conhecimento [28]. Regras de produção são enunciados compostos de duas partes. A primeira parte da regra, denominada antecedente, expressa uma situação ou premissa enquanto que a segunda parte, denominada consequente estabelece uma ação particular ou conclusão que é aplicada se a situação ou premissa é verdadeira. Uma regra típica é ilustrada a seguir:

IF a taxa de erros > LIMIAR

THEN desencadear inspeção de linha

A ação, consequência ou conclusão enunciada na parte THEN é válida se a parte IF da regra é verdadeira.

Conforme anteriormente referido, os sistemas de racicínio ba-

seados em regras podem ser do tipo :

encadeamento para frente ou progressivo (forward-chaining)

encadeamento para trás ou reversivo (backward-chaining)

Nos sistemas de encadeamento para frente o raciocínio começa com uma evidência e tenta chegar a hipóteses, testando a validade de condições IF contra fatos conhecidos para produzir conclusões a partir das cláusulas THEN.

Sistemas encadeamento para trás criam uma solução hipotética e tentam prová-la satisfazendo as condições IF. Se os fatos necessários não estão presentes então outra hipótese é gerada e verificada.

Num sistema onde as síndromes podem ser agrupadas, encadeamento para frente é mais eficiente do que tentar testar todas as possíveis condições embora esta seja uma técnica usada como sub-estratégia na resolução de certas metas intermediarias no processo de raciocínio.

Frames também são usados para armazenar descrições de atributos de componentes e outros objetos. Frames lembram estruturas de dados mas tem o poder de expressar valores default e métodos para adquirir valores. Um tipo genérico de frame pode ser definido e instanciações subsequentes do tipo herdam o tipo do tipo pai. Assim, hierarquias são facilmente representadas por uma estrutura baseada em frames. Em adição, componentes do frame podem conter ***demons***, ou funções que podem ser invocadas automaticamente em certas condições do frame.

O modelo conceitual sugere muitos sub-problemas no processo de diagnóstico. Uma vez que diferentes sub-problemas usam um conjunto diferente de conhecimento, o uso de mais de uma base de conhecimento pode ser um meio de focalizar cada diagnóstico separadamente conforme sugerido em [102].

Uma vez que a arquitetura da rede contém uma hierarquia de componentes, a melhor maneira de representar a descrição lógica e física de rede é com uma hierarquia de frames. Frames são usados para capturar a informação numa mensagem de síndrome.

Por outro lado, o reconhecimento de rajadas é etapa importante no processamento de um sistema especialista. Para facilitar o reconhecimento de rajadas é importante elaborar uma base de conhecimento de rajadas. A base de conhecimento de rajadas contém regras IF-THEN baseadas no instante em que a síndrome ocorreu e o tipo de síndrome. Padrões de rajadas são complicados pela sobreposição de múltiplos problemas. A ausência de certas síndromes também é informativa, reduzindo possiveis linhas de raciocínio.

As regras IF-THEN usadas para mapear rajadas em problemas constituem um nível na base de conhecimento para análise de problemas. Regras adicionais, usando informação da arquitetura da rede são usadas para posteriormente isolar o problema numa análise de mais alto nível [81].

Sugestões de manutenção podem ser derivadas de regras que mapeiam problemas alvos em ações de manutenção na base de conhecimento. Tais regras incorporam ações de manutenção em função de hora do dia, nível de servico prestado pelo componente, natureza do problema, possiveis problemas múltiplos e histórico de consertos. O histórico de consertos também é importante para determinar a ação sugerida. Um conserto recomendado que não tenha funcionado, não deve ser repetido e uma nova linha de raciocínio usando aquele fato deve ser empregada para isolar o problema [22].

4.2.2.4 Implementação

Durante a implementação, o engenheiro do conhecimento transfor-

ma o conhecimento formalizado num programa de computador. Engenheiro do conhecimento é a pessoa que projeta e constroi um sistema especialista; são especialistas de computação/IA que adquirem o conhecimento de todas as fontes possíveis, incluindo peritos humanos, organizando o conhecimento numa base de conhecimento

A implementação deve ser feita logo a seguir porque uma da razões para implementar um protótipo inicial é testar a eficiência do processo de decisões projetado nas etapas anteriores do processo. Isto significa que existe uma alta probabilidade de que o código inicial tenha que ser revisado ou descartado durante o desenvolvimento.

4.2.2.5 Teste

O teste do sistema é um processo em andamento. A natureza dinâmica do desenvolvimento do sistema especialista requer que codificação das regras, teste de novas regras e possível reprojeto subsequente sejam conduzidos num processo interativo.

Teste envolve avaliação da performance e utilidade do programa protótipo e sua revisão se necessário. Avaliar a performance do sistema significa responder aos seguintes tipos de perguntas:

-O sistema toma as decisões que os especialistas geralmente consideram apropriadas?
-As regras de inferência são corretas, consistentes e completas?
-A estratégia de controle permite ao sistema considerar os itens na ordem natural preferida pelos especialistas?
-As explicações do sistema são adequadas para descrever como e porque as conclusões estão sendo obtidas?
-O teste de problemas cobre o domínio, manipulando casos reais e complexos?

Para avaliar a utilidade do sistema, usa-se um conjunto diferente de questões:

-A solução do problema auxilia o usuário de modo significativo?

-As conclusões do sistema são apropriadadmente organizadas, ordenadas e apresentadas com o nível adequado de detalhe?

-O sistema é rápido suficiente para satisfazer o usuário?

-O interface é suficientemente amigável?

4.2.3 Estágios de um sistema especialista

A maioria dos sistemas especialistas começa com um protótipo de demonstração que é um pequeno programa capaz de manipular uma parcela do problema que vai ser abordado. Este tipo de programa é útil para testar idéias sobre a definição do problema, seu escopo e representar o domínio. Um protótipo típico para demonstração contém de 50 a 100 regras e funciona adequadamente em um ou dois casos de teste [103].

O passo seguinte da maioria dos sistemas especialistas é sua evolução para protótipo de pesquisa, um programa de tamanho médio, capaz de evidenciar performance aceitável em um bom número de casos de teste. Tais sistemas tendem a ser frágeis, falhando em situações mesmo dentro de seu escopo pois ainda estão em fase de crescimento. Um protótipo deste tipo contém tipicamente 200 a 500 regras.

Alguns sistemas especialistas evoluem para o estágio de protótipo de campo. Tais sistemas são programas de tamanho medio a grande e foram revisados através de teste em problemas reais na comunidade de usuários. São moderadamente confiáveis, contem interfaces amigáveis e tratam das necessidades do usuário final. Um

protótipo de campo pode conter 500 a 1000 regras.

Alguns poucos sistemas especialistas evoluiram para o estágio de protótipo de produção. Tais sistemas são programas grandes que foram extensivamente testados em campo e possivelmente reimplementados em uma linguagem mais eficiente para aumentar sua velocidade e reduzir as necessidades de armazenamento. Um típico sistema de produção baseado em regras pode conter de 500 a 1500 regras e prove apoio preciso, rápido e eficiente à tomada de decisões.

Somente raros sistemas especialistas evoluiram até hoje para o estágio de sistema comercial, usados rotineiramente em ambientes comerciais. Podem ter mais de 3000 regras e levam a conclusões corretas de 90 a 95 das vezes [103].

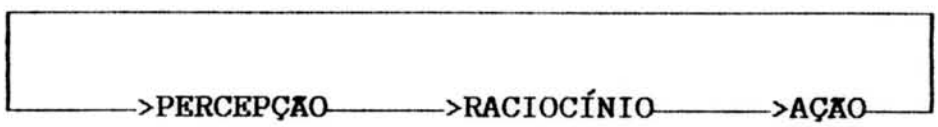
4.3 Problemas inerentes à representação do conhecimento

A escolha de uma metodologia para representação do conhecimento, em sistemas especialistas é importante porque existem muitos possiveis paradigmas representacionais. As consequências de uma escolha inadequada podem ser limitantes em estágios mais avançados de um projeto pois podem implicar na impossibilidade de representar informações críticas.

Um problema grave encontrado na escolha de um método para representar o conhecimento advém do fato de que os critérios que determinam a escolha de uma ou outra metodologia geralmente não estão claros no inicio do projeto.

Considere-se um agente inteligente cujo padrão de raciocínio seja como o da figura seguinte:

Figura 4.7: Padrão de raciocínio



O agente inteligente está perpetuamente engajado num *loop* infinito, percebendo coisas, raciocinando sobre elas e atuando.

A maior tarefa para um agente como este é adquirir um modelo do mundo real em que está envolvido e manter este modelo suficientemente consistente. O conhecimento do mundo real consiste de:

-fatos que são ou que foram verdade (o estado conhecido)

-regras para predizer mudanças ao longo do tempo, consequências de acões e coisas não observadas que podem ser deduzidas de outras observacões.

Um fato inevitável é que o modelo sempre será incompleto, mesmo no melhor caso, pois o mundo real é demasiado complexo e dinâmico. As diferenças entre o modelo interno e o mundo real advém de varios fatores:

-mudanças ocorridas no mundo real desde que o agente inteligente registrou algum fato sobre o mesmo;

-a incapacidade do agente para aprender num prazo de tempo limitado, tudo o que seria possível aprender;

-limitações no sistema de representação do conhecimento que precludem a conceitualização de certas coisas sobre o mundo.

4.3.1 Percepção e racionínio

O raciocínio dedutivo aplicado à informação provida por percepção é essêncial para a análise da situação e para o planejamento das ações, nos sistemas especialistas. Tal raciocínio é necessario para levar a conclusões, a partir de fatos, para conectar fatos conhecidos, habilitando condições para ações e expectativas, para determinar a aplicabilidade das ações em certas situações, para avaliar ações hipotéticas alternativas, para determinar quando expectativas e hipóteses foram contraria das, para estruturar

planos para atingir metas e para detectar inconsistências entre metas e resolve-las.

Muitas destas atividades envolvem a aplicação de regras de inferência dedutivas válidas, como num cálculo predicativo formal. Outros processos de raciocínio que podem ser aplicados a tal informação são:

a)derivando conclusões não-monotômicas, que, conforme [112] são conclusões obtidas em situações onde existe carência de conhecimento suficiente e que não seriam derivadas se conhecimento suficiente estivesse disponivel.

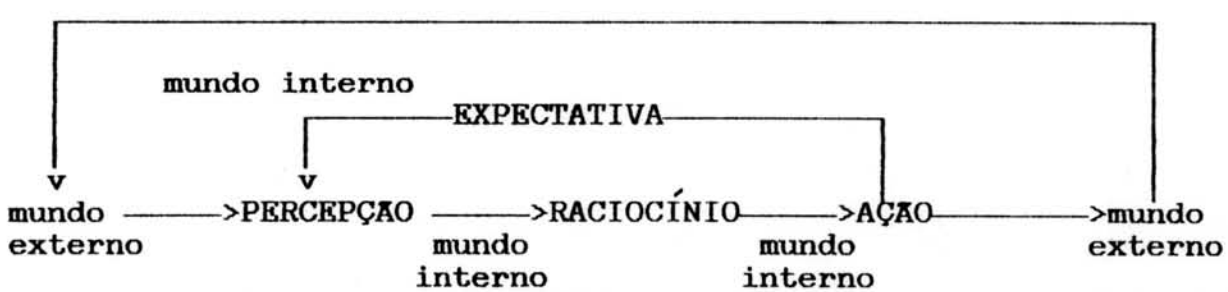
b)determinando graus de confiabilidade de um suposição através de raciocínio probabilístico ou análise Bayesiana.

c)conceitualizando e nomeando novas entidades, através de ações construtivas, tais como planejamento e projeto

d)maximizando ou otimizando processos que busquem a melhor escolha entre alternativas

O raciocínio efetuado envolve o modelo do mundo real e as conclusões resultantes levando a ações, podem afetar tanto o mundo externo quando o mundo interno (o modelado). Assim a atuação do agente inteligente poderia, ser representada, de forma mais precisa, conforme a figura 4.8:

Figura 4.8 Atuação de um agente inteligente



O efeito do processo de EXPECTIVA é registrar o efeito esperado para a ação.

No presente sistema a percepção que o agente inteligente pode ter é limitada pela conjunto de eventos que são relatados, segundo

o padrão OSI, ao SMAP, pelos diversos gerenciadores de nivel. A ação fica limitada aos comandos que o SMAP pode encaminhar e o raciocínio pelas regras de inferência e conhecimento acumulado na base de conhecimento.

A melhor alternativa para o agente inteligente é uma evolução gradual para um modelo mais e mais completo e realista, à medida em que adquire "experiência" com o próprio mundo e com sua necessidade de predizer. Logo, um quesito básico para o modelo deve ser sua capacidade de permitir uma evolutiva aquisição de novos modelos, mais confiáveis que possam ser mais efetivamente aplicados as tarefas e metas desejadas.

Alguns dos objetivos almejados no esquema de representação selecionado são:

-como estruturar o sistema representacional tal que seja capaz de fazer todas as distinções importantes;

-como permanecer a par de detalhes que não possam ser resolvidos (como um teste num componente do sistema que não pode ser atingido);

-como capturar generalizações tais que fatos que possam ser generalizados não tenham que ser aprendidos e armazenados individualmente (tal como as características de um modem que é usado em diversos pontos da rede);

-como notificar eficientemente quando novo conhecimentos contradizem ou modificam o conhecimento ou as hipóteses existentes e como saber/descobrir/decidir o que fazer a respeito;

-como representar atributos dependentes de tempo;

-como adquirir conhecimento dinamicamente sobre a vida do sistema;

No que concerne à máquina de inferência:

-como gerar e pesquisar um conjunto de hipóteses possiveis sem explosão combinatorial;

-como encontrar os relacionamentos entre elementos que tenham sido identificados e regras que poderiam ser aplicáveis numa percepção mais ampla

-como reconhecer que uma hipótese é duplicata de outra

-como encontrar a melhor caracterização da situação

-como lidar com erros de entrada ou percepções incompletas ou distorcidas

No que concerne ao planejamento da ação:

-como estruturar um plano para permitir a monitoração até sua conclusão

-como representar e disparar plano de contingência e dinamicamente replaneja-los quando algo sai errado

-como simular e avaliar um plano

-como registrar as expectativas e objetivos que motivam um plano e reconhecem quando um plano não é mais relevante

-como planejar para objetivos múltiplos e possivelmente competitivos.

4.3.2 Organização dos sistemas especialistas

Muitos sistemas são organizados em torno de regras de produção. Quando o número de regras cresce muito são necessárias técnicas para evitar testar todas as regras em cada situação.

Uma solução consiste em agregar hierarquicamente todas as regras, garantindo que conceitos mais gerais sejam acessiveis a um conceito mais específico. Numa taxonomia como esta um conceito pode ser armazenado no nível mais geral e indiretamente acessado por conceitos mais específicos que herdam aquela informação. Com uma

estrutura taxonômica deste tipo, as ações dos sistemas de regras podem ser conectadas aos nodos conceituais, como segmentos de orientação que são aplicáveis em situações descritas para àquele conceito.

4.4 Validação de sistemas especialistas

Existe uma grande dificuldade para assegurar que um sistema especialista, que tenha um comportamento adequado quando analisado face a alguns casos, evidencie a mesma qualidade de comportamento em todos as situações.

Algumas possíveis modalidades em que o sistema pode falhar são descritas em [68]. Ele relaciona os seguintes modos de falha de um sistema especialista:

- Submissão de informações incorreta: o sistema especialista produz uma resposta errada, não porque existe um problema nele mas porque ocorreu um erro ou inconsistência na informação que o usuário forneceu ou solicitou. A solução para isto é uma completa validação em toda informação oferecida pelo usuário mas isto implica num esforço extra de computação.O uso de conjuntos de teste do sistema especialista, gerados automaticamente, é a forma efetiva de validar este aspecto do sistema especialista.

- Regras incorretas no conjunto de regras: nessa situação o perito declarou um fato incorreto inadvertidamente. O sistema especialista trabalha corretamente mas dá uma resposta correta embora 100% consistente com o conjunto de regras. O desenvolvimento de um conjunto de teste automatizado, automaticamente gerado, também é uma solução para este problema. Miller [68] sugere que com cuidado na escolha de casos limites e não usuais para testar o acesso mais comum ao sistema especialista, pode-se eliminar tais deficiências

no mesmo.

- Regra incorretamente colocada no conjunto de regras: aqui encontra-se três casos. Regras omissas, regras extras e regras erradas. Uma análise das regras que são usadas pelo sistema especialista permite detectar a maioria dos casos de regras erradas. Se as sequências de teste são suficientemente sofisticadas podem detectar de 50% a 70% dos casos de regras omissas. Regras extras, isto é, que não são utilizadas também podem ser detectadas por verificação de regras usadas na fase de testes.

- Problema de redução de regras são os casos que o erro está no ambiente do sistema especialista ou em outro componente de software. A solução para isto é o uso de métodos de teste de convergência convencionais basadas em exercício compreensivo das funções e da estrutura do sistema porém isso só pode ser feito quando se tem acesso ao código fonte do sistema especialista.

- Validação das saídas: atualmente, estima-se que é possível uma redução de cerca de 100:1 na complexidade do processamento do sistema especialista com métodos de linearização. Porém como implementar isto efetivamente ainda não está totalmente pesquisado e desenvolvido. Uma estratégia passível de uso poderia ser a análise do fluxo da lógica e agrupamento eficiente das regras.

Durante a aquisição do conhecimento, isto é, na transferência do conhecimento de um perito para um programa, vários problemas podem ocorrer:

- o conhecimento do perito pode ser incompleto ou inconsistente;
- a representação do mesmo de uma forma computacional pode não ser adequada e a implementação introduz erros.

Diferentes métodos foram propostos para auxiliar o processo de transformação do conhecimento. Em alguns casos, cada regra é veri-

ficada sintáticamente quando é inserida ou editada. Uma verificação semântica limitada também pode ser executada no sistema, comparando cada regra nova ou alterada com as existentes. Contradições são relatadas (por exemplo, duas regras com a mesma premissa mas diferentes conclusões). Isso é necessário para a construção incremental de uma base de conhecimento. Em alguns sistemas existem ferramentas de software que automaticamente verificam a consistência de uma base de conhecimento. É difícil e não é genéricamente possível provar a validade de um sistema sem um teste sistemático. Os métodos de verificação e as propriedades verificáveis dependem essencialmente da estrutura da base de conhecimento.

Quando o conhecimento é representado usando regras de produção os seguintes erros podem ocorrer no sistema especialista:

- Incompleto: uma situação na área de domínio em que certa inferência é requerida mas não existe uma regra na base de conhecimento que produza as conclusões desejadas.

- Conflito: esta é uma situação na área de domínio em que ao menos duas regras com decisões conflitantes podem ocorrer. Isto é detectável se é possível examinar essas regras e determinar a situação em que elas seriam acionadas.

- Redundância: é uma situação na área de domínio em que ao menos duas regras existem com a mesma decisão. A redundância pode ocasionar esforços de pesquisa adicionais porque numa situação de retrocesso pode ocasionar o exame de uma sub-árvore de regras sem resultado efetivo algum. No sistema que usa fatores de certeza, a redundância é um problema real porque pode causar a mesma evidência ser contada duas vezes e levar a diferenças nos pesos das conclusões [79].

No sistema XTEL descrito em [18], o teste e a validação consistiu em usar o sistema especialista em uma rede pequena e compa-

rar suas conclusões com aquelas produzidas por especialistas no assunto. Durante o desenvolvimento dessas experiências, o XTEL chegou a um ponto onde ele gerou um super conjunto das recomendações do perito. Entre as lições aprendidas no desenvolvimento do sistema XTEL foi destacado que, à medida em que os especialistas compreendem o processo de desenvolvimento das regras do engenheiro de conhecimento, podem passar a gerar outras regras por si próprios sem a participação do engenheiro do conhecimento. Por outro lado, o sistema desenvolvido a partir da colaboração de múltiplos peritos pode experimentar problemas não encontrados quando somente um perito participa. Por exemplo, a sobreposição de conjuntos de regras desenvolvidas por diferentes peritos pode, em alguns casos, interferir uma com a outra de modo a não produzir as consequências desejadas. Um exemplo disso é "looping". Cada perito pode desenvolver um conjunto de regras tal que em cada um não exista "loop", porém quando os conjuntos são combinados o "loop" pode ocorrer. A solução para isto é uma abordagem baseada em busca de consenso. No XTEL, todas as regras submetidas por peritos individuais foram revisadas e acatadas pelo grupo antes de serem inseridas na base de conhecimento.

4.5 Ferramentas de apoio ao trabalho do sistema especialista

A distinção entre hipóteses e resultado situa-se em sua interpretação lógica. A hipótese é uma afirmativa sobre um objeto da rede que pode ser verdadeiro ou não. Um resultado é uma afirmativa correta ou testada sobre um objeto da rede. Hipóteses são traduzidas para manipular raciocínio plausível.

Em consequência da manipulação de problemas múltiplos, o processo de raciocínio cria e usa muitas deduções não relacionadas.

Essas poderiam ser registradas numa estrutura de dados *ad hoc* desconectadas dos objetos da rede, uma espécie de grande área de trabalho. Uma abordagem ótima para o processo de dedução consiste em associar uma história com cada objeto selecionado de acordo com o seguinte critério:

- A função do componente na rede

- A importância do componente na estrutura da rede: alguns componentes dividem a rede em unidades funcionais independentes.

- O tamanho do componente: um componente com grande números de sub-componentes pode registrar dados históricos para todos seus sub-componentes.

A história pode conter qualquer número de deduções. As deduções que não são mais válidas devem ser removidas. Em princípio, todas deduções relativas a um problema poderiam ser removidas quando uma conclusão é atingida. Contudo, o registro das conclusões é muito útil para o diagnóstico direto de problema que ocorre repetidamente no mesmo componente da rede. Por exemplo, se uma conexão apresentou falhas devido ao sub-componente (um modem por exemplo), na próxima vez que ocorrer uma falha nessa conexão, pode-se suspeitar diretamente deste sub-componente.

Uma alternativa para representar o conhecimento a ser adicionado no processo de diagnostico consiste em usar técnicas de associação de procedimentos a objetos estruturados ("demons" ou demônios na terminologia de "frames", os quais são conceitualmente, um procedimento que observa, até alguma condição tornar-se verdadeira e, então, ativa um processo associado). "Demons" de tratamento de eventos são definidos para todos os tipos e objetos de rede. Quando um evento ocorre, o correspondente "demon" é adicionado [3].

A solução de problemas de rede é um processo de raciocínio acionado pelos eventos no sentido de que o que dispara o mecanismo

de inferência é sempre um evento externo da rede. Uma mensagem relatora do evento é enviada para cada objeto. Na recepção da mensagem as regras selecionadas são tentadas sequencialmente podendo por seu turno derivar outras regras para continuar a inferência tentando classificar o problema. Nem todas as regras são testadas; somente aquelas que se aplicam à particular classe de objetos de rede que pode ser afetados.

A codificação do conhecimento, normalmente implícita e de difícil compreensão, é um processo por si só esclarecedor que leva a muitas novas descobertas. Neste processo, os peritos podem ampliar seu conhecimento, no seu próprio campo de excelência, o que pode levar ao fato de que o conhecimento adquirido na construção do sistema especialista seja até maior do que o conhecimento transferido para o sistema especialista. Contudo, muitas vezes, os programas necessários para implementar um sistema especialista são muito complexos. As rotinas computacionais para implementar a estrutura do conhecimento e a estratégia de controle podem requerer considerável porção do tempo dedicado à construção do sistema especialista. O aparecimento de ambientes de desenvolvimento de sistema especialistas no mercado, tal como referido em [57], auxilia a minimizar o tempo de desenvolvimento de um sistema especialista e permite ao projetista dedicar maior atenção ao domínio da aplicação do que aos detalhes da implementação.

Os sistemas de apoio ao desenvolvimento são genericamente denominados de ***shell*** e permitem a construção de programas a partir das ferramentas e facilidades propiciadas. Tem sido usada algumas vezes uma estrutura de apoio denominada "quadro-negro" ("blackboard") para apoiar a organização do conhecimento em varios contextos, tal como descrito em [3 e 100]. Os sistemas ***shell*** orientados a sistemas especialistas normalmente permitem ao usuário pro-

ver as regras concernentes ao domínio de conhecimento em foco e rapidamente construir e testar protótipos. Até mesmo o usuário final pode interagir com tais sistemas e, com pouco conhecimento específico de inteligência artificial, usa-los e ampliar a base de conhecimento [57].

Contudo, podem surgir dificuldades quando a estrutura das tarefas do domínio de conhecimento não se ajusta aos padrões previamente previstos no sistema ***shell***. A fim de poder codificar métodos de solução de problemas intrínsecos do domínio de conhecimento normalmente é requerido uma facilidade de escape do ambiente de desenvolvimento que permita o uso de outros recursos de software (rotinas escritas em alguma linguagem mais adequada para tratar um determinado aspecto do problema ou até mesmo acesso a um sistema gerenciador de banco de dados).

O sistema EXPERT da ARITY [2] tem a vantagem de, além de prover uma coletânea de facilidades que simplificam a implementação de um sistema especialista, permitir o uso de rotinas escritas em outras linguagens. Isto abre o leque de opções concernentes à busca da solução ótima em termos computacionais.

Por este motivo, foi escolhido este software para servir de ferramenta para a construção do protótipo do sistema especialista para a gerência de rede.

5 - A MIB-Management Information Base

A Management Information Base-MIB, é o conjunto de dados de gerenciamento num sistema aberto, que está disponível para o ambiente OSI. O gerenciamento OSI comunica estes dados gerenciais entre sistemas abertos [43].

Este conceito não implica em qualquer forma de armazenamento da informação, segundo a ISO e sua implementação é assunto de interesse local, e fora do escopo dos padrões OSI. Os dados na MIB são estruturados de acordo com as exigências dos processos de gerenciamento que necessitam acessá-los.

Embora, no modêlo para gerenciamento do ambiente OSI, a MIB tenha outros usos (inclui dados usados pelas entidades de nível **n** e pelas entidades gerenciadoras de nível n), para fins de sistema de gerenciamento, a MIB pode ser considerada tão somente como uma definição da informação que deve ser transferida. Neste sentido, a definição da MIB é idêntica a definir o protocolo que descreve a informação transferida pelos sistemas de gerenciamento. Por isto, a definição das informações da MIB pode ser feita usando a linguagem ASN.1 [40 e 41] que atualmente é a opção ISO e CCITT para descrição formal de PDU:Protocol Data Units, conforme destacado em [14]. No presente trabalho, foi usada também a linguagem de taxionomia do sistema de desenvolvimento ARITY PROLOG usado para especificar as regras do projeto do sistema especialista para gerência de rede. Isto permite seu uso direto pela máquina de inferência. Assim, embora numa situação real de gerência de rede as informações sejam intercambiadas entre os SMAP usando a forma estabelecido pelas regras de codificação básicas definidas em [41], no protótipo usado para testar as idéias propostas não haverá real intercâmbio de PDUs entre SMAPs, mas apenas uma simulação disto. Assim, para tornar mais direto o trabalho de manipulação das infor-

mações de gerenciamento, foi pressuposto que elas virão com a definição adequada ao manuseio pelo sistema especialista.

Destaque-se que as transferências entre os sistemas de gerenciamento podem ser informação sendo reportada ou informação que causará a ocorrência de certas ações de controle.

Em linhas gerais, o gerenciamento como um todo pode ser modelado como sendo efetuado por um conjunto de processos que não estão necessariamente localizados no mesmo local e sim distribuidos por uma certo número de sistemas. Quando processos de gerenciamento necessitam comunicar-se com outros que não são residentes no mesmo sistema usam os protocolos de gerenciamento definidos no modelo OSI para gerenciamento.

A seleção do conteúdo da MIB foi feita segundo uma abordagem recomendada na metodologia da engenharia de informações, isto é, em função da necessidade de quem vai utilizar a informação. Como o SMAP é o processo que mais intensamente dependerá de tais informações, a análise começou com a coleta de sintomas relevantes para este processo e, mais especificamente, em relação ao gerenciamento da camada de transporte. O processo SMAP é detalhado na seção seguinte.

5.1 O processo de gerenciamento

Em cada sistema aberto gerenciado deve existir um processo de gerenciamento, denominado SMAP, System Management Application Process (SMAP), conforme descrito na seção 2.4.

Conforme ilustrado na figura 5.1, os processos de gerenciamento (SMAPs) que apoiam a atividade de gerenciamento recebem informação de controle de:

a) pessoas e/ou software atuando como agentes administrativos locais ao processo de gerenciamento;

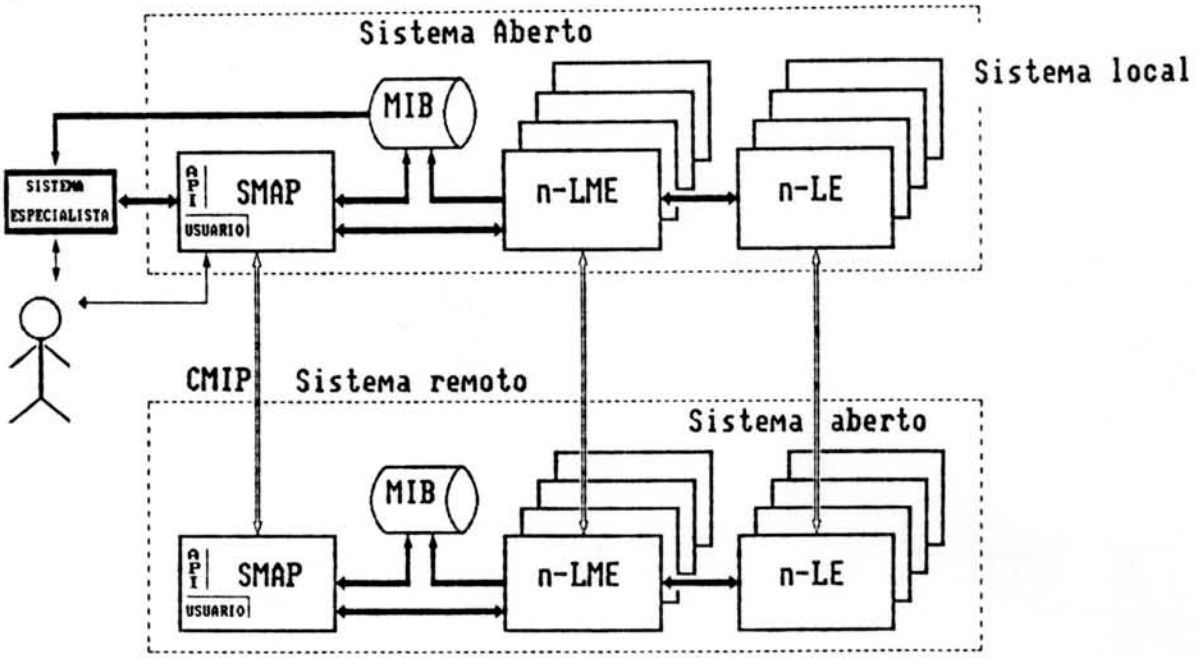
b) de sistemas remotos atraves de seus:

i) SMAEs

ii) entidades de gerenciamento de cada nível

iii) entidades de nível (N)

Figura 5.1: Fluxo das informações de gerenciamento



A MIB contém toda a informação relevante à operação de gerenciamento do ambiente OSI. Esta informação pode ser visualizada como uma coleção de objetos gerenciados, cada um dos quais tem atributos e pode ter eventos e ações inerentes à sua operação ou au seu uso. Tais informações podem envolver:

- Informações sobre eventos

  contadores (erros, time-outs, ...)

- Informação sobre a estrutura da rede

  diretório da rede (o que existe na rede)

  topologia (o que está instalado onde e ligado a que)

- Atributos

  parâmetros (tamanho de janela, etc...)

A informação na MIB pode ser provida a:

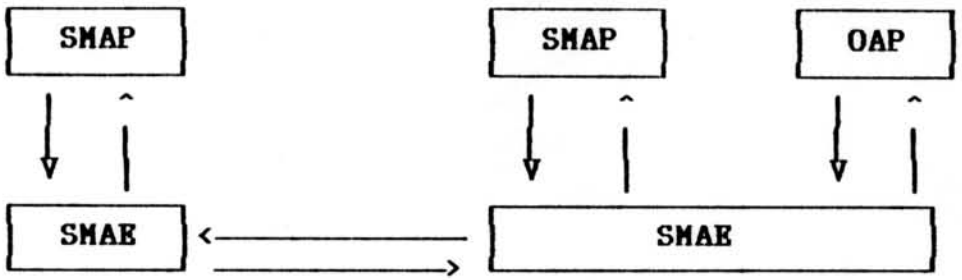
-processos de aplicação locais (de forma não sujeita à qualquer padronização)

-sistemas abertos remotos, através do gerenciador do sistema,

dos protocolos de gerenciamento em cada nível e através dos próprios protocolos de cada nível.

Em cada sistema envolvido na transferência de informação de gerenciamento existirá ao menos um System Management Application Process (SMAP). Este SMAP comunicar-se-á com um SMAP remoto ou com um Processo da Aplicação Comum (OAP-Ordinary Application Process), para fins de transferência de informação de gerenciamento, através da Systems Management Application Entity (SMAE), tal como ilustrado na figura 5.2.

Figura 5.2 : Interação entre os SMAPs e OAPs



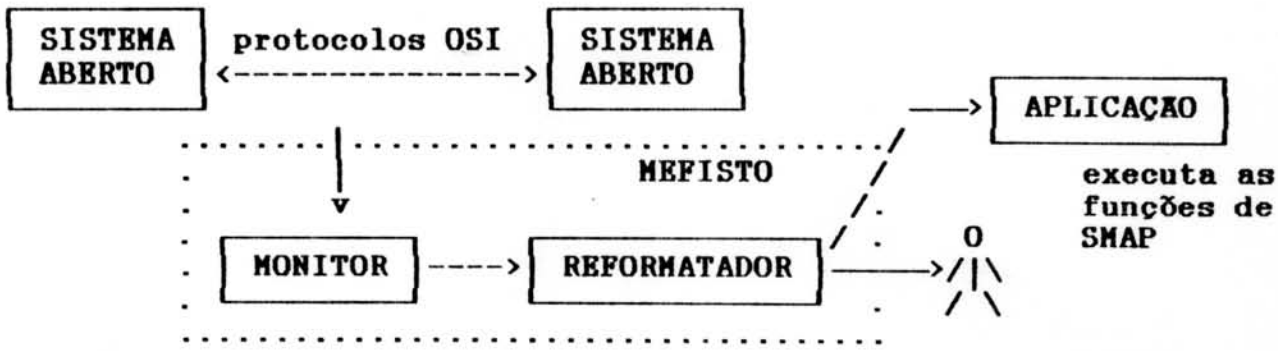
Se um sistema aberto não tem SMAP, um processo aplicativo comum (OAP-Ordinary Application Process) pode interagir com outros SMAPs, cumprindo, de forma limitada, o papel de um SMAP.

Esta limitação será, provavelmente, uma impossibildade de **atuar** sobre o sistema aberto. Esta aplicação que cumpre o papel de processo gerenciador, poderá observar o sistema aberto e relatar a um SMAP remoto sua observações mas não poderá alterar parâmetros do sistema aberto observado. Numa situação como essa, se for configurada a necessidade de alterações no sistema aberto, a intervenção humana seria necessária e o sistema, provavelmente, teria que ser descontinuado para ser alterado e então reiniciado.

É previsível que muitos sistemas abertos sejam colocados em uso

sem um SMAP. Assim, foi projetado e está sendo implantado um sistema de apoio à captação de informações sobre a atuação dos sistemas abertos, a partir de observação externa [96] (com monitoração de suas interações com outros sistemas abertos). Com este suporte será possível obter informações sobre a atividade do sistema aberto que não tem um SMAP e poderá ser implantado um processo de aplicação comum (OAP) que desempenhe esta função (figura 5.3).

Figura 5.3: Captação extrena de dados sobre o sistema aberto



SMAPs usarão a SMAE ou outros servicos de aplicação para transferir as informações entre si. Uma SMAE é composta de vários elementos: Common Application Service Elements (CASE), outros Applications Service Elements e os Managements Application Service Elements (MISE). Para intercâmbio de informação, tanto um SMAP quanto um OAP podem utilizar um SMAE.

Os intercâmbios de informação de gerenciamento são operações ponto-a-ponto, entre dois processos gerenciadores (SMAPs), com um deles fazendo o papel de iniciador e o outro de respondedor. Um SMAP iniciador inicia uma atividade de controle encaminhando um pedido para que uma específica função de controle ocorra. A resposta ao pedido de controle é retornada pelo SMAP respondedor. O iniciador pode também encaminhar um pedido de informação específica. Notificações de eventos são criados e enviados por um iniciador e podem ter seu recebimento confirmado ou não, pelo respondedor.

São definidos dois tipos de MISE. O primeiro, denominado de

Common Management Information Service Element (CMISE), provê um conjunto generalizado de serviços para transferência de informação de gerenciamento e controles (tabela 5.1).

Tabela 5.1: CMISE-Common Application Service Element

M-INITIALIZE, M-TERMINATE e M-ABORT: estabelecer e encerrar ou interromper associações de aplicação
M-EVENT-REPORT: relatar um evento sobre um objeto gerenciado
M-GET: solicitar informações a uma entidade de gerenciamento par
M-SET: solicitar à entidade par a alteração de alguma informação de gerenciamento
M-ACTION: solicitar à entidade par a execução de uma ação
M-CREATE: requisitar à entidade par a criação de outra instância de um objeto gerenciado;
M-DELETE: solicitar à entidade par a deleção de uma instância de um objeto gerenciado

O segundo tipo, denominado Specific Management Service Element (SMISE) é definido em termos dos vários tipos de informação de gerenciamento identificados (gerenciamento de problemas, contabilização, configuração e segurança). A tabela 5.2 apresenta os elementos de serviço específicos, orientados à gerência de problemas.

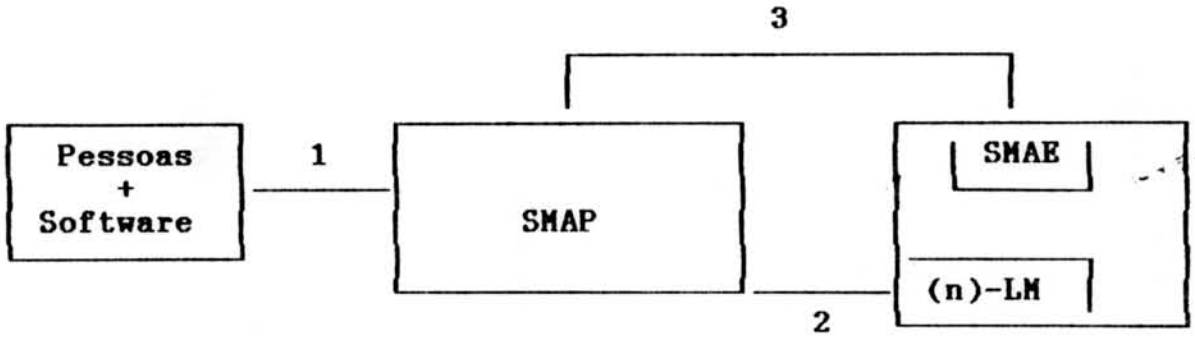
Tabela 5.2: SMISE:Specific Service Application Service Element

FM-ERROR-REPORTING: um SMAP notifica outro SMAP sobre a ocorrência de um erro em seu dominio de gerenciamento;
FM-GET-ERROR-COUNTERS e FM-ZERO-ERROR-COUNTERS permitem a um SMAP reportar estatísticas de erro acumuladas;
FM-SET-THRESHOLD permite enviar alarmas quando determinados liniares são ultrapassados;
FM-INITIATE-TEST é uma função de controle e ocasiona o envio de um comando para o SMAP do sistema que está gerenciando o recurso a ser testado e uma resposta daquele SMAP confirmando sua intensão de fazer algo a respeito. É usado o serviço CMISE M-ACTION para intercambiar a informação de gerenciamento concernente;
FM-TEST-REPORT é uma operação assíncrona relativa ao pedido que invocou o teste. Testes podem tomar uma quantidade de tempo significativa. Outros elementos de serviço relacionados com teste são: FM-TEST-STATUS, FM-TEST-ABORT, FM-TEST-SUSPEND e FM-TEST-RESUME;
FM-TRACE serve para solicitar ao SMAP no sistema destinatário para testar a operação da estação e dos links a ela conectados

A arquitetura OSI não prescreve qualquer distribuição das funções de gerenciamento OSI em particular. Num sistema aberto que admite conexões entre usuários OSI em diferentes sistemas abertos, as funções de gerenciamento são organizadas em domínios. Domínios são associações entre conjuntos de usuários relacionados. Cada domínio de gerenciamento conterá um par, um grupo ou grupos de SMAPs.

Um SMAP pode intercambiar informação de gerenciamento nos modos ilustrados na figura 5.4. Diz-se que ele tem 3 **limiares** de interação com outras entidades

Figura 5.4: Intercâmbio de informação do SMAP



O limiar 1 é o interface entre o SMAP e as pessoas e software que requisitam serviços do SMAP, invocando uma ou mais funções de gerenciamento. Este limiar é de escopo local e não está sujeito à padronização no âmbito dos padrões OSI. No caso em foco, este será o limiar entre o sistema especialista e o SMAP.

O limiar 2 é o interface entre o SMAP e os gerenciadores individuais de cada nível, denominados (n)-LM (layer-managers). Os gerenciadores de cada nível são responsáveis por funções de monitoração, controle e coordenação dos recursos daquele nível. Por exemplo, o SMAP pode passar um parâmetro contendo o limite de retransmissões para o nível de enlace. O fluxo de informações entre o SMAP e o gerenciador de um dado nível conterá: pedidos para ler, setar e efetuar ações concernentes a valores, contadores, status etc. conforme relacionado na seção 2.4. Também conterá respostas a consultas feitas pela entidade de gerenciamento do nível ao SMAP e dados do gerenciamento de nível de outros sistemas. O fluxo de informações do gerenciador de um nível para o SMAP conterá: respostas a pedidos de ler, setar e comandos de ações provenientes daquele SMAP, pedidos para enviar dados ao gerenciador de nível em outro sistema aberto, pedidos para colocar

dados na MIB e pedidos para obter informações da MIB.

O limiar 3 é o interface entre o SMAP e a SMAE. A SMAE é um tipo de entidade de aplicação e é quem se comunica, via protocolo de gerenciamento, com outras SMAEs em outros sistemas abertos. Dados e informações de controle passam através deste interface. O protocolo usado é denominado CMIP-Common Management Information Protocol [45].

As responsabilidades de um SMAP são classificadas em dois contextos: local e globais. As responsabilidades locais incluem inicializar o sistema aberto, servir como meio para o intercâmbio de informação entre os níveis; inicializar o gerenciador de nível em cada nível, após a ativação do sistema, servir como gerente da informação comum a vários níveis ou que é fornecida externamente. As responsabilidades globais envolvem: prover suporte para o intercâmbio de informação entre os gerenciadores de nível de um único nível (de modo que estes não necessitem protocolos separados para tais trocas) coordenar as atividades dos vários SMAPs num conjunto de sistemas abertos.

5.2 Definição das informações da MIB

Os requisitos do sistema de gerenciamento que se pretende implementar é que determinarão as informações de gerenciamento a serem coletadas. Tais informações deverão permitir derivar conclusões sobre a situação dos recursos gerenciados. As informações são especificadas de acordo com os padrões ISO, mesmo quando estes padrões ainda não tenham um caracter definitivo. Foi feita uma seleção de informações que atendesse os requisitos de gerenciamento considerados importantes no gerenciamento de problemas.

5.2.1 Tipos básicos de informação

Em [43 e 47] é apresentada uma relação de tipos básicos de informação de gerenciamento e sua estrutura lógica. A ISO ainda não tem padrão que defina ou especifique completamente os elementos de informação que possam estar presentes numa particular MIB (Manegement Information Base). Uma definição neste sentido foi elaborada no decorrer deste trabalho de pesquisa.

Os tipos relacionados pela ISO são:

-status

-contador

-medidor

-marcador de máximo

-limiar

-informação de tratamento de evento

-informação de controle de relatos

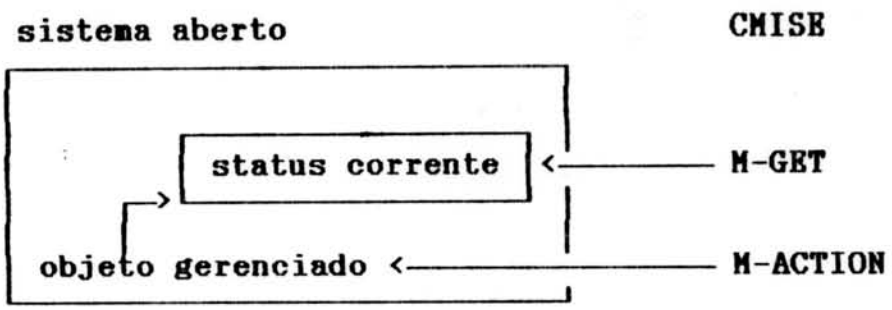
-log

-relacionamento

-informação não especificada

Um elemento **status** consiste de um único item de enumeração de tipo que seleciona um valor num conjunto discreto. O status pode ser lido por meio do serviço M-GET e eventualmente poderá ser alterado pelo serviço M-ACTION. O conjunto do valores possíveis é definido na especificação do objeto gerenciado ao qual o status se aplica. Por exemplo, o status de uma entidade de nível N pode ser o estado em que se encontra na tabela de estados que define seu comportamento. Neste caso, o status poderia ser alterado pelo serviço M-ACTION (tal como ilustrado na figura 5.5) pois pode ser comandado um reinício para uma entidade de nível N que provoque uma troca de estado.

Figura 5.5 Alteração do status



Um status tem um valor corrente, e especificação de propriedades (objeto ao qual se aplica, conjunto de valores que o item pode assumir e ações requeridas para mudar seu estado).

Contadores proporcionam o suporte para a coleta de estatísticas de vários tipos sobre erros, aspectos de performance, contabilização de uso etc... O elemento contador só pode ser lido e pode acumular informação de diversos tipos tais como, erros ocorridos, performance, contabilização etc. Um contador é um item de tipo composto e tem associado a ele os seguintes itens:

-um valor corrente (inteiro),

-propriedades inerentes:

evento interno que é contado,

direção de evolução (ascendente ou descendente)),

valor máximo

um instante de inicialização.

O valor do contador é incrementado de 1 (ou decrementado se é um contador decrescente) sempre que ocorrer um evento interno ao qual está associado. Ele pode assumir qualquer valor até o alcance. Se atinge o alcance (ou 0 em contagem decrescente) ele continua evoluindo na mesma direção e a informação que excede é perdida. O instante de inicialização é um horário. Sempre que um contador for re-inicializado, a informação que detinha será perdida.

Medidores apoiam a monitoração do valor de variáveis dinâmicas, tal como o número de conexões num SAP (Service Access Point). O elemento medidor também só pode ser lido e visa permitir a monito-

ração de uma variável dinâmica, tal como, o número de conexões num SAP. Também é um elemento composto sendo integrado pelos seguintes itens: variável dinâmica que é medida, valor corrente (inteiro sem sinal) e alcance (valor mínimo e máximo medido). Medidores são incrementados ou decrementados pela ocorrência de eventos internos com o que estiverem associados. O incremento não precisa ser apenas 1 e ao atingir o alcance o medidor fica parado naquele valor, não ultrapassando-o.

Um **marcador de máximo** registra o valor máximo (ou mínimo em caso de contadores decrescentes) atingido por uma variável dinâmica (por exemplo, o número máximo de conexões num dado SAP). Medidores de máxima ou de mínima proveem um mecanismo para registrar o valor máximo ou mínimo atingido por uma variável dinâmica durante um período de medida. São itens que somente podem ser lidos embora possam ser também reinicializados. São itens compostos dos seguintes tipos de elementos:

- informação à qual se aplica (variável dinâmica medida),
- valor corrente (inteiro),
- direção (para cima ou para baixo),
- alcance (valor mínimo e máximo)
- valor de reinicialisação (inteiro sem sinal dentro do alcance)
- hora de reinicialização.

Limiar é o mecanismo genérico para gerar eventos a partir de mudanças nos valores de status, contadores, medidores ou marcador de máximo. É composto por:

- um identificador do item ao qual se aplica,
- um estado (ativo ou inativo)
- um modo de operação (único ou contínuo),
- um critério de comparação (<, <=, =, >= ou >),
- o valor corrente da comparação (quando passa de falso para

verdadeiro determina a geração do evento associado a este limiar),

- valor de recarga para o limiar (determinando novo limiar a ser considerado, mais restringente ou mais tolerante em relação ao anterior),
- um identificador do evento associado.

O limiar é o mecanismo geral para gerar eventos internos a partir de mudanças nos demais elementos. Um limiar pode ser associado a um status, a um contador, a um medidor etc. Quando o limiar está no estado inativo nenhum evento é gerado por este mecanismo. O critério de comparação define as circunstâncias em que o evento decorrente será gerado (quando o valor do item ao qual se aplica for <, <=, =, >= ou > que o valor corrente do limiar). No caso de um limiar aplicado a um contador, o valor de deslocamento é usado para registrar o valor inicial do contador. Quando a comparação entre o valor do item e o limiar não atende mais ao critério adotado, um evento é gerado. Uma vez que o evento tenha sido gerado o valor de recarga do limiar é usado para recarregá-lo. Se a operação do limiar era de informação única (um só tiro) o estado do limiar passa a inativo. No modo contínuo o valor de recarga do limiar permite um segundo e outros eventos serem gerados, com critérios que podem diferir do inicial, permitindo operação em forma diferente na qual os critérios sejam mais ou menos rigorosos.

A **Informação de tratamento do evento** é associada com todos os eventos internos e especifica a identificação do evento e ação(ões) interna(s), a serem disparadas quando o evento ocorrer. Estas ações podem incluir: envio de relato a destinatários especificados, incrementar contadores, decrementar contadores, mudar o status de valores, registrar (log) o evento; ou seja, a ação pode conter uma lista de ações de comprimento arbitrário.

A **informação de controle de relato** especifica para onde e com

que conteúdo deve ser enviado um relato a ser enviado em decorrência da ocorrência de algum evento interno. Assim, o elemento informação de controle deve conter os seguintes itens: lista de destinação (lista de endereços aos quais o relato deve ser enviado (endereços de P-SAP, títulos de AEs e nomes internos de logs) e lista de conteúdo (lista de identificadores dos itens de informação que constituirão o relatório). Esta estutura sendo passível de modificação por operações M-SET permitirá adaptação dinâmica e controle do nível de relato emitido.

O **log** é um registro de eventos que tenham ocorrido em algum período de tempo. Os particulares eventos cuja ocorrência é registrada no log são determinados pela informação de tratamento dos eventos. Um elemento de log tem só um item de informação que é uma lista de identificadores de eventos. Este item pode ser lido como um todo, por meio do serviço M-GET ou pode constituir parte de um relato (se consta na informação de controle do relato). O log também pode ser reinicialisado, isto é, esvaziado por meio de uma operação M-ACTION.

O **relacionamento** indica uma forma de relacionamento entre dois recursos. Deve incluir para cada possível relação o objeto com o qual existe a relação e o tipo de relação.

A **informação** não especificada é um elemento de informação que pode ter qualquer definição desejada por quem quiser utilizá-la. permite a definição de quaisquer outras informações requeridas numa particular implementação.

5.2.2 Identificação dos objetos gerenciados

A identificação dos objetos gerenciados será composta por três sub-componentes:

- o identificador do sistema

- um componente reservado

- um identificador específico.

O **identificador do sistema** é em geral também hierarquisado em componentes e, se for incluida, a designação do domínio, um sistema aberto real pode ter mais de um identificador, correspondendo aos diferentes planos de enfoque sob os quais pode estar sendo visto num dado instante. O componente reservado pode ser usado para designar separadamente objetos que não sejam parte do sistema aberto cujo identificador é a raiz da hierarquia.

O **identificador específico** tem alguns sub-componentes: identificador do nível e identificador de padrão. Quando o identificador de nível identifica um nível OSI (pelo seu número), o identificador de padrão identifica o particular padrão (padrão ISO ou recomendação CCITT) em que o objeto é especificado. Isto pressupõe que sub-divisões adicionais serão as especificadas por aqueles padrões. Quando o objeto gerenciado não é classificado como pertencente a um particular nível, o identificador de nível é setado como nulo.

5.2.3 Especificação dos objetos gerenciados

Cada componente ou sub-componente do identificador do objeto gerenciado é especificado por declaração de atributos [46]. Uma declaração de atributos é uma combinação de um tipo de atributo e um valor de atributo. Uma ou mais declarações de atributos identificam um objeto gerenciado no sistema aberto. Quando informação concernente a um objeto gerenciado é intercambiada, mediante o uso de protocolos de gerenciamento de rede, a identificação do objeto gerenciado pode ser feita de forma incompleta se as declarações de atributo omitidas podem ser deduzidas pelo contexto.

As informações de gerenciamento sobre os objetos gerenciados são identificadas por um identificador do tipo de informação de

gerenciamento e/ou por uma chave primária. O identificador do tipo é uma declaração de atributo de tipo NULO e valor igual a um dos definidos na tabela que os descreve. Na sua ausência uma consulta solicitada se aplica a todos os tipos de informação de gerência associados com o objeto gerenciado identificado. A chave primária identifica um sub-conjunto de elementos de informação de um dado tipo ou de todos os tipos (se esta identificação está ausente). A chave primária consiste de uma ou mais declarações de atributo, cada uma consistindo de um nome de campo (tipo do atributo) e o correspondente valor (valor do atributo). Quando a chave primária está ausente a consulta aplica-se a todas as instâncias de informação de gerenciamento associadas ao objeto gerenciado.

5.3 Informações requeridas para gerenciar a camada de transporte

A especificação da MIB para o sistema ora proposto foi orientada ao objetivo de poder gerenciar os problemas relacionados na seção 3.1.3. Nesta seção serão apresentadas os objetos gerenciados e seus atributos selecionados.

Um **objeto gerenciado** é um componente operacional do ambiente OSI, cuja configuração, status, comportamento e uso é descrito pela informação contida na MIB. Os objetos gerenciados são classificados em categorias para as quais são definidas propriedades comuns que instâncias específicas de um objeto gerenciado herda.

A análise que levou à seleção dos objetos gerenciados e seus atributos visou definir um conjunto mínimo de atributos capazes de apoiar o gerenciamento pretendido pois sabe-se que um número muito elevado de objetos gerenciados torna mais cara e difícil a implementação dos mecanismos inerentes à sua manipulação.

Assim, os critérios usados na seleção levaram em conta:

-Grau de necessidade daquele objeto gerenciado e seus atributos para fins de gerenciamento de problemas nos termos definidos em 2.4.3;

-Evitar redundância de informações evitando a inclusão de objetos que possam ser derivados de outros.

-Minimizar o registro de informações que tivessem que ser feitos durante a execução de seções críticas da atividade da entidade de transporte.

Conforme anteriormente justificado, os objetos gerenciados foram definidos usando ASN.1. Cada tipo de objeto tem um **nome**, uma **sintaxe** e uma **codificação**. O **nome** é representado de acordo com as regras definidas em [43]. A **sintaxe** para um objeto define a estrutura abstrata correspondente àquele tipo de objeto (inteiro, cadeia de caracteres etc...). A **codificação** especifica os valores que podem ser associados a instâncias do objeto sendo definido.

Adicionalmente, foram agregados alguns comentários à definição para explicar o significado dos códigos.

Algumas **definições básicas** usadas na especificação da sintaxe dos objetos selecionados são as seguintes:

```
DiaHora::= SEQUENCE {dia IMPLICIT INTEGER {--2 dígitos--},
                     mes IMPLICIT INTEGER {--2 dígitos--},
                     ano IMPLICIT INTEGER {--4 digitos--},
                         HoraGenerica  }
HoraGenerica ::= SEQUENCE {
                     hora INTEGER { -- <= 24 },
                     minuto INTEGER { < 60 },
                     segundo INTEGER { < 60 },
                     centesimoDeSegundo { <100 }
```

```
Filtro ::= CHOICE { igual [0],
                    maior [1]
                maiorOuIgual [2],
                    menor [3],
                menorOuIgual [4],
                maiorOuMenor [5]  }
```

Informações sobre a entidade de transporte como um todo:

```
EntidadeDeTransporte ::= SEQUENCE {
  classesAceitas            SET OF {Classe}
  tamanhoMaximoTPDU         INTEGER,
  numeroMaximoConexoes      INTEGER,
  temporizadorJanela        INTEGER {--em centesimos de segundo},
  statusEntidadeTransporte  StatusEntidadeDeNivel
  grauDeImportancia         INTEGER {(0) normal, (1) critico}
  identificadoresTSAP       SET OF { OCTET STRING }
  contadoresAssociados      SET OF {Contador},
  medidoresAssociados       SET OF {Medidor}   }

Classe ::= CHOICE { naoOrientadoAConexao [1],
                    orientadoAConexao    [2] INTEGER {0,1,2,3,4}
StatusEntidadeDeNivel ::= INTEGER CHOICE {operacional(0),
                                          inoperante (1)}
Contador  ::= SEQUENCE {
  eventoContado       TipoContador,
  inicioContagem      DiaHora,
  valorDoContador     INTEGER,
  direcaoEvolucao     INTEGER {ascendente (0) default,
                               descendente (1) }
      valorMaximo [1] INTEGER OPTIONAL }
```

```
TipoContador ::= CHOICE {
  conexoesNormais                 [1] INTEGER,
  desconexoesNormais              [2] INTEGER,
  desconexoesanormais             [3] INTEGER,
  conexoesRecusadas               [4] INTEGER,
  errosNaoEspecificados           [5] INTEGER,
  tpduDescartadosPorErroChecksum [6] INTEGER,
  tPDUTransmitidas                [7] INTEGER,
  tPSURecebidas                   [8] INTEGER,
  tPDURetransmitidas              [9] INTEGER,
  octetosTransmitidos             [10] INTEGER,
  octetosRetransmitidos           [11] INTEGER,
  octetosRecebidos                [12] INTEGER,
  tPDUsInvalidos                  [13] INTEGER,
  errosDeProtocolo                [14] INTEGER }
Medidor ::= SEQUENCE {
  eventoMedido        TipoMedidor,
  inicioMedida        DiaHora,
  valorDoMedidor      INTEGER,
  valorMaximo     [1] INTEGER OPTIONAL }
TipoMedidor ::= CHOICE {
  conexoesAbertas                 [1] INTEGER,
  tamanhoFilaEnvio                [2] INTEGER,
  tamanhoFilaEsperaConfirmacao    [3] INTEGER,
  tempoEstabelecimentoConexao     [4] INTEGER, --em centésimos de
                                                  segundo
IdentificadorLimiar ::= SET OF { SEQUENCE {
  contadorAssociado   TipoContador,
  estado              BOOLEAN --TRUE significa ativo
  modoOperacao        INTEGER {continuo (0), umTiro (1)},
                      -- ocorre somente um aviso de ultrapassagem
```

```
                        do limiar  ou vários, de forma continua
  criterioDeComparacao Filtro,
  estadoCorrente       BOOLEAN,  -- FALSE significa não ultrapas.
  valorDoLimiar        Limiar  }
Limiar  ::= CHOICE { INTEGER, SEQUENCE OF INTEGER }
```

Informações sobre as conexões

```
ConexaoTransporte ::= SEQUENCE {
       nivel           INTEGER {4},
       dadosDaConexao Conexao }
Conexao ::= SEQUENCE {
       usuarios            SET OF {IdentificadorUsuarioSAP },
       horaInicioDaConexao HoraGenerica,
       contadoresDaConexao SET OF {Contador },
       referenciaDaConexao OCTET STRING OPTIONAL  }
IdentificadorUsuarioSAP ::= SEQUENCE {
       nivel INTEGER {1,2,3,4,5,6,7},
       nomeUsuario OCTET STRING
       statusDaConexao  StatusConexao   }
StatusConexao ::= CHOICE {inexiste (0), ativa (1)}
```

Informações sobre os problemas detectados pela LME do nível de transporte

```
IncidenteRelatado ::=  SET OF {InformacaoErro}
InformacaoErro  ::= SET {
       [0] CausaProvavel OPTIONAL,
       [1] Severidade OPTIONAL,
       [2] IndicacaoTendencia OPTIONAL,
       [3] StatusDegradado OPTIONAL,
       [4] Informacao Diagnostico ANY OPTIONAL,
```

```
        [5] AcaoReparadoraProposta ANY OPTIONAL,
        [6] InformacaoLimiar OPTIONAL,
        [7] MudancaEstado OPTIONAL,
        [8] outraInformacao ANY OPTIONAL }
CausaProvavel ::= CHOICE {
        [0] ComunicacaoCausaErro,
        [1] QOSCausaErro,
        [2] ErroProcessamentoCausaErro,
        [3] EquipamentoCausaErro,
        [4] AmbienteCausaErro }
QOSCausaErro:= INTEGER { tempoRespostaExcessivo       (0),
                         tamanhoFilaExcedido          (1),
                         janelaReduzida               (2),
                         taxaRetransmissoesExcessivas (3)}
ComunicacaoCausaErro ::= INTEGER {
                falhaEstabelecimentoConexao (0),
                errosResiduais              (1),
                conexaoDescontinuada        (2)}
ErroProcessamentoCausaErro ::= INTEGER {
                erroDeProtocolo (0) }
Severidade ::= INTEGER {
        tolerável     (0), -- indice de disponibilidade > .9
        intermitente  (1), -- índice de disponibilidade > 0.
        inoperante    (2), -- índice de disponibilidade = 0 mas
                           -- passível de ser reinicializado
        irrecuperavel (3) }-- não reinicializável
IndicacaoTendencia ::= BOOLEAN OPTIONAL, -- TRUE indica piorando
StatusDegradado ::= BOOLEAN  -- TRUE indica que ja houve reducao
                             -- no nivel de servico
InformacaoSobreEvento ::= SEQUENCE {
```

```
        tipoDeErro  IncidenteRelatado,
        horaAtual   HoraGeneralizada}
Erro ::= SEQUENCE {TipoErro,
                   ValorContadorErro,
                   HorarioInicializacao}
RelatoDeEvento ::= SEQUENCE {
  identificacaoDoEvento  TipoEvento,
  horaDoEvento           HoraGenerica,
  parametroEvento        SET OF {Parametro}
Parametro ::= CHOICE {
  identificacao   [1] CodigoIdentificadorTipo,
  caracteristicas [2] CodigoTipoCaracteristica,
  status          [3] CodigoTipoStatus,
  eventoLog       [4] CodigoTipoLog,
  contador        [5] CodigoTipoContador,
  acao            [6] CodigoTipoAcao          }
LogEvento ::= SEQUENCE {
  tipoLog          FiltroDeEvento,
  horaInicioLog    HoraGenerica,
  entradasLog      SET OF LogEvento     }
FiltroDeEvento  ::= SET OF SEQUENCE {
  menorCodigoEvento  TipoEvento,
  maiorTipoEvento    TipoEvento    }
```

Informações sobre testes

```
Teste::= SEQUENCE {
  tipo ANY --compativel com a entidade que o executara --,
  parametrosTeste OCTAL STRING,
  instanciaDoTeste INTEGER,
  StatusTeste,
  resultadoTeste ANY {--definido em funçao do teste--}  }
```

```
StatusDoTeste ::= INTEGER {naoConhecido(0),
                           iniciado (1),
                           completado (2)}
```

Informações sobre atividade de manutenção relatada pelo operador da rede

```
RelatoManutencao ::= SEQUENCE {
        IdentificacaoManutencao,
        IdentificacaoObjetoGerenciado,
        inicioManutencao DiaHora,
        terminoManutencao DiaHora,
        TipoManutencao,
        outrasInformacoes ANY,
        ComponentesAfetados ,
        StatusManutencao}
TipoManutencao ::= SEQUENCE {
        reinicializar       [1] Reinicializar OPTIONAL,
        substituir          [2] Substituicao OPTIONAL,
        redefinirParametros [3] ParametrosRedefinidos OPTIONAL,
        consertar           [4] Conserto OPTIONAL,
        removerComponente   [5] Remocao OPTIONAL,
        adicionarComponente [6] Adicao OPTIONAL }
ComponentesAfetados ::= {SET OF {IdentificadorObjetoGerenciado}}
StatusManutencao ::= INTEGER {
        solicitada (1) OPTIONAL,
        completada (2) OPTIONAL,
        pendente   (3) OPTIONAL,
        postergada (4) OPTIONAL}
```

```
Reinicializar ::= SET OF SEQUENCE  {
            objetoReinicializado  IdentificacaoObjetoGerenciado,
            valorReinicializacao  INTEGER  OPTIONAL }  --Valor  de
recarga possivelmente definido para o objeto reinicializado
Substituicao ::= SET OF SEQUENCE  {
            objetoSubstituido IdentificacaoObjetoGerenciado,
            objetoSubstituto  IdentificacaoObjetoGerenciado }
Conserto ::= SET OF SEQUENCE {
            objetoConsertado  IdentificacaoObjetoGerenciado,
            autorConserto     IdentificacaoPessoal }
ParametrosRedefinidos ::= SEQUENCE OF {
         Parametro          AtributoObjetoGerenciado,
         valorAnterior      INTEGER,
         novoValorParametro INTEGER  }
```

O conjunto de informações acima relacionado constitui apenas um conjunto inicial e não o conjunto definitivo. Contudo as informações previstas são suficientes para que o SMAP possa avaliar se os valores que entram no cálculo da qualidade do serviço estão aquém do limiar. Estes parâmetros são os discutidos na seção 3.1.1, que permitem calcular:

-retardo no estabelecimento da conexão

-probabilidade de falha no estabelecimento

-throughput

-retardo de trânsito

-taxa de erros residual

-probabilidade de falha na transferência

-retardo na liberação

-probabilidade de falha na liberação

-resistência

Considerando que a metodologia de trabalho é baseada em proto-

tipagem e acréscimo de funcionalidades de forma incremental, espera-se que outras informações irão sendo agregadas à definição da MIB, na medida em que for sendo detectada sua necessidade. Com os objetos e atributos já definidos a LME e o SMAP podem monitorar a operação da entidade de transporte e gerar as devidas notificações, quando os limiares forem excedidos ou quando ocorrer algum dos erros previstos. Tais informações seriam enviadas para o operador da rede que poderia consultar sobre outras informações disponíveis na MIB para optar por algum procedimento paliativo/corretivo quando percebe problemas na rede. Dado o alto volume de eventos relatados, possivelmente o operador da rede termine por definir filtros que impeçam o relato de uma boa parcela dos eventos. Isto pode ocasionar problemas pois eventos que deveriam ter sua atenção podem ser filtrados por uma definição inadequada dos critérios de filtragem. Por exemplo, o operador não quer ser informado se houve retransmissão intermitente de DT-TPDU mas precisaria ser informado se a quantidade de retransmissões está aumentando e ultrapassa o limiar estabelecido para este contador.

Na seção 6.3 é apresentado um conjunto significativo das regras definidas para o sistema especialista orientado ao gerenciamento de problemas com a entidade de transporte num contexto OSI.

Estas informações serão passadas para o Sistema Especialista para que este, acione a máquina de inferência e derive conclusões e recomendações que apresentará ao operador da rede.

6 - SISTEMA ESPECIALISTA PARA GERÊNCIA DE PROBLEMAS EM REDES OSI

As redes de computadores podem ser sub-divididas em dois segmentos distintos: uma sub-rede de transporte (uma rede operando com protocolo X.25 por exemplo), e uma rede periférica composta de processadores (ou aplicações) interconectadas via esta sub-rede. Cada segmento tem sua própria função supervisora: o sistema especialista integra informação provenientes de ambos. Os eventos que chegam ao sistema especialista podem ser alarmes de redes espontâneas, resultados e teste, ou informação fornecida manualmente pelo operador. O sistema analisa cada evento e quando detecta um mau funcionamento adverte o supervisor dando orientação e indicação para as tarefas manuais que necessitam ser executadas.

A supervisão de uma rede de computadores grande usualmente é efetivada a partir de um centro de controle, conforme descrito no capitulo 1. Todos as exceções ocorridas na rede são reportadas a este centro de controle através da própria rede. Esses eventos devem ser interpretados pelos operadores humanos para detectar e diagnosticar o problema. Neste contexto o problema é uma situação anormal séria o suficiente para perturbar a comunicação entre os componentes da rede. A complexidade da detecção de problemas é devida a vários problemas, destacando-se a correlação de eventos, segundo [66]. Isto advém do fato de que um evento isolado nem sempre é significativo. Uma condição anormal frequentemente gera um grande número de eventos, cada um contendo uma parcela da informação. A análise da situação e sua caracterização devem ser feitas por um processo que integre esses vários eventos. Numa primeira vista, a correlação dos eventos pode parecer simples, com os eventos reportados sendo divididos em grupos e componentes relacionados; a sequências de eventos resultantes será então comparada com as sequências padrões caracterizando um problema então levando a

uma conclusão sobre o problema. Mas, nem sempre é assim tão simples, devido a uma série de razões tais como:

- Correlação com o tempo: Os eventos relativos a uma dada situação são dispersos num intervalo de tempo que pode ser longo: horas, mesmo dias para alguns problemas de degradação. Além disso, a sequência em que tais eventos ocorrem pode ter importância; por exemplo flutuação no comportamento de um modem em uma hora não é séria mas dez vezes num minuto é.
- Correlação espacial: Uma situação anormal pode induzir erros em vários componentes da rede: no componente com problemas, em componentes relacionados hierarquicamente e em componentes interconectados através da rede. O problema algumas vezes só pode ser detectado indiretamente, a partir dos eventos relativos relatados por outros componentes e não o componente que está com problemas.
- Eventos redundantes: Muitos eventos são uma consequência direta de outros e não dão qualquer informação adicional. Eventos podem ser relatados através da rede mesmo depois que o problema tenha sido consertado. Tais eventos somente distraem os operadores e devem ser subtraídos.

Para solucionar os vários aspectos na correlação de eventos especialmente no que tange ao correlaçãos espacial o sistema especialista necessita conhecimento estrutural:

- Sobre a rede incluindo seus componentes e as relações entre eles
- Sobre os eventos que podem ocorrer para estes componentes

Além destes fatores, existem outros, que também devem ser considerados no projeto de um sistema especialista para gerência de redes, tais como os comentados a seguir:

a- Capacidade de atuação com dados incompletos

A resolução do problema em redes sempre começa e frequentemente prossegue com informação incompleta. A informação é obtida incrementalmente à medida que mais eventos são reportados. A interpretação dos eventos nem sempre leva a uma conclusão definitiva: eventos podem ser perdidos ou não reportados porque o centro de comunicação, o centro de controle ou a comunicação estão paralisados e diferentes problemas algumas vezes iniciam com o mesmo conjunto de eventos. Na maioria dos casos, somente se pode derivar hipóteses sobre uma dada situação, quando os eventos que vão ocorrendo adicionam informação e, neste caso, essas hipóteses devem ser reconsideradas ou refinadas. Então o mecanismo de inferência deve ter raciocínio permitindo a possível revisão das suas crenças. A revisão de uma crença não é somente ocasionada por eventos mas também pela progressão de tempo. A ausência de eventos pode algumas vezes ser tão significante quanto seu aparecimento e deve ser interpretado como tal. Por exemplo, eventos que levem a suspeitar de alguma degradação de um componente implicam em que outros eventos devam ocorrer em algum espaço de tempo posterior para confirmar a degradação. Se nenhum evento é reportado a hipótese de degradação deve ser removida.

b- Manipulação de múltiplos problemas

Qualquer rede de computadores usualmente tem vários problemas pendentes não resolvidos. Eventos correspondentes gerados para cada um deles serão entremeados. Algumas situações anormais são mais críticas que outras e devem ser atendidas primeiro. Uma vez que a informação é obtida progressivamente o raciocínio relativo a um dado problema é adquirido em vários estágios descontínuos no tempo e ocasionados pela chegada dos eventos. Entre esses instantes, a atividade de raciocínio pode ser focalizada em outras situações anormais. Em cada estágio um mecanismo deve registrar todas dedu-

ções já feitas para habilitar o raciocínio a seguir adequadamente.

c- Explosão de eventos

O número de eventos aumenta rapidamente com o tamanho da rede. Essa situação implica em requisitos de eficiência para um mecanismo de raciocínio quando a rede tiver que funcionar em tempo real.

Para atentar para todos estes aspectos, o projeto do sistema especialista foi elaborado tendo em vista as seguintes preocupações:

a- Representação do conhecimento

Três tipos de conhecimento foram considerados:

- Conhecimento estrutural

- Deduções: os tipos de dados criados e manipulados durante as atividades de raciocínio

- Conhecimento sobre a detecção do problema diagnóstico que especifica como interpretar os eventos de rede, como reconhecer situações problemáticas e como isolar componentes com problema.

O conhecimento estrutural sobre a rede e os eventos da rede tendem por si próprios a uma organização hierárquica que permite a herança de propriedades. Cada tipo de objeto de rede e de evento de rede é representado por uma classe numa hierarquia de classes. Uma classe filha é considerado ser uma especialização na classe pai. A classe filha herda as propriedades do pai mas pode adicionar ou alterar as propriedades. Alguns tipos de objetos incluídos na hierarquia são:

Componentes da rede

Propriedades representando as relações entre os objetos (tal como o relacionamento componente/sub-componente ou o relacionamento entre objetos que são fisicamente interconectados);

Propriedades são usadas pelo processo de raciocínio tal como status, que é um resumo da situação corrente do objeto. **Eventos da**

rede

As principais propriedades servem para identificar um evento no tempo e no espaço (hora foi informado, hora em que a informação foi recebida, máquina que enviou, etc) e para definir as características do evento (por exemplo, limiar ou intervalo)

As deduções podem pertencer a três tipos básicos: sintomas, hipóteses e resultados. O sintoma representa um conjunto de eventos que pode ser requerido para futura manipulação. Sintoma e evento diferem em sua utilização e em sua interpretação. Um evento ocorre num determinado momento de tempo e desaparece. Um sintoma é registrado para futura utilização no processo de raciocínio. Sintomas se alteram ao longo do tempo. Desta maneira eles tem uma hora de início uma hora de parada. Isto permite que sintomas representem uma sequência de eventos de um mesmo tipo e que tenham ocorrido durante o intervalo de tempo.

6.1 O desenvolvimento do sistema especialista

No decorrer deste trabalho foi definido um sistema especialista para gerência de rede orientado aos problemas identificados em 3.1.3. Este projeto foi denominado SEREIA-Sistema Eficaz de Rede Empregando Inteligência Artificial.

Conforme referido no capítulo 4, o primeiro passo no desenvolvimento do sistema especialista consiste na identificação do conhecimento a ser incorporado no sistema e determinar o modo em que deve ser estruturado. Como tanto a metodologia de projeto e o ambiente de desenvolvimento eram novos o projeto começou com a implementação de um protótipo de sistema especialista bem simples, orientado a problemas ocorridos em linhas de comunicação de dados numa rede de teleprocessamento [109]. Suas caracteríticas são descritas na próxima seção.

6.1.1 A versão zero do projeto SEREIA

O projeto SEREIA (Sistema Eficaz de Rede Empregando Inteligência Artificial) visa o desenvolvimento de ferramentas para apoiar a gerência da rede num âmbito inicialmente mais restrito, voltado à rede de Teleprocessamento da UFRGS, que é baseada no computador A-10 da UNISYS, pertencente ao CPD da UFRGS. O sistema se propõe a analisar os diagnósticos e estatísticas de erros provenientes deste sistema de comunicação de dados.

A comunicação entre o computador A-10 (sistema central) e os componentes remotos (terminais,impressoras) é realizada através do subsistema de comunicação de dados (Data Comm Subsystem). O Data Comm Subsystem é composto por processadores "front-ends" chamados Network Support Processor (NSP) e Line Support Processor (LSP), que são responsáveis por funções tais como:

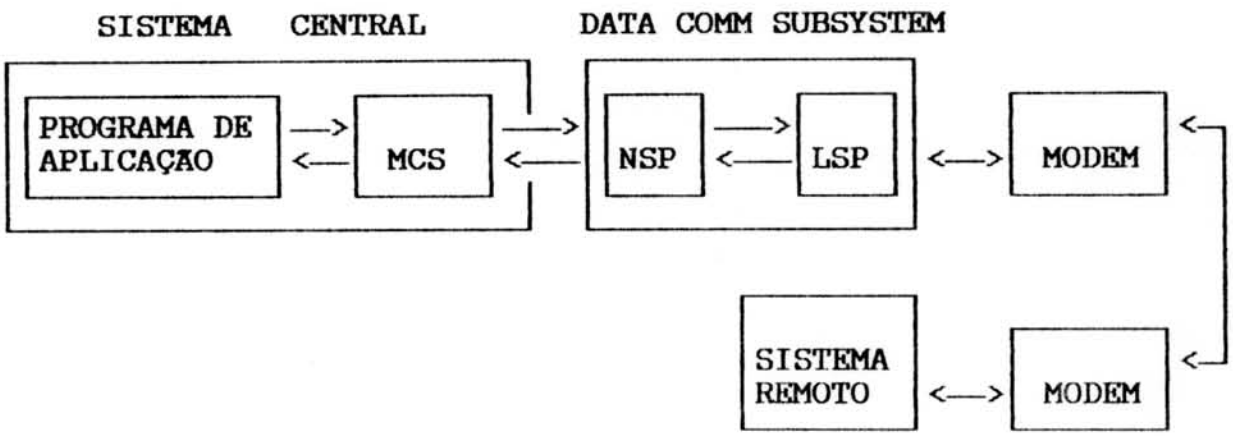
- Executar o algoritmo referente ao protocolo POLL-CONTENTION;
- Manipulação de linhas multiponto;
- Controle de modem e equipamento de chamada automática;

Os processadores NSP e LSP são programados usando-se NDL II - Network Definition Language II, uma linguagem de programação e definição de alto nível que é usada para definir a rede de comunicação física, logica e funcionalmente.

No sistema central existem elementos de software que controlam o fluxo de mensagem com o Data Comm Subsystem, que são chamados de Message Control System (MCS), escritos em DCALGOL. Cada estação pertencente ao A-10 deve possuir um MCS controlador.

A tragetória dos dados do sistema central ao sistema remoto e mostrado na figura 6.1.

Figura 6.1: Tragetória dos dados



A ligação entre modens é half-duplex com dois fios e é realizada por três meios na UFRGS:

- cabo interno no próprio predio do CPD
- cabo externo privado da UFRGS interligando os campi
- cabo alugado da companhia telefônica .

Adicionalmente é usado em âmbito local uma conexão direta, sem modem (TDI-Two Wire Direct Connect)

O programa que controla o Data Comm Subsystem está preparado para detecção de erros e tentativas de recuperação. Quando as tentativas falharem um número N de vezes, uma mensagem será passada para o respectivo MCS da estação que originou o problema e este passará a mensagem de erro para um LOG (LOGDCP) junto com a data, horário e número da estação envolvida.

Existem vinte e dois tipos de erros que podem ser sinalizados na rede, no entanto, apenas oito são mais frequentes e são relacionados a seguir, com uma breve explicação de seu significado:

-VERTICAL PARITY: ocorre quando o número de bits "1", incluindo o bit de paridade, foi impar numa conexão assíncrona ou par numa conexão síncrona.

-HORIZONTAL PARITY: o caracter de paridade longitudinal (BCC-Block Check Character), num texto recebido, não estava

correto.

-TIMEOUT: foi excedido o tempo de resposta em uma recepção, ou o tempo entre a recepção de um caracter e outro na linha.

-STOP BIT: em uma conexão assíncrona houve perda do STOP BIT.

-BREAK ON INPUT: a execução de um comando de recepção encontrou a linha (Física) em condição de espaço por um tempo maior que 2,5 caracteres.

-FORMAT ERROR: em um comando de recepção, o caracter recebido não era o esperado (não estava de acordo com o protocolo)

-LOSS OF CARRIER: perda do sinal DCD (Data Carrier Detect) no modem.

-DISCONNECT: perda do sinal DATA SET READY.

A simples indicação da ocorrência de um dos erros não esclarece a causa da ocorrência. As vezes pode haver problemas transientes, que podem ser ignorados. Outras vezes ocorrem eventos que se confundem com um problema mas que foram provocados pela operação de um equipamento numa forma diferente do previsto no NDL II. Por exemplo, basta que o modem do sistema central seja desligado para que o computador central registre no log o erro de DISCONNECT.

Por outro lado, dependendo da periodicidade com que ocorrem os eventos e das condições de controle existentes o(s) evento(s) sinalizado(s) pode(m) indicar necessidade de manutenção. Idealmente, isto deveria ser percebido pela equipe de gerência da rede e as devidas providências deverão ser tomadas, antes que o usuário reclame.

No estágio em que o projeto SEREIA se encontra, não se tem condições de detectar todos os problemas comuns em teleprocessamento por falta de informações no LOGDCP e no DCSTATUS. Mas com o desenvolvimento do projeto MEFISTO [96] pretende-se ampliar as formas de sensoriamento da rede, buscando informações em outros pontos

tais como:

-Analisar a saída produzida por um analisador de protocolos para avaliar tempos de resposta e carga de transmissão na linha.

-Tratamento de mensagens de erro dentro do MCS para avaliar erros de software (programa de aplicação que não está disponível ou que cai ou tentativa de acesso não autorizado) ou ainda usuário mal treinado.

A análise de todos os eventos é complexa e requer investigação cuidadosa, em várias fontes (manuais,programas fontes) além de uma boa experiência em teleprocessamento, que permita identificar a causa provável dos eventos anormais sinalizados. Este know-how somente existe em pessoas com muitos anos de experiência, as quais, nem sempre estão disponíveis. Com este projeto, foi iniciado um processo de transferência deste conhecimento (ou pelo menos de uma parte relevante dele), para um sistema especialista que fosse capaz de prover recomendações equivalentes para solucionar os problemas da rede. A próxima seção descreverá alguns aspectos da construção deste sistema especialista.

6.1.1.1 O ambiente de implementação do SEREIA na fase zero

O ambiente ARITY [2] foi o escolhido para servir de suporte ao desenvolvimento do sistema SEREIA, porque possui a flexibilidade de, a partir do seu interpretador PROLOG, anexar módulos funcionais tais como ARITY/EXPERT, um "shell" para construção de sistemas especialistas e ARITY/SQL, uma linguagem para a manipulação de um banco de dado relacional.

Os módulos (ARITY/EXPERT e ARITY/SQL) são anexados ao interpretador PROLOG em tempo de ligação dos módulos. Um programa PROLOG pode usufruir de todos os serviços prestados por estes módulos,

através de predicados especiais pré-definidos.

A principal característica do ARITY/EXPERT é o uso concomitante de duas formas de representação de conhecimento: frames e regras de produção.

A representação em frames determina a relação entre conceitos do domínio do problema, junto com seus atributo e respectivos valores. Usa-se uma linguagem específica deste ambiente de desenvolvimento para descrever estes relacionamentos. A figura 6.2 apresenta a definição do conceito REDE com atributo MEIO_DE_LIGAÇAO e seus possíveis valores, usando a linguagem do ARITY/EXPERT.

Figura 6.2: Exemplo de definição de um conceito

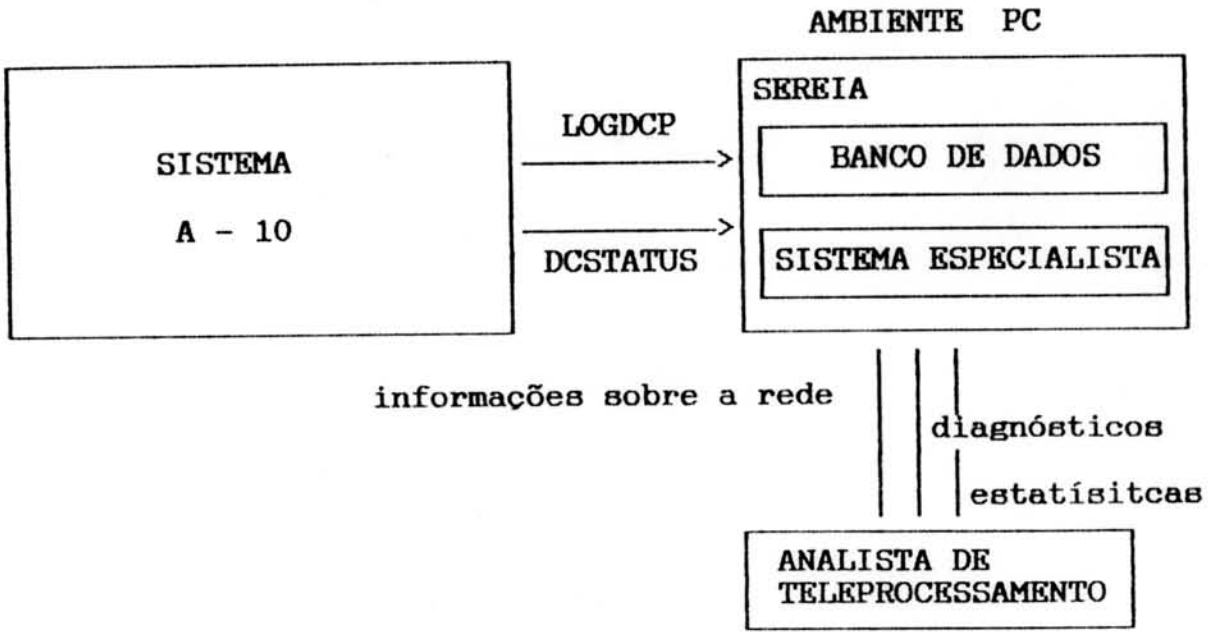
```
define REDE with
  MEIO_DE_LIGAÇÃO=[TDI,MODEM_INTERNO,MODEM_EXTERNO_CPD,
                   MODEM_EXTERNO_RP].
```

Todos os elementos usados nas regras devem ser previamente declarados em forma de frames.

6.1.1.2 Características da implementação feita

O sistema SEREIA foi implementado num microcomputador PC. Os dados colhidos pelo sistema A-10, referentes à atividade da rede, são enviados ao PC que os processa e apresenta ao analista de TP uma série de recomendações concernentes à solução dos problemas sinalizados, tal como diagramado na figura 6.3

Figura 6.3: Diagrama geral



Os arquivos LOGDCP e DCSTATUS contém os dados resultantes do registro das atividades da rede e são transferidos do A-10 para o PC. O arquivo LOGDCP contém os registros das mensagens dos erros provenientes da rede, e o DCSTATUS, gerado pelo compilador NDL, contém informações relevantes sobre a configuração da rede, especificando características da cada estação. Foi elaborado e implementado no computador A-10, um programa para filtrar os dois arquivos e gerar outros dois com formatos mais adequados para serem recebidos no PC e trabalhados pelo sistema SEREIA.

O sistema SEREIA foi escrito em PROLOG, usando facilidades do ARITY/EXPERT e ARITY/SQL. O ARITY/SQL é uma outra ferramenta, usada para construir um banco de dados relacional em cima da base de dados do interpretador. A manipulação do banco de dados é feita através do uso da SQL (Structured Query Language). O ARITY/SQL foi utilizado para permitir uma melhor organização do grande volume de dados provenientes dos arquivos transferidos e de outros informações necessárias para a gerência da rede.

Foram criados três tabelas que possuem as mensagens de erro da rede, a configuração da rede e o histórico dos erros

diagnósticados. A grande vantagem do uso do SQL é o acesso simples aos dados, tanto para elaboração de tabulações e outros cálculos como para uso pelas regras de produção.

O ARITY/EXPERT foi usado para a construção da base de conhecimento que representa o conhecimento do especialista em Teleprocessamento no ambiente UNISYS. A sintaxe usada para descrever o sistema especialista é aquela descrita na seção 6.2.

Assim, o primeiro procedimento do sistema é isolar erros críticos, isto é, erros que ocorrem de forma insistente em uma mesma estação ou linha, originados pela mesma causa durante um certo período de tempo. Os parâmetros de pesquisa do erro crítico são definidos pelo especialista através do uso de ***frames***. Na figura 6.4 são representados os parâmetros dos erros de paridade vertical e horizontal. O atributo CODIGO representa o código do erro que virá junto com a mensagem no LOGDCP. PESQUISA_CODIGOS representa os códigos dos erros que possuem a mesma causa e o INTERVALO_SEG e NUMERO_CRITICO determinam o número de ocorrências que deve num certo intervalo de tempo para que o ocorrido seja caracterizado como erro crítico.

Figura 6.4 Caracterização de erro crítico

```
define PARIDADE_HORIZONTAL with
       CODIGO=[100040] and
       PESQUISA_CODIGOS=[100080,100040] and
       INTERVALO_SEG=20 and
       NUMERO_CRITICO=5.

define PARIDADE_VERTICAL with
       CODIGO=[100080] and
       PESQUISA_CODIGOS=[100080,100040] and
       INTERVALO_SEG=20 and
       NUMERO_CRITICO=5.
```

Com base nestas definições, quando uma mensagem de erro de paridade vertical for lida do LOGDCP, o sistema varrerá a tabela que contém as mensagens de erro obtidas até o momento para procurar

todos os erros de paridade (vertical e horizontal) que aconteceram dentro do intervalo de 20 segundos antes do horário lido. Se o número achado for maior ou igual a cinco, se configurará um erro crítico.

Após ser evidenciada a ocorrência de um erro crítico, serão ativadas as regras definidas pelo especialista. Na **figura 6.5** são apresentados, em linguagem semi-natural (orientada à sintaxe das regras no ARITY/EXPERT), dois exemplos de regras para isolar as causas do erro e as recomendações para erradicá-lo.

Figura 6.5 **Manipulação de um erro**

```
causa do erro é INTERFACE ELÉTRICA/30%
                LINHA/10%
                TERMINAL/60%
             se
                ERRO É DE PARIDADE e
                MEIO DE LIGAÇÃO É TDI e
                NÃO HOUVE ERRO NA LINHA MULTIPONTO

solução do erro é
             "1.VERIFICAR SE O MODEM DO SISTEM ESTA LIGADO
              2.LIGAR CABO AO MODEM DO SISTEMA"
               se
                  CAUSA DO ERRO É DATA SET NOT READY
```

O sistema apresenta ao usuário a possível causa do erro e sua solução, com a possibilidade de gerar a explicação sobre a linha de raciocínio seguida para chegar a estas conclusões.

Os erros críticos detectados são gravados na tabela do histórico dos erros diagnósticados para futura verificação da reincidência de um mesmo tipo erro que não tenha ainda sido solucinado.

Está previsto um desenvolvimento suplementar que permitirá ao sistema a opção de gerar estatísticas sobre os erros e informações sobre a rede. Algumas perguntas para as quais o usuário poderá ter resposta do SEREIA são, por exemplo:

- Quais os erros ocorridos na linha 10 ?

- Quantos erros de paridade ocorreram na estação 23 a partir do horário 10:23:00 ?

- A estação 17 pertence a qual linha ?

- Houve algum erro crítico na estação 15 ?

O sistema SEREIA está sendo construido segundo a metodologia de trabalho proposta por [103], em que a prototipagem é a base do desenvolvimento que evolui de forma incremental, a partir da análise dos resultados obtidos em cada estágio.

A fase seguinte do projeto envolveu a orientação do SEREIA para o ambiente OSI. Os detalhes desta fase serão descritos em seções posteriores.

6.1.2 A versão OSI do SEREIA

Neste protótipo optou-se por considerar inicialmente a gerência de problemas, associada a dois níveis: o nível de transporte primariamente com algumas informações adicionais do nível de sessão.

Estes dois níveis foram escolhidos por duas razões, uma estratégica e outra pragmática. A razão estratégica para a escolha do nível de transporte deriva do fato de que o nível de transporte provê o serviço mais básico, independente de rede, ou seja, é o primeiro nível em que se estabelece um protocolo fim-a-fim. Como se considera que será usada uma rede de comutação de pacotes para prover os serviços correspondentes aos níveis inferiores, a gerência da rede terá interesse especial em saber como os sistemas fim se comportam. Alem disso, considerando que a rede de comutação de pacotes ser á provavelmente uma rede pública, terá seguramente um esquema de gerenciamento próprio e independente. Mas a mera gerência do nível de transporte não satisfaz porque há problemas que são detectados no nível de transporte mas a causa é originada no nível superior. Assim a gerência das conexões de sessão constitui

também um ponto importante a incluir em qualquer sistema de gerência de rede. A razão pragmática para seleção destes dois níveis deriva do fato de que alem de serem importantes são mais bem conhecidos do que os níveis superiores a eles. Portanto, fica mais facil iniciar o trabalho com dois níveis em cuja implementação se tem alguma experiência conforme descrito em [93 e 110].

Os problemas associados com estes dois níveis podem ser adicionalmente subdivididos, tal como será ilustrado posteriormente. Na determinação do exato problema que está ocorrendo ou que tenha ocorrido, o sistema especialista necessita informação que permita identificar a sub-classe de problema concernente. As vezes não há informação suficiente para tal. Neste caso, o sistema solicitará informação adicional. Em alguns casos esta informação adicional será solicitada a outro sistema aberto, mediante o uso do protocolo de gerenciamento de nível apropriado ou mediante o uso do protocolo de gerenciamento no nível de aplicação (CMIP). Outras vezes o sistema especialista terá que solicitar informações adicionais ao operador humano.

Quando for solicitado ao operador humano uma resposta sobre algo (condição de um componente etc...) será também solicitado o grau de certeza dele quanto àquela resposta. Para tornar mais ergonômica a interação, ser-lhe-á apresentado um menu com palavras que indiquem seu grau de certeza quanto à resposta fornecida. Internamente tais respostas serão mapeadas conforme o quadro da figura 6.6.

Figura 6.6: Grau de certeza nas respostas

Palava	Coeficiente de certeza
certo	1.0
quase certo	0.8
bem possível	0.6
possível	0.5
talvez	0.4
alguma chance	0.2
impossível	0.0

Quando o sistema apresentar ao usuário afirmativas acompanhadas de grau de confiança, os valores numéricos indicativos do grau de confiança poderão ser também substituidos pelas palavras do quadro da figura 6.6.

A interação do sistema especialista com o processo SMAP ocorrerá de duas formas básicas: interação entre dois processos ou interação entre máquinas. No primeiro caso o sistema especialista estará sendo executado na mesma máquina em que o SMAP. Nesta situação uma fila simples entre eles podera constituir o meio de intercomunicação. A forma de interação entre o SMAP e o sistema especialista não está sendo objeto de padronização pela ISO, tal como declarado por [86]. Se o sistema especialista estiver sendo executado num outro computador, o SMAP deverá passar as informações apropriadas para o mesmo, através de um interface serial V.24.

Em função das informações sobre o problema, o sistema especialista proverá orientação ao usuário humano sobre como agir com relação ao problema em foco.

Um dos aspectos mais difíceis na criação de um sistema especialista, na opinião de muitos autores, consiste em delimitar o domínio de conhecimento sobre a area afetada pelo problema [3, 60,

70, 80, 87]. Para adequar o problema a uma forma passível de uso pelo sistema especialista Expert da Arity [2] a melhor forma encontrada foi construir um esquema básico do conhecimento e ir elaborando-o pouco a pouco, adicionando mais conhecimento ao sistema. Assim, iniciou-se o trabalho relacionando as informações específicas concernentes aos problemas que o sistema deverá ser capaz de manipular. Por exemplo, com respeito ao nível de transporte, pode-se subdividir os problemas em função do momento em que podem ocorrer:

-na tentativa de abertura de uma conexão

-durante o andamento da conexão

-durante o encerramento da conexão

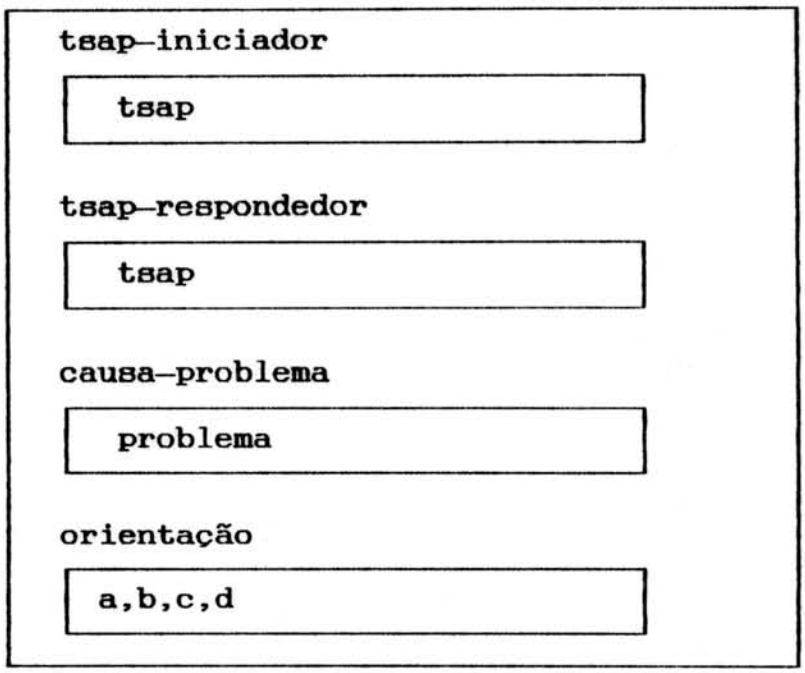
A especificação do sistema especialista será feita na seção 6.3, de acordo com o paradigma apresentado na seção seguinte.

6.2 A taxionomia usada

A taxionomia do sistema tem como elemento básico da base de conhecimento o CONCEITO. Um conceito é uma classe de objetos ou idéias que é representado como uma unidade.

Um conceito pode ser descrito através de um quadro(*frame*) que tem uma janela (slot) para cada elemento que o descreve, tal como ilustrado na figura 6.7.

Figura 6.7 Frame simples do conceito de conexão-de-transporte



Os termos tsap-iniciador, tsap-respondedor, causa-problema e orientação são os nomes para as janelas do frame. Os termos tsap, problema e a lista de abreviações representam as restrições para os valores nos slots.

Cada uma das qualidades definidas no frame poderá ser, por sua vez, descrita através de uma definição própria. Assim, a taxionomia não apenas define os conceitos mas também os relacionamentos entre eles.

Um conceito pode ser classificado como: PRIMITIVO ou DEFINIÇAO. Um conceito primitivo é um estado ou condição para o qual existe orientação mas não pode ser dada uma definição absoluta. Por exemplo, caminhão pode ser usado como um primitivo. Não se pode definir completamente o que constitui um caminhão, embora o conceito de caminhão descreva um grupo de veículos que podem ser categorizados univocamente e separadamente de outros veículos.

Uma definição é um conceito para o qual existe um conjunto de condições suficientes que definem o conceito. Um "advogado" é um exemplo de uma definição. Um advogado pode ser definido como uma

pessoa cuja ocupação é a prática da lei. Observe-se que os termos usados para descrever uma definição podem, frequentemente, ser classificados também como primitivos.

Na base de conhecimento do sistema especialista construido com o sistema de desenvolvimento EXPERT/ARITY [2], a informação é estruturada em termos de conceitos. A informação dos conceitos, por sua vez, é estruturada em termos de propriedades e papeis. Propriedades e papeis descrevem as características que constituem um conceito em particular. As janelas ("slots") num quadro ("frame") de um conceito representam os possíveis valores das propriedades e papéis do conceito.

Uma propriedade representa uma qualidade especifica de um conceito. Por exemplo, o conceito vinho pode ter as propriedades cor, corpo e paladar. Cada uma destas propriedades descrevem uma particular característica do conceito vinho e um item pertencente ao conceito vinho deverá ser classificado de acordo com estas propriedades. As propriedades tem valores que são definidos em termos de restrição de valores, as quais descrevem os valores que uma dada propriedade pode ter para um dado conceito. Uma restrição de valor pode ser um número ou números, ou um conjunto de átomos.

A linguagem de taxionomia usada no sistema EXPERT/ARITY prevê que as propriedades e suas restrições de valores sejam especificados por meio de declarações ***type*** (tipo). As declarações de tipo para propriedades podem ter três diferentes formas. Se a propriedade tem um valor específico não numérico, então cada possível valor da propriedade deve ser listado na declaração, tal como no exemplo:

type severidade = [irrecuperavel, inoperante, intermitente, toleravel] .

Propriedades também podem ter um valor numérico ou do tipo ca-

racteres. Para propriedades numéricas é usada a palavra chave **numeric**, tal como no exemplo a seguir:

type contador = numeric.

Para propriedades para as quais não é prático prover um valor específico, por exemplo, poder-se-ia ter uma restrição de valor que fosse um nome e o nome poderia ser um dentre milhares de possibilidades. Neste caso tal propriedade seria declarada como de tipo **string** usando a palavra chave **string**. Neste caso qualquer string seria aceito como uma restrição de valor.

Ex. type nome_sistema = string.

Normalmente indivíduos pertencendo a um conceito podem somente ter um valor único para uma particular propriedade. Em situações onde se deseja permitir a um indivíduo ter mais do que um valor para uma propriedade em particular, pode-se faze-lo usando a declaração **multivalue.**

Ex. type tipo_manutencao = multivalue [reinicializar,
substituir, redefinir, consertar,
remover_componente, adicionar_componente].

Após esta declaração mais de um tipo de manutencao poderá ser atribuído à propriedade tipoManutencao de um objetjo gerenciado. Contudo, propriedades numéricas podem ter um único valor.

Uma propriedade descreve uma qualidade muito específica de um conceito. Os valores de restrição, tal como a,b,c,d no conceito descrito na figura 6.7 são valores específicos do atributo que descrevem.

Papéis, tal como propriedades, também representam os valores de um conceito. Contudo, papéis representam conjuntos de valores que descrevem mais detalhadamente do que uma propriedade. O valor de um papel pode ser um quadro de um conceito. Como tal, o valor de uma papel pode ter, por sua vez, propriedades e papéis.

Frequentemente, papéis representam os componentes que compreen-

dem um conceito. Por exemplo, o conceito Bicicleta pode ter os papéis Rodas, Assento e Freios. Atribuindo estes papéis ao conceito bicicleta, estão sendo indicadas as características que cada membro do conceito bicicleta deve ter.

Um papel é definido em termos de uma restrição de valor e um número de restrições. A restrição de valor descreve os valores que um dado papel pode ter para um dado conceito e o número de restrições define o número de partes que um papel pode ter. O número de restrições estabelece o número mínimo e o máximo de partes que o papel pode ter. Se este valor é omitido na definição do papel é assumido ser igual a 1.Para definir os papéis de um conceito na linguagem de taxionomia do EXPERT/ARITY usa-se o formato exemplificado a seguir:

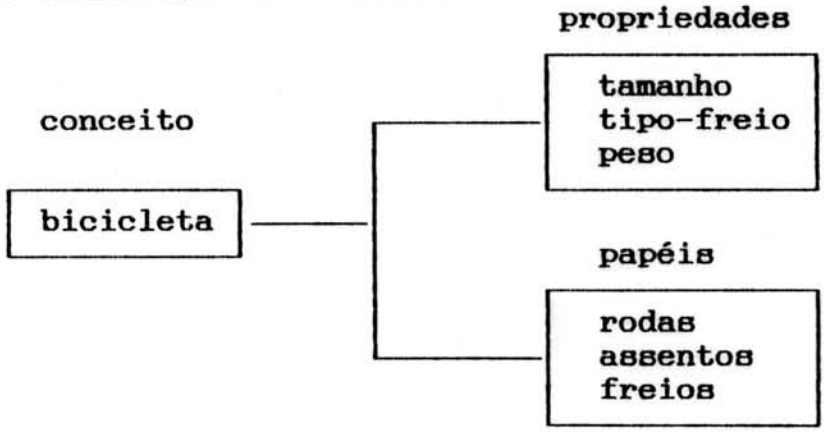
```
define primitive bicicleta with
        rodas = roda and
        number-of rodas=2 and
        assentos = assento and
        number-of assentos = (1,2) and
        freios = freio and
        number-of freios = (1,2).
```

Isto indica que o conceito bicicleta tem duas rodas, um ou dois assentos e um ou dois freios.

Em resumo, um valor pode ser representado como uma propriedade ou como um papel dependendo do contexto em que se deseja usá-lo. Propriedades descrevem características específicas de conceitos. Quando se considera que descrevendo um valor como uma propriedade não leva ao grau de detalhamento desejado, pode-se descrevê-lo como um papel. Papéis descrevem uma característica geral de um conceito, a qual por sua vez é constituida de características específicas. Desse modo, um papel pode ter, por sua vez propriedades e

papéis. Por exemplo, o conceito bicicleta pode ter as propriedades e os papéis indicados na figura 6.8.

Figura 6.8 Construção de conceitos



Declarações de tipo devem ser também inseridas no arquivo de taxonomia para identificar os papéis usados na definição dos conceitos. As declarações de tipo para papéis usam a palavra chave ***role***, tal como ilustrado a seguir:

type tsap_iniciador = role

type tsap_respondedor = role

type causa_problema = role

6.2.1 Descrição de conceitos

Todos os conceitos usados pelo sistema especialista devem ser descritos. Isto significa defini-los em termos de seus componentes bem como definir os relacionamentos entre eles. Estas definições também são incluidas no arquivo que contém a taxionomia. A primeira parte do arquivo é composta das declarações de tipos indicadas na seção anterior e a segunda parte é constituida de declarações de conceito que tem o seguinte formato básico:

define conceito.

Os conceitos podem ser primitivos e não terem propridades ou papéis. Neste caso seriam definidos como primitivos, da seguinte maneira:

define primitive conceito.

Quando um conceito tem propriedades, estas também são inseridas na sua definição, tal como no exemplificado a seguir:

define conexão_de_transporte with

tsap_iniciador = tsap and

number_of tsap_iniciador_item = 1 and

tsap_respondedor = tsap and

number_of tsap_respondedor_item = 1 and

causa_problema = problema and

number_of causa_problema = 1 and

orientação = [a,b,c,d].

Esta definição inclui vários objetos: conexão_de_transporte, tsap_iniciador, tsap_respondedor, tsap, causa_problema, problema e orientação. O objeto com tais definições é conexão_de_transporte. Os outros objetos descrevem qualidades do objeto conexão_de_transporte. Logo, conexão_de_transporte é um conceito definido.

A lista de itens para orientação [a,b,c,d] são abreviações que representam as ações recomendadas para lidar com problemas relativos à conexão de transporte. A definição do significado de tais abreviacões é colocada num outro arquivo que também conterá a base de regras. Assim o sistema pode apresentar ao usuário as recomendações na sua forma extensa mas internamente é usada a forma abreviada para economizar espaço.

Os termos tsap_iniciador, tsap_respondedor, causa_problema e orientação são os nomes para os slots do frame. Os termos tsap, problema e a lista de abreviações representam as restrições para os valores nos slots.

A descrição das propriedades de um conceito consistem do nome da propriedade seguido de um sinal de = e o valor ou valores da propriedades, tal como mostrado a seguir: **propriedade = valor.**

propriedade = [valor1, valor2,...].

Quando uma propriedade tem mais de um valor eles são envolvidos por colchetes. Se a propriedade tem um valor numérico usa-se o formato:

propriedade = número.

Exemplo: **define concentrador with portas = 16.**

Quando uma propriedade tem um valor que pode ser um intervalo de números deve-se incluir o número mínimo e o máximo do intervalo, tal como mostrado a seguir:

propriedade = (número_mínimo, númro_máximo).

Ex.: **define concentrador with portas = (8,32).**

Os número podem ser inteiros ou de ponto-flutuante.

Também se pode indicar que uma propriedade poderá ter qualquer dos valores definidos pela declaração de tipo, usando a palavra chave **any** Suponha-se que tivesse sido declarado o tipo cor da seguinte maneira:

type cor = [azul, vermelho, verde, amarelo, laranja].

Se a propriedade de um conceito pode ter qualquer uma destas cores sua definição seria a seguinte:

define cor_fio with any cor.

Para definir os papéis procede-se de maneira similar à usada para definir propriedades. Contudo, na definição de papéis é preciso indicar o número das restrições de valores, além das próprias restrições de valores, tal como mostrado a seguir:

define primitive livro with

divisões = capítulo and

componente_principal = papel and

identificaçâo = título.

As relações entre os conceitos também precisam ser definidas. Existem relacionamentos de subordinação e sub-classes.

6.2.1.1 Subordinação de conceitos

Uma relação de subordinação permite transferir informação entre conceitos descritos na taxionomia. Um conceito B é dito ser subordinado ao conceito A se todas as propriedades (e suas restrições de valor) e todas os papéis (e seus valores e restrições de número) do conceito A podem ser aplicadas ao conceito B.

Na linguagem de taxionomia, uma subordinação direta seria definida com o formato seguinte:

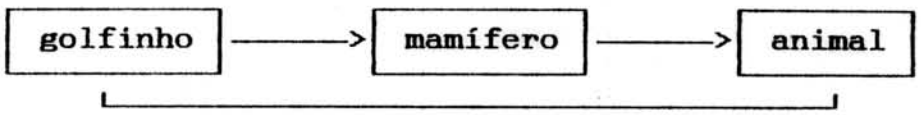
define concept-B as a concept-A

Em função desta declaração o conceito B passa a ter as propriedade e papéis de A, além das suas próprias, ou seja, o conceito B herda os valores do conceito ao qual é subordinado, em situações onde seus próprios valores não sejam definidos. Um conceito pode ser subordinado a qualquer número de outros conceitos. Neste caso a subordinação a mais de um conceito seria indicada na seguinte forma:

define {primitive} conceito as a conceito_englobante_1 and a conceito_englobante_2.

A subordinação pode ser indireta, tal como no exemplo seguinte:

Figura 6.9 Subordinação indireta



Neste caso, se a taxionomia estabelece que mamífero é um tipo de animal e a definição de golfinho estabelece que ele é um tipo de mamífero, o sistema pode deduzir que golfinho é tanto mamífero quanto animal. Neste sentido pode ser estabelecido que todas as propriedades e regras que se aplicam a animal, também se apliquem a golfinho.

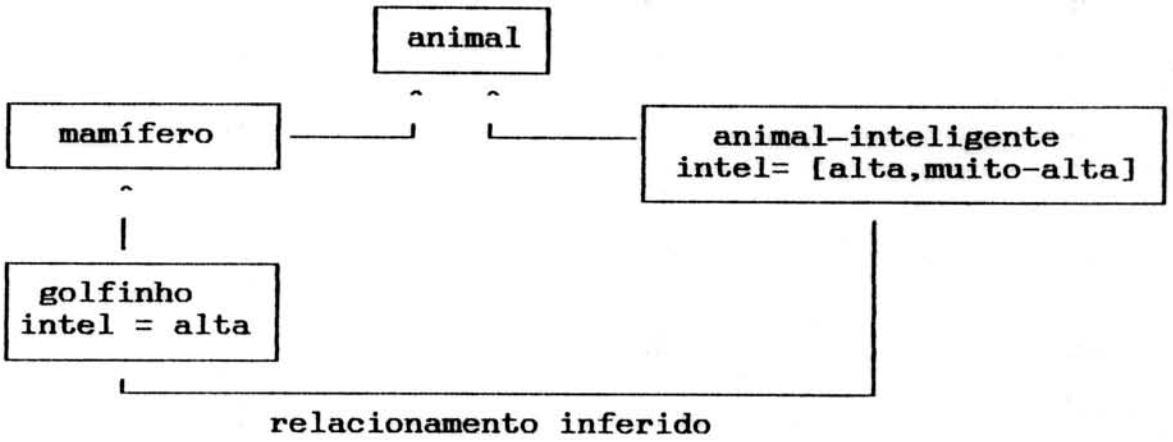
Uma pecularidade importante do sistema de desenvolvimento

EXPERT/ARITY é sua habilidade de inferir relações de subordinação ou inclusão que não sejam explicitamente definidas, mas que podem ser determinadas pela informação que é definida. Por exemplo, supondo que a taxionomia inclui, além da definição de golfinho como mamífero e como animal, o conceito de animal-inteligente que é definido como sub-classe de animal com a proprie-dade inteligência para a qual os valores possíveis são alto e muito alto. Suponha que definição de golfinho estabelece que ele é um mamífero com a propriedade inteligência com valor alto.

Uma vez que foi definido que mamífero é uma sub-classe de animal, o fato de que golfinho tenha a propriedade inteligência com valor alto estabelece que animal-inteligente é um conceito que inclui golfinho, tal como diagramado na figura 6.10.

É possível inibir a capacidade do sistema de inferir relacionamentos de inclusão ou subordinação. Quando se comandar a classificação dos conceitos, é preciso indicar específicamente, que as inferências são desejadas.

Figura 6.10 : relacionamentos inferidos

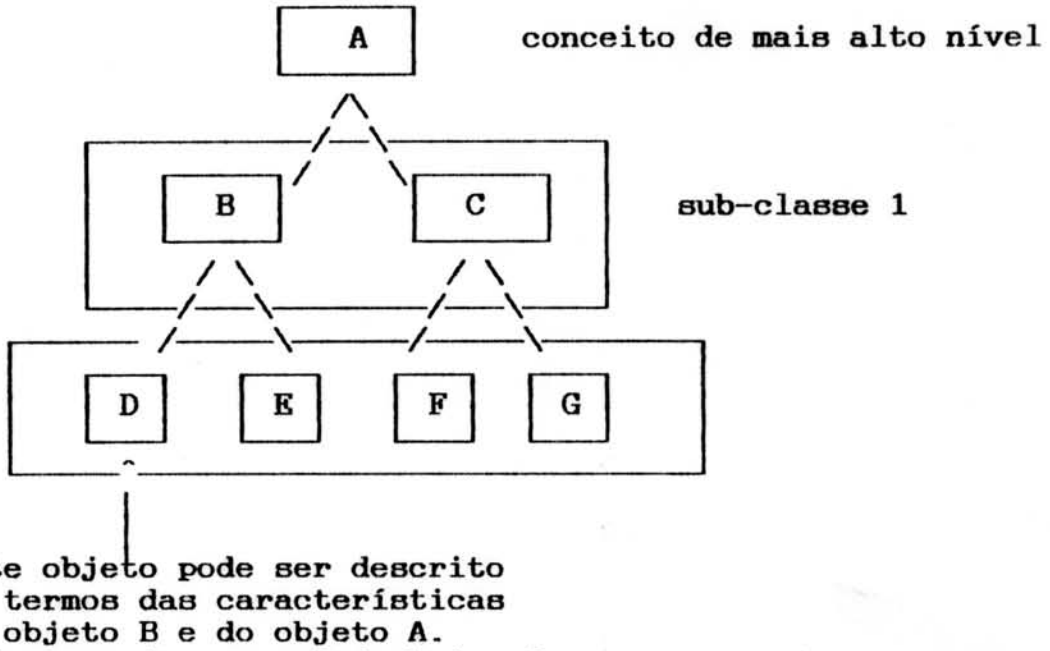


6.2.1.2 Relacionamento hierárquico (sub-classes)

Conceitos podem ser grupados em relacionamentos hierárquicos usando-se sub-classes. Um conceito numa hierarquia pode herdar qualidades dos conceitos que estão acima dele na hierarquia. A fi-

gura 6.11 ilustra uma representação hierárquica.

Figura 6.11. Uma representação de subclasses



Um conceito que está abaixo de outro conceito numa hierarquia é dito ser subordinado pelo outro conceito. Assim, o conceito D é subordinado seu conceito pai, o conceito B.

Usa-se subclasses para refinar a definição de objetos e elas são definidas de maneira similar à definição de um conceito tal como no exemplo:

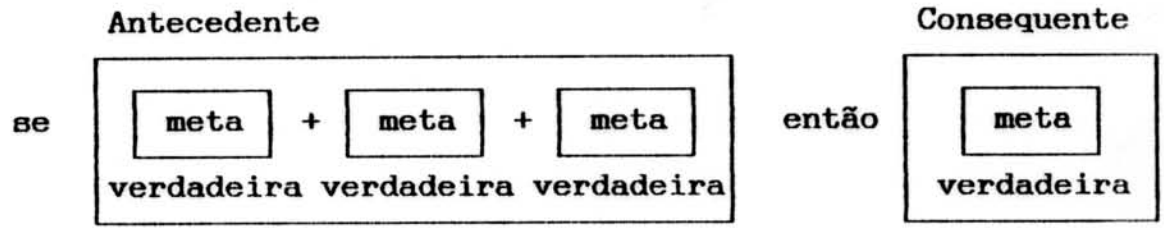
define tipo_problema as a subclass of problema.

Assim, poder-se-ia definir tipos de problema e causas associadas a cada tipo. Se nada é dito sobre as causas elas seriam derivadas ou herdadas da definição de causa de problemas genérica que ficou num nível hierárquico maior. Sub-classes permitem criar uma hierarquia multi-nivel.

6.2.2 Definição das regras

As regras permitem ao sistema descobrir a solução de um problema. Uma regra consiste de um antecedente e um consequente. Tanto o antecedente quanto o consequente são formados de metas. Uma meta é uma condição que pode ser provada ser falsa ou verdadeira.

Figura 6.12 REGRA



O consequente de uma regra é verdadeiro se cada uma das metas que compõe o antecedente é verdadeira. Uma meta indica que uma propriedade de um conceito tem um valor particular. As metas são construidas usando-se a taxionomia construida conforme descrito na seção 6.2.

O sistema EXPERT da Arity é um sistema de encadeamento para trás (backward chaining). Isto significa que quando é executada uma consulta ao sistema especialista, a conclusão da regra é considerada primeiro e cada uma das metas que levam à conclusão é avaliada. Os valores das propriedades ou classes listadas nas metas do antecedente de uma regra são verificadas em face dos valores de propriedades ou identidades de classes conhecidos para a instância do conceito sendo avaliado na consulta.

No sistema EXPERT/ARITY, o consequente da regra sempre aparece primeiro e é composto de somente uma meta. O antecedente pode ser composto de uma ou mais metas. Sua definição tem o seguinte formato básico:

consequente (meta)

if

meta {and

meta and

meta and

meta }.

Quando o valor de uma instância de um conceito não é conhecido, o sistema formula uma questão tentando determinar seu valor. A resposta do usuário é usada para instanciar um conceito ou pro-

priedade.

As metas são usadas para determinar uma propriedade de um conceito tem um valor particular, tal como no exemplo seguinte:

the causa of the tipo-problema is a.

Uma regra também pode ser usada para isolar um membro específico de uma particular sub-classe:

the causa-especifica of the problema is x.

Quando o valor de uma instância de um conceito não é conhecido, o sistema formula uma questão tentando determinar seu valor. A resposta do usuário é usada para instanciar um conceito ou propriedade.

O formato básico de uma meta é o seguinte:

{the an a} propriedade of {the an a} sinônimo indicativo da posição na hierarquia ou conceito is valor.

O uso dos artigos ***the, an*** ou ***a*** é opcional. Uma meta simples estabelece que um valor se aplica a um conceito em particular. O conceito ao qual o valor se aplica é descrito em termos de sua posição na hierarquia. A posição na hierarquia de conceito é usada para assegurar que não haja ambiguidade no que tange ao conceito referido na meta. A posição na hierarquia representa o contexto em que foi usado algum conceito sendo referido. A posição na hierarquia descreve o conceito em termos de seu relacionamento com a meta de mais alto nível. A posição na hierarquia pode ser indicada de forma nais sucinta através de sinônimos. Tais sinônimos são declarados no arquivo que contem as regras, antes das regras que os utilizem. A forma para definir um sinônimo para uma posição na hierarquia é a seguinte:

set(posição na hierarquia) = sinônimo.

O sinônimo deve ter um nome diferente de qualquer outro conceito descrito na taxionomia mas pode ter o mesmo nome de algum papel

que ele descreva.

Metas usadas nas regras podem referir listas de valores:

the cor of the parte-principal of the conjunto is [azul, verde]

Metas podem descrever a sub-classe a que um conceito pertence:

the tipo_específico of the problema is congest.

ou

the tipo_específico of the problema is [congest,bug_software]

Fatores de confiança podem ser associados às metas:

the tipo_específico of the problema is congest with_cf 0.6.

Quando o sistema encontra uma meta como esta, ela será considerada verdadeira somente se o fator de confiança associado ao fato de o problema ser congest seja maior do que 0.6. Se nenhum fator de confiança é associado a uma regra, o mínimo fator de confiança para que ela seja aceita como positivo é 0.2 e para metas negativas é -0.2. Quando um fator de confiança é atribuido ao consequente de uma regra isto significa a confiança associada à conclusão da regra, se o antecedente for verdadeiro. Fatores de confiança podem ser associados aos valores incluidos no antecedente de uma regra:

the cor of the parte_principal of the conjunto is [azul/0.7, verde/0.4]

Quando uma lista de valores é incluida no antecedente de uma meta ele representa uma disjunção de metas, isto é, o valor da meta pode ser qualquer um da lista. Por exemplo, no caso acima, é indicado que a cor da parte_principal do conjunto pode ser azul com limiar de confiança 0.7 ou verde com limiar de confiança 0.4.

Quando uma lista de valores é incluida no consequente de uma regra, ele representa uma conjunção de metas. Por exemplo:

the causa of maufuncionamento is [congest/0.6, bug_soft/0.3]

indica que a conclusão da meta é tanto congest com confiança 0.6 quanto bug_soft com confiança 0.3. Pode-se selecionar somente

o valor como mais alto fator de confiança, usando a declaração ***mostlikely***.

the mostlikely causa of maufuncionamento is [congest/0.7,

bug_soft/0.3]

A palavra reservada ***unknown*** pode ser usada nas metas no antecedente de uma regra para indicar que o valor de uma propriedade é desconhecido, o que permite escrever regras para tratar tais situações.

6.3 A regras do SEREIA-OSI

O SEREIA pode ter dois tipos básicos de reação, quando recebe relato de evento do SMAP: reagir apresentando conclusão e/ou recomendação sobre curso de ação ou simplesmente registrar a informação para uso posterior.

Os problemas que requerm a atenção imediata do responsável pela rede são os que implicam na paralização de algum componente crucial. Até o momento apenas a entidade de transporte em si foi classificada como tendo um grau de importância crucial no contexto deste projeto. Por outro lado, quando chegar um relato de evento que não e crucial isoladamente, mas que agregado aos relatos anteriores permita detectar uma degradação crescente de performance, também haverá manifestação sobre o sintoma ao responsável pela rede. Neste caso, uma extrapolação da evolução do MTBF (Mean Time Between Failures) permitirá calcular a probabilidade de falha do componente e alterar a severidade do problema inerente, conforme os níveis definidos na seção 5.3.

Problemas que não impedem o funcionamento de segmento crítico da rede devem apenas ser anotados e os contadores associados incrementados e comparados com os limiares estabelecidos, quando for o caso.

Por outro lado, o responsável pela rede pode requisitar a indicação de todos os componentes com problemas que tenham grau de severidade intermitente (diz-se que estão num estado semi-crítico) para determinar manutenção preventiva. Neste caso, a apresentação pode ser por ordem de importância do elemento (quantos dependem dele), por ordem alfabética do nome do componente, por regiãa (para economizar a movimentação da equipe de manutenção) ou ainda por domínio.

A seguir são apresentados alguns exemplos de regras integrantes do SEREIA. As regras são apresentadas numa forma coerente com as das regras na sintaxe do ARITY/EXPERT mas numa linguagem mais similar à linguagem natural para melhor legibilidade e compreensão das idéias que representam.

O problema é congestionamento_local se
usuário solicitou T-CON-req e
entidade-transporte encaminhou T-DISC-req e
causa é falta-recursos-locais.

O problema é congestionamento_remoto se
entidade de transporte enviou TPDU.CR e
recebeu TPDU-DR e
razao é 129.

O problema é erro-de-software se
a entidade de transporte recebeu DR-TPDU e
razao é [130, 131, 132, 133, 135, 138].

O problema é saturacao-nivel-rede se
a entidade de transporte recebeu TPDU-DR e
razao é 136.

O estado de um componente é semi-crítico se
a qualidade-serviço não é atingida e
o serviço que usa tal componente esta operacional.

A qualidade de serviço não é atingida se
retardo-estabelecimento é alto.

O retardo-estabelecimento é alto se
retardo > limiar-retardo-estabelecimento.

Necessário consultar entidadede transporte par se
retardo-estabelecimento é alto e
culpa-retardo-estabelecimento é remota.

Obs.:Nesta situação o SEREIA solicitará ao SMAP esta informação e este usará o elemento M-GET para obter o valor deste contador na entidade par.

Culpa-retardo-estabelecimento é remota se
retardo-estabelecimento é alto e
o componente local do retardo-estabelecimento é aceitável.

O estado de um componente é crítico se
a qualidade-de-serviço não é atingida e
o serviço que usa tal componente não está operacional.

Quem define se um componente está operacional é a entidade de gerenciamento de nível (LME) e coloca tal informação na MIB. Portanto, para obtê-la o sistema especialista deve acionar uma cláusula PROLOG que leia a MIB e retorne a condicao de operacional ou não daquele componente.

A qualidade de serviço é uma informação determinada pelo entidade de gerenciamento de nível, sempre que algum componente por ela gerenciado termina de ser usado (em condições normais ou

anormais). Esta informação também é passada para a MIB.

Numa primeira abordagem pode-se decidir que o SMAP recalcule o retardo médio no período, compare com o limiar e se estiver acima do limite estabelecido, passe a informação para o sistema especialista que vai decidir o que fazer:

-consultar o SMAP do sistema (indiretamente, via SMAP local) para saber se a culpa do retardo no estabelecimento é remota ou local

-alertar o operador da rede de que a entidade de transporte está congestionada, se já não houve alerta anterior. Um arquivo com os alertas sinalizados no período, concernentes a determinados objetos de gerenciamento.

Após o encerramento de um período, os alertas são copiados para um arquivo histórico e o arquivo de alertas recolocado em estado inicial.

Em uma outra alternativa o SMAP compara a QOS informada pela LME com o valor aceitável, indicado na MIB. Se está bom, recalcula a média de QOS (no que concerne ao tempo de estabelecimento de conexão que é o caso considerado nesta discussão) e grava na MIB. Se está ruim, alerta o sistema especialista passando-se as seguintes informações:

tipo de evento

valor da QOS

tsap iniciador

tsap respondedor

Ao definir o funcionamento destas entidades veio a tona uma dúvida concernente ao melhor local para armazenar o limiar da QOS, se na MIB ou no sistema especialista. A decisão a ser tomada deve ser baseada na modalidadede gerenciamento de atualizações que se desejar implementar. Se todos os parâmetros da MIB são atualizados diretamente (sem o uso do sistema especialista) então este e

outros limiares devem ficar na MIB. Se os parâmetros vão ser alterados somente mediante a intermediação do sistema especialista então algumas informações podem ficar no âmbito dele. Contudo, neste caso, se um SMAP remoto solicitasse esta informação, ela não estaria diretamente acessível ao sistema especialista o que seria inconveniente porque em termos de eficiência é previsível que o sistema especialista trabalhe mais lentamente do que um programa normal.

Assim, concluiu-se que seria mais eficiente deixar os parâmetros na MIB e fazer o sistema especialista solicita-los, quando necessário.

O sistema especialista foi projetado pressupondo-se que não seria do tipo ativo, sendo acionado somente quando necessário. Em caso contrário, o SMAP atua sozinho pois, o SMAP e o sistema especialista são processos cooperantes mas independentes. O sistema especialista entra em cena para dar orientação em caso de exceção. Quem decide se é caso de exceção ou não é o SMAP e ele faz isto sempre que receber do LME alguma informação ou relato de evento.

Algumas dentre o conjunto de recomendações veiculadas para os problemas detectados na entidade de transporte seriam:

Aumentar tempo-maximo-espera pelo estabelecimento-de-conexao
se
retardo-estabelecimento é muito alto e
culpa é da entidade-remota e
não existe alternativa de sistema a usar

Aumentar numero-maximo-conexoes-abertas se
taxa de falhas no estabelecimento de conexão é alta e
retardo-de-transito é baixo

Mover serviços para outra maquina se
retardo de transito é alto e
probabilidadede falha no estabelecimento é alta

Utilizar transporte-classe-4 com checksum se
taxa de erros residual é alta

Utilizar opção de confirmação explicita se
probabilidade de falha na transferencia é alta

Obs.: Em todas estas regras, ser **alta** significa ter ultrapassado o limiar.

Procurar outro sistema para usar se
resistencia da conexão-de-transporte é intoleravel e
codigo-causa-desconexao é [129]
Obs.: Está havendo congestionamento remoto sinalizado

Aumentar acessos de rede se
problema é saturacao-nivel-de-rede

Requisitar manutencao de software se
problema é erro-software

6.4 Interfaces com o sistema

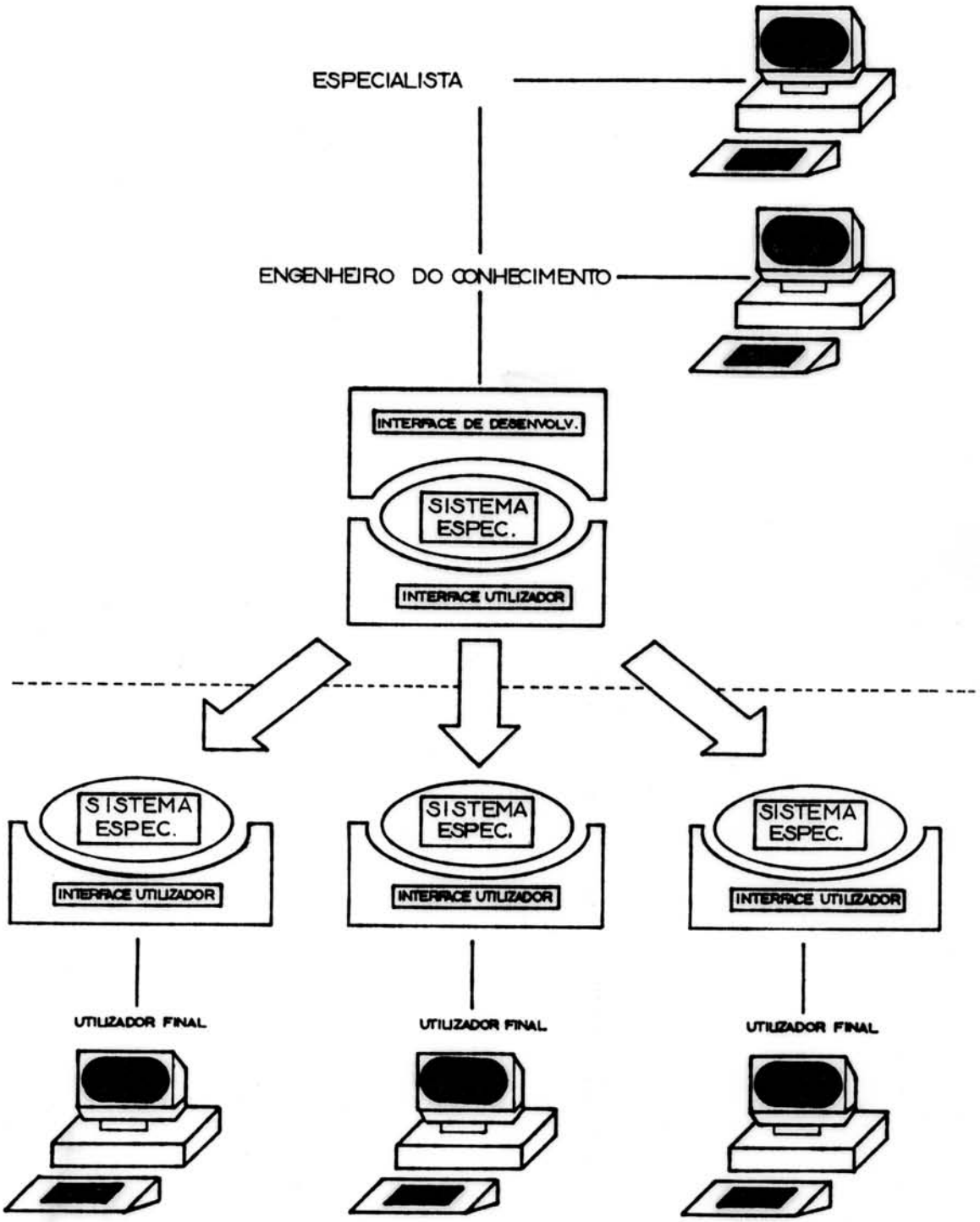
Tal como descrito em [4] em muitos sistemas especialistas é necessário prover auxílio que envolve raciocínio baseado em conhecimento incompleto sobre as restrições vigentes. Em função das informações inicialmente vigentes certas hipóteses são delineadas e apresentadas ao usuário visando verificar com o mesmo se aquilo atende às suas metas. Isso pode resultar num aumento do número de restrições aplicáveis quando o usuário indica o que se ajusta e o que não se ajusta aos seus intentos no cumprimento de uma particu-

lar tarefa. Esse conjunto de restrições que se aplica num determinado instante é armazenado sob a forma de características inerentes a um dado contexto. Existe, pois, a necessidade de registrar o contexto de interação entre o sistema especialista e o usuário. No decorrer da interação o usuário pode querer voltar atrás para um contexto anteriormente vigente. Logo é necessário manter o histórico dos contextos para que o usuário possa efetuar esse retrocesso. Isto ocorre quando por exemplo o usuário tinha uma idéia a respeito do possível elemento da rede ocasionador de um determinado defeito e em virtudes de testes ordenados concluiu que o componente não era o culpado e deve voltar a uma etapa anterior de condições e iniciar outras hipóteses.

Por outro lado, em se tratando de um sistema que está evoluindo, novas regras são continuamente adicionadas à base de conhecimento ou regras antigas são removidas ou modificadas. Esta alteração na base de conhecimento pode ser feita diretamente pelo especialista humano ou pelo engenheiro de conhecimento que recebe a informação neste sentido do especialista e codifica-a de forma apropriada para inclusão. As interações dos diversos tipos de usuários do Sistema Especialista estão ilustradas na figura 6.13.

Por outro lado, alguns usuários finais podem estar localizados em máquinas remotas e o sistema especialista pode ser distribuido redundantemente nas diversas maquinas onde será necessario para otimizar o acesso ao mesmo. Numa situação como esta, tudo o que se necessita é que o sistema remoto passe para o sistema especialista as informações no formato que ele espera receber. Se o sistema aberto não for o mesmo para o qual o sistema especialista foi originalmente projetado, será necessário adicionar um módulo que capture as informações da maneira que for possível, reformatando-as para transferí-las ao sistema especialista.

Figura 6.13: Interações com o sistema especialistas



6.5 Impacto da ação de manutenção

Podbury e Dillon [78] determinam que na organização do processo de manutenção deve-se distinguir entre:

a- Paralizações pré-planejadas especificadas pela manutenção

b- Paralizações forçadas resultantes de equipamento falho ou erro de operação

c- Paralizações curtas resultantes em deficiências nos recursos necessários para o funcionamento da rede

Em decorrência da paralização eventos serão relatados na rede. Assim um processo de filtragem inicial deve ser o de comparar se os eventos relatados não ocorrem em virtude de uma paralização comandada pela manutenção. Nesse caso tais eventos devem ser desconsiderados. Assim para cada objeto de rede deve constar a informação se o mesmo está em manutenção ou não. Outra informação que pode ser útil é se este objeto de rede vai entrar em manutenção programada ou não.

7. CONCLUSÕES

O desenvolvimento deste trabalho constituiu um processo de aprendizagem que era esperado mas que mesmo assim surpreendeu, tanto de forma positiva quando de forma negativa.

A surpresa positiva foi constatar que a abordagem era diferenciada e inovadora, mesmo no contexto internacional, e como tal recebeu atenção de pesquisadores propiciando intercâmbios muito produtivos em todas as reuniões e congressos em que teve algum de seus múltiplos aspectos apresentado. Constatou-se que a experiência acumulada com a implementação de sistemas OSI auxiliou muito a definir os aspectos que deveriam merecer estudo mais aprofundado, isto é, o que é realmente importante gerenciar.

Por outro lado, foi um tanto frustante perceber que o conhecimento atualmente existente a nível nacional ou internacional, sobre problemas numa rede OSI, praticamente inexiste. Ainda é pequeno o número de sistemas OSI operantes que tenham registro de anormalidades ou que tenham um sistema de gerenciamento que permita sequer detectar o que está ocorrendo internamente em cada nível. Na implantação do protótipo desenvolvido como parte deste trabalho, foi detectada esta dificuldade; falta de controle sobre as informações emitidas pelo sistema aberto, concernente à sua operação e aos problemas que encontra. Quando a implentação é local torna-se possível alterar o software para que emita as notificações necessários mas nos demais casos isto não é viável.

Assim para permitir que o sistema especialista pudesse ser portado para outros ambientes, auxiliando qualquer operador de outro sistema aberto, uma dentre as duas abordagens seguintes deve ser usada:

-Todos os sistemas abertos passem a emitir as notificaçoes

sobre eventos de forma padronizada

-Implanta-se um mecanismo que abstraia tais informações a partir de observação externa do comportamento do sistema aberto.

A primeira abordagem está sendo objeto de discussões a nível internacional tendo sido criado uma associação denominada OSI/NM Forum, integrada por diferentes fornecedores, com vistas à obtenção de um consenso sobre as informações de gerenciamento a serem intercambiadas num contexto OSI, integrado por sistemas produzidos por diferentes fabricantes. Este grupo definiu que o protocolo CMIP seria usado para o intercâmbio de informações e estão tentando chegar a um consenso sobre o conteúdo da MIB para uma implementação coerente em todos os sistemas.

A segunda abordagem, independe de outros padrões ou acordos, e consiste em implementar um processo de aplicação capaz de simular o comportamento de um SMAP e de algum mecanismo de coleta das informações sobre o comportamento das entidades de nível n do sistema aberto. Esta abordagem foi a selecionada para dar continuidade ao projeto SEREIA, pelo fato de assegurar uma maior independência de decisões externas (de fornecedores de software OSI) e o software necessário está sendo desenvolvido, no âmbito do projeto REDURGS [110], para permitir a observação de sistemas OSI e dar suporte para análise dos nPDUs intercambiados pelas entidades par de qualquer nível [96]. Pretende-se que este sistema poder coletar as informações sobre a operação de um sistema aberto apenas com a monitoração do tráfego por ele gerado. Tais informações serão repassadas aos sistema especialista para análise e provimento de recomendações de otimização.

Espera-se com esta nova ferramenta criar condições para um trabalho progressivo e de aumento da base de conhecimento do SEREIA. Acredita-se que este trabalho é extremamente necessário pois a cada dia surgem novos equipamentos, gerando notificaçoes de

modo diferenciado e o sistema especialista deve ser capaz de ir absorvendo gradativamente o conhecimento necessário para prover aconselhamento sobre toda a rede, mesmo com os novos equipamentos e sistemas [106, 107 e 108]. A especificação dos objetos gerenciados precisa ser agilizada pois começa a aparecer no mercado sistemas com um número cada vez maior de camadas OSI implementadas em placas, de forma bem menos flexível. Se não for inserida logo a provisão para atender aos comandos M-GET, M-SET, M-ACTION não haverá como extrair informações das entidades dos níveis implementados nesta modalidade. Tampouco será possível alterar com a flexibilidade necessária, os parâmetros que comandarão o funcionamento daquelas camadas OSI.

Neste sentido, espera-se que este trabalho possa também servir como um elemento de disseminação de informações sobre o gerenciamento de rede no contexto OSI e com este objetivo, tal tema foi abordado com bastante extensão neste texto.

Um outro aspecto a comentar, sobre a abordagem usada no desenvolvimento do trabalho, diz respeito ao ambiente de desenvolvimento utilizado. Embora flexível para adaptar-se às necessidades de quem implementa um sistema especialista, o sistema EXPERT/ARITY não tem um interface ergonômico para quem o utiliza para desenvimento. A adição ou alteração de novas regras é um processo um tanto complicadao, usando-se um editor de texto convencional e posteriormente compilando as definições num processo lento e trabalhoso. Ja são conhecidos e usados no exterior alguns ambientes de desenvolvimento de sistemas especialistas que oferecem uma gama maior de facilidades para que trabalha na implementacao, tais como: editor orientado ao contexto, interpretador, verificador de consistências das definições que atua no momento em que as mesmas estão sendo

submetidas ao sistema, auxiliando o usuário (engenheiro do conhecimento) a verificar suas alterações, provendo informações sobre regras similares ou redundantes. Ambientes com tais funcionalidades, que são usados com um contexto de janelas facilitam a criação e expansão da base de conhecimento e melhoram a produtividade dos projetistas envolvidos no processo.

No momento, cada nova regra é adicionada ao SEREIA, após extensa análise pelo(s) especialista(s) mas já foi detectado um outro aperfeiçoamento necessário ao sistema, para que possa passar para um estagio não experimental; consiste em dotá-lo de capacidade de aprendizagem isto é, que a partir da observação dos problemas e das soluções determinadas pelos especialistas humanos (devidamente registradas para conhecimento do sistema especialista), ele possa aumentar sua base de conhecimento derivando novas regras que adicionem as novas linhas de raciocínio ao seu processo dedutivo. Para que isto seja possível é necessário um estudo conjunto no sentido de avaliar, projetar e implantar mecanismos eficientes que assegurem ao sistema especialista condições para aprender, derivando automaticamente novas regras. Esta é uma área nova, mesmo para os pesquisadores da Inteligência Artificial e acredita-se que os sistemas de gerenciamento de redes de computadores apoiados por inteligência artificial precisarão dispor desta capacidade para poderem sair dos laboratórios onde estão sendo desenvolvidos e se tornarem efetivas ferramentas para uso num ambiente real.

REFERÊNCIAS BIBLIOGRAFICAS

[1] AGRE,J. A message-based fault diagnosis procedure. SIGCOM'86

[2] ARITY, The Arity/Expert Development Package, Arity Corp., Concord, Massachusetts, 1986, 8p.

[3] BERGQVIST J., LOUNAMAA P. Organizing knowledge bases using a context mechanism. In: 7th International Workshop Expert Systems & Their Applications, Avignon, France, 13-17 May 1987, p.1133-1138.

[4] BOSE, P. & Padala, A. Reasoning in an incomplete knowledge in an interactive personal flight planning assistant. In:7th International Workshop Expert Systems & Their Applications, Avignon, France, 13-17 May 1987, p.1077-92.

[5] Brachman,Ronald. e HENIG, Fran. The emergence of artificial intelligence technology.AT&T Technical Journal.Jan/Feb 1988,p.2-6.

[6] BUCHANAN,B. Expert systems: working systems and the research literature. Report No STAN-CS-86-1094, Stanford University, Stanford, 1985.

[7] CALLAHAN,Paul. Expert Systems for AT&T switched network maintenance. AT&T Technical Journal. JAN/FEB 1988, p.93-103.

[8] CHIKTE, S. et alii. A decision suport systems for transmission facility maintenance. Proccedings IEEE Internatinal Conference on Communications, Amsterdam, May, 1984,p.487-490.

[9] CLOCKSIN,W. & MELLISH,C. Programming in PROLOG. Springer-Verlag, New York, 1984.

[10] CUPELLO, James e MISHELEVICH David. Manageing Prototype

Knowledge/expert system projects. Communications of the ACM. 31 (5):534-541, may, 1988.

[11] CURRIE, W. Network status display systems. Computer Communications, 5 (1):35-41, feb., 1982.

[12] DATSKOVSKY,G. Natural language interfaces to expert systems. CUCS-169-85, Columbia University, 1985. ,4 p.

[13] DAVIS,R. Diagnostic based on structure behavior. AI MEMO 739, M.I.T., Cambridge, 1984. 54 p.

[14] DESCHON, Annette. A survey of data representation standards. ARPANET Request for Comments, Nº 971. SRI International, Menlo Park, California, January 1986

[15] ENNIS,R. et alii. A continuous real-time expert system for computer operations. IBM J.Res.Develop. 30 (1):14-28, jan., 86.

[16] ERIKSSON, Bengt. Experiences from expert system projects in network management. In:INDC90 Information Network and Data Communication, IFIP, Lillehammer, Norway, March 26-29, 1990 p.4-1 a 4-1

[17] FIKES, Richard and KEHLER, Tom. The role of frame-based representation in reasoning. Communications of the ACM. 28 (9):904-920, sep., 1985.

[18] FEINSTEIN, J. et alii. XTEL: an expert system for designing theatherwide telecommunications architectures. In: 7th International Workshop Expert Systems and Their Applications, Avignon, France,13-17 May 1987, p.313-333.

[19] FININ,T et al. FOREST: an expert system for automatic test equipment. MS-CIS-84-16, Univ. Pennsylvania, Philadelphia,

1984, 18 p.

[20] FININ, F. & KLEIN, D. On the requirements of active expert systems. In: 7th International Workshop Expert Systems and Their Applications, Avignon, France, 13-17 May 1987, p.1199-1207.

[21] FRENZEL,L. Crash course in artificial intelligence and expert systems. Howard W.Sams & Co., Indianopolis, 1987.

[22] FU,Li-Min and BUCHANAN,B. Inductive knowledge acquisition for rule-based expert system. Report No STAN-86-1116, Stanford University, Stanford, 1985. 34 p.

[23] GAAG,L. PROLOG:an expert system building tool. Report CS-R8616, Centre for Mathematics and Computer Science, Amsterdan, April 1986. 13 p.

[24] GEORGEFF,M. and LANSKY,A. A system for reasoning in dynamic domains: fault diagnosis on the space shuttle. SRI International, Tech. Note 375, Menlo Park, Jan 1986. 97 p.

[25] GEORGEFF,M. Procedural expert systems. Tech.Note 314, SRI, Menlo Park, Dec 1983. 21 p.

[26] GODO, L. et alii. Managing linguistically expressed incertainty in MILORD - application to medical diagnosis. In: 7th International Workshop Expert Systems and Their Applications, Avignon, France, 13-17 May 1987,p.571-596.

[27] HART, Larry. For network managers, finding faults is no easy task. Data Communications, Sep., 1983, p.189-192.

[28] HAYES-ROTH, Frederick. Rule-based systems. Communications of the ACM. Vol. 28, nº 9, Sep., 1985, p.921-932.

[29] HENDRIX,G. The lifer manual - A guide to building Practical

natural language interface. Tech.Note 138, SRI International, Menlo Park, 1977, 68 p.

[30] HENDRIX,G. Encoding Knowledge in partitioned networks. Tech. Note 164, SRI International, Menlo Park, 1978.

[31] HORTON, Elizabeth, et alii Interactive Repair Assistant: A Knowledge-Based System for Providing Advice to Field Technicians. Communications of the ACM. 31 (5), may, 1988.

[32] ISO 8327. Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Protocol Specification, 1983.

[33] ISO 8073. Information Processing Systems - Open Systems Interconnection - Connection Oriented Transport Protocol Specification,

[34] ISO/DIS 8649/2. Information Processing Systems - Open Systems Interconnection - Service Definition for Common Application Service Elements - Part2 : Association Control.

[35] ISO/TC 97/SC 26/N 4057. OSI Management Framework - Seventh Working Draft. 7/05/1986

[36] ISO/TC 97/SC 26/N 979. OSI Draft Proposal of Management Information Service Definition - Part 2: Common Management Information Service Definition. 10/Sept/1986.

[37] ISO/TC97/SC21/WG4. Draft proposal of Management Information Service Definition -Part 1: Overview.

[38] ISO/TC97/SC221/WG4. Information Processing - Open Systems Interconnection - Part 3: Fault Management Service Definition, Jan.86

[39] ISO/IEC 7498-4. Information processing systems - Open Systems

Interconnection - Basic Reference Model - Part 4: Management Framework 1989

[40] ISO/DIS 8824.2. Information Processing Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)

[41] ISO/DIS 8825.2. Information Processing Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)

[42] ISO. Draft proposal of Management Information Service Definition -Part 1:Overview.ISO/TC97/SC21/WG4, Sept.86

[43] ISO/JTC1/SC21 WP Information Processing - Open Systems Interconnection - Management Information Services - Structure of Management Information. Rome output 1987

[44] ISO/IEC DIS 9595-2 Information Processing Systems - Open Systems Interconnection - Management Information Service Definition - Part 2: Common Management Information Service definition. 28/Dec/1988.

[45] ISO/IEC DIS 9596-2 Editors Draft of Information Processing - Open Systems Interconnection - Management Information Protocol Specification - Part 2: Common Management Information Protocol. 28/Jun/1988

[46] ISO/JTC1/SC21 WP N3509. Information Processing - Open Systems Interconnection - Management Information Services - SMI Part 4: Guidelines for the Definition of Managed Objects. May 1989

[47] ISO/JTC1/SC21 DP 10165-1. Information Processing - Open Systems Interconnection - Management Information Services -

Structure of Management Information. Part 1: Management Information Model. May 1989

[48] JACKSON,P. Forging tools: knowledge representation for expert systems. DAI Research Paper No 224, Univ. Edinburgh, 1984. p.1-6

[49] JONES,K. User models and expert systems. Tech.Report No 61, Univ. Cambridge, 1984. p.1-43

[50] KAELBLING,L. An architecture for intelligent reactive systems. SRI International. Tech.Note 400, Menlo Park, Oct., 1986, 14 p.

[51] KING,J. Artificial intelligence techniques for device troubleshooting. Tech.Note CSL-82-9, Hewlett Packard, Palo Alto-CA, Aug. 1982.

[52] KLEIN, D. An expert systems approach to realtime, active management of a target resource. MS-CIS-85-40 Univ. Pennsylvania, 1985. 95 p.

[53] KOBAYASHI, Yoshikazy. Standardization issues in integrated network management. In: Integrated Network Management, 1, Boston-MA, IFIP,. 14-17 maio 1989. p.70-90.

[54] KOWALSKI Thaddeus. The VLSI design automation assistant: a synthesis expert. AT&T Technical Journal, Jan/ Feb 1988, p.81-103.

[55] KOSAK, Kerry. Designing network control centers for greater productivity. Data Communications, April, 1988.

[56] KUZMAK Sylvia. et alii Knowledge-based signal interpretation. AT&T Technical Journal. JAN/FEB 1988, p.104-120.

[57] LEE, Newton, Knowledge shell programing: developing high-

level knowledge constructs. In: 7th International Workshop Expert Systems & Their Applications, Avignon, France, 13-17 May 1987, p.809-819.

[58] LENOX, Tim, DEAN, Rod. Improve problem management step by step. Data Communications, June, 1984, p.187-200.

[59] LIEBOWITZ, Jay. Expert Systems Applications to Telecommunications. Wiley Interscience. New York, 1988

[60] LUCAS,P. Knowledge representation and inference in rule-based systems. Report CS-R8613, Centre for Mathematics and Computer Science,Amsterdan, April, 1986. 17 p.

[61] MACLEISH,K. et alii. Expert systems in central office switch. IEEE Comm. Magazine, Vol.24, No 9. Sept.1986, p.26-33.

[62] MANTELMAN,Lee. AI carves inroads: Network design, testing and management. Data Communications, Jul,1986,p.106-123.

[63] MARCUS,C. Prolog programming, Addison-Wesley, Menlo Park, 1986.

[64] MARTINS, J. et al. O centro de Supervisão e Controle do Sistema COMPAQ. In: 5º Simpósio Brasileiro de Redes de Computadores, São Paulo, 13-15/abril 1987, p.359-371.

[65] MARQUES, Todd. A symptom-driven expert system for isolating and correcting network faults. Communications of the ACM. 31 (5), may, 1988.

[66] MATHONET, R. et alii. Dantes: An expert system for real-time network troubleshoting.In:7th International Workshop Expert Systems & Their Applications, Avignon, France, 13-17 May 1987, p.469-488.

[67] MILLIKEN K. et alii. YES/MVS and the automation of operations

for large computer complexes. IBM Systems Journal 25 (2): 159-180, 1986.

[68] MILLER, Edward. Expert system validation: issues and approaches. In: 7th International Workshop Expert Systems and Their Applications, Avignon, France, 13-17 May 1987, p. 233-236.

[69] MINES, J. et alii. Integrating performance monitoring with other alarm informations to enhance digital transmissions systems. Proceedings of IEEE International Conference on Data Communications, Amsterdam, May 1984, p.471-475.

[70] MOORE,R. A formal theory of knowledge and action, Tech. Note 320, SRI, Menlo Park, Jul 1984. p.1-84

[71] MOORE, R. The role of logic in artificial intelligence. Tech. Note 335, SRI, Menlo Park, Jul 1984. 29 p.

[72] MOTO-OKA,T., FUCHI,K. The architectures in the fifth generation computers. IFIP 9th World Computer Congress. Paris, 1983, p.589-602.

[73] MOTODA,H. & YAMADA,N. & YOSHIDA,K. A knowledge based systems for plant diagnosis. International Conference on FIFTH Generation Computer Systems 1984, ICOT, Tokyo, 1984, p.582-588.

[74] NEUFELD, G. et alii. EAN: an X.400 message system. In: Computer Message Systems, Proceeding of the IFIP TC6 International Symposium on Computer Message Systems, Washington - DC, 5-7 Sep/85.

[75] NORMAN, Harrel. A user´s guide to network design tools. Data Communication, April 1988.

[76] PICKEREING, G. Just managing a network is not enough. In: INDC90 Information Network and Data Communication, IFIP, Lillehammer, Norway, March 26-29, 1990 p. 2a/1-1 a 1-12

[77] PERRAULT C. & GROSZ,B. Natural-language interfaces. Tech.Not. 393, SRI, Menlo Park, Aug 1986. 40 p.

[78] PODBURY, C. & DILLON, T. Maintenance scheduler: a frame based expert system. In: 7th International Workshop Expert Systems and Their Applications, Avignon, France, 13-17 May 1987, p.649-666.

[79] PUURONEN S. A tabular rule-checking method. In: 7th International Workshop Expert Systems and Their Applications, Avignon, France, 13-17 May 1987, p.257-268.

[80] REBOH,R. Knowledge engineering techniques and tools in the prospector environment. Tech Note 243, SRI, Menlo Park, Jun 1981. 149 p.

[81] REICH,H. & VAN HARMELEN,F. Relevant criteria for choosing an inference engine in expert systems. DAI reasearch paper No 270, Univ. Edinburgh, 1985. 19 p.

[82] ROBINSON, W.Maintenance of digital radio using remote performance monitoring. Proceedings IEEE International Conference on Communications. Amsterdam, May, 1984,p.456-461.

[83] ROSENBERG, R. "Are users up in the air over network management?" Data Communications, Dec 1987.McGraw-Hill Magazine.

[84] SACERDOTTI,E. Problem solving tactics. Tech.Note 189, SRI International, Menlo Park, July 1979. 22 p.

[85] SCHWARTZ,T. Artificial intelligence in the personal computer environment, today and tomorrow. ACO Annual Conference,

Denver 1985.

[86] SLUMAN, Chris. Network and systems management in OSI. Telecommunications, Jan., 1988, p.32-39.

[87] SMITH,G & STRAT,T. A knowledge-based architecture for organizing sensory data. SRI International, Tech.Note No 399, Menlo Park, Dec 1986. 12 p.

[88] STABILE,L. Frame-based computer network monitoring. National Conference on Artificial Intelligence, Pittsburg, 1982, p.327-330.

[89] STOLFO,S. and VESONDER,G. ACE:expert system supporting analysis and management decision making. Separata do "Proceedings of the Eight International Joint Conference on Artificial Intelligence", 8-12 August, 1983, Karlsruhe. 22 p.

[90] SULLIVAN. T. Communication Network Management enhancements for SNA networks: An overview. IBM Systems Journal. Vol 22 Nº 1/2 1983, p.129-142.

[91] TAROUCO, Liane. Gerenciamento de Problemas em Redes. In: PANEL'86 EXPODATA - Conferência Latino-Americana de Informática, 12. Montevidéo-Uruguai, 3-7/novembro/1986. Anais. pág.117-129.

[92] TAROUCO,Liane. Network Management. In:IBERICOM 87-Iberiam Conference on Data Communicatoins, 1. API - Associação Portuguesa de Informática, Lisboa-Portugal and IFIP-TC6, 19-21 Maio, 1987. Palestrante Convidada. Anais. p.217-226.

[93] TAROUCO, Liane. Padronização de servidores de arquivo em redes. 6º Simpósio Brasileiro de Redes de Computadores,Belo Horizonte, 28-30/abril/1988,.

[94] TAROUCO, Liane M.R. Intelligent Network Management. In: Integrated Network Management, 1, Boston-MA, IFIP,. 14-17 maio 1989. p.141-155

[95] TAROUCO, Liane, Suport to Network Management, In: INDC90 Information Network and Data Communication, IFIP, Lillehammer, Norway, March 26-29, 1990 p. 2a/2-1 a 2-12

[96] TAROUCO, Liane & DOTTI, Fernando. MEFISTO - Mechanism Efficient to Foster the Implementation of Software Totally OSI. IN:International Symposium on Local Communications Systems Management, IFIP and The University of Kent at Canterbury, 18-19 September, 1990, Canterbury, UK.

[97] TELEBRAS. Definição do serviço de transporte do modelo de referência de interconexão de sistemas abertos - Modelo ISA. Sistema de Práticas Telebrás, X.214.1, Janeiro 1987.

[98] TELEBRAS. Especificação do protocolo de transporte para interconexão de sistemas abertos. Sistema de Práticas Telebrás, X.224.1, Agosto 1987.

[99] TEMPLE, Nigel. Network management for automated banking.Computer Communications, Vol.7, Nº 4, Aug 1984

[100] TYNOR, S. et alii. GEST: The anatomy of a blackboard expert system tool. In: 7th International Workshop Expert Systems & Their Applications, Avignon, France, 13-17 May 1987, p.793-808.

[101] WALKER, Charles. Network management in the post divestiture era. Data Communications, Feb., 1984, p.109-116.

[102] WALDES,P. et al ii. Are maintenance expert systems practical now? CUCS-166-85, Columbia University, New York, 1985.

[103] WATERMAN,D. A guide to expert systems. Addison-Wesley, Menlo Park, 1986.

[104] WAKID, Shukri et alii. Coming to OSI: network resource management and global reachability. Data Communications, Dec., 1987. McGraw-Hill Magazine.

[105] WEINGARTEN. R & IACOBUCCI, E. Logical problem determination for SNA networks. IBM Systems Journal. 22 (4):387-403, 1983.

[106] WESTPHALL, Carlos & TAROUCO,Liane, WAGNER,Jaime. Medidas para Avaliar Desempenho em Redes. In: Congresso Nacional de Informática,20. SUCESU/SP, São Paulo-SP, 31 Agosto-06 Setembro 1987. Anais. pag. 722-726.

[107] WESTPHALL,Carlos & TAROUCO,Liane. Redes de Computadores com Função de Gerenciamento. In: Congresso Nacional de Informática,20. SUCESU/SP, São Paulo-SP, 31 Agosto - 06 Setembro.1987. Anais. pag. 791-796.

[108] WESTPHALL, Carlos B. e TAROUCO, Liane M.R. Proposição em Gerência de Comunicação de Dados. In: 7º Simpósio Brasileiro de Redes de Computadores (20-22 março 1989). Anais. p. 366-380.

[109] WEISSHEIMER,Cleber, CHIAPIN,Claudio, TAROUCO, Liane, SEREIA-Sistema Eficaz de Rede Empregando Inteligência Artificial. In: 7º Simpósio Brasileiro de Redes de Computadores (20-22 março 1989). Anais. p. 829-838.

[110] WILKENS, Maria Janilce e TAROUCO, Liane M.R. A Implementação do Modelo OSI na UFRGS. In: 7º Simpósio Brasileiro de Redes de Computadores (20-22 março 1989). Anais. p. 296-307.

[111] WITEN I. & MACDONALD B., Concept learning: a practical tool for knowledge acquisition. In: International Workshop Expert Systems and Their Applications, 7th. Avignon, France, 13-17 May 1987, p. 1535-1555.

[112] WOODS,W. Important issues in knowledge representation. Proceedings of the IEEE, 74 (10):1322-1334, oct., 1986.

[113] WRIGHT, Ray. Network Management - the open system future. Telecommunications, Jan., 1988, p.42-73

[114] YEMINI, Y et alii. Network fault Management: a user´s view. In:Integrated Network Management I, Boston-MA, IFIP,. 14-17 maio 1989. p.101-107.

[115] ZADEH, Lofti. Fuzzi logic. IEEE Computer,April 1988,p.83-92.