Universidade Federal do Ceará

Centro de Ciências

Departamento de Computação

Pós-Graduação em Ciência da Computação

Dissertação de Mestrado

SAGRES

Um Sistema Baseado em Conhecimento para Apoio à Gerência de Falhas em Redes de Computadores

Por

Raimir Holanda Filho

raimir@mcc.ufc.br

Orientador : Prof. Dr. Mauro Oliveira

Co-Orientadora : Prof[a]. MsC. Suzana Ramos

UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

Raimir Holanda Filho

SAGRES

Um Sistema Baseado em Conhecimento para Apoio à Gerência de Falhas em Redes de Computadores

Este trabalho será apresentado à Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação.

Orientador : Prof. Dr. Mauro Oliveira

Co-Orientadora: Prof[a]. MsC. Suzana Ramos

Resumo

Este trabalho apresenta o SAGRES, um Sistema baseado em conhecimento para Apoio à Gerência de falhas em REdeS de computadores. São descritas a arquitetura funcional, a arquitetura física, a modelagem e a implementação do SAGRES. Sua arquitetura é caracterizada pelo uso de SGRBCs (Sistemas de Gerenciamento de Redes Baseados em Conhecimento), utilizando, portanto, a diagnose heurística no tratamento de falhas. Os SGRBCs representam a visão de um especialista em gerência de redes e são capazes de analisar situações a partir de um conhecimento sintetizado e das condições existentes. No desenvolvimento do SAGRES, foi utilizada a metodologia DAG (Desenvolvimento de Aplicações de Gerenciamento).

Abstract

This work presents SAGRES, a knowledge-based system supporting network fault management. Its functional and physical architecture, the modelling and its implementation are described. Its architecture is characterized by the use of KBSNM (Knowledge-Based System for Network Management), using therefore a heuristic diagnosis in the treatment of faults. KBSNM's are systems which represent a specialist's vision on network management, formalize the knowledge, being able to analyse the situation from this knowledge. We have used the methodology DAG.

Conteúdo

Lista de Figuras

Capítulo 1

Redes de Computadores

Neste capítulo, apresentamos a evolução da tecnologia de redes de computadores desde o surgimento dos *mainframes* até a utilização das redes de alta velocidade. Em seguida, demonstramos a necessidade do gerenciamento de redes de computadores e a motivação para a realização deste trabalho.

1.1 Evolução das Redes de Computadores

1.1.1 Do Mainframe às Redes de Computadores

A utilização mais efetiva de computadores iniciou-se na década de 1950, época em que existiam apenas os *mainframes* ou computadores de grande porte. Essas enormes máquinas eram utilizadas apenas por poucas universidades, laboratórios de pesquisa ou grandes empresas. Todos os sistemas eram processados em lote (*batch*) e não havia nenhuma forma de interação direta entre os usuários e a máquina.

Os avanços em *hardware* e, principalmente nos sistemas operacionais, na década de 1960, possibilitaram o desenvolvimento dos primeiros terminais interativos, permitindo aos usuários acesso em tempo real ao computador central. Usuários passavam a ter, então, um mecanismo que possibilitava a interação direta com o computador, ao mesmo tempo em que os avanços nas técnicas de processamento davam origem a sistemas de tempo compartilhado (*time-sharing*), permitindo que as várias tarefas de diferentes

usuários ocupassem simultaneamente o computador central, através de uma espécie de revezamento no tempo de ocupação do processador.

Com o propósito de conectar diversos computadores bem como permitir o compartilhamento de informações e recursos computacionais, surgem as redes de computadores. Computadores de diferentes marcas e modelos passam a cooperar através de mecanismos de conectividade e interoperabilidade.

Devido a demora e a complexidade de implementação da solução OSI/ISSO, o Governo dos Estados Unidos, através da Agência de Projetos de Pesquisa Avançados de Defesa (DARPA – *Defense Advanced Research Projects Agency*), começou a estimular pesquisas sobre o tema de redes de computadores através de contratos com os departamentos de computação de várias universidades americanas. Essa pesquisa levou a uma rede experimental, denominada ARPANET, tendo suas operações iniciadas em dezembro de 1969.

A Internet surgiu da ARPANET com a filosofia de tratar a informação transportada, independente do tipo de hardware. Seus principais protocolos, responsáveis pela flexibilidade desta rede, são denominados TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*). Para prover a interconexão destas redes, tais protocolos preconizam um padrão de endereçamento compacto e eficiente, possibilitando o "Serviço de Comunicação Universal".

A aceitação da Internet assumiu, então, proporções inesperadas dentro de vários segmentos da sociedade, interligando instituições de pesquisa, universidades e inúmeras corporações. O sucesso recente da Internet deve-se, basicamente, ao desenvolvimento da tecnologia de transmissão de dados, cada vez mais rápida e segura, e o aparecimento do WWW (*World Wide Web*), um serviço baseado na tecnologia de hipertexto que fornece uma interface gráfica de fácil navegação para visualização de documentos

multimídia. Este serviço alavancou a presença de inúmeras empresas com propósitos comerciais permitindo realizar inúmeras transações a partir da rede e contribuindo para uma maior utilização da Internet.

1.1.2 Arquiteturas OSI x Internet

Para reduzir a complexidade de projeto, a maioria das redes é organizada em camadas ou níveis, umas sobre as outras. O número de camadas, o nome, o conteúdo e a função de cada camada diferem de uma rede para outra. No entanto, em todas as redes, o propósito de cada camada é oferecer certos serviços às camadas superiores, protegendo essas camadas dos detalhes de como os serviços oferecidos são de fato implementados.

O conjunto de camadas e protocolos é chamado de arquitetura de rede [Tan89]. A especificação da arquitetura deve conter informação suficiente para que um implementador possa escrever o programa ou construir o hardware de cada camada de tal forma que obedeça corretamente ao protocolo apropriado.

O modelo da arquitetura OSI (Figura 1.1) se baseia em uma proposta desenvolvida pela ISO como um primeiro passo para a padronização internacional dos diversos protocolos. O modelo é chamado de Modelo de Referência OSI ISSO [Soa95] para Interconexão de Sistemas Abertos.

O modelo OSI tem sete camadas assim definidas:

- Camada Física;
- Camada de Enlace de Dados;
- Camada de Rede;
- Camada de Transporte;
- Camada de Sessão;

- Camada de Apresentação;
- Camada de Aplicação.

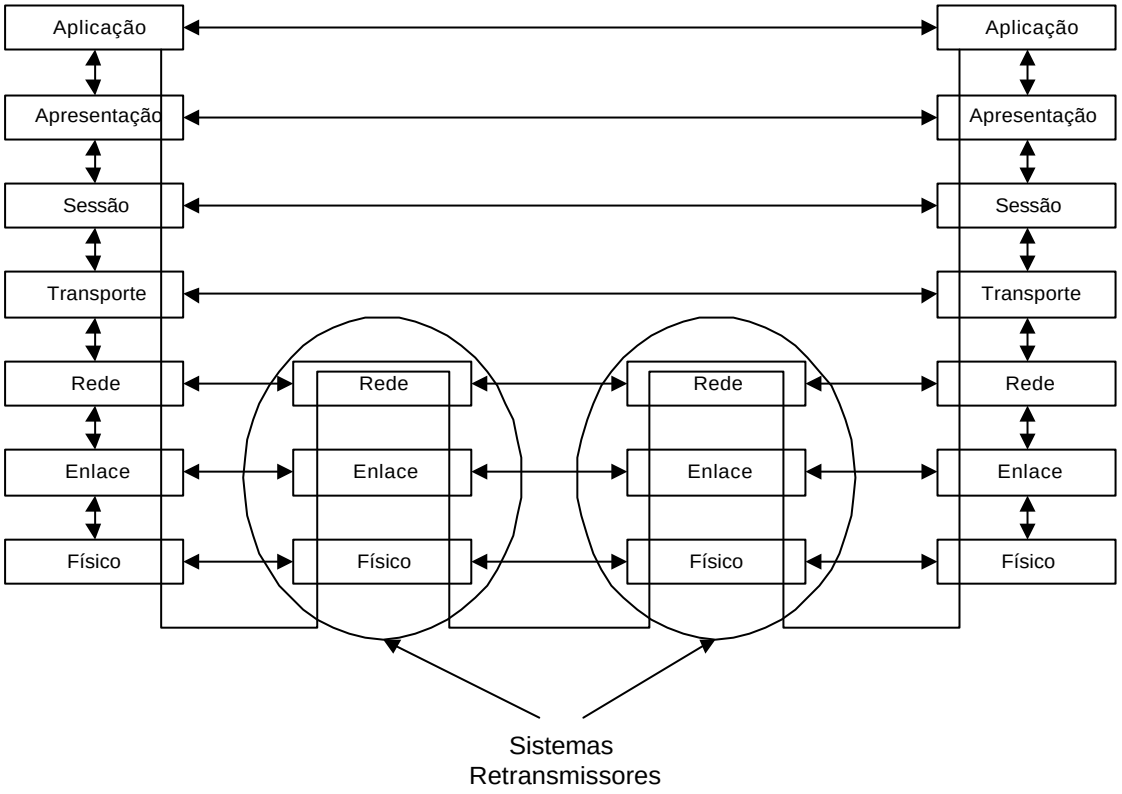


Figura 1.1: Arquitetura OSI

O desenvolvimento da arquitetura Internet TCP/IP foi patrocinado pela DARPA. A arquitetura baseia-se principalmente em um serviço de transporte orientado a conexão, fornecido pelo protocolo TCP, e em um serviço de rede não orientado a conexão, fornecido pelo protocolo IP.

A arquitetura Internet TCP/IP dá uma ênfase toda especial à interligação de diferentes tecnologias de redes. Sua arquitetura é organizada em quatro camadas conceituais

construídas sobre uma Quinta camada que não faz parte do modelo, a camada intra-rede. A figura 1.2 apresenta as camadas da arquitetura Internet.

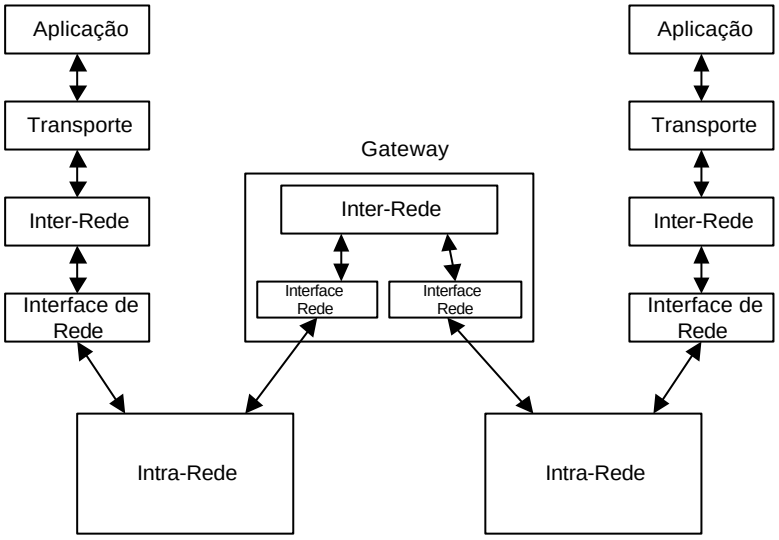


Figura 1.2: Arquitetura Internet

1.1.3 Redes Locais de Computadores

A primeira motivação no surgimento das redes locais nos anos 80 foi o compartilhamento dos recursos de hardware, notadamente do disco rígido, que na época era um recurso relativamente caro. Assim, surgia o Servidor de Disco, um software capaz de alocar logicamente partições de um disco a diversos usuários da rede.

A necessidade de permitir o compartilhamento de software e de dados deu origem ao conceito de Servidor de Arquivos. Esse servidor aceita solicitações que chegam de usuários da rede local e retorna arquivos de dados ou aplicações, de forma compartilhada. Esse recurso de software transformaria a rede local em um sistema

computacional de grande utilidade , porém longe de ser competitivo com o *mainframe*, do ponto de vista custo/benefício.

A Figura 1.3 detalha o funcionamento do Servidor de Arquivos a partir de uma aplicação de Banco de Dados. Neste exemplo, uma aplicação numa estação solicita uma base de dados armazenada num Servidor de Arquivos e a processa localmente. Vale observar que há um tráfego intenso na rede durante a transmissão da base de dados.

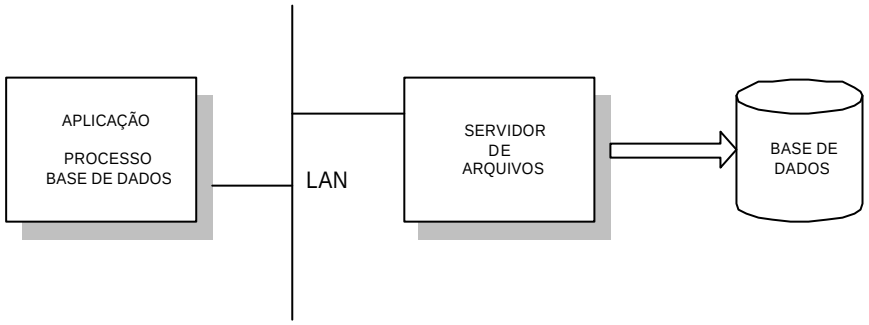


Figura 1.3: Servidor de Arquivos

A etapa seguinte na evolução das redes de computadores foi o surgimento do modelo Cliente/Servidor, uma arquitetura composta por dois processos distintos e cooperantes: o processo Cliente e o processo Servidor. Nesse modelo, a origem da conversação entre os processos cooperantes define se um determinado processo é cliente ou servidor. O processo que inicia a comunicação é chamado cliente. A cada vez que uma aplicação cliente é executada, o processo cliente gerado contacta um processo servidor, envia um pedido por serviço e fica bloqueado aguardando, uma resposta. Quando a resposta chega do processo servidor, o processo cliente pode, então, prosseguir com a sua execução. Servidores aceitam pedidos vindos da rede, executam seu serviço, e retornam resultados para os requisitantes.

A Figura 1.4 mostra o uso do modelo Cliente/Servidor numa aplicação de banco de dados. Nesse modelo, diferentemente do servidor de arquivo o tratamento da solicitação é feito no servidor de banco de dados, sendo enviado ao cliente apenas o resultado da solicitação [Orf96]. Conclui-se que nesse modelo, o tráfego na rede é bem menor, pois não é enviada toda a base de dados mas apenas o resultado da solicitação.

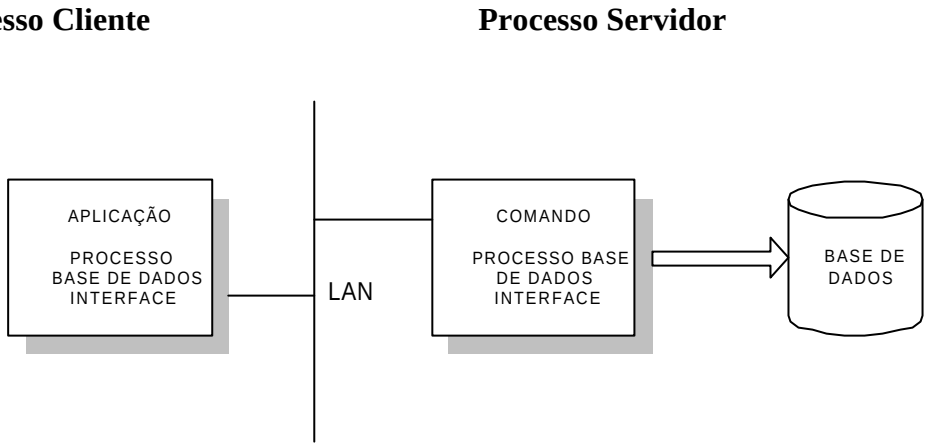


Figura 1.4: Modelo Cliente Servidor

As tecnologias para o desenvolvimento de aplicações Cliente/Servidor podem ser:

- Banco de Dados
- Monitores de TP (*Transactions Processing*):
- *Groupware*:
- Objetos Distribuídos:
- Tecnologia WEB:

Um refinamento do conceito de Cliente/Servidor é o *Remote Procedure Call* (RPC) ou Chamada de Procedimentos Remotos. Ele é o mecanismo clássico para a construção dos chamados Sistemas Distribuídos [Tan95].

A idéia básica do RPC é fornecer ao desenvolvedor de aplicações distribuídas um ambiente no qual a semântica da chamada de procedimentos remotos seja idêntica à de

procedimentos feitos nos sistemas centralizados. Assim feito, a existência de processos espalhados em uma rede de computadores torna-se transparente para o usuário.

O mecanismo chave do RPC é o Stub, ilustrado na Figura 1.5. Graças ao Stub, tanto a parte da aplicação Cliente quanto a parte da aplicação Servidor comunicam-se localmente, ficando ao encargo do mesmo resolver ao nível do sistema de comunicação o problema de localização física dos processos cooperantes.

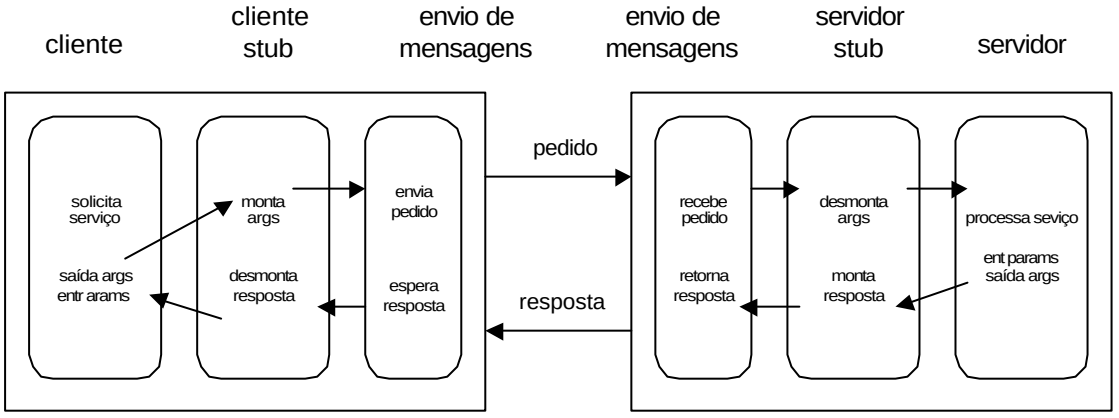


Figura 1.5: Representação RPC

Com o objetivo de fornecer uma interoperabilidade entre aplicações em diferentes máquinas em ambientes heterogêneos distribuídos o OMG (Object Management Architecture) adotou uma tecnologia denominada ORB (*Object Request Broker*).

O ORB tem as seguintes características:

- prover mecanismos pelos quais os objetos fazem pedidos e recebem respostas de forma transparente; e
- prover interoperabilidade entre aplicações em diferentes máquinas com ambientes distribuídos e heterogêneos.

CORBA foi a tecnologia ORB adotada pelo OMG. Ela define uma estrutura para que diferentes implementações de ORBs possam prover serviços e *interfaces* comuns para suportar clientes e implementações de objetos portáveis.

Qualquer objeto desenvolvido em conformidade com o padrão ORB deve garantir portabilidade e interoperabilidade de objetos sobre uma rede de sistemas heterogêneos. Basta que o ORB seja definido por suas *interfaces* e, então, qualquer implementação adequada às *interfaces* se torna aceitável. Tais *interfaces* são organizadas nas seguintes categorias:

- operações padrão para toda implementação de ORB;
- operações específicas a um tipo de objeto;
- operações específicas a um estilo de implementação de objetos.

1.1.3 Redes de Alta Velocidade

Até pouco tempo, cada tipo de aplicação necessitava de um tipo específico de rede de comunicação para ser implementada. Por exemplo, o serviço de voz utiliza rede comutada de circuito, enquanto o serviço de dados utiliza a rede comutada de pacotes. Atualmente deseja-se que uma rede simultaneamente suporte todos esses serviços, o que é chamado de Rede de Integração de Serviços (RDSI).

Com o advento das redes de comunicação de alta velocidade, há a possibilidade de se integrar esses serviços, e ainda disponibilizar outros serviços de tempo-real que eram impraticáveis em tecnologias passadas. Tais serviços, possuem freqüentemente perfis de tráfego totalmente diferentes dos tradicionais, o que levou ao estudo de novas tecnologias de rede como ATM [Soa95].

ATM é uma tecnologia de comunicação que utiliza um protocolo comutado por circuito de alta velocidade, que oferece a capacidade de transmissão de dados a altas velocidades com "delay" mínimo e qualidade de serviço (QoS) garantida. O ATM transfere pacotes de tamanho fixo de 53 bytes, denominados células.

Por ser uma tecnologia de transmissão mais flexível, ATM oferece a promessa de integração de diversos serviços (ex. voz, vídeo e dados). ATM também possibilita a utilização de transmissões ponto-a-ponto e multiponto, enquanto provê largura de banda de maneira escalonável e oferece a promessa de integrar redes de longa distância (WAN) e redes locais (LAN).

Vários órgãos tem definido padrões para redes ATM, tais como ITU-T, ATM Forum, Belcore e ANSI. Este processo de padronização assegura uma alta probabilidade de interoperabilidade entre os produtos dos fornecedores.

1.2 Necessidade de Gerenciamento em Redes de Computadores

Gerenciar qualquer sistema consiste, basicamente, nas atividades de monitorar os elementos, analisá-los à luz de uma política previamente estabelecida e atuar sobre esses elementos, de modo a manter o sistema funcionando dentro de padrões aceitáveis.

A necessidade de gerenciamento das redes atuais deve-se, dentre outros fatores, à complexa estrutura de implementação dessas redes, onde pode-se encontrar diversas LANs interligadas entre si local ou remotamente (via WANs ou MANs) através de variados equipamentos de interconexão. A gerência de uma rede é, então, uma tarefa extremamente complexa pois envolve uma quantidade significativa de variáveis associadas a softwares, hardwares e meios de comunicação. A complexidade no gerenciamento de redes de computadores pode ser observada sob dois aspectos:

- primeiramente, devido a heterogeneidade dos componentes de diversos sistemas provenientes de diferentes fornecedores dentro de um mesmo domínio organizacional (empresas, departamentos, etc.), resultando na convivência de protocolos e filosofias de gerenciamento incompatíveis, não operando de uma maneira integrada. Como conseqüência, temos o aumento do custo e ineficiência na operação e manutenção do sistema; e
- em segundo lugar, temos a questão da interoperabilidade entre diversos domínios organizacionais, cada uma com suas necessidades e políticas de gerenciamento próprias. Como conseqüência, organismos de pesquisa e normalização, bem como fabricantes de redes de computadores, têm atentado para a necessidade da adoção de estratégias padrões, no sentido de contornar os problemas provenientes dessa complexidade a nível intra-domínio (heterogeneidade) e de inter-domínio (interoperabilidade).

Entre os aspectos de heterogeneidade nas redes que influenciam a atividade de gerência, vale ressaltar:

- a heterogeneidade no nível das arquiteturas dos sistemas interligados - as redes interconectadas são com freqüência compostas de sub-redes distribuídas que possuem arquiteturas diferentes (LAN, MAN, WAN) e que necessitam de mecanismos de gerência específicos. Como em quase todos os sistemas de engenharia, os pontos fracos localizam-se nos elementos de interconexão. Esses elementos, lógicos ou físicos, são compartilhados pelas redes adjacentes que os gerenciam diferentemente, necessitando, portanto, de um tratamento especial para a unicidade da atividade de gerência; e
- a heterogeneidade no nível dos dados a serem transmitidos - as novas tecnologias disponíveis permitem o transporte nas redes de imagem, voz e dados, cada um exigindo técnicas de gerência bem particulares.

O gerenciamento integrado de sistemas multifornecedores é portanto, um problema importante a ser resolvido, pois se de um lado a aquisição de componentes de um único fornecedor resolve o problema causado pela heterogeneidade, por outro lado causa um indesejável problema de dependência a este fornecedor.

1.3 Motivação do Trabalho

A popularização e o crescimento de redes de computadores têm ocasionado um aumento substancial de problemas e a conseqüente necessidade de gerenciamento cada vez mais eficiente dos recursos destas redes. Dado a complexidade e a sofisticação de redes, seu gerenciamento não pode ser realizado somente com o esforço humano, exigindo soluções semi-automatizadas para superar tais dificuldades.

Uma alternativa de automação do processo de gerência seria a utilização de sistemas baseados em conhecimento. Sistemas baseados em conhecimento são sistemas capazes de representar a visão de um especialista, de sintetizar o conhecimento e as condições existentes, e de analisar situações a partir deste conhecimento. Muitas razões justificam o uso de sistemas baseados em conhecimento em gerência de redes, incluindo o tratamento eficiente da crescente complexidade da rede, o tratamento consistente de problemas de gerência e o aproveitamento do conhecimento de pessoal especializado.

Esta dissertação propõe um sistema baseado em conhecimento para apoiar a gerência de falhas de redes de computadores. Seu principal objetivo é reduzir o nível de complexidade inerente à atividade de gerência. Este sistema, denominado SAGRES foi inserido no contexto da metodologia DAG (Desenvolvimento de Aplicações de Gerenciamento).

O esquema da figura 1.6 mostra os elementos constituintes do modelo de concepção do SAGRES. Estes elementos, abordados nos capítulos que se seguem, podem ser agrupados em dois blocos:

- Metodologia SAGRES
  - Contexto (Capítulo 2)
  - Infraestrutura Conceitual (Capítulos 3,4,5)
  - Arquitetura Funcional (Capítulo 6)
- Implementação SAGRES (Capítulo 7)
  - Arquitetura Física
  - Especificação Formal
  - Prototipagem

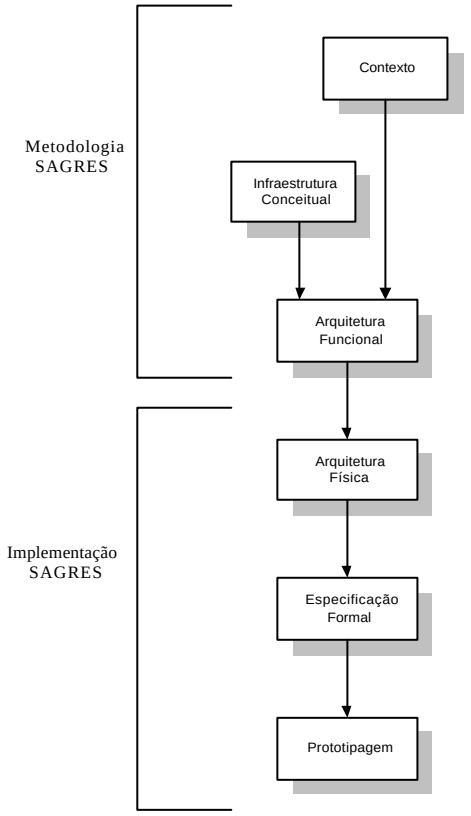


Figura 1.6: Modelo de Concepção do SAGRES

1.4 Estrutura da Dissertação

Esta dissertação foi dividida em 8 (oito) capítulos, sendo composta dos seguintes temas: redes de computadores, gerenciamento de redes, gerência de falhas, metodologia DAG, sistemas de gerenciamento de redes baseados em conhecimento, metodologia SAGRES, implementação do SAGRES e conclusões.

O Capítulo 1 apresenta a evolução ocorrida nas redes de computadores, a necessidade de gerenciá-las e a motivação para a realização deste trabalho.

O Capítulo 2 expõe uma visão geral dos principais conceitos de gerenciamento de redes, descrevendo os modelos de gerência OSI e Internet. São enfocados os conceitos de elementos gerenciados, agentes, gerentes, áreas funcionais e as características de alguns sistemas de gerenciamento de redes presentes no mercado.

O Capítulo 3 explora os conceitos de gerência de falhas. São apresentados os problemas e funções de monitoramento de falhas e os modelos de diagnose de falhas.

O Capítulo 4 descreve a metodologia DAG (Desenvolvimento de Aplicações de Gerenciamento). São apresentadas suas três fases e os módulos constituintes de cada uma delas.

O Capítulo 5 apresenta os conceitos de SGRBCs (Sistemas de Gerenciamento de Redes Baseados em Conhecimento) e como são classificados estes sistemas. Uma arquitetura típica de um SGRBC é apresentada, sendo especificados os benefícios de sua utilização. É apresentado também o estado atual da pesquisa, no Brasil, em sistemas baseados em conhecimento. São analisados os sistemas OLHO VIVO, Agente 6 e I-DREAM.

O Capítulo 6 apresenta o SAGRES, um sistema baseado em conhecimento para apoio à gerência de falhas em redes de computadores. Nesse capítulo é descrito sua arquitetura funcional.

O Capítulo 7 trata da implementação do SAGRES. São apresentados sua arquitetura física, sua modelagem e a prototipagem. Ao final do capítulo é realizado um estudo de caso.

O Capítulo 8 apresenta as considerações finais desta dissertação. São relatados as alternativas de trabalhos futuros bem como as conclusões deste trabalho.

Capítulo 2

Gerenciamento de Redes

Neste capítulo são apresentados os conceitos sobre gerência de redes, com ênfase nos modelos OSI e Internet. A figura 2.1 mostra o modelo de concepção do SAGRES (Figura 1.6) com detalhamento do bloco contexto.

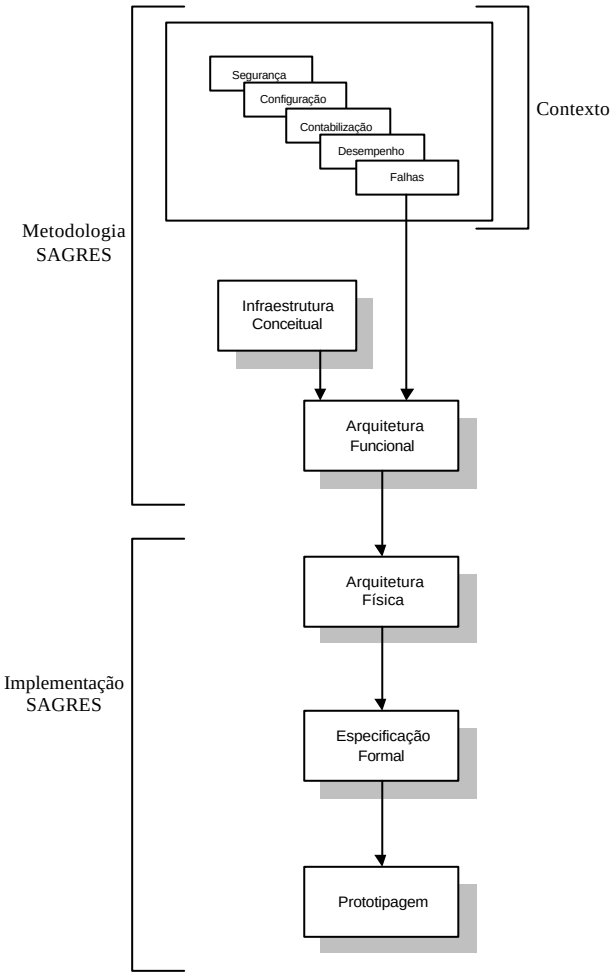


Figura 2.1: Modelo de Concepção do SAGRES - Contexto

2.1 Conceitos Gerais

Uma rede deve propiciar os tempos de resposta requeridos para cada aplicação, alta disponibilidade e custos compatíveis com os serviços oferecidos.

Dada a importância vital de uma rede corporativa para suportar os sistemas de informação, as organizações investem altas quantidades de recursos financeiros, humanos e materiais, que precisam ser gerenciados com eficiência.

A gerência de redes de computadores provê mecanismos para que uma corporação mantenha sob controle, e de forma integrada, os recursos que compõem a sua infra-estrutura tecnológica para o tratamento da informação. Compreende a monitoração, análise e resolução de problemas, dentre outras atividades necessárias para a manutenção de uma rede com qualidade de serviços adequada aos objetivos dos sistemas de informação.

A atividade de gerência cresce em importância e complexidade na proporção em que se diversificam o número de tecnologias de sistemas operacionais, de protocolos de rede e de elementos necessários para interconectar todos estes componentes. Com a rápida evolução da tecnologia de redes, aumenta também a freqüência com que surgem novos elementos agregados à rede corporativa, constituindo-se em novos elementos de rede a serem gerenciados.

A solução para possibilitar a gerência integrada dos elementos de diferentes tecnologias e de diferentes fabricantes em uma rede passa pela utilização de uma arquitetura aberta de gerência de rede. Uma arquitetura aberta especifica um conjunto de protocolos não proprietários. Com isso, um mesmo sistema de gerência pode coletar e tratar informações de maneira uniforme e consistente e executar operações sobre um determinado elemento na rede, não importando o seu tipo ou seu fabricante. Novos elementos podem ser incorporados também a qualquer momento.

2.1.1 Funções de Gerenciamento de Redes

As funções de gerenciamento a serem executadas numa rede podem ser agrupadas nas seguintes categorias [Sta93]:

- Monitoramento: é essencialmente uma função de leitura onde são coletadas informações sobre os elementos de rede gerenciados;

- Análise: a partir dos dados coletados e em função de uma política de gerência, decisões são tomadas; e
- Controle: é uma função tipicamente de escrita, relacionada com a alteração da configuração dos componentes da rede.

2.1.2 Objeto Gerenciado

Gerenciamento de redes tem como base o conceito de Objeto Gerenciado (*Managed Object*). Em geral, a definição de um Objeto Gerenciado apresenta dois aspectos [Sta93]:

- onde ele se situa (localização dentro do sistema sendo gerenciado); e
- sua natureza (representada por seus atributos).

Em sendo o Objeto Gerenciado a representação de um recurso físico ou lógico de rede, qualquer elemento de rede não modelado como um Objeto Gerenciado é invisível ao sistema de gerenciamento. A Figura 2.2 ilustra esse conceito.

Dois tipos de interações entre as aplicações de gerenciamento e os objetos gerenciados são identificadas:

- a primeira é relativa às informações ditas estáticas e corresponde ao uso das informações de gerenciamento que não apresentam alteração em seu conteúdo; e

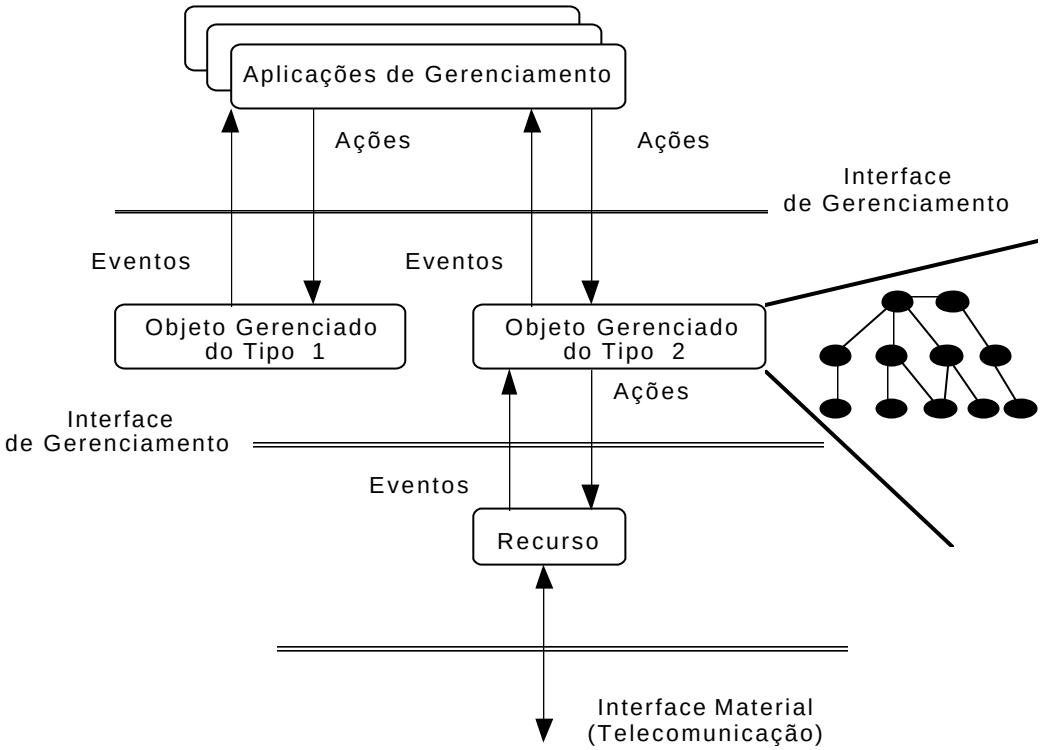


Figura 2.2 : Relação entre objetos gerenciados e recursos reais de rede

- a segunda é relativa às informações ditas dinâmicas e corresponde à manutenção da coerência das informações de gerenciamento entre os recursos reais e seus respectivos objetos gerenciados. Os recursos reais podem gerar eventos que são transmitidos aos objetos gerenciados para atualizarem os atributos devidos. No outro sentido, quando uma aplicação envia uma ação atualiza-se o objeto gerenciado antes de aplicar a referida ação sobre o recurso real. Desse modo, uma visão coerente dos recursos gerenciados é garantida.

A classificação dos recursos reais de rede é feita de acordo com as seguintes regras:

- um recurso real é a unidade básica de um sistema gerenciado do ponto de vista de gerenciamento;
- um recurso real possui uma interface de gerenciamento através da qual é feita a comunicação com o sistema de gerenciamento;
- os recursos reais têm a capacidade de detectar a ocorrência de eventos e de reagir às ações de gerenciamento;
- os recursos reais podem ser programas (*software*) ou materiais (*hardware*), ou ambos; e
- os recursos reais podem ser agrupados de acordo com as necessidades da atividade de gerenciamento.

Os recursos reais devem ser claramente distinguidos de suas representações em termos de objetos gerenciados. O conjunto de axiomas abaixo define a correlação entre os objetos gerenciados abstratos e os recursos reais que eles representam:

- um objeto gerenciado único pode representar um recurso real único;
- um objeto gerenciado único pode representar um certo número de recursos reais inter-relacionados;
- vários objetos gerenciados podem representar um único recurso;
- podem existir recursos que não possuem objetos gerenciados que os representem. Nesse caso, esses recursos não podem ser gerenciados através da interface de gerenciamento e, portanto, não são visíveis pelo sistema;
- um objeto gerenciado pode representar uma relação entre recursos; e
- um objeto gerenciado pode representar outro (s) objeto (s) gerenciado (s).

2.1.3 Paradigma Gerente-Agente

Sob o enfoque de gerenciamento de redes, dois ou mais processos podem associar-se para prover uma instância de aplicação de gerenciamento. As interações que ocorrem entre tais processos são modelados como operações de gerenciamento (Figura 2.3). Para uma determinada associação de gerenciamento, os processos de aplicação envolvidos podem assumir os seguintes papéis [ISO/IEC DIS 10040] :

- Gerente: responsável pelo envio de operações para serem realizadas pelos agentes sobre os elementos gerenciados. Pode existir um ou mais sistemas que implementam processos gerente em uma rede, operando de forma integrada; e
- Agente: processo de aplicação que é implementado em cada elemento de rede. Tem como finalidade prover uma interface entre o gerente e o elemento a ser gerenciado. O processo agente executa as operações de gerenciamento emitidas pelo processo gerente e fornece uma visão dos elementos gerenciados. Ele também pode emitir notificações que espelhem o comportamento dos mesmos.

A seguir citamos alguns exemplos de interação entre gerente e agente em uma rede:

- Um gerente requisita uma operação para ligar ou desligar um equipamento na rede. Esta solicitação é interpretada pelo agente, que interage com o respectivo equipamento, fazendo com que este execute os procedimentos físicos para cumprir a solicitação; e

- Um agente implantado em um determinado roteador na rede monitora constantemente o tráfego em todas as suas portas de comunicação e quando uma delas excede o limite pré-estabelecido, emite uma notificação para o gerente.

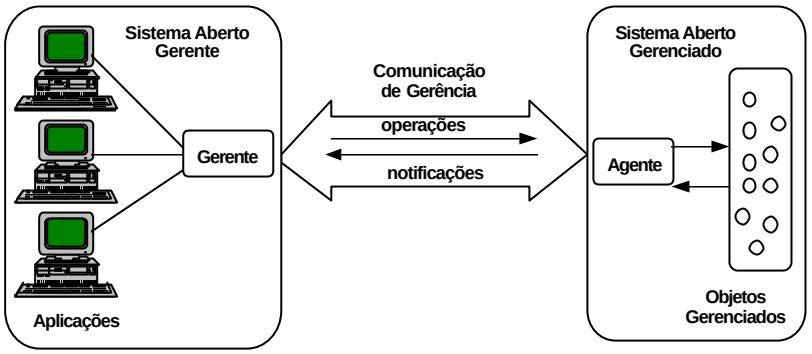


Figura 2.3 : Modelo Gerente-agente

2.1.4 MIB

O gerenciamento de uma rede de computadores necessita de uma base para armazenar os Objetos Gerenciados. Assim, um componente fundamental em um sistema de gerenciamento é a Base de Informações de Gerenciamento ou MIB (*Management Information Base*) [ISO/IEC 7498-4] a qual é um repositório destes objetos. Quando um gerente solicita a um agente informações relativas a um determinado elemento gerenciado, o agente responde à consulta enviando um relatório baseado no conteúdo de sua MIB.

O conceito de MIB não impõe nenhuma condição ao nível de normalização de sua estrutura interna ou ao nível de organização local de uma base de informação. O essencial é que todo o sistema seja capaz de identificar corretamente os objetos constituintes da MIB. Em outras palavras, a MIB é o repositório conceitual de todos os Objetos Gerenciados, não importando qual seja o meio para armazenamento físico das informações de gerenciamento.

Objetos Gerenciados OSI são definidos em termos de:

- Atributos: são propriedades dos Objetos Gerenciados;
- Operações: são as operações a que estão submetidos os Objetos Gerenciados;

- Notificações: são informações que os Agentes podem emitir ao Gerente para informar sobre a ocorrência de eventos nos Objetos Gerenciados; e
- Relações com outros objetos.

A Figura 2.4 apresenta um exemplo de definição para objetos gerenciados OSI. Os objetos são definidos em forma de templates, que é um formato padrão utilizado.

```
pduCounterObject MANAGED OBJECT CLASS
        DERIVED FROM "CCITT REC.X.721 (1992)|ISO ...;
        CHARACTERIZED BY
                basePackage PACKAGE – in-line PACKAGE definition
                        ATTRIBUTES    pduCounterName
                                      GET;
                                    pduCounter
                                      INITIAL VALUE syntax.initialZero
                                      GET
                ; -- End of in-line PACKAGE definition
        ; -- End of CHARACTERIZED BY construct
        CONDITIONAL PACKAGES additional Package
                PRESENT IF *enable/disable control is required*;
REGISTERED AS {object-identifier 1}
```

Figura 2.4: Template OSI

A cláusula "DERIVED FROM" indica que a definição deste atributo é derivada de uma outra definição já existente. A cláusula "Characterized By" permite que um ou mais pacotes obrigatórios sejam incluídos na definição da classe do objeto. A cláusula "CONDITIONAL PACKAGES" permite que um ou mais pacotes condicionais sejam incluídos.

Um atributo tem um valor que pode ter uma estrutura simples ou complexa. Parte da definição de um Objeto Gerenciado é a especificação do conjunto de operações de gerenciamento que ele pode executar, bem como o

efeito que estas operações têm sobre outros Objetos Gerenciados e seus atributos. Em um cenário típico, o Gerente solicita uma operação ao Agente de acordo com o esquema conceitual dos Objetos Gerenciados. É responsabilidade do Agente fazer com que a operação sobre o Objeto Gerenciado afete o recurso que este modela.

Alguns aspectos importantes são a definição de uma estrutura lógica, a compreensão das ações a serem executadas sobre os Objetos Gerenciados na MIB, bem como os eventos que eles podem gerar. Se por um lado a informação é a base para todo e qualquer processo de gerenciamento, em um processo de coleta de dados é necessário adotar padrões para que diversos usuários destes dados possam falar a mesma linguagem. A forma de criação de um item de dado segue uma Estrutura de Informação de Gerenciamento ou SMI (*Structure of Management Information*), que consiste num conjunto de regras a ser obedecido para definir e identificar variáveis na MIB.

A SMI determina que todos os nomes e tipos de variáveis da MIB devem ser definidos e referenciados em ASN.1 (*Abstract Sysntax Notation 1*), que é uma linguagem formal padronizada pela ISO (*International Organization for Standardization*). A ASN.1 determina regras tanto para a linguagem humana quanto para a sua representação codificada, que é utilizada nos protocolos de comunicação.

2.1.5 Áreas Funcionais de Gerenciamento de Redes

Segundo a ISO, as diversas atividades de gerenciamento de redes podem ser divididas em cinco áreas funcionais (Figura 2.5) específicas denominadas por gerenciamento de falhas, gerenciamento de desempenho, gerenciamento de configuração, gerenciamento de contabilização e gerenciamento de segurança [ISO/IEC 7498-4].

A seguir, são caracterizadas as áreas funcionais de gerenciamento de redes propostas pela ISO :

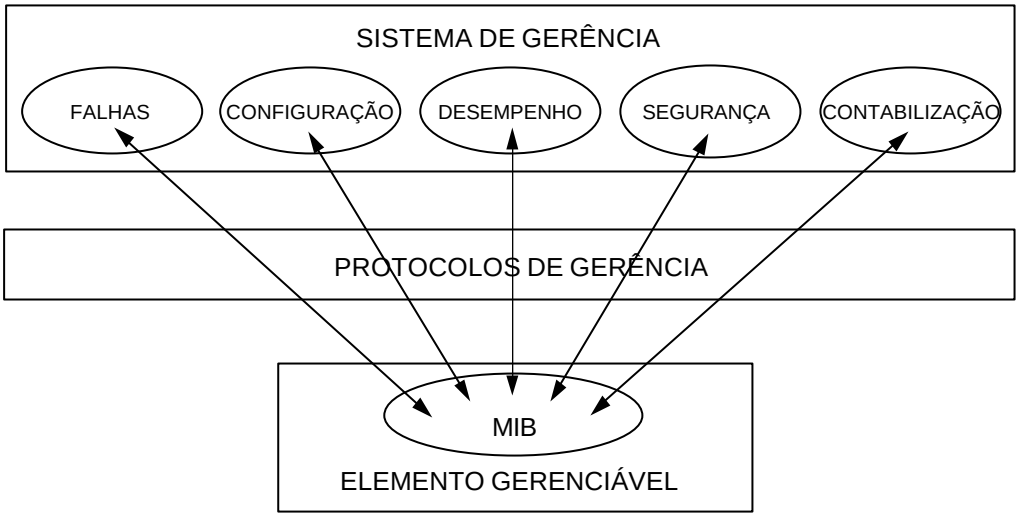


Figura 2.5 : Áreas Funcionais no Sistema de Gerência de Rede

- Gerência de Falhas: corresponde à área funcional que permite a detecção, o isolamento e a correção de operações anormais na rede. Os recursos de gerenciamento de falhas mostram ao administrador de rede, o número, tipos, horas de ocorrência e localizações de erros na rede. Quando ocorrem falhas em uma rede, é importante que os seguintes procedimentos sejam seguidos: localizar a falha, isolar a falha do restante da rede e reparar os componentes em falha, de forma a retornar a rede ao seu estado normal;
- Gerência de Desempenho: os elementos que compõem uma rede precisam ser monitorados de forma constante com a finalidade de avaliar o seu comportamento. Tais informações podem ser utilizadas para fins de planejamento e controle da qualidade de serviço na rede. Através de estatísticas de desempenho, pode-se promover ações para antecipar-se a problemas que venham a ocorrer pela degradação crescente dos tempos de resposta, motivado por problemas ou por saturação de capacidade dos equipamentos ou dispositivos na rede. O sistema deve prover o estabelecimento de limiares de comportamento permitidos para cada elemento, de forma que sejam emitidas notificações para motivar uma ação quando estes valores forem atingidos;
- Gerência de Configuração: compreende o conjunto de facilidades que lidam com a instalação, inicialização, modificação e registro de parâmetros de configuração. As redes de computadores podem constituir-se de milhares de equipamentos e dispositivos dispersos por vários locais físicos diferentes em uma organização, muitos deles envolvidos em mudanças freqüentes de localização. Para a gerência da rede é fundamental que sejam conhecidas a localização de cada equipamento ou dispositivo, suas especificações técnicas e configurações, os responsáveis pela manutenção ou correção de problemas, dentre outras informações;

- Gerência de Contabilização: registra as informações sobre a utilização dos recursos da rede com o objetivo de quantificá-los para efeito de distribuição de custos, de tarifação, de planejamento de capacidade, e verificação de cotas de utilização; e
- Gerência de Segurança: corresponde ao conjunto de funções responsáveis pela criação e supressão de mecanismos de segurança na rede. Uma rede de computadores de uma organização somente deve ser acessada por pessoas ou aplicações que possuam a devida autorização. As informações sensíveis para o negócio da corporação devem ser mantidas de forma segura, prevenindo-se contra os acessos indevidos, seja para leitura ou para alteração da informação. Os procedimentos de uma gerência de segurança devem incluir a identificação dos pontos de acesso em uma rede, definição dos procedimentos de segurança e, principalmente, manter estes pontos seguros, informando inclusive as tentativas de ataque para uma ação preventiva. Estão inseridos neste contexto os procedimentos de autenticação dos usuários para acesso aos sistemas de processamento, a implementação de *Firewalls*, as técnicas de criptografia para o transporte da informação, a geração e manutenção de cópias de segurança dos arquivos dentre outros.

2.2 Arquiteturas para Gerenciamento de Redes

Conforme dito no capítulo 1 existem atualmente duas arquiteturas padronizadas para interconexão de sistemas abertos: o Modelo de Referência OSI e o Ambiente de Protocolos TCP/IP. Ambos possuem em suas definições um conjunto de protocolos abertos para implantação de sistemas de gerência de rede e são conhecidos como gerenciamento OSI e gerenciamento Internet.

2.2.1 Gerenciamento OSI

2.2.1.1 Modelo de Gerenciamento OSI

O modelo de gerenciamento OSI define um sistema integrado composto de gerentes, de agentes e de objetos gerenciados que modelam os recursos de interesse do sistema de gerenciamento [Sta93,ISO/IEC 7498-4,Uda96].

O responsável pelas funções de gerenciamento num ambiente OSI é o Processo de Aplicação de Gerenciamento do Sistema ou SMAP (*System Management Application Process*). Como mostra a Figura 2.6, o SMAP pode atuar tanto como gerente ou como agente.

A troca de informações entre sistemas é realizada pelas Entidades de Aplicação de Gerenciamento de Sistemas, SMAE (*Service Management Application Entity*). A SMAE residente no sistema agente comunica-se com a SMAE no sistema gerente através do protocolo de comunicação CMIP (*Common Management Information Protocol*).

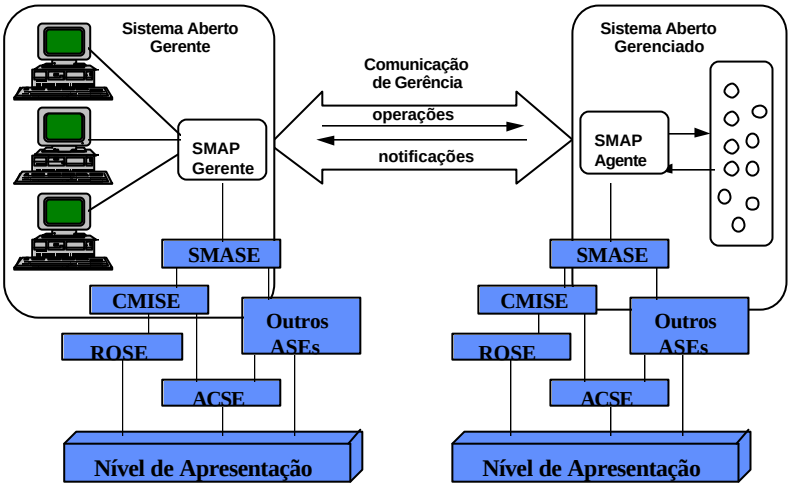


Figura 2.6 : Modelo de Gerenciamento OSI

Assim, o SMAP utiliza os serviços de gerenciamento fornecidos pelo SMAE com o objetivo de executar funções de gerenciamento e transporta as operações de gerenciamento, as notificações, bem como os resultados da execução destas operações através do CMIP.

No modelo de gerenciamento OSI, cada camada é gerenciada pela Entidade de Gerenciamento da Camada (LME - *Layer Management Entity*) (Figura 2.7), a qual é responsável pelo mapeamento dos objetos de gerenciamento na MIB em objetos e eventos reais relativos à camada que gerencia.

Em relação à sua estrutura de gerenciamento, o modelo OSI é subdividido em três áreas denominadas: Gerenciamento de Sistema, Gerenciamento de Camada e Operação de Camada; a seguir detalhadas:

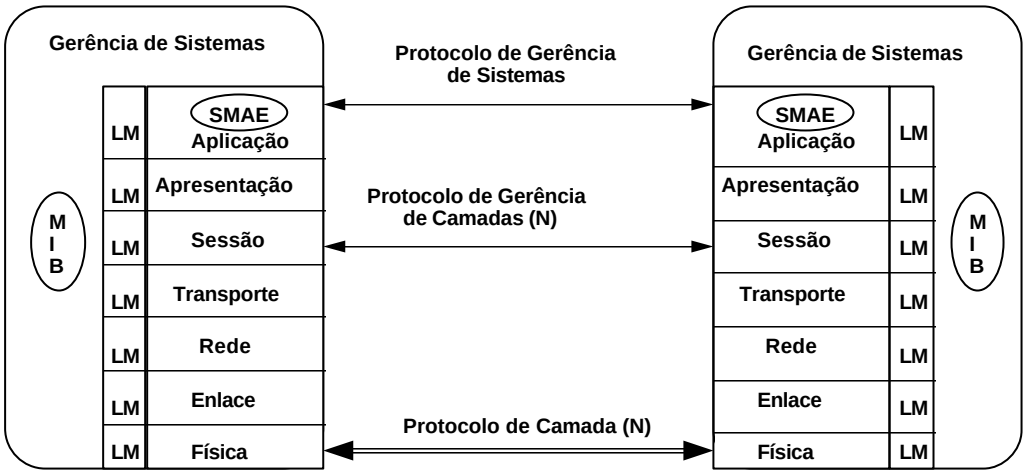


Figura 2.7: Estrutura de Gerenciamento OSI

- Gerenciamento de Sistemas (SM - *System Management*) - Fornece os mecanismos para a supervisão, o controle e a coordenação dos objetos gerenciados em um sistema aberto. Estes objetos podem pertencer a uma ou várias camadas, sendo o SM, portanto, o único meio de realizar o gerenciamento das várias camadas OSI. É, portanto, através dos protocolos do SM da camada de aplicação que quaisquer objetos pertencentes ou associados a um sistema aberto podem ser gerenciados, o que exige funções de apoio em todas as sete camadas. As comunicações entre SMs são realizadas através das entidades de aplicação SMAE. Este gerenciamento faz uso do protocolo do nível de aplicação SMP (*System Management Protocol*).

- Gerenciamento de Camadas (N - *Layer Management*) - Fornece os mecanismos que permitem a supervisão e a coordenação dos objetos de comunicação associados a uma determinada camada N, ou seja, este gerenciamento é realizado sobre objetos relacionados com atividades na mesma camada. Para realizar este gerenciamento, um protocolo de propósito especial denominado LMP (*Layer Management Protocol*) é utilizado.

- Operação de Camada (N - *Layer Operation*) - Fornece os mecanismos necessários a supervisão e ao controle de uma única instância de comunicação. Este tipo de gerenciamento não necessita de um protocolo específico para a troca de informações de gerenciamento, como no caso do Gerenciamento de camadas. Estas informações são trocadas fazendo-se uso do protocolo normal de cada camada.

2.2.2 Gerenciamento Internet

O ambiente TCP/IP define um sistema de gerência mais simplificado do que o modelo padronizado pelo RM-OSI. Este sistema é implementado como um processo de aplicação do Ambiente de Protocolos TCP/IP, utilizando o SNMP (Simple Network Management Protocol) [Ros91, Sta93,Uda96].

Devido a algumas deficiências no modelo SNMP, foi proposto no início de 1993, uma nova versão para este modelo. O modelo SNMP passou a ser referenciado como modelo SNMPv1 (versão 1) e sua nova versão, como modelo SNMPv2 (versão 2). A seguir, detalharemos os modelos SNMPv1 e SNMPv2.

2.2.2.1 SNMP versão 1

No Ambiente de Protocolos TCP/IP o modelo de informação de gerência é mais simples que no Modelo de Referência OSI. A construção das MIBs obedece as especificações contidas nos documentos denominados SMI que foram padronizados em [RFC1155] e [RFC1212] para o SNMPv1. Estas especificações impõem um alto grau de simplicidade na descrição das MIBs para tornar o processo de gerência de fácil implementação.

Embora não exista a definição de classes como encontrado no RM-OSI (Figura 2.4), os objetos das MIBs SNMP são organizados em grupos que representam uma determinada funcionalidade de um elemento de rede, podendo em cada grupo existir tabelas de objetos limitadas no máximo a duas dimensões.

A exemplo dos objetos gerenciados OSI a definição de objetos gerenciáveis para composição das MIBs SNMP utiliza a linguagem ASN.1. No entanto, a macro OBJECT-TYPE especificada em [RFC1155] [RFC1212] utilizada nos objetos SNMP são bem mais simples do que a equivalente OSI. A figura 2.8 exemplifica uma template SNMP.

SysDescr OBJECT-TYPE

SYNTAX DisplayString (SIZE)0..255))

ACCESS read only
STATUS mandatory
DESCRIPTION
"A textual description of the entity"
::{system 1}

Figura 2.8: Template Internet

Em uma fôrma SNMP, podem ser especificadas as seguintes cláusulas:

- sintaxe: sintaxe abstrata que define o tipo do objeto. Justificado na filosofia da simplicidade de implementação, é utilizado apenas o sub-conjunto dos tipos padrão da norma ASN.1 constituído de: *integer, octet string, object identifier, null e sequence.* São definidos na SMI tipos específicos para a aplicação de gerência SNMP que são: *ipaddress*, endereço de rede; *counter*, inteiro que vai até um máximo e volta a zero; *gauge*, inteiro que pode aumentar e diminuir porém não aumenta além de um valor máximo; *timeticks*, que conta o tempo em centésimos de segundos a partir de uma referência inicial especificada; e *opaque*, que representa um tipo arbitrário qualquer;
- acesso: define a maneira como uma instância do objeto pode ser acessada através do SNMP, existindo como opções: somente leitura, somente gravação, leitura/gravação e não acessível, onde o objeto não pode ser lido nem gravado, como é o caso das tabelas de objetos, por exemplo;
- status: determina se um objeto é mandatório ou opcional na implementação de um agente. Existem outras duas situações de status, uma delas denominada *deprecated*, que indica que o objeto será removido da MIB na próxima versão e outra denominada *obsolete* que indica que os agentes já não precisam mais implementar este objeto, permanecendo na MIB apenas para utilização dos objetos já existentes;
- descrição: definição da semântica do objeto, contendo as informações necessárias para a compreensão do objeto descrito, possuindo apenas valor documentacional;
- referência: elemento opcional que contém uma informação para estabelecer uma relação entre o objeto em pauta e um outro objeto qualquer definido em outra MIB, também com valor documentacional.

- índice: nos casos das tabelas bidimensionais de objetos, onde cada linha representa uma instância do recurso gerenciado, o índice identifica a linha e consequentemente a instância do objeto correspondente àquele recurso;
- valor default: elemento opcional que define valores iniciais para uma instância de objeto quando valores específicos não forem informados quando da instanciação; e
- identificação de Objeto: nome pelo qual o objeto será registrado na árvore de registro da ISO e pelo qual será identificado e acessado pelo sistema de gerência.

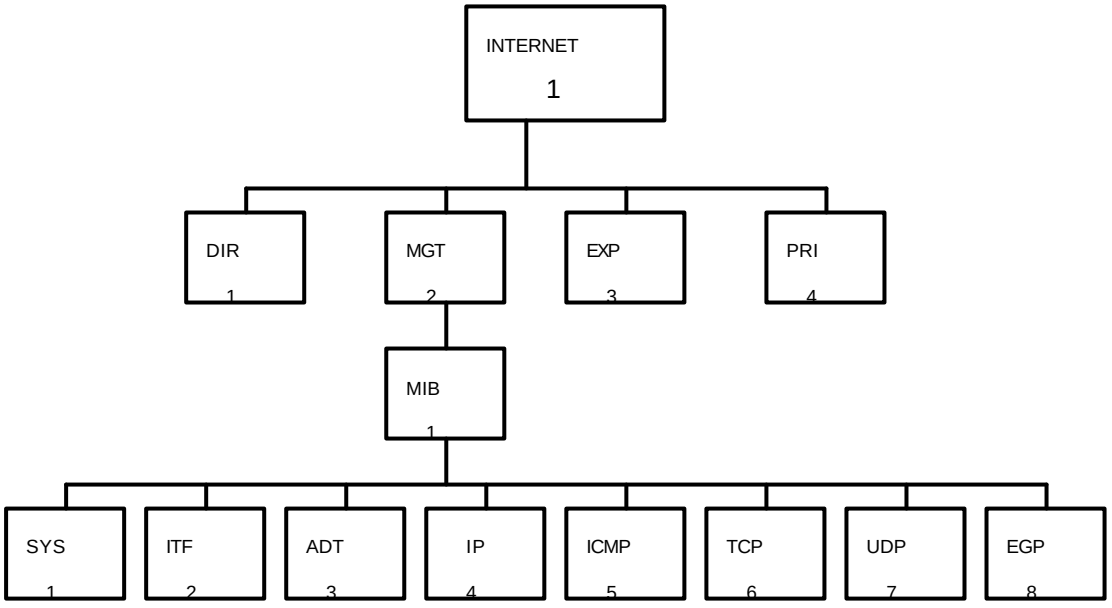
A árvore de instanciação dos objetos gerenciáveis SNMP segue a própria árvore de registro de objetos. Quando o objeto é um simples atributo, a instância é identificada pelo valor zero acrescido ao OID. Quando um objeto pertence a uma entrada de uma tabela, a diferenciação entre instâncias do mesmo objeto ocorre através do valor do *índice* definido na descrição da tabela, o qual é acrescido ao OID do objeto. A árvore de objetos instanciados em uma MIB no ambiente TCP/IP é portanto estática e determinada quando da definição da respectiva MIB. O acesso somente é permitido para objetos que sejam folha na árvore da MIB. Isto significa dizer que nenhum objeto representado por uma tabela pode ser acessado pela referência à tabela ou a uma das suas entradas, mas apenas através de cada um dos seus componentes individualmente. O mesmo ocorre com o acesso aos grupos de objetos.

Para minimizar os problemas causados pelo *overhead* de tráfego excessivo ocasionado pelo acesso individual a cada objeto folha em uma MIB, o SNMP permite que múltiplos objetos possam ser relacionados em uma operação, de forma a serem transportados em uma mesma PDU, porém limitando-se o escopo a uma única linha no caso de uma tabela. Na arquitetura SNMP, é definida em [RFC 1213] uma MIB básica padrão para gerência a qual foi denominada MIB-2. Esta MIB é composta por objetos que representam abstratamente um conjunto de funcionalidades que são encontradas nos diversos elementos de rede, possuindo portanto característica de uso geral e de aplicação direta. São os seguintes os grupos definidos para a MIB-2 (Figura 2.9):

- System: informações gerais sobre o sistema gerenciado no qual o agente está implementado;
- Interfaces: informações genéricas sobre as interfaces físicas do sistema com a rede ou sub-rede na qual está conectado, incluindo informações e estatísticas sobre os eventos que ocorrem em cada interface;
- AdrressTranslation: tabelas de tradução de endereços que permitem mapear de forma uni-direcional os endereços de rede IP para endereços internos da sub-rede, como por exemplo para endereços MAC em redes

locais. Seu status é *deprecated* pois estas informações devem passar para o respectivo grupo de cada protocolo de rede, pela necessidade de mapeamento bi-direcional e na relação de um para vários;

- Ip: informações relevantes para a operação dos protocolos IP nos elementos de rede, incluindo os endereços IP para cada interface física, as tabelas de roteamento além de outras informações para avaliação de desempenho e de falhas;
- Icmp: diversos contadores para disponibilização de informações sobre os vários tipos de mensagens enviadas e recebidas pelo ICMP-*Internet Control Management Protocol*;
- Tcp: informações a respeito das conexões de transporte ativas mantidas pelo protocolo TCP em um elemento de rede, incluindo tabela com os endereços de rede e número das portas locais e remotas para cada conexão, com seus respectivos status, além de uma série de estatísticas e informações a respeito da operação do protocolo;
- Udp: informações a respeito de datagramas enviados e recebidos através do protocolo de transporte UDP, incluindo uma tabela que indica os endereços IP e as portas locais para o elemento de rede;
- Egp: informações sobre a operação de um EGP-*External Gateway Protocol* em um elemento de rede, incluindo além das informações sobre mensagens recebidas e enviadas, uma tabela identificando cada um dos *gateways* vizinhos por ele conhecidos;
- Transmission: criado como raiz para novos grupos de objetos que devem ser definidos para modelar informações sobre os meios de transmissão para cada tipo de interface de um elemento de rede; e
- Snmp: informações quantitativas sobre as mensagens trocadas entre gerente e agente em um sistema de gerência de rede implementado no Ambiente de Protocolos TCP/IP.



LEGENDA:

DIR - DIRECTORY MGT - MANAGEMENT EXP - EXPERIMENTAL PRI - PRIVATE

Figura 2.9: MIB Internet

A arquitetura do modelo SNMPv1 é composta pelos seguintes componentes:

- estação de gerenciamento: corresponde a sistemas que monitoram o estado da rede, coletando, periodicamente, informações de cada nó gerenciado. Para as estações de gerenciamento, cada nó gerenciado é visto como um conjunto de objetos gerenciados. Lendo-se valores de variáveis, um nó gerenciado é monitorado; mudando-se valores de variáveis, um nó gerenciado é controlado;
- nós gerenciados: correspondem a sistemas e dispositivos (pontes, hubs, modens, etc) que implementam agentes; e
- protocolo de gerenciamento: permite a troca de informações de gerenciamento entre estações de gerenciamento e nós gerenciados.

A exemplo do modelo de gerenciamento OSI, num quadro típico, o Agente SNMP aceita os comandos vindos do Gerente, executa as operações indicadas e retorna ao Gerente a resposta devida. Portanto, o comportamento do Agente SNMP pode ser sumarizado nos seguintes passos:

- ele recebe comandos da aplicação no papel de Gerente;
- traduz o comando para um formato interno;

- pesquisa na MIB o Objeto Gerenciado referenciado no comando SNMP;
- executa a operação solicitada pelo Gerente sobre o Objeto;
- traduz a resposta para um formato externo; e
- envia a resposta.

Para implementar a comunicação entre a estação de gerenciamento e os agentes instalados nos nós de gerenciamento, o protocolo SNMPv1 utiliza basicamente dois tipos de comandos: um para recuperar o valor de um atributo, outro para alterar o valor deste atributo. Estes comandos são traduzidos nas seguintes operações (figura 2.10):

- operação Get-request: permite a leitura de atributos dos Objetos Gerenciados armazenados na MIB utilizando como argumentos o nome do nó Agente e o identificador do Objeto Gerenciado;
- operação Get-next-request: recebe como argumento o nome do nó Agente, além do identificador do Objeto Gerenciado ou de um grupo de Objetos. Esta operação realiza uma leitura na lista de objetos percorrendo a árvore de identificação;
- operação Set-request: altera um valor de um atributo sobre um Objeto Gerenciado. Para tanto são passados como argumentos: o nome do nó sobre o qual a alteração será feita, a identificação do Objeto Gerenciado, o novo valor do atributo afetado;
- operação get-response: responde a uma operação ger-request, get-nest-request ou set-request; e
- operação Trap: o Agente informa ao gerente sobre eventos anormais que se produzem sobre o Objetos Gerenciados. Desde que um evento crítico aconteça sobre o nó gerenciado, o Agente envia imediatamente um comando trap ao Gerente sem esperar que este lhe solicite informações. Por exemplo, um Agente pode ser programado para enviar ao Gerente uma mensagem de alarme sempre que houver a perda de uma conexão.

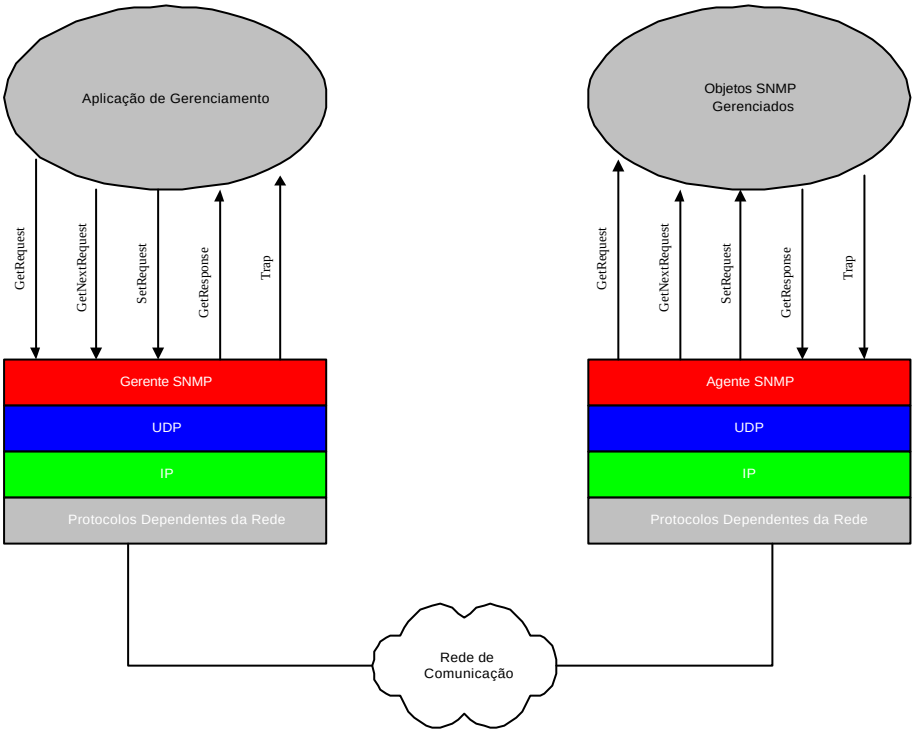


Figura 2.10: Operações SNMP

2.2.2.2 SNMP Versão 2

O protocolo SNMP versão 2 é o resultado de trabalhos isolados na busca de corrigir as deficiências funcionais do SNMPv1 [Uda96]. As principais inovações fornecidas pelo SNMPv2 localizam-se nos seguintes pontos:

- estrutura da informação de gerenciamento (SMI - *Structure of Management Information*);
- operações do protocolo;
- comunicação gerente-a-gerente; e
- segurança.

A SMI do SNMPv2 expande a SMI do SNMPv1 para incluir novos tipos de dados e melhorar a documentação associada a objetos, através de modificações na macro OBJECT-TYPE. A SMI é agora dividida em quatro partes: definição de objetos, tabelas conceituais, definições de notificações e módulos de informação.

As mudanças no nível do protocolo são as inclusões de duas novas PDUs (*Protocol Data Unit*): a **GetBulkRequest**, e a **InformRequest.** O objetivo da GetBulkRequest é retornar um grande número de informações gerenciadas permitindo ao gerente SNMPv2 requisitar uma resposta seja tão grande quanto possível, dada as restrições de tamanho de mensagem. No InformRequest, uma PDU é enviada por uma entidade SNMPv2 agindo como gerente, para uma outra entidade agindo também como gerente.

O modelo SNMPv2 admite a existência de um gerenciamento distribuído, com estações de gerenciamento configuradas para exercer o papel de gerentes e agentes, e com a possibilidade de comunicação entre gerentes para a troca de informações de gerenciamento. Cada gerente pode administrar diretamente um conjunto de agentes, e quando o número de agentes cresce ao ponto de causar problemas relativos a seu gerenciamento, uma estação de gerenciamento pode delegar a tarefa de gerenciamento a gerenciadores intermediários. O gerenciador intermediário exerce o papel de gerente para monitorar e controlar os agentes sob sua responsabilidade e exerce o papel de agente para enviar e receber informações de controle de uma estação de gerenciamento hierarquicamente superior.

Três novas MIBs foram definidas especificamente para a estruturação do SNMPv2, as quais estão alocadas na árvore de objetos sob o identificador *snmpModules* :

- SnmpMIB: a MIB do SNMPv2 foi especificada em [RFC1907] e redefine os grupos *system* e *snmp*, adaptando-os para as facilidades da versão 2, assim como redefine também os *traps* padrão. Implementa um objeto para coordenação dos comandos *set* requeridos ao mesmo tempo por vários gerentes sobre um mesmo objeto;
- snmpM2M: denominada *Manager-to-Manager*, esta MIB [RFC1451] tem como objetivo modelar a troca de informações entre dois gerentes SNMPv2 em um mesmo sistema, quando da implementação de gerência distribuída. É composta por dois grupos: alarme, especificando os objetos a serem monitorados e seus valores limite, e eventos, descrevendo os eventos associados aos alarmes; e

- partyMIB: definida em [RFC1447], a MIB denominada *Party* modela os objetos concernentes aos aspectos de administração e segurança, sendo composta por três grupos: identificação e autenticação dos parceiros no sistema, definição dos contextos do sistema onde os parceiros estão inseridos e definição de acessos e privilégios para os parceiros sobre os objetos gerenciados.

O SNMPv2 incorpora as especificações contidas no conjunto de documentos conhecido como S-SNMP (*Secure SNMP*), as quais foram submetidas em Julho/1992 como propostas de padrão às especificações já existentes no mundo Internet. Estas especificações fornecem os seguintes serviços:

- autenticação: utilizado para assegurar que uma mensagem recebida foi realmente transmitida pelo emissor que aparece como fonte no cabeçalho de mensagem;
- confidencialidade: baseia-se na criptografia, e é usado nas mensagens trocadas entre gerentes e entre gerentes e agentes. Confidencialidade é a proteção dos dados transmitidos contra escutas. A confidencialidade exige que o conteúdo de qualquer mensagem seja oculto, de modo que somente o receptor pretendido possa recuperá-la; e
- controle de acesso: determina o tipo de acesso que uma determinada fonte pode ter sobre as informações de gerenciamento. O controle de acesso assegura que somente usuários autorizados tenham acesso a um banco de informações de gerenciamento em particular.

2.2.2.3 RMON

Os monitores de rede remotos (RMON – *Remote Network Monitors*) (Figura 2.11), também conhecidos como analisadores de rede, são extremamente úteis à atividade de gerenciamento de redes [Uda96]. Eles constituem-se na mais importante adição ao conjunto de padrões SNMP.

Através da MIB-II, o administrador da rede pode obter apenas informações localizadas nos elementos da rede. Uma estação de gerenciamento SNMP pode por exemplo, descobrir informações acerca do tráfego que entra ou sai em cada elemento da rede, mas não pode ser informado do tráfego da rede como um todo. Os monitores remotos portanto, são empregados para estudar o tráfego em uma rede, consistindo basicamente na atividade de coleta e análise de pacotes. Tipicamente, estes monitores operam em uma rede no modo promíscuo, visualizando todos os pacotes que passam através dela.

As especificações do RMON (RFC 1757) constituem-se basicamente na definição de uma MIB. Neste documento, são definidas as funções e *interfaces* para comunicação entre uma console de gerenciamento SNMP e monitores remotos.

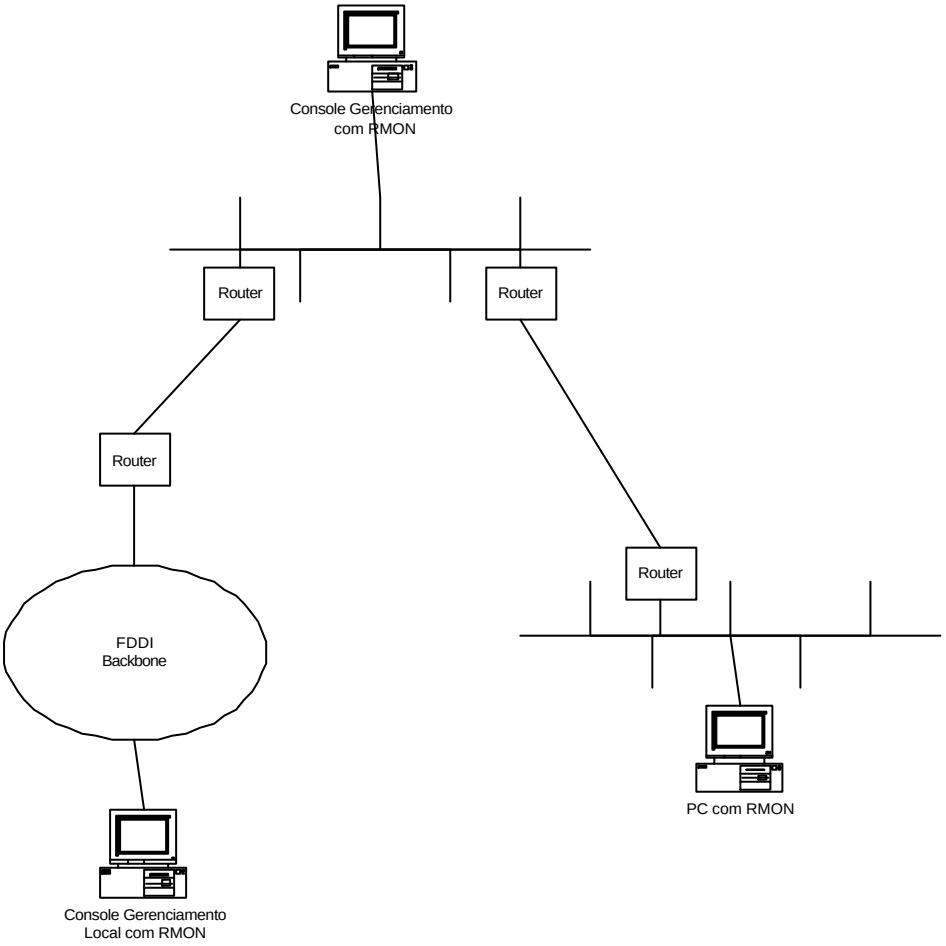


Figura 2.11: RMON

Segundo ainda a RFC1757, os cenários onde RMON pode ser utilizado são os seguintes:

- operações *offline*: são as operações nas quais uma estação de gerenciamento não necessita estar em contato direto com seus dispositivos de monitoração remotos. Essa função na RMON MIB permite que os agentes sejam configurados para realizar diagnósticos e coletar estatísticas continuamente, mesmo que a comunicação entre a estação de gerenciamento não seja possível ou não seja eficiente;

- monitoração pró-ativa: os recursos disponíveis nos monitores são potencialmente úteis para continuamente executar diagnósticos e manter *logs* do desempenho da rede. Essas informações são importantes para desenvolver a função *baseline*. A função *baseline* refere-se ao fato de manter um histórico da operação normal de uma rede por um tempo estendido, com o objetivo dessas informações posteriormente serem analisadas para identificar problemas potenciais numa rede;
- detecção e registro de problemas: o monitor remoto pode fazer o reconhecimento de determinadas condições, realizando constantes averiguações;
- valorização dos dados coletados: o monitor pode executar análises específicas nos dados coletados em suas subredes; e
- múltiplos gerentes: na configuração de uma rede pode haver mais de uma estação de gerenciamento como forma de oferecer maior nível de disponibilidade ou para executarem diferentes funções.

O conjunto de especificações do RMON é direcionado para a definição da RMON MIB. Esta MIB foi incorporada à MIB-II como uma subárvore. A RMON MIB é formada por nove grupos, descritos a seguir :

- Grupo Statistics: contém estatísticas medidas pelo monitor para cada interface monitorada no dispositivo. O grupo consiste de uma única tabela, com um registro para cada interface monitorada;
- Grupo History: registra amostras estatísticas periodicamente e as armazena para uma posterior recuperação. Este grupo consiste de duas tabelas: historyControlTable que especifica a interface, e a etherHistoryTable que grava os dados.
- Grupo Alarm: os objetos deste grupo periodicamente obtêm amostras estatísticas e as comparam com os limiares previamente configurados. Este grupo consiste de uma única tabela e cada registro da tabela especifica uma variável a ser monitorada, um intervalo e um parâmetro limite.
- Grupo Host: contém estatísticas associadas a cada host da rede.Os endereços fonte e destino desses hosts são mantidos numa lista e atualizados a partir dos pacotes bons que foram promiscuamente recebidos pela interface. Este grupo consiste de três tabelas: uma tabela de controle e duas tabelas de dados.
- Grupo HostTopN: é usado para preparar relatórios que especificam os principais hosts de uma lista ordenada por algum parâmetro. As estatísticas que são geradas por este grupo são derivadas dos dados do grupo "Host". Este grupo consiste de uma tabela de controle e uma tabela de dados.

- Grupo Matrix: armazena a estatística do tráfego e número de erros entre pares de hosts. A informação é gravada em forma de uma matriz. Este grupo consiste de três tabelas: uma tabela de controle e duas tabelas de dados.
- Grupo Filter: permite que os pacotes sejam capturados com uma expressão de filtro arbitrária. Existem dois tipos de filtros: um filtro de dados e um filtro de status. Estes filtros podem ser combinados usando os operadores lógicos AND e OR formando complexos testes para serem aplicados aso pacotes.
- Grupo Capture: permite que os pacotes sejam capturados sobre a correspondência de um filtro e que o sistema de gerenciamento crie múltiplos *buffers* de captura, controlando se o buffer de monitoração (*trace buffer*) continua ou interrompe a captura de pacotes quando estiver cheio. O grupo Capture consiste de duas tabelas: bufferContrlTable que especifica os detalhes da função *buffering*, e captureBufferTable que coloca os dados no *buffer*.
- Grupo Event: controla a geração e notificação de eventos. O grupo Event consiste de uma tabela de controle e uma tabela de dados. A tabela de controle eventTable contém definições de evento. Cada linha da tabela contém os parâmetros que descrevem um evento para ser gerado quando certas condições acontecerem.

2.2.3 Comparações entre Gerência OSI X Gerência Internet

Apesar do Gerenciamento OSI e do Gerenciamento Internet fazerem uso de conceitos similares, tais como Gerente x Agente, Objeto gerenciado x MIB, eles apresentam características que os diferenciam.

O Gerenciamento Internet é mais simples do que o gerenciamento OSI uma vez que os conceitos no gerenciamento OSI são complexos e de difícil implementação. Como consequência de sua simplicidade, o Gerenciamento Internet tem dominado o mercado de SGRs (Sistemas de Gerenciamento de Redes). Apresentaremos a seguir, algumas similaridades e diferenças entre o Gerenciamento OSI e o Gerenciamento Internet.

2.2.3.1 Similaridades

Tanto o modelo de Gerência OSI quanto o modelo de gerência Internet têm, naturalmente, o mesmo objetivo: enviar comandos e receber resultados e notificações de tal modo a coordenar (monitorar e controlar) os recursos lógicos e físicos de uma rede. As principais similaridades entre os dois são:

- paradigma Gerente x Agente;
- ambos usam o conceito de objeto gerenciado x MIB; e
- usam protocolos de arquiteturas abertas (não proprietárias).

2.2.3.2 Diferenças

As diferenças entre os modelos de Gerenciamento OSI e Internet podem ser resumidas pelas características dos protocolos CMIP e SNMP:

- acesso a dados: o SNMP é orientado mais a recuperação individual de itens de uma informação, enquanto o CMIP é mais orientado a recuperação de informações agregadas;
- *Polling* x *Reporting*: o SNMP trabalha por *polling* (o gerente regularmente pergunta ao agente sobre seu *status*), enquanto o CMIP usa *Reporting* (o Agente informa ao gerente quando o *status* do Objeto Gerenciado mudou). Esta filosofia de gerenciamento do CMIP é vantajosa sobre o SNMP quando se tem um número considerável de Agentes a serem consultados;
- tamanho e desempenho: o agente SNMP é menor e mais rápido. O agente CMIP requer maior capacidade de processamento, mais memória. Esta característica é associada a filosofia: *polling* requer menos inteligência dos dispositivos sendo gerenciados;
- funcionalidade: o CMIP tem mais características e capacidades específicas;
- nível de transporte: o SNMP requer somente datagramas não confiáveis, o que implica que ele pode ser implementado em várias redes. Em comparação, o CMIP exige um nível de transporte orientado a conexão.
- padronização: o CMIP, a exemplo dos demais protocolos OSI, é um padrão internacional, sujeito, portanto, a testes de conformidade. O SNMP, por sua vez, somente pode dispor de testes de interoperabilidade.

2.3 Sistemas de Gerenciamento de Redes

Um sistema de gerenciamento de redes (SGR) é uma coleção de ferramentas que possibilita monitorar e controlar redes [Sta93]. Tais ferramentas se propõem a:

- oferecer uma interface única com um conjunto de comandos para execução das tarefas de gerenciamento; e
- integrar ao máximo equipamentos e programas desempenhando funções que auxiliam a gerência de redes.

As atuais arquiteturas de sistemas de gerenciamento de redes apresentam algumas diferenças em função das funcionalidades que são oferecidas. A Figura 2.12 apresenta um modelo genérico de um sistema de gerenciamento de redes.

Em arquiteturas de SGRs, pelo menos um dos computadores na rede é designado para executar a aplicação de gerenciamento de rede. Estas aplicações caracterizam-se por incluir uma interface única para os usuários e por serem constituídos de módulos que desempenham funções específicas. Os módulos de acesso à MIB da figura 2.12 representam as funcionalidades dos agentes.

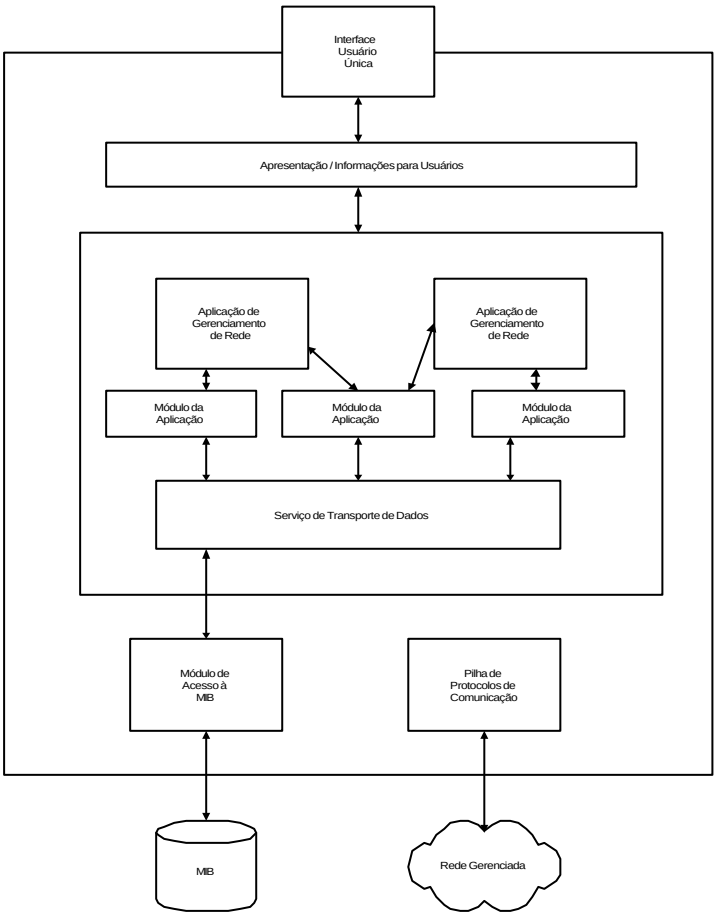


Figura 2.12: Modelo de um SGR

Sistemas de gerenciamento de redes têm sido alvo do desenvolvimento de inúmeras ferramentas comerciais. A seguir, são apresentadas algumas destas ferramentas disponíveis, descrevendo suas principais funcionalidades.

2.3.1 NetView (IBM)

O NetView foi introduzido pela IBM em 1986, como sendo um produto para o diagnóstico e controle de redes de comunicação. Ele foi concebido com o objetivo de substituir uma variedade de produtos similares já existentes, provenientes do mesmo construtor. Esses produtos estavam integrados em boa parte no ambiente IBM, mas limitados a tarefas particulares.

O mais conhecido desses produtos é o NCCF (*Network Communication Control Facility*), tido como um dos primeiros sistemas de controle de rede. Ele funciona como uma aplicação VTAM (*Virtual Telecommunication Access Method*), que fornece uma visão do estado da rede através de um monitor de controle e uma base de dados. Outros produtos de controle são o NPDA (*Network Problem Determination Application*) e o NLDM (*Network Logical Data Manager*), que operam na detecção de problemas e gerenciamento de dados lógicos, respectivamente.

O principal objetivo do NetView foi então reunir as funcionalidades desses produtos em um só, melhorando o desempenho e introduzindo novos conceitos no gerenciamento de redes, mostrados a seguir:

- Ponto Focal - corresponde ao recurso que fornece uma administração centralizada da rede. Ele inclui funções de administração tais como o gerenciamento de mudanças na configuração da rede e operações de controle;
- Ponto de Entrada - corresponde a um recurso endereçável que permite a comunicação entre a administração dos elementos de uma rede SNA da IBM e o ponto focal da rede. Os dados enviados ao ponto focal são dados de gerenciamento;
- Ponto de Serviço - corresponde ao recurso que permite o encaminhamento de informações de gerenciamento provenientes de recursos não-SNA para o ponto focal. O ponto de serviço é endereçável e pode aceitar solicitações de gerenciamento do ponto focal, traduzi-las, enviá-las para os elementos não-SNA da rede, esperar pelas respostas e enviá-las de volta ao ponto focal.

O primeiro componente, NetView, também chamado um produto Ponto Focal, foi lançado em Maio/86. Ele reúne os produtos já conhecidos e inclui novas funcionalidades de gerenciamento. As ferramentas disponíveis são :

- uma aplicação de controle que exerce a função de gerenciamento;
- uma aplicação de supervisão de material que coleta informações relativas a panes ou eventos significativos capazes de afetar o desempenho da rede;
- uma aplicação de supervisão de sessão que coleta informações sobre as sessões lógicas da rede;
- uma aplicação de supervisão que exibe no monitor o estado dos recursos da rede e suas relações;
- uma ferramenta de ajuda ao usuário no manuseio de instruções, códigos e mensagens de gerenciamento; e
- uma ferramenta de configuração do programa NetView, que permite ao administrador customizá-lo de acordo com suas necessidades.

O segundo componente, NetView/PC, também chamado Ponto de Serviço, foi introduzido em Setembro/1986. Ele foi concebido para complementar o NetView, fazendo com que o gerenciamento pudesse atingir elementos externos à arquitetura SNA (*System Network Architecture*) da IBM. O objetivo final era atingir uma administração total de uma instalação qualquer de rede. Este componente foi projetado para rodar numa estação de trabalho e se comunicar diretamente com outra conectada, a fim de realizar as funções de gerenciamento. Está incorporada nesse produto uma linguagem de comandos, chamada API/CS (*Application Programming Interface/Communication Services*), que permite ao administrador desenvolver interfaces para o gerenciamento de elementos externos.

A estrutura do Netview permite um gerenciamento hierárquico como mostrado na figura 2.13. Nesta estrutura, existem operadores locais que gerenciam seus próprios ambientes, mas recebendo comandos e reportando alarmes para um operador geral.

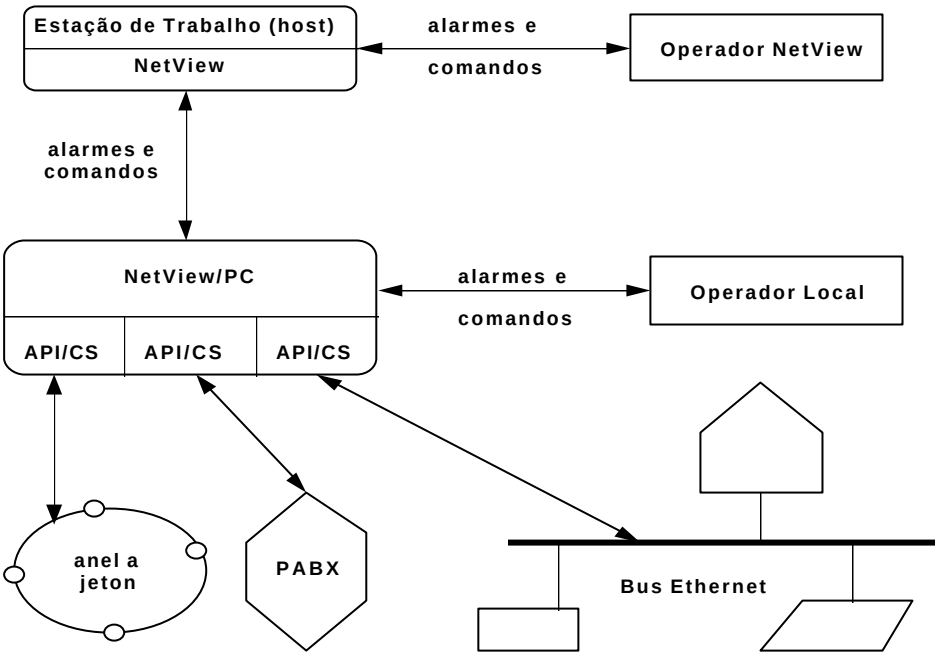


Figura 2.13: Arquitetura Netview

Numa versão mais atualizada, a IBM lançou o Netview/6000, como uma ferramenta de gerenciamento e administração em redes TCP/IP. Esta ferramenta é formada por uma família de aplicações que disponibilizam as seguintes funcionalidades: gerenciamento de redes heterogêneas, descoberta automática dos nós da rede, interface gráfica única, integração com bancos de dados relacionais, gerenciamento distribuído, gerenciamento SNMP, ferramentas para gerenciamento da MIB, distribuição de software e inventário.

2.3.2 HP OpenView

O sistema HP Open View de propriedade da empresa HP (Hewlett Packard), possui uma completa família de produtos, oferecendo um Sistema Integrado de Rede e Gerenciamento de Sistema (INSM-*Integrated Network System Management*) com soluções para diversos fabricantes. Na realidade, esta ferramenta trata-se de um pacote de módulos capaz de gerenciar redes TCP/IP, Netware e SNA (*System Network Architecture*) da IBM. Este trio é incorporado numa única ferramenta de gerenciamento. Suas principais características são: automático *discovery* e mapeamento de toda a rede, incorporação de todos os devices SNMP, controle completo em uma única visão,

configuração de ações automáticas, definição de prioridades em categorias de eventos, interface gráfica, emissão de relatórios em tempo real e baseado nos padrões IP,UDP e SNMP.

A arquitetura do sistema HP OpenView é constituída dos seguintes componentes:

- Development System;
- HP-UX Controller;
- Unix Target System;
- PC Controller;
- PC File Server; e
- PC Target.

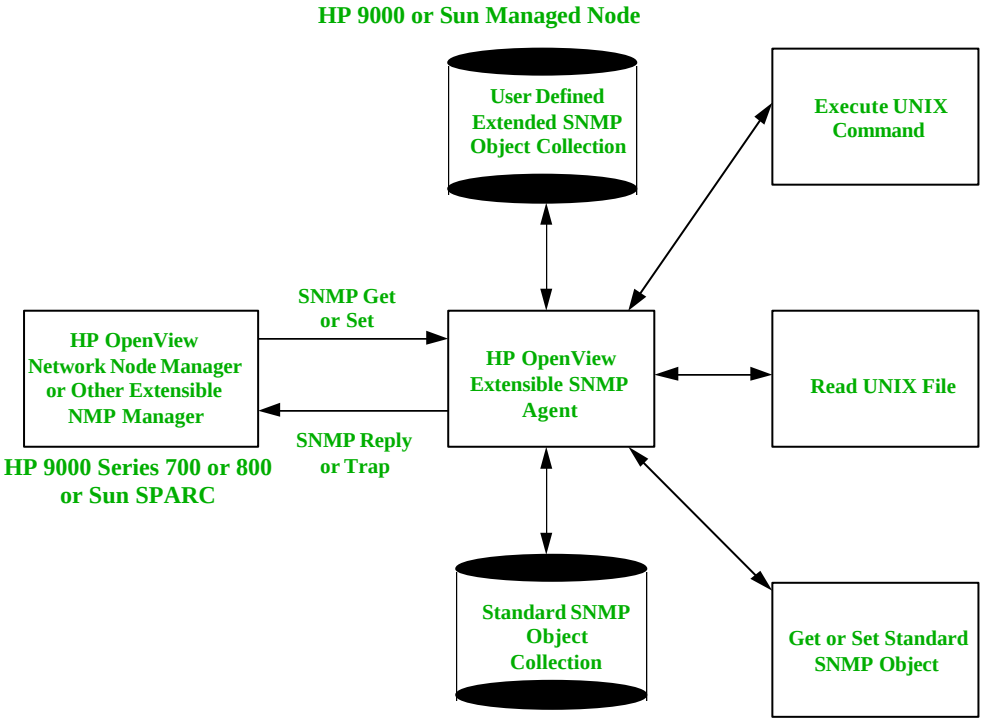


Figura 2.14 : Funcionalidades HP OpenView

As funcionalidades do HP OpenView (Figura 2.14) estendem-se além do gerenciamento dos objetos padronizados SNMP, permitindo o desenvolvimento de novos agentes bem como uma interação direta com o ambiente UNIX.

2.3.3 SunNet Manager

O SunNet Manager é um sistema de gerenciamento distribuído que oferece um ambiente amigável para sistemas independentes do protocolo e soluções de gerenciamento de redes. Essa ferramenta [Sun91] apresenta uma interface de usuário gráfica aberta e intuitiva. Oferece, também, um conjunto completo de ferramentas gráficas para descoberta automática, isolamento de falhas, diagnóstico e análise de desempenho. Essa ferramenta possui os seguintes componentes: gerenciamento de aplicações, que gerencia dados coletados sobre os nós da rede gerenciada; agentes que coletam dados sobre os nós a pedido do módulo de gerenciamento de aplicações; agentes de *proxy* que são similares aos agentes comuns com duas exceções pois usam técnicas de RPC (*Remote Procedure Call*) para gerenciar múltiplos protocolos e são capazes de obter informações sobre múltiplos nós, tornando a centralização possível. Por fim, esta ferramenta segue o modelo gerente-agente descrito nas especificações OSI e no protocolo de comunicação TCP/IP.

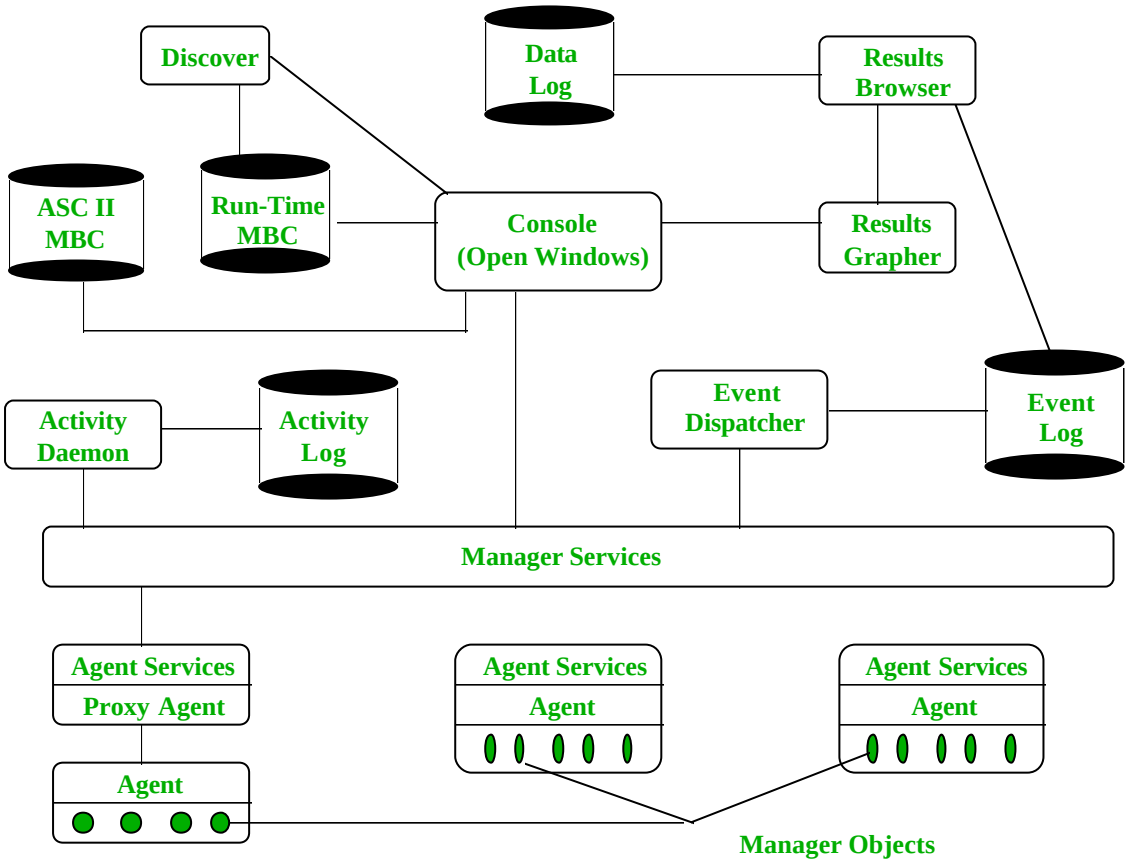


Figura 2.15 : Arquitetura SunNet Manager

A Figura 2.15 mostra o modelo de funcionamento do SunNet Manager, sendo constituídos dos seguintes elementos:

- Módulo Console: Inicia as tarefas de gerenciamento e recebe de volta as informações. Possui as funções de inicializar solicitações por Data Reporting e por Event Reporting. Suas principais ferramentas são: Discover (descoberta automática da topologia da reda), Results Browser (examinar os arquivos de Log) e Results Grapher (visualização de relatórios e arquivos de Log);
- Data Reporting: Permite dirigir agentes para enviar reports de gerenciamento de dados brutos de forma periódica;
- Event Reporting: Permite dirigir agentes para reports apenas quando condições especificadas são encontradas; e

- Management Database (MDB): Realiza a descrição do objeto gerenciado (Agent Schema) , sendo estes arquivos descritos no formato ASC II.

Capítulo 3

Gerência de Falhas

Neste capítulo, é apresentado o problema tratado nesta dissertação, o gerenciamento de falhas em redes de computadores. Inicialmente, são apresentados os conceitos gerais sobre falhas sendo em seguida analisados diagnósticos de falhas baseados em Modelo e em Heurísticas. A figura 3.1 mostra o modelo de concepção do SAGRES – Contexto (Figura 2.1) adicionado do refinamento do elemento Infraestrutura Conceitual.

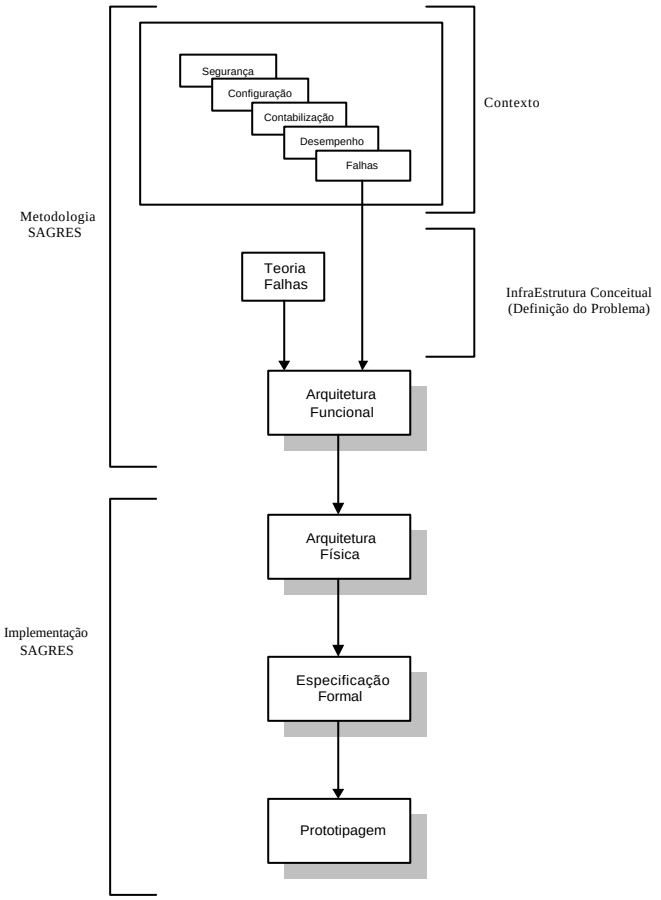


Figura 3.1: Modelo de Concepção do SAGRES – Teoria de Falhas

3.1 Generalidades sobre Falhas

3.1.1 Problemas no Monitoramento de Falhas

O monitoramento de falhas tem como objetivo determinar a ocorrência de falhas da forma mais rápida possível e identificar as causas dessa falha. Identificada a causa, ações devem ser tomadas para solucionar o problema[Sta93].

Os seguintes problemas são associados à ocorrência de falhas:

- Falhas não observadas: certas ocorrências de falhas são difíceis de serem observadas através de observação local. Por exemplo, a existência de deadlock entre processos cooperantes distribuídos pode não ser observado localmente. Outras falhas podem não ser observadas devido a impossibilidade do equipamento registrar a ocorrência da falha;
- Falhas observadas parcialmente: uma falha em um elemento de rede pode ser observada, porém a observação pode ser insuficiente para identificar com precisão o problema;
- Observações inexatas: sempre que observações detalhadas de falhas são possíveis, podem existir incertezas ou inconsistências associadas às observações.

Após as falhas serem observadas, é necessário que cada falha seja isolada. Para que estas falhas sejam isoladas, surgem alguns problemas, dentre eles:

- Múltiplas fontes: quando várias tecnologias estão envolvidas, os locais e tipos de falhas aumentam significativamente. Isso torna mais difícil a localização da fonte da falha. Como mostrado na Figura 3.2, dados transmitidos entre um cliente e um servidor passam por uma rede, um roteador, um multiplexador, e um sistema de transmissão. Se a conexão é perdida, ou se a taxa de erros é muito alta, o problema pode ter sido gerado em algum desses subsistemas;

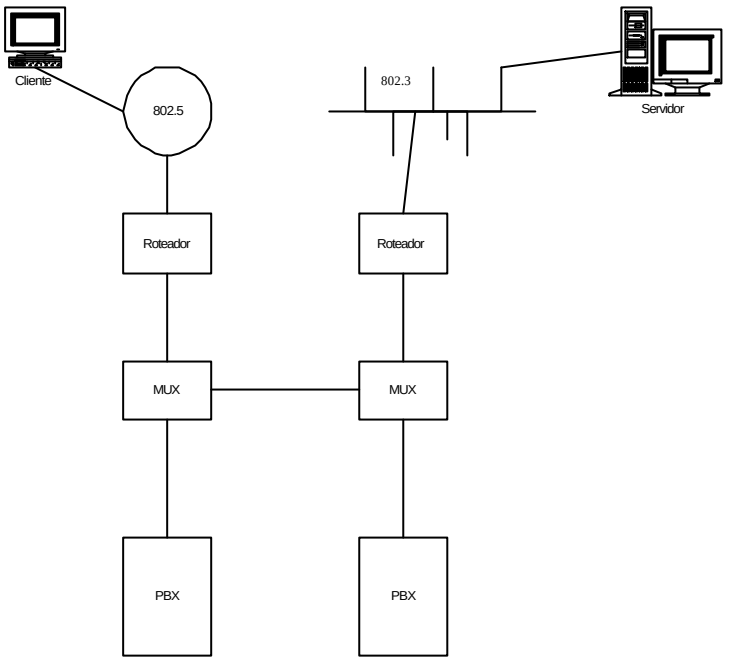


Figura 3.2: Fontes de Falhas

- Várias Observações relacionadas: uma falha na linha de comunicação da Figura 3.2 pode afetar toda a comunicação entre as estações conectadas em uma rede padrão 802.5 e as estações conectadas à uma rede padrão 802.3, como também a comunicação de voz entre os PBXs. Entretanto, uma falha em uma camada da arquitetura OSI pode causar degradação ou falhas em todas as camadas de nível mais alto, como mostrado na Figura 3.3. Por exemplo, uma falha na linha de comunicação da Figura 3.2 será detectado no roteador como uma falha no link de comunicação e nas estações como falha na aplicação. Isto acontece porque uma única falha pode gerar muitas outras falhas secundárias;

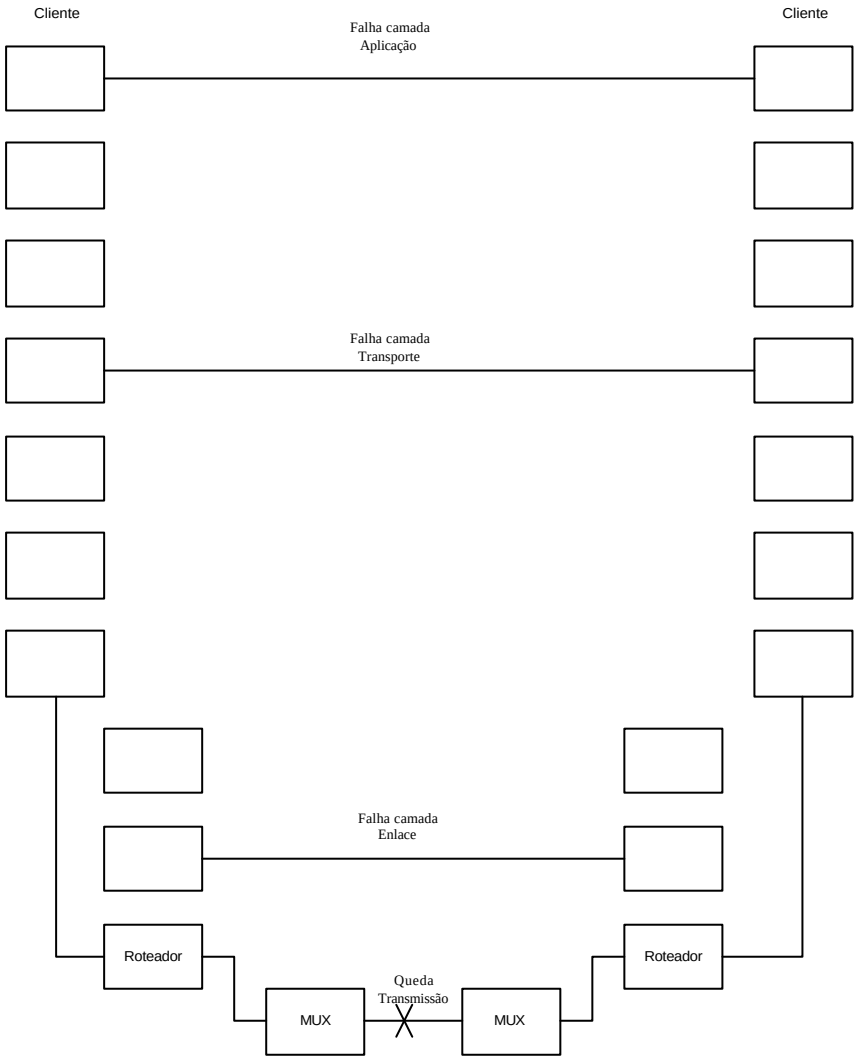


Figura 3.3: Falhas x Camadas da Arquitetura de rede

- Interferência de procedimentos de reparação local em diagnose: procedimentos de correção local podem destruir importantes evidências referente à natureza da falha, impossibilitando a diagnose;

- Ausência de ferramentas de teste automatizadas: testes para isolar falhas são difíceis e custam caro para o administrador.

3.1.2 Funções da Gerência de Falhas

A primeira exigência em um sistema de gerência de falhas é que ele detecte e informe a ocorrência de falhas. No mínimo, um agente de monitoramento de falhas mantém um arquivo de log com os eventos e erros mais significativos. Tipicamente, um agente de monitoramento de falhas tem a capacidade para independentemente informar a ocorrência de erros para um ou mais gerentes. Para evitar um congestionamento na rede, alguns critérios para informar as falhas devem ser estabelecidos.

Além de informar sobre alguns tipos de falhas, um bom sistema de gerência de falhas deve ser capaz de antecipar-se à falha. Geralmente, isto é feito estabelecendo limites (threshold) e uma vez atingidos estes limites o sistema emite um alarme. Por exemplo, se a quantidade de pacotes com erro excede um certo valor, isto pode indicar algum problema de comunicação.

Um sistema de gerência de falhas deve também permitir a diagnose da falha e os procedimentos para recuperação. Exemplos de testes que um sistema de gerência de falhas deve realizar são os seguintes:

- Teste de conectividade;
- Teste de integridade dos dados;
- Teste de integridade dos protocolos;
- Teste de congestionamento da conexão;
- Teste de tempo de resposta;
- Teste de diagnóstico.

Talvez, mais importante que em outras áreas de gerenciamento, uma boa interface para o usuário é necessária para o monitoramento de falhas. Em situações complexas, falhas podem ser diagnosticadas, isoladas, e mais recentemente corrigidas com a contribuição de um software monitor.

3.2 Diagnose de Falhas

A maneira de detecção de falhas consiste, em geral, na comparação entre um comportamento esperado (normal) e um comportamento apresentado[Dav84]. Discrepâncias entre estes comportamentos indicam que o sistema está

com problemas. Confirmada a discrepância, deve-se determinar as causas do problema ou diagnose. Assim, o objetivo da diagnose é determinar os elementos responsáveis pelo mal funcionamento do sistema.

As discrepâncias entre o comportamento esperado e o comportamento observado são utilizados para guiar a pesquisa pela diagnose. Existem dois tipos de diagnose[Dav84]:

- diagnose baseada em modelo; e
- diagnose heurística, descritos abaixo.

3.2.1 Diagnose baseada em Modelo

O princípio básico da abordagem baseada em modelo para diagnose pode ser entendido como uma interação entre o comportamento esperado para o sistema que está sendo diagnosticado e a observação do sistema no estado atual (Figura 3.4).

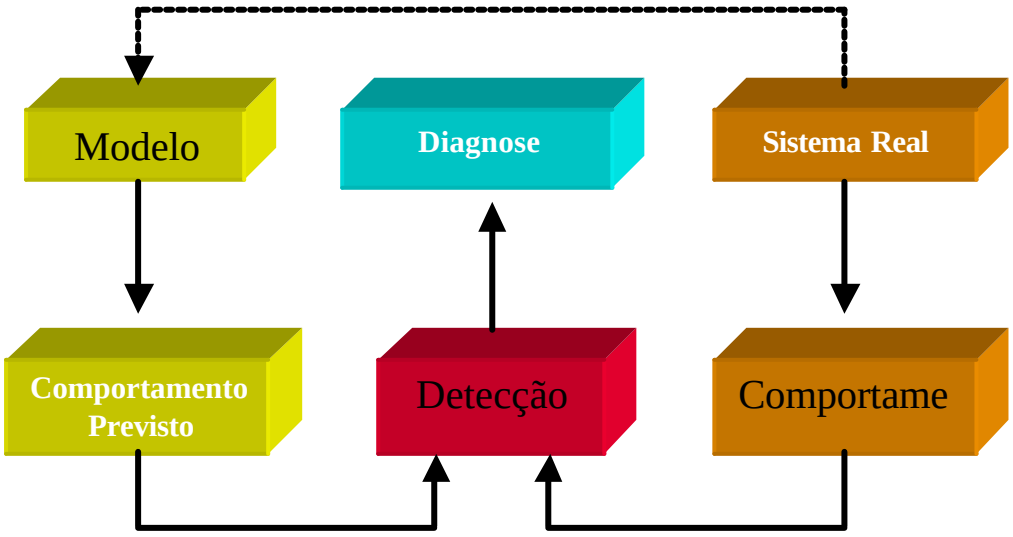


Figura 3.4: Diagnose por Modelo

O modelo permite definir o comportamento esperado para o sistema. As observações sobre o sistema informam como o sistema atualmente está se comportando que é também chamado de comportamento observado. Discrepâncias entre o comportamento observado e o comportamento esperado indicam que o sistema não está se comportando como esperado.

Uma hipótese fundamental na diagnose baseada em modelo é que o modelo é completamente correto. Entretanto, o modelo trabalha com uma quantidade de hipóteses simplificadas e aproximações que possivelmente não representam os exatos relacionamentos do mundo real. Em geral, se a aproximação é boa o suficiente, a abordagem baseada em modelo tem se mostrado como uma boa técnica de diagnóstico.

A pesquisa em diagnose baseada em modelo tem trabalhado em dois segmentos: o modelo de comportamento correto e o modelo de comportamento falho. O modelo comportamental correto define como o sistema normalmente trabalha, enquanto que o modelo de comportamento falho especifica como ele trabalha se determinadas falhas ocorrem.

Resumidamente, a diagnose baseada em modelo segue os seguintes passos:

- Descrição do comportamento esperado de um sistema de interesse (modelo);
- Observação de um comportamento real de um sistema que está em conflito com o esperado (Detecção de discrepância ou falha); e
- Determinação dos componentes do sistema que em hipótese de falha explicam tal discrepância (Diagnose).

A diagnose baseada em modelo utiliza um formalismo apropriado para determinar o comportamento esperado do sistema de interesse, por exemplo lógica de primeira ordem [Rei87].

As formas de se fazer diagnose baseada em modelo, depende do conhecimento que se tenha do comportamento do sistema, e se classificam em:

- Diagnose baseada em consistência: é baseada em um modelo que descreve o comportamento esperado do sistema; e
- Diagnose baseada em Abdução: é baseada em um modelo que descreve o comportamento falho do sistema.

Para exemplificar a diagnose baseada em modelo, utilizamos a formulação do seguinte problema:

Considere o sistema um par (DS,Componentes), onde:

- DS: é a descrição do sistema, sendo constituído de um conjunto de sentenças de primeira ordem;
- Componentes: são os componentes do sistema, sendo representado por um conjunto finito de constantes.

A seguir é apresentado a descrição do sistema:

$E(x) \wedge \neg AB(x) \supset out(x) = e(ent1(x),ent2(x))$

$X(x) \wedge \neg AB(x) \supset out(x) = oux(ent1(x),ent2(x))$

$O(x) \wedge \neg AB(x) \supset out(x) = ou(ent1(x),ent2(x))$

Os componentes do sistema seguem a seguinte representação:

E(E1), E(E2), X(X1), X(X2), O(O1)

sai(X1) = ent2(E2); sai(X1) = ent1(X2); sai(E2) = ent1(O1); ent1(E2) = ent2(X2); ent1(X1) = ent1(E1); ent2(X1) = ent2(E1); sai(E1) = ent2(O1).

O passo seguinte consiste em realizar algumas observações sobre o sistema. Uma observação de um sistema é um conjunto de sentenças de primeira ordem. Podemos denotar (DS,Componentes,OBS) para um sistema (DS,Componentes) com a observação OBS.

A diagnose é determinada da seguinte maneira:

Se um sistema (DS, $\{c_1,...,c_n\}$) está falhando, significa que há uma observação OBS em que o comportamento do sistema não foi o esperado.

$SD \cup \{\neg AB(c_1), \ldots, \neg AB(c_n)\}$ representa o comportamento do sistema na hipótese de que todos os seus componentes estejam funcionando corretamente.

A observação do sistema mostrou uma discrepância entre o comportamento observado e o esperado para ele, portanto:

$SD \cup \{\neg AB(c_1), \ldots, \neg AB(c_n)\} \cup OBS$ é Inconsistente.

3.2.2 Diagnose Heurística

Para alguns problemas, a solução através de procedimentos exatos simplesmente não existe ou são computacionalmente inviáveis. Uma alternativa para esse problema consiste na utilização de procedimentos que oferencem soluções consideradas boas mas em alguns casos pode não ser a melhor solução. Este método é chamado de heurística.

Uma classe mais geral ao método de heurística é chamado de metaheurística. Algumas metaheurísticas têm sido propostas, especialmente projetadas para evitar que o procedimento fique preso em armadilhas de ótimos locais. Podem ser considerados como metaheurísticas os seguintes procedimentos:

- Algoritmos genéticos[Bra85];
- Simulated Annealing[Met53]; e
- Tabu search[Glo86] [Glo87].

A diagnose heurística usa o conhecimento de especialistas e o conhecimento obtido através da observação de uma significativa quantidade de dados. Tipicamente, este conhecimento pode ser expressado através de regras associando sintomas com as falhas observadas.

Diagnose heurística tem sido utilizado principalmente na medicina. O principal exemplo de diagnose heurística é o sistema MYCIN que realiza diagnose e terapia para doenças causadas por infecção no sangue[Buc84].

Vários problemas no entanto, têm sido identificados quando se faz uso da abordagem heurística:

- A aquisição do conhecimento de especialistas humanos é uma tarefa difícil e consume muito tempo;
- O conhecimento é muito dependente de um ambiente específico e não é reutilizável;
- A manutenção de uma grande base de regras é difícil; e
- Apenas o conhecimento sobre o comportamento do sistema até a data atual pode ser utilizado, e portanto alguns tipos de falhas raras podem são ser diagnosticados.

3.3 Área Funcional Gerência de Falhas

O gerenciamento de falhas é composto dos seguintes elementos[Uda96]:

- Detecção de Problemas: inclui relatório de problemas ocorridos e um log de falhas e erros. Correlação de alarmes e eventos também são importantes funções, servindo para se antecipar a futuros problemas;
- Diagnóstico do Problema: tem como função diagnosticar e testar procedimentos para detecção de problemas;
- Correção do problema: envolve métodos manuais e automáticos para resolver problemas;
- Localização do Problema: usado para localizar a origem do problema.

A classificação de componentes acima não é uma definição OSI, mas é útil para o entendimento dos conceitos de gerenciamento de falhas. Uma falha envolve uma situação anormal e é resultante de um problema operacional. Erros, entretanto, podem ocorrer normalmente e eles podem não resultar em uma operação anormal de um recurso.

Em grande redes de computadores onde existem centenas de estações de trabalho e diferentes componentes, torna-se muito difícil isolar um problema. Quando um problema ocorre, é importante conhecer a causa exata para corrigí-lo. Então, quando um problema ocorre, deve-se recorrer a um arquivo de log de problemas.

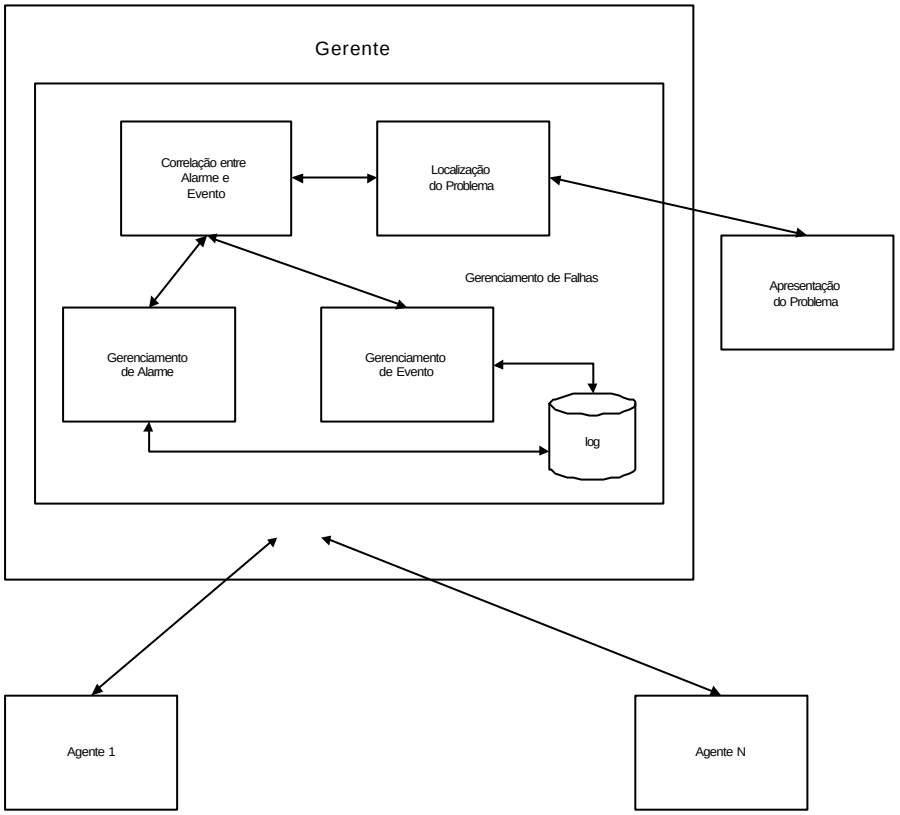


Figura 3.5: Gerenciamento de Falhas

Entretanto, um log de problemas por si próprio não é suficiente. Detalhes da provável causa e um diagnóstico com ações recomendadas são necessários, e devem ser imediatamente disponibilizados para os gerentes. Os problemas podem ser resultado de várias causas, e também é possível que um problema possa ser responsável por diferentes alarmes gerados. Então é necessário correlacionar estes problemas em uma única causa se existe uma possibilidade de múltiplas mensagens serem geradas.

O gerenciamento de falhas pode ser representado de acordo com a figura 3.5. Conforme a figura, agentes enviam alarmes de problemas para um gerente, os quais estes alarmes precisam estar correlacionados e filtrados para serem armazenados em um arquivo de log. Estes dados devem ficar disponíveis para o usuário. Para reduzir o tráfego na rede, agentes devem filtrar as notificações que eles enviam para os gerentes. Quando da implementação, restrições

como tamanho de memória, capacidade de armazenamento, velocidade dos processadores, e questões relacionadas à performance das estações de trabalho usadas pelos agentes e gerentes devem ser consideradas.

Documentos OSI relacionados à gerência de falhas têm sido utilizados para a construção de inúmeros produtos. Estes documentos são: Função de Registro de Alarmes (ISO 10164-4), Função de Gerenciamento do Registro de Eventos (ISO 10164-5) e Função de Controle de log (ISO 10164-5).

3.3.1 Função Registro de Alarmes

Notificações são mensagens emitidas por objetos gerenciados. Alarmes constituem um subconjunto de notificações e são gerados quando condições não usuais ocorrem. Eles podem ser gerados em função de condições anormais que foram detectadas, como por exemplo, quando existe uma degradação de um determinado serviço e ele ultrapassa um certo valor limite.

Alarmes podem ser gerados por mais de uma razão, então, para isolar as fontes, os alarmes devem ser correlacionados. Desses alarmes correlacionados, a fonte da condição de alarme deve ser identificada. Esses alarmes são relacionados em uma maneira padrão, e devem conter informações para identificar a natureza e a fonte do problema. Se alguns problemas ocorrem freqüentemente, informações adicionais devem ser utilizadas para analisar e estudar as tendências.

A função registrar alarmes (ISO 10165-4) leva em consideração as necessidades de serviços dos usuários, o protocolo necessário para oferecer esses serviços, e os parâmetros usados nos alarmes. Esses alarmes são empacotados no serviço de registro de alarme, que está presente nos agentes e no gerente.

Alarmes são muito importantes para a determinação de problemas. Os dados gerados pelos alarmes carregam não apenas uma ajuda para a determinação da fonte do problema, mas alguns deles podem indicar os passos do diagnóstico que podem ser iniciados. O serviço de registro de alarme é realizado conforme a figura 3.6.

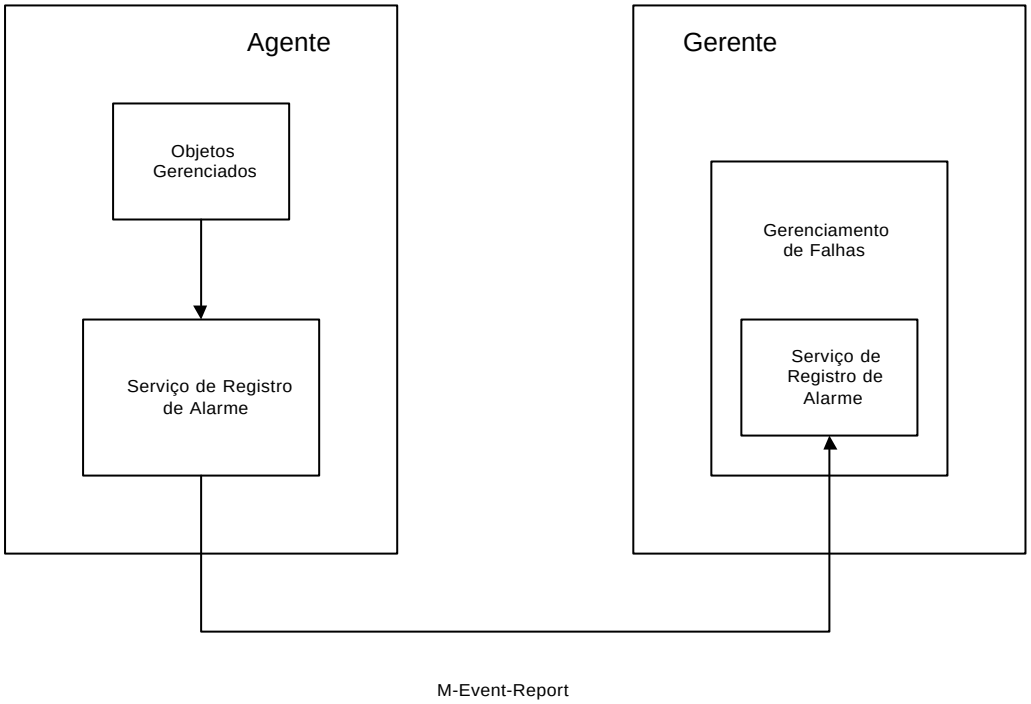


Figura 3.6: Serviço Alarme

3.3.2 Função Gerenciamento de Eventos

Notificações emitidas pelos objetos gerenciados devem ser seletivamente manipuladas para escolher qual delas devem ser enviadas para um ou mais gerentes. Também a freqüência do envio de notificações para o gerente deve ser flexível. Pode ser necessário enviar essas tais notificações para diferentes destinos. Para manipular a maneira como as notificações são informadas, dois objetos gerenciados: discriminator e eventForwardingDiscriminator, tiveram que ser definidas na Definition of Management Information (ISO 10165-2), e a função de gerenciamento de eventos é descrita em ISO 10164-5.

A principal idéia por trás da criação destas classes de objetos gerenciados (discriminator e eventForwardingDiscriminator) é adicionar flexibilidade e controlar a maneira com que notificações são convertidas em eventos.

Notificações de objetos gerenciados, após serem processadas, devem ser convertidas em potenciais eventos. Um potencial evento tem informações sobre a notificação emitida por um objeto gerenciado e informações derivadas de um processamento local que foi feito. Potenciais eventos informados servem de entrada para o módulo repassador de eventos (Figura 3.7). Esse módulo determina de onde os eventos foram enviados.

A função de gerenciamento de registro de eventos, como pode ser observado na figura 3.7, interage com o módulo repassador de eventos. As operações de gerenciamento que podem ser feitas sobre este módulo são inicializar, concluir e suspender.

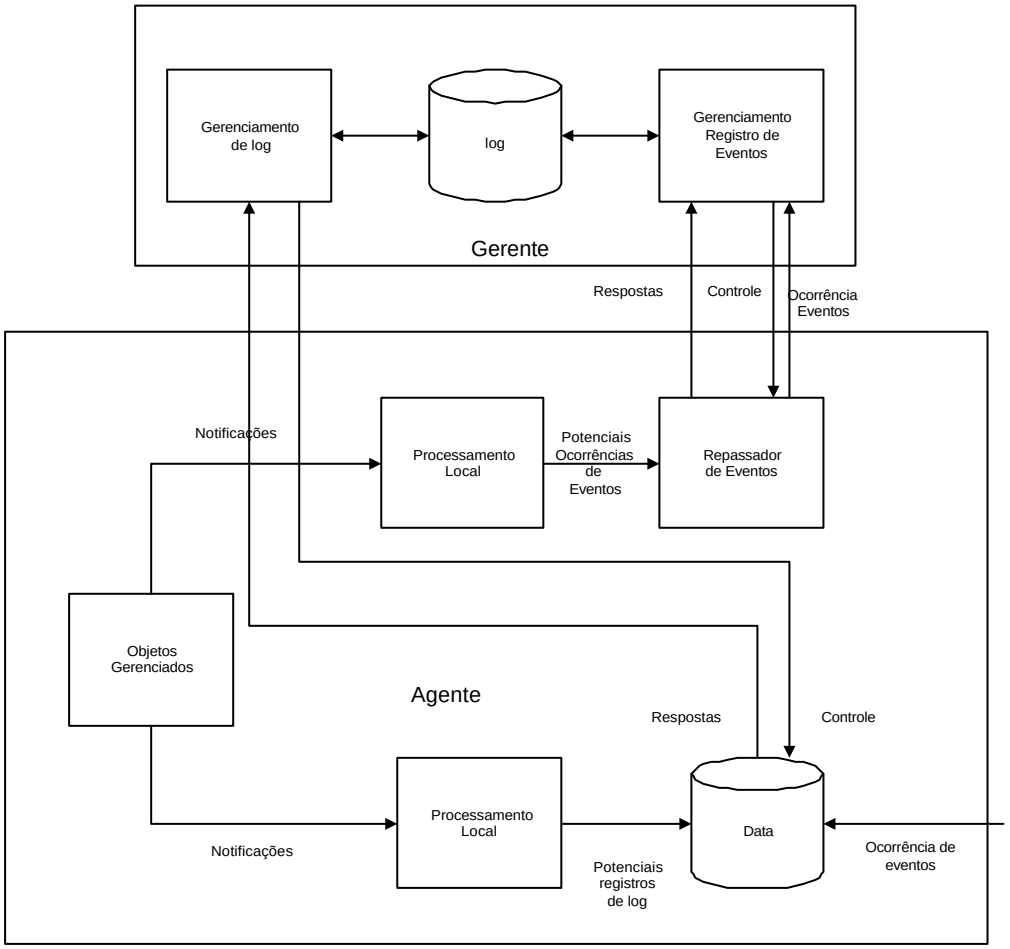


Figura 3.7: Gerenciamento de Eventos e log

3.3.3 Função Controle de Log

O documento função de controle de log (ISO 10164-6) cobre os requerimentos de usuário, serviços oferecidos, e o protocolo necessário para oferecer os serviços de registro de log. Eventos e notificações que são recebidas, precisam ser registradas para serem usadas posteriormente e algumas vezes para analisar algum problema. Este repositório é denominado log.

Os objetos gerenciados que emitem notificações, através de algum processamento podem gerar potenciais registros de log (Figura 3.7). Potenciais registros de log são enviados para um ou mais arquivos de log, e passam através de filtros antes de serem gravados no arquivo. Os filtros têm um conjunto de regras que determinam que registros de log seram gravados.

Capítulo 4

Metodologia DAG

Este capítulo descreve a metodologia DAG (Desenvolvimento de Aplicações de Gerenciamento) [Ram94]. Essa metodologia é parte da Infraestrutura Conceitual do Modelo de Concepção do SAGRES. São apresentadas as fases e módulos que compõem a metodologia e em seguida são definidas as funcionalidades pertinentes a cada módulo. A figura 4.1 mostra o modelo de concepção do SAGRES – Teoria Falhas (Figura 3.1) adicionada do bloco DAG.

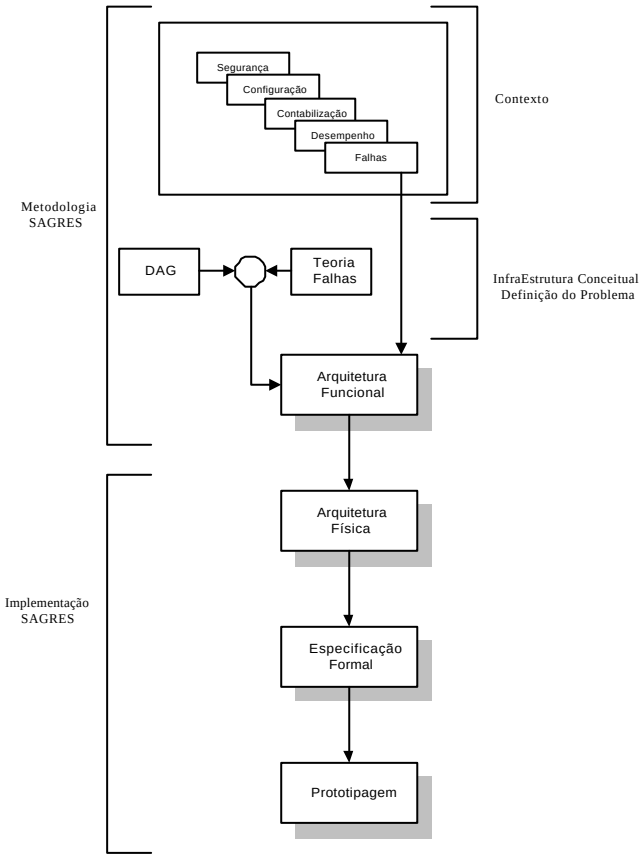


Figura 4.1: Modelo de Concepção do SAGRES - DAG

4.1 Introdução

A metodologia DAG (Figura 4.2) congrega um conjunto de atividades que facilita a análise e o desenvolvimento de aplicações de gerenciamento de redes de computadores.

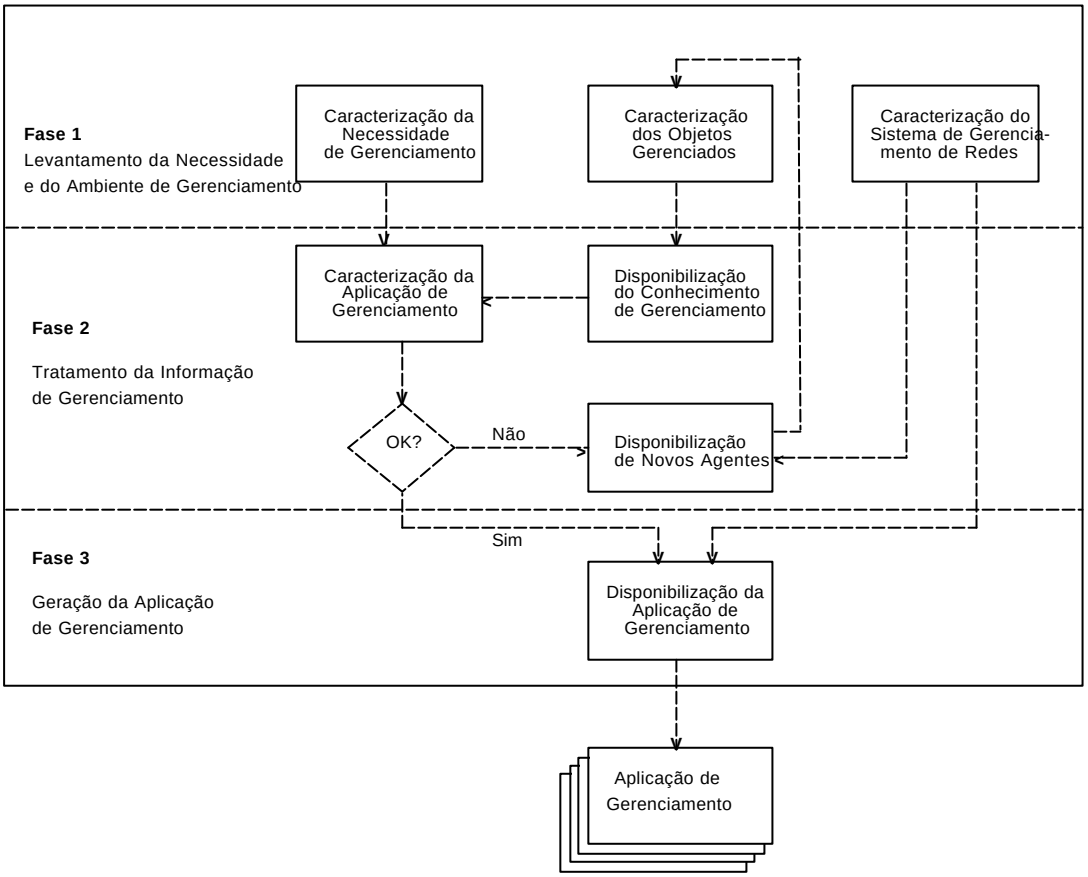


Figura 4.2: Arquitetura DAG

A escolha dessa metodologia no desenvolvimento do sistema proposto nessa dissertação deveu-se basicamente a dois fatores. Primeiro pelo fato dela contemplar os fatores requeridos em todas as fases do desenvolvimento e segundo por ter sido uma metodologia desenvolvida no âmbito do mesmo grupo de pesquisadores ao qual faz parte o autor dessa dissertação.

Estruturada em módulos e disposta em fases, a metodologia DAG especifica, de forma disciplinada, que atividades administradores de redes devem realizar para desenvolverem aplicações de gerenciamento. Neste contexto, o termo módulo agrupa uma ou mais atividades a serem desenvolvidas para o alcance de um determinado objetivo.

Como podemos abservar na Figura 4.2, a metodologia DAG está disposta em três fases denominadas:

- Fase 1 ou Levantamento da Necessidade e do Ambiente de Gerenciamento;
- Fase 2 ou Tratamento de Informações de Gerenciamento; e
- Fase 3 ou Geração da Aplicação de Gerenciamento.

A seguir, estão caracterizadas cada uma das fases da metodologia DAG.

4.2 Fases da Metodologia

4.2.1 Levantamento da Necessidade e do Ambiente de Gerenciamento

Conforme ilustra a Figura 4.2, a fase Levantamento da Necessidade e do Ambiente de Gerenciamento compreende os seguintes módulos:

- Caracterização da Necessidade de Gerenciamento;
- Caracterização dos Objetos Gerenciados; e
- Caracterização do Sistema de Gerenciamento de Redes.

O módulo Caracterização da Necessidade de gerenciamento consiste na elaboração textual de uma descrição informal da necessidade de gerenciamento. Nesta descrição, administradores devem explicitar, de forma clara e objetiva, o que se deseja gerenciar. A aplicação a ser desenvolvida está diretamente relacionada com a necessidade de gerenciamento.

O módulo Caracterização dos Objetos Gerenciados tem por finalidade organizar informações sobre objetos gerenciados contidos na MIB do ambiente. Especificamente, para cada objeto gerenciado, administradores devem relacionar sua descrição sucinta e classificá-lo de acordo com os seguintes critérios:

- critério comportamental: segundo esse critério, um objeto gerenciado pode representar informações estáticas ou dinâmicas;
- critério de modificação: um objeto gerenciado pode permitir ou não a alteração de seu conteúdo. Se um objeto permitir a alteração de seu conteúdo, ele é classificado como alterável, caso contrário, ele é classificado como não alterável; e
- critério funcional: Um objeto gerenciado pode ser associado a uma ou mais áreas funcionais de gerenciamento propostas pela ISO. Assim, um objeto pode ser associado ao gerenciamento de configuração, falhas, desempenho, contabilização e/ou segurança.

A classificação de objetos segundo os critérios comportamental e funcional deve ser baseada, respectivamente, na descrição do objeto e na definição das áreas funcionais de gerenciamento propostas pela ISO. Segundo o critério de modificação, sua classificação é, em geral, encontrada em documentos que descrevem a MIB.

No desenvolvimento do módulo Caracterização dos Objetos Gerenciados, é proposta a elaboração da tabela "Objetos de Gerenciamento" (Tabela 4.1). As colunas CC, CM e CF desta tabela representam, respectivamente, os critérios comportamental, de modificação e funcional.

Objeto	**Descrição Sucinta**	**CC**	**CM**	**CF**

Tabela 4.1 Objetos de Gerenciamento

No módulo Caracterização do Sistema de Gerenciamento de Redes, administradores devem relacionar as funcionalidades das ferramentas do SGR que tem em disponibilidade. Para tanto é proposta a elaboração da tabela "Ferramentas do SGR" (Tabela 4.2). A funcionalidade das ferramentas, geralmente, está descrita na documentação do SGR. A Tabela 4.2 é utilizada como entrada no módulo Disponibilização da Aplicação de Gerenciamento da fase 3.

Ferramenta	Função

Tabela 4.2 Ferramentas do SGR

4.2.2 Tratamento de Informações de Gerenciamento

A fase Tratamento de Informações de Gerenciamento compreende, os seguintes módulos:

- Caracterização da Aplicação de Gerenciamento;
- Disponibilização do Conhecimento de Gerenciamento; e
- Disponibilização de Novos Agentes.

No módulo Caracterização da Aplicação de Gerenciamento, administradores devem, inicialmente, determinar a natureza da aplicação a ser desenvolvida baseados na
descrição informal da necessidade de gerenciamento elaborado no módulo Caracterização da Necessidade de Gerenciamento. A natureza de uma aplicação de gerenciamento pode ser definida como de monitoramento, de controle, ou de monitoramento e controle.

A seguir, apresentamos as atividades a serem realizadas de acordo com a natureza da aplicação de gerenciamento, distribuídas em três casos:

Caso 1 : Monitoramento
Se a natureza da aplicação for de monitoramento, administradores devem realizar as seguintes atividades:

- Atividade 1: determinar parâmetros de monitoramento; e
- Atividade 2 : caracterizar condição de evento e a forma de detecção desta condição de evento.

Caso 2 : Controle
Se a natureza da aplicação de gerenciamento for de controle, administradores devem realizar as seguintes atividades:

- Atividade 1: determinar os objetos gerenciados cujos conteúdos devem ser alterados;
- Atividade 2 : estabelecer novos valores para tais objetos.

Na caracterização de aplicações de controle, é proposta a elaboração da tabela "Parâmetros de Controle" (Tabela 4.3).

Objeto Gerenciado	**Novo Conteúdo**

Tabela 4.3 Parâmetros de Controle

Caso 3 : Monitoramento e Controle

À medida em que aumenta a complexidade de uma aplicação de gerenciamento é provável a existência de várias atividades de monitoramento e uma ou mais atividades de controle. Se a natureza da aplicação de gerenciamento for de monitoramento e controle, administradores devem realizar as atividades propostas isoladamente para o desenvolvimento de aplicações de monitoramento e controle.

O módulo Disponibilização do Conhecimento de Gerenciamento apóia a determinação de objetos a serem monitorados e/ou controlados, e de condições de eventos do módulo Caracterização da Aplicação de Gerenciamento. No contexto da metodologia DAG, o módulo Disponibilização do Conhecimento de Gerenciamento detém o conhecimento (*expertise*) de um profissional em gerenciamento de redes e pode ser implementado por sistemas computacionais inteligentes. A tabela Objetos de Gerenciamento (Tabela 3.1), elaborada no módulo Caracterização dos Objetos Gerenciados, contribui para a formação do conhecimento do módulo Disponibilização do Conhecimento de Gerenciamento. Muitas vezes, objetos a serem monitorados e/ou controlados em uma aplicação são determinados pela coluna Descrição Sucinta da tabela Objetos de Gerenciamento.

O módulo Disponibilização de Novos Agentes é utilizado nas situações em que não é possível caracterizar a aplicação de gerenciamento em decorrência da inexistência de objetos gerenciados na MIB. Quando isto ocorre, administradores devem disponibilizar novos agentes. A disponibilização de novos agentes corresponde à aquisição ou ao desenvolvimento de novos agentes. Agentes podem ser adquiridos de fabricantes de software e hardware ou

desenvolvidos no próprio ambiente de gerenciamento utilizando recursos do SGR e do sistema operacional do ambiente de gerenciamento. Em geral, SGRs que permitem o desenvolvimento de agentes oferecem APIs (*Application Program Interfaces*) que facilitam a implementação de serviços utilizados por agentes.

4.2.3 Geração da Aplicação de Gerenciamento

A fase Geração da Aplicação de Gerenciamento compreende o módulo Disponibilização da Aplicação de Gerenciamento. No módulo Disponibilização da Aplicação de Gerenciamento, administradores devem, inicialmente, verificar na tabela Ferramentas do SGR (Tabela 4.2), elaborada no módulo Caracterização do Sistema de Gerenciamento de Redes, se o SGR disponível no ambiente oferece ferramentas que possibilitem a implementação da aplicação de gerenciamento necessária.

Se existirem ferramentas do SGR que possibilitem a geração da aplicação necessária, administradores devem fazer uso dessas ferramentas. Caso contrário, administradores devem elaborar um programa em uma linguagem de programação cuja sintaxe permita a implementação do diagrama de fluxo da aplicação de gerenciamento sugerido no módulo Caracterização da Aplicação de Gerenciamento. Em alguns casos, a necessidade de gerenciamento pode ser atendida pelo uso de ferramentas do SGR e programas auxiliares, que têm como propósito suprir deficiências das ferramentas utilizadas.

4.3 Utilização da Metodologia DAG

Nesta seção, exemplificamos a utilização da metodologia DAG em uma aplicação de gerenciamento desenvolvida para indicar interfaces sem comunicação no roteador de uma rede.

Fase 1: Levantamento da Necessidade e do Ambiente de Gerenciamento

- Módulo Caracterização da Necessidade de Gerenciamento

Neste módulo, foi elaborada a seguinte descrição da necessidade de gerenciamento:

Conhecimento, em tempo real, da existência de interfaces sem comunicação no roteador do ambiente.

- Módulo Caracterização dos Objetos Gerenciados

Neste módulo, foi elaborada a tabela Objetos de Gerenciamento, que relaciona alguns objetos gerenciados contidos na MIB do ambiente. Nesta tabela, a descrição sucinta dos objetos gerenciados e a classificação destes segundo o critério de modificação foram baseados no documento RFC (Request For Comments) 1066.

Objeto	Descrição Sucinta	CC	CM	CF
SysDescr	Descrição do sistema gerenciado	E	NA	Cf
IfNumber	Número de interfaces do elemento	E	NA	Cf
IfIndex	Valor de identificação para cada interface	E	NA	Cf
IfDescr	Informações sobre a interface	E	NA	Cf
IfPhysAddress	Endereço físico da rede	E	NA	Cf
IfOperStatus	Estado operacional corrente da interface. Ativada(1), desativada(2) Em teste(3)	D	NA	Cf,F
IfInOctets	Número total de octetos recebidos na interface	D	NA	D
IfOutOctets	Número total de octetos transmitidos	D	NA	D
IfOutQLen	Tamanho da fila de pacotes de saída	D	NA	D

- Módulo Caracterização do Sistema de Gerenciamento de Redes

Neste módulo, foi elaborada a tabela Ferramentas do SGR, que relaciona as ferramentas do SunNet Manager.

Ferramenta	Função
Quick Dump	Permite a coleta de dados em um determinado instante de tempo
Data Report	Permite a coleta de dados durante um período de tempo

Event report	Possibilita que o administrador receba uma sinalização (notificação) sempre que uma determinada condição ocorrer em um componente da rede.
Set Request	Possibilita a alteração do conteúdo de objetos de gerenciamento
Browser	Permite a visualização textual de dados coletados
Discovery	Permite a descoberta automática de nodos da rede
Grapher	Permite a visualização gráfica de dados coletados

Fase 2: Tratamento de Informações de Gerenciamento

- Módulo Caracterização da Aplicação de Gerenciamento

Neste módulo, identificou-se a natureza da aplicação como sendo de monitoramento. Em seguida, forma preenchidas as seguintes templates:

Objetos a serem monitorados: ifOperStatus
Período de Monitoramento: 12 horas
Intervalo de Monitoramento: meia hora

Condição de Evento: ifOperStatus igual a 2
Forma de Detecção: Imediata

O objeto gerenciado a ser monitorado (ifOperStatus) e a condição de evento foram determinados com o apoio do módulo Disponibilização do Conhecimento de Gerenciamento. Fez-se uso da experiência de um profissional em gerenciamento de redes que, neste caso, foi o próprio administrador.

Fase 3: Geração da Aplicação de Gerenciamento

- Módulo Disponibilização da Aplicação de Gerenciamento

De acordo com a natureza da aplicação e os dados dos templates, considerou-se a utilização do diagrama do fluxo de monitoramento com detecção imediata. Logo após, verificou-se que a necessidade de gerenciamento poderia

ser atendida pela utilização da ferramenta Event Report do SunNet Manager(SNM). Assim, foi desenvolvida uma rotina no SNM que testava, a cada meia hora o estado operacional das interfaces do roteador.

4.4 Disponibilização do Conhecimento

O módulo "Disponibilização do Conhecimento de Gerenciamento" definido na fase 2 da metodologia DAG (Figura 4.2) tem sua funcionalidade baseada na definição, comportamento e relacionamento entre si dos objetos gerenciados do ambiente. Para cada objeto gerenciado ou conjunto de objetos está associado, em uma base de conhecimento, um conjunto de regras.

A Tabela Objetos de Gerenciamento, elaborada no módulo caracterização dos Objetos Gerenciados da metodologia DAG, contribui para a formação destas regras.

As regras compreendem condições de eventos que podem ou não implicar em ações de gerenciamento e são estabelecidas através da definição de objetos gerenciados. Uma ação pode ser composta por comandos get e/ou set. Na metodologia DAG, uma necessidade de gerenciamento deve ser confrontada com o diagnóstico de regras para obtenção da condição de evento e dos objetos a serem monitorados e/ou controlados.

O módulo Disponibilização do Conhecimento de Gerenciamento é composto por quatro componentes denominados interface do usuário, gerador de regras, analisador de regras, e base de conhecimento, descritos a seguir :

- interface do usuário: fornece um meio comum para a apresentação de telas baseadas em janelas voltadas para uma interação facilitada e intuitiva com os usuários;

- gerador de regras: mapeia informações contidas na Tabela de Objetos de Gerenciamento, particularmente a descrição sucinta dos objetos gerenciados, em regras que são armazenadas na base de conhecimento;

- analisador de regras: é responsável pela confrontação das regras contidas na base de dados com a necessidade de gerenciamento. Uma condição de evento a ser utilizada na metodologia DAG pode envolver várias regras da base de conhecimento; e

- base de conhecimento: é o repositório das regras a serem utilizadas para a definição da aplicação de gerenciamento.

A Figura 4.3 a seguir representa o detalhamento do módulo Disponibilização do Conhecimento de Gerenciamento da metodologia DAG. A base de conhecimento, na Figura 3.2, estabelece a ligação da metodologia DAG à arquitetura típica de um SGRBC (Sistema de Gerenciamento de Redes Baseado em Conhecimento), o qual será tratado com mais detalhes no próximo capítulo.

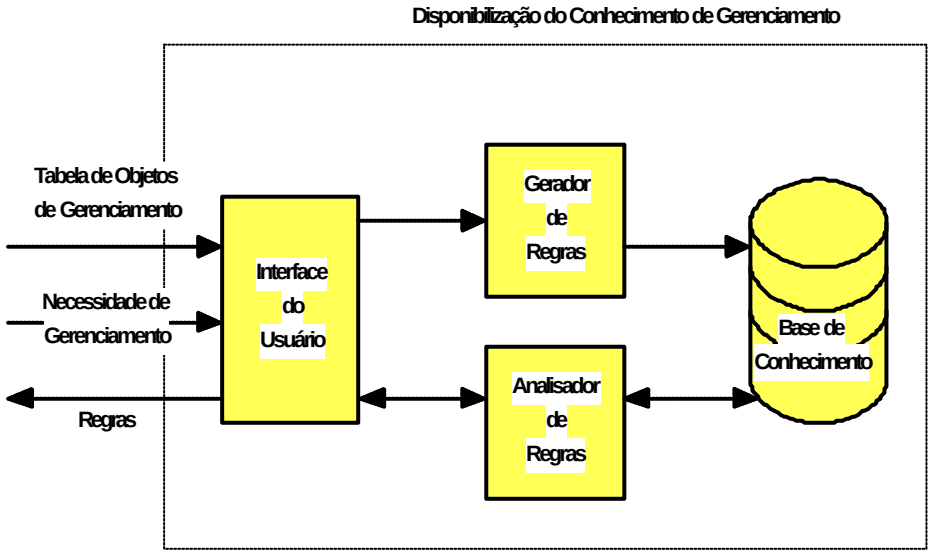


Figura 4.3: Disponibilização do Conhecimento de Gerenciamento

Capítulo 5

SGRBCs

Neste capítulo apresentamos os conceitos de SGRBCs (Sistemas de Gerenciamento de Redes Baseados em Conhecimento) [Mur94]. A figura 5.1 mostra o modelo de concepção do SAGRES – DAG (Figura 4.1) adicionado do bloco SGRBC.

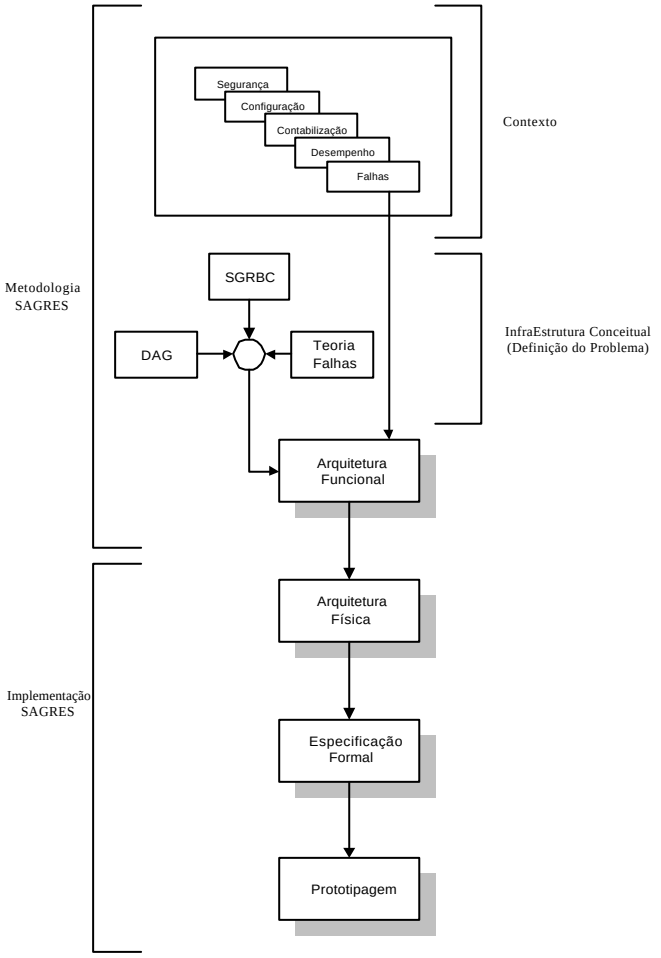


Figura 5.1: Modelo de Concepção do SAGRES - SGRBC

5.1 Limitações dos SGRs

A atividade de gerenciamento dita convencional não atende às exigências inerentes à complexidades das redes atuais. Em função disto, constatamos um esforço de pesquisa e desenvolvimento de ferramentas apoiadas em recursos não convencionais, dentre eles o uso de sistemas inteligentes.

Os principais problemas encontrados nas soluções existentes são :

- alguns fabricantes fornecem ferramentas de gerência que são destinadas a usuários inexperientes, contemplando porém poucas funcionalidades, sendo totalmente inadequadas para gerenciamento de plataformas de grande porte;
- as ferramentas comerciais que disponibilizam algumas das funcionalidades descritas neste trabalho são extremamente caras, impossibilitando o acesso às pequenas e médias empresas; e
- várias ferramentas desenvolvidas por grandes empresas, apesar de contemplarem inúmeras funcionalidades, não dispõem de mecanismos inteligentes para apoio à atividade cotidiana de gerenciamento, ficando limitado seu uso a usuários experientes.

5.2 Classificação dos SGRBC's

Sistemas de gerenciamento de redes baseados em conhecimento (SGRBCs) são sistemas capazes de adquirir a visão de um especialista em gerência de redes, de sintetizar o conhecimento e as condições existentes, e de analisar situações a partir deste conhecimento. Muitas razões justificam o uso de SGRBCs, incluindo o tratamento eficiente da crescente complexidade da rede, o tratamento consistente de problemas de gerência e o treinamento e o aproveitamento do conhecimento de pessoal especializado.

Para fazer a representação do conhecimento nos SGRBC's existem algumas técnicas disponíveis, dentre elas a Manutenção do Raciocínio, o Raciocínio Temporal, Sistemas de produção e Planejadores.[Mur94]

A técnica Manutenção do Raciocínio busca garantir que um banco de dados de assertivas seja mantido de maneira consistente. Essa consistência evita a possibilidade de existirem assertivas verdadeiras e falsas simultaneamente.

Existem duas maneiras possíveis para tratar a técnica Manutenção do Raciocínio, os sistemas de manutenção de verdade e sistemas de manutenção de verdade baseados em suposição[Mur94].

O Raciocínio Temporal é aplicado em sistemas de tempo real nos quais o fator tempo possui uma importância relevante. Para tratar dos sistemas de tempo real, a técnica de Raciocínio Temporal faz uso dos seguintes recursos:

- banco de dados temporal;
- linguagem de acesso ao bando de dados;
- capacidade de projeção; e
- capacidade de manutenção do raciocínio temporal.

Os Sistemas de Produção consistem de uma série de regras e uma estratégia de controle para analisar estas regras. Cada regra possui um conjunto de condições e uma assertiva que pode ser verdadeira ou falsa.

Planejadores representam a técnica mais complexa de construção dos sistemas baseados em conhecimento. O Planejador inicia com um conjunto de objetivos e uma biblioteca de ações disponíveis para ele. Objetivos consistem em estados que o sistema deseja alcançar por alguma razão. Ações podem ser imaginadas como a parte do programa que afetará o ambiente. O Planejador seleciona uma série de ações que quando executadas, levarão o Planejador do estado atual para o estado desejado.

Os SGRBC's em função do modelo que operam, podem ser classificados em SGRBC's assistentes e SGRBC's autônomos.

SGRBC's assistentes constituem o tipo mais simples de sistemas de gerência inteligentes. Caracterizam-se por trabalharem *off-line*, auxiliando usuários na gerência de redes. Um administrador, ao detectar a ocorrência de um problema na rede, utiliza a base de conhecimento para solucioná-lo. A base de conhecimento faz uma série de perguntas a respeito da rede e das condições observadas e, em seguida, diagnostica o problema, fornecendo uma possível solução. Uma vantagem desse enfoque é que a base de conhecimento pode ser usada para treinar usuários, além de fornecer diagnósticos.

Os SGRBC's autônomos trabalham *on-line*, tomando decisões de gerenciamento. Estes sistemas estão ativos continuamente na rede e, geralmente, decidem o que fazer quando detectam problemas. Tais características exigem uma alta capacidade de desempenho, pois um SGRBC autônomo monitora o contexto de alterações constantes da rede, e decide, por si mesmo, se algo precisa ser feito. Por estar ativo continuamente na rede, um SGRBC autônomo pode detectar e isolar problemas mais rapidamente que os seres humanos, o que implica em um melhor desempenho e disponibilidade para a rede.

Independente da classificação acima, um SGRBC auxilia na detecção de eventos críticos em uma rede, e fornece possíveis soluções para os mesmos. Seu principal objetivo é reduzir o nível de experiência (*expertise*) exigida de administradores para que se possa diagnosticar e corrigir problemas de maneira rápida e confiável.

5.3 Arquitetura SGRBC

A arquitetura dos SGRBC's foi concebida de maneira a permitir alta modularidade [Mur94]. Os sistemas baseados em conhecimento podem ser desenvolvidos independentemente e cada um deles conter conhecimento próprio, permitindo melhorar os níveis de desempenho e disponibilidade. Uma arquitetura típica de um SGRBC é apresentada na Figura 5.2, e os seguintes elementos de informação, abaixo conceituados, são necessários para o entendimento das funcionalidades de cada módulo da arquitetura:

- Contadores: indicam o número de ocorrência de erros ou de outros eventos relevantes na rede;
- eventos: representam ocorrências significativas na rede;
- diagnóstico: corresponde a um esquema que descreve um evento na rede;
- dado temporal: constitui uma estrutura que representa eventos e diagnósticos que podem ser armazenados na base de dados temporal; e

- imagem da rede: corresponde a um conjunto de esquemas que descreve a rede e seus elementos constituintes. Tais elementos podem ser, por exemplo sistemas sub-redes e equipamentos de interconexões (pontes, roteadores e *gateways*).

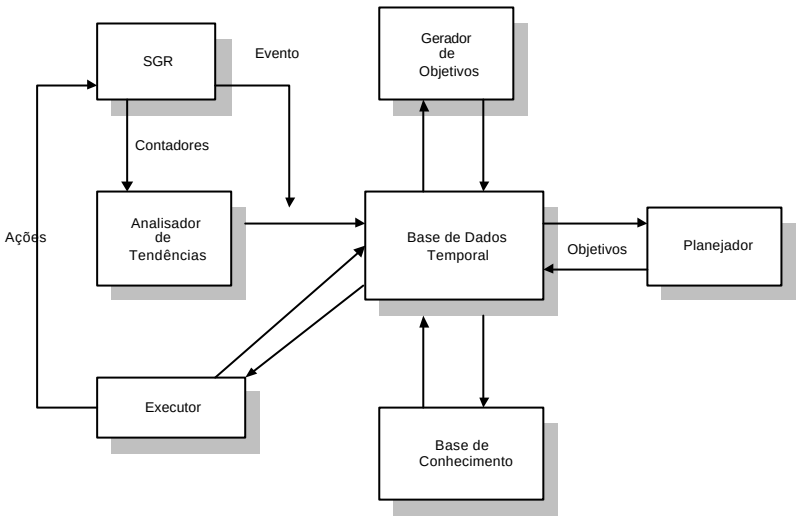


Figura 5.2 Arquitetura SGRBC

A seguir, descrevemos os principais componentes da arquitetura dos Sistemas de Gerenciamento de Redes Baseados em Conhecimento conforme ilustrada na Figura 5.2:

- analisador de tendências: identifica anomalias de comportamento na atividade da rede indicadas por mudanças excessivas nos valores de contadores. Os algoritmos usados neste componente são baseados em métodos de análises estatísticas que determinam quando uma alteração no valor de um contador varia dentro de um período pré-estabelecido ou tem relevância estatística, por exemplo aumentos repentinos ou de longa duração em valores de um contador. Se o resultado da análise mostrar que o contador está dentro de um limite aceitável, nenhuma ação precisa ser tomada, senão, um esquema é criado para descrever as ocorrências de um evento. Este evento é, em seguida, enviado para a base de dados temporal;
- base de dados temporal (BDT): implementa a base de dados central do sistema e contém assertivas e informações sobre intervalos de tempo nos quais estas assertivas são consideradas verdadeiras. Adicionalmente, a BDT apresenta uma funcionalidade de simulação que é usada para prever eventos que espera-se que venham a ocorrer. A BDT também processa consultas vindas de outros componentes do sistema;

- base de conhecimento: consiste de regras que especificam os eventos. Tais especificações podem ser o diagnóstico de um novo problema ou uma explicação de um evento em termos de diagnósticos anteriores. A única entrada para a base de conhecimento são os eventos vindos da BDT. Quando um novo evento é gerado na BDT, a base de conhecimento tenta explicar o evento. Os eventos podem ser explicados pelo diagnóstico de um problema, e um esquema de diagnóstico é criado e enviado para a BDT;
- gerador de objetivos: este componente tem a responsabilidade de decidir que ações devem ser tomadas para identificar eventos críticos. Ele monitora a BDT em busca de problemas que precisam ser solucionados e gera os objetivos para solucioná-los. Essa atividade inclui a conclusão de diagnósticos parciais, a determinação de como se reconfigurar a rede após um evento crítico, a solução de falhas e a geração de relatórios para o administrador da rede. Os objetivos são identificados, priorizados, e enviados para a BDT;
- planejador: cria planos para ações a serem tomadas pelo SGRBC, incluindo a reconfiguração da rede após uma falha e a localização e correção de problemas. O planejador busca objetivos do gerador de objetivos armazenadas na BDT. Quando um objetivo é armazenado, o planejador gera um plano para viabilizar o objetivo dentro de um intervalo de tempo específico. Após a geração do plano, o mesmo é enviado para a BDT, onde ele é executado pelo executor; e
- executor: executa os planos gerados pelo planejador. Isto requer a geração das ações de gerenciamento a serem enviadas para o SGR, e também o completo monitoramento do plano para assegurar que ele funcione a contento. Esse componente monitora a BDT, aguardando que um plano esteja pronto para ser executado. Quando isto acontece, o plano é lido na BDT e cada ação é enviada para o SGR para que a mesma seja executada.

5.4 Estado Atual dos Sistemas Baseados em Conhecimento

Neste item, analisamos três ferramentas desenvolvidas no Brasil que fazem uso de sistemas baseados em conhecimento como apoio às atividades de gerenciamento ou administração de redes de computadores. Para cada sistema, serão apontadas suas potencialidades e deficiências.

5.4.1 Sistema Olho Vivo

O sistema OLHO VIVO [Art96] foi construído para coletar informações da rede observando os indicadores de degradação de desempenho e buscando soluções para estes problemas. O sistema faz uso do agente RMON conhecido como BTNG (*Beholder The Next Generation*) e de um módulo inteligente.

O agente RMON BTNG é um software de domínio público que foi desenvolvido pelo grupo de pesquisa DNPAP (*Data Network Performance Analysis Project*) da Universidade de Delft na Holanda. Este agente implementa a RMON MIB completa e alguns outros objetos que esse grupo de pesquisa adicionou. A estrutura do BTNG constitui-se de um conjunto de coletores, onde cada coletor vai ser responsável por um grupo de objetos RMON MIB.

O sistema Olho Vivo está dividido em módulos conforme Figura 5.3.

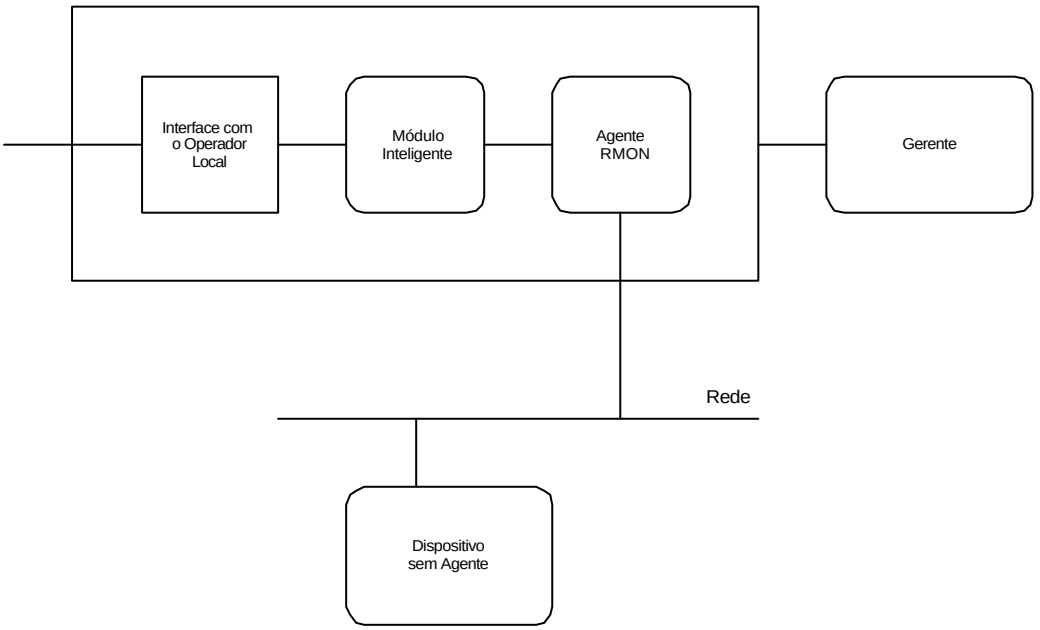


Figura 5.3: Sistema Olho Vivo

O módulo inteligente é o principal componente do sistema. Esse módulo analisa os dados coletados pelo monitor (BTNG) e emite sugestões ao administrador da rede sobre certos parâmetros que podem causar problemas significativos no desempenho da rede. Esse módulo foi desenvolvido em *Perl Scripts* [Wal96] [Sri97]. O próprio

software BTNG traz consigo uma ferramenta que contém os comandos que fazem as requisições SNMP ao monitor.

No sistema Olho Vivo , uma estação de gerenciamento SNM (SunNet Manager) é utilizada como gerente. Para integrar essa estação de gerência com o agente RMON, é necessário que o SNM acesse o BTNG.

A comunicação do sistema com o administrador da rede é através de mensagens eletrônicas (e-mail). Quando uma das regras é verdadeira, é emitido uma mensagem descrevendo o problema identificado. Isso pode ser posteriormente analisado pelo administrador, o qual verifica se a rede está trabalhando dentro dos parâmetros de operação que garantem um desempenho normal .

Apesar de apresentar muito da funcionalidade desejada, este sistema utiliza como gerente uma ferramenta proprietária como interface (SunNet Manager), trata apenas variáveis da RMON MIB e não trata da correção de problemas de forma automática, limitando-se a fornecer sugestões ao administrador da rede.

5.4.2 Agente 6

Uma outra ferramenta disponível que utiliza mecanismos inteligentes é o sistema Agente 6 [Roc96]. Ele foi criado com o objetivo de analisar o congestionamento em um barramento ethernet. Através da monitoração do volume de tráfego e da quantidade de erros, tem-se uma medida da qualidade e desempenho dos serviços de comunicação da rede. O Agente 6 opera em dois modos básicos, a monitoração de dados e monitoração de eventos.

Pela monitoração de dados, o Agente 6 fica em execução contínua, fazendo uma consulta a cada intervalo de tempo. Ao final de cada consulta, os dados recuperados são enviados ao gerente (SNM - SunNet Manager). Esse envio periódico de dados permite a geração de gráficos por parte do SNM e também o armazenamento dos dados recuperados em consultas anteriores no próprio agente, que então pode gerar estatísticas sobre o andamento do sistema de comunicações.

A monitoração de eventos funciona basicamente como a monitoração de dados, porém não tem a finalidade de mostrar ao usuário através de números ou gráficos todos os dados recuperados, mas somente determinados eventos

sobre esses dados. Esses eventos são gerados basicamente quando um dado monitorado atinge ou ultrapassa determinado valor (*threshold*). Dessa forma, é possível controlar se um dado recuperado está dentro de uma faixa de valores tidos como aceitáveis. Caso esse dado esteja fora dessa faixa, o evento informa ao administrador, de algum modo, para que ele possa verificar a situação e tomar a medida necessária.

Com o objetivo de determinar o comportamento médio padrão da rede a ser gerenciada, deve-se inicialmente construir um *baseline* desta rede. O módulo denominado diagnóstico, que se encontra associado ao sistema, aciona regras cada vez que a medida lida esteja fora dos parâmetros padrões contidos no *baseline* e funciona da seguinte forma :

- para cada medida lida dos agentes de monitoração que chegar ao alcance da média da hora em que ocorreu, se dispara o módulo diagnóstico que passa a verificar se existem problemas de acordo com as regras do módulo;
- se o módulo diagnóstico verificar que se trata de um problema de queda de performance ou congestionamento, faz uso de regras para estabelecer quais os motivos que causam o evento; e
- em um terceiro momento, após verificados os anteriores, tomam-se atitudes para corrigir o problema, reportando ao administrador da rede, via e-mail ou pela interface, as anomalias encontradas sugerindo procedimentos corretivos.

O módulo diagnóstico foi elaborado com todas as características de um sistema especialista. Todo o conhecimento está representado por fatos e regras. Os fatos são gerados a partir dos arquivos de monitoração e *baseline*. A implementação desse sistema foi feita em linguagem Prolog.

No protótipo, o mecanismo de inferência é invocado para cada conjunto de regras, de forma seqüencial. Essa estratégia, bastante procedimental, torna a inferência mais fácil de ser implementada.

A arquitetura do sistema (Figura 5.4) é composta dos seguintes módulos:

- Blocos de conversão;
- Base de conhecimento;
- Motor de inferência;
- Explanação; e
- Interface.

O módulo de conversão é responsável por receber os arquivos de monitoração e da *baseline* e convertê-los para o formato de fatos Prolog. Esses fatos irão formar a base de conhecimento.

O motor de inferência é utilizado para, a partir da base de conhecimento, analisar os valores dos parâmetros e inferir um diagnóstico. Esse diagnóstico é apoiado pela explanação que indica os motivos que levaram a tal situação problema, além de sugerir possíveis formas de resolução desses problemas.

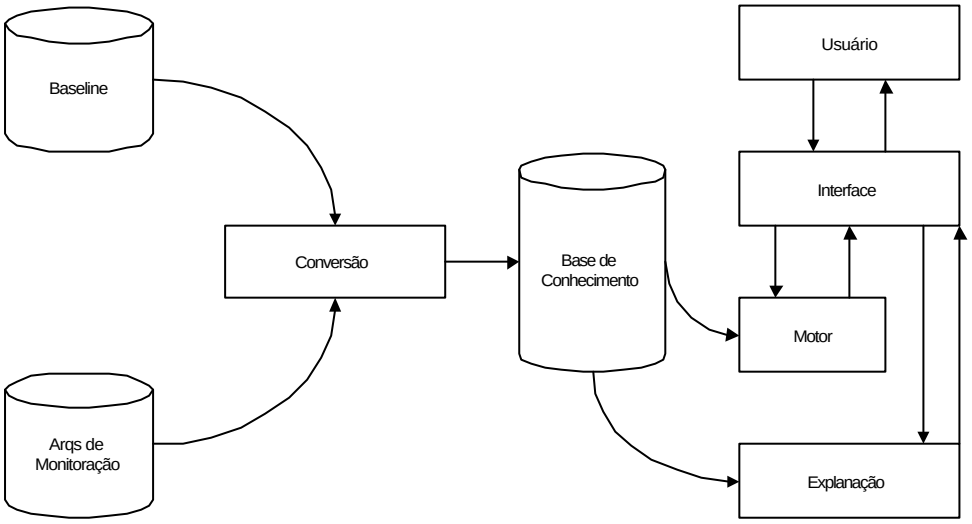


Figura 5.4: Sistema Agente 6

Pela interface, o administrador da rede recebe as informações do sistema bem como as sugestões para a resolução dos problemas detectados.

Apesar da utilização de um mecanismo inteligente (Sistemas Especialistas) no diagnóstico dos problemas, este sistema apresenta alguns inconvenientes. O primeiro é quanto à dependência de uma ferramenta proprietária como interface (SunNet Manager), além de não disponibilizar uma interface para inclusão, deleção ou alteração de regras por parte do usuário. Apesar de serem fornecidos diagnósticos sobre o estado da rede, o sistema limita-se a informar ao administrador as possíveis soluções, sendo incapaz de resolver os problemas.

5.4.3 I-DREAM

O I-DREAM (*Intranet baseD REsource and Application Monitoring system*) [Fra97] é um sistema de monitoração de recursos e aplicações baseados em tecnologia intranet.

O modelo proposto para o I-DREAM visou o desenvolvimento de uma ferramenta que suportasse os seguintes requisitos:

- suporte distribuído: deve ser possível particionar a monitoração em diversos componentes da rede por razões de escalabilidade, eficiência e robustez. Além disso, os resultados da monitoração distribuída devem ser integrados para permitir uma visão única do sistema. Por exemplo, os agentes que coletam informação sobre a taxa de utilização da CPU podem estar distribuídos em cada máquina, coletando informações sobre as mesmas, independentemente um dos outros;
- suporte integrado: é necessário que a ferramenta apresente uma visão integrada dos componentes da rede apresentados no item anterior e seja capaz de correlacionar as informações coletadas sobre cada um deles;
- suporte uniforme: além da ferramenta ser gráfica, a mesma deve apresentar uma mesma interface para todos os componentes. Além disso, os vários componentes de monitoração devem ser capazes de se comunicarem através de um protocolo de comunicação padrão;
- suporte escalável: a ferramenta deve ser capaz de crescer para se adaptar à demanda do ambiente onde será usada. Por exemplo, caso o ambiente necessite da aquisição de um novo serviço a ferramenta deve ser capaz de incluir uma nova ferramenta para monitorá-lo;
- suporte seguro: toda comunicação entre os diversos componentes que fazem a monitoração deve ser feita de maneira segura;
- suporte que atenda a padrões: a maioria das ferramentas existentes não possuem qualquer padrão. É importante que a ferramenta siga algum padrão, unifique a administração de sistemas e o gerenciamento de redes em ambientes heterogêneos;
- coleta de informações: é necessário que a ferramenta seja capaz de coletar dados sobre o comportamento real da rede. Isto inclui fornecer informações sobre os *hosts* que estão funcionando e os que não estão, fornecer informações sobre os serviços e aplicações disponíveis, fornecer informações sobre os recursos, etc;

- análise de informações: além de coletar informações, a ferramenta deve ser capaz de analisar os dados coletados. Para tanto é necessário que a ferramenta possua parâmetros para realizar comparações, que dizem respeito ao comportamento desejado da rede;
- identificação e solução de problemas: de posse dos dados sobre o comportamento real da rede e dos dados sobre o comportamento realmente desejado, a ferramenta deve ser capaz de realizar uma comparação para identificar os problemas, seja diretamente ou seja através da correlação de dados;
- correção de problemas: a correção de alguns problemas, muitas vezes, é trivial e pode ser automatizada. A ferramenta deve ser capaz de realizar correções nestes casos. Para tanto, é necessário que a ferramenta possua uma base onde regras e ações sejam definidas. A ferramenta deve possuir também agentes que sejam capazes de realizar tais ações uma vez que sejam solicitadas;
- disponibilidade dos componentes: com a monitoração constante dos componentes da rede, os problemas estarão sendo corrigidos, seja automaticamente ou seja porque o administrador foi informado de sua ocorrência e já pode tomar providências para repará-lo;
- custo mais acessível: as ferramentas comerciais possuem custo muito elevado para instalações de pequeno e médio porte. É importante que a ferramenta ofereça solução para tais redes a custos mais acessíveis;
- monitoração de uma rede local de maneira segura a partir de qualquer *host* da Internet: a ferramenta deve usar recursos que permitam a possibilidade da monitoração de recursos a partir de qualquer *host* na Internet. A melhor maneira de realizar tal atividade é através do uso de protocolos da família TCP/IP e de interfaces baseadas em servidores e clientes WWW;
- várias visões da rede para diferentes tipos de usuários: cada tipo de usuário deve possuir uma visão diferente do sistema. Ao administrador de sistemas é permitido o acesso à interface que permite a inserção de dados sobre o comportamento desejado da rede e a visualização dos dados coletados e do diagnóstico fornecido sobre o comportamento real da rede.

O modelo conceitual do I-DREAM descreve os subsistemas necessários para suportar as atividades tradicionais de monitoração. A Figura 5.5 descreve o modelo conceitual utilizado pelo I-DREAM. Na figura, as setas indicam troca de informações.

O modelo conceitual do I-DREAM é composto de cinco subsistemas:

- Controle: é o subsistema responsável pela ativação e desativação das monitorações e pela utilização das definições para estabelecer o comportamento das ferramentas de monitoração e interagir com o subsistema Coleta e Atuação para ativar e desativar seus agentes;
- Definição: é o subsistema que mantém as definições da configuração e monitoração. O administrador de sistemas fornece estas definições. As informações contidas na Definição são usadas pelo Controle para configurar as ferramentas de monitoração. Estas informações também podem ser usadas pelo subsistema Análise e Diagnóstico para comparar as informações coletadas da rede com os parâmetros de monitoração armazenados na definição;
- Visualização: é o subsistema que fornece aos usuários e aos administradores de sistema as informações sobre o comportamento e a configuração real da rede e os resultados das análises realizadas pelo subsistema Análise e Diagnóstico;
- Análise e Diagnóstico: é o subsistema que usa informações coletadas da rede e parâmetros de monitoração, regras e ações fornecidas pela Definição para detectar problemas no comportamento da rede. Este subsistema pode decidir por três ações: diretamente ativar a Atuação (do subsistema Coleta e Atuação) para corrigir problemas, ativar alarmes para alertar o administrador de sistemas ou simplesmente armazenar a informação para que esta seja visualizada posteriormente; e
- Coleta e Atuação: é o subsistema que coleta informação na rede. As informações coletadas podem ser visualizadas pelos usuários e administradores de sistema através da Visualização e é usada pelo subsistema Análise e Diagnóstico para fornecer uma análise do comportamento do sistema. Este subsistema pode também agir na rede para modificar seu comportamento e configuração por uma ordem do controle.

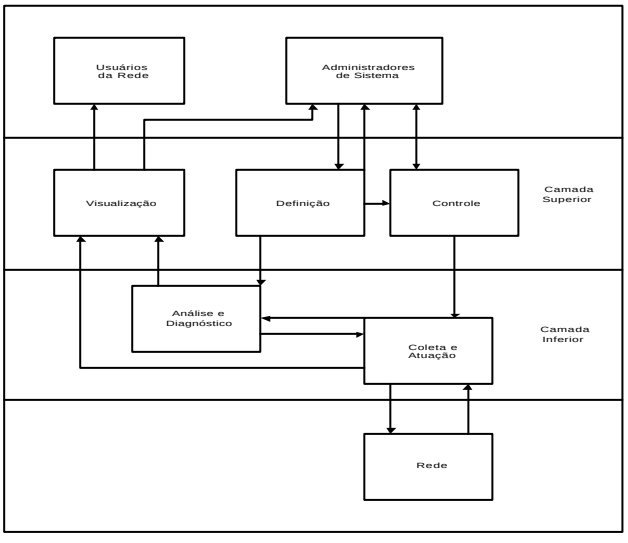


Figura 5.5: Modelo Conceitual I-DREAM

Através do modelo conceitual descrito anteriormente, foi demonstrado a funcionalidade do sistema. As características de implementação são apresentadas através da descrição do diagrama de arquitetura. A arquitetura do sistema (Figura 5.6) é composta dos seguintes módulos:

- Interface Usuário/Sistema (IUS): oferece a interface necessária para que haja a interação entre os usuários/administrador e os módulos SGBD, Caixa de Ferramentas e Gerente. É através da interação desta interface com o Gerente que se ativa e desativa as ferramentas de monitoração;
- Gerente: controla a monitoração dos componentes da rede. Esta monitoração é feita utilizando as ferramentas disponíveis na Caixa de Ferramentas. Estas ferramentas são utilizadas na rede por agentes de monitoração do módulo Sociedade de Agentes;
- Caixa de Ferramentas: mantém as ferramentas de monitoração que estão disponíveis no I-DREAM, ou seja, contém as ferramentas necessárias para fazer as monitorações;
- Sociedade de agentes (SA): é o módulo que concentra a *inteligência* do sistema. A SA é responsável por implementar a funcionalidade dos subsistemas Coleta e Atuação e Análise e Diagnóstico descritos no modelo conceitual.
- SGBD: mantém a definição das configurações, parâmetros de monitoração e várias outras informações sobre a rede e seu comportamento esperado, além de armazenar as regras definidas pelo administrador que serão usadas pelo Agente de Raciocínio.

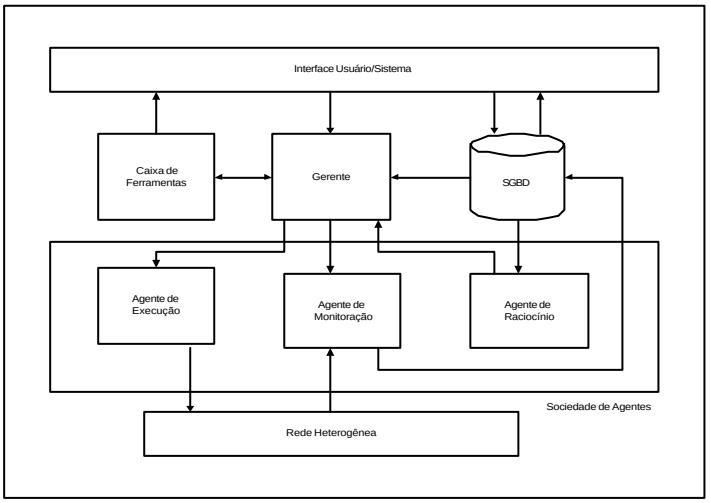


Figura 5.6: Arquitetura do I-Dream

Neste sistema, vemos incorporados inúmeros conceitos necessários para tratar o nível de complexidade que hoje está presente nas redes de computadores. Entretanto, ele foi concebido para tratar especificamente os problemas de administração não se atendo a resolver os problemas de gerência de redes de computadores.

Capítulo 6

Metodologia SAGRES

Neste capítulo é apresentada a metodologia SAGRES. Esta metodologia constitui-se em uma proposta para tratar do gerenciamento de falhas em redes de computadores. A figura 6.1 mostra o modelo de concepção do SAGRES – SGRBC (Figura 5.1) adicionado do refinamento do elemento Arquitetura Funcional.

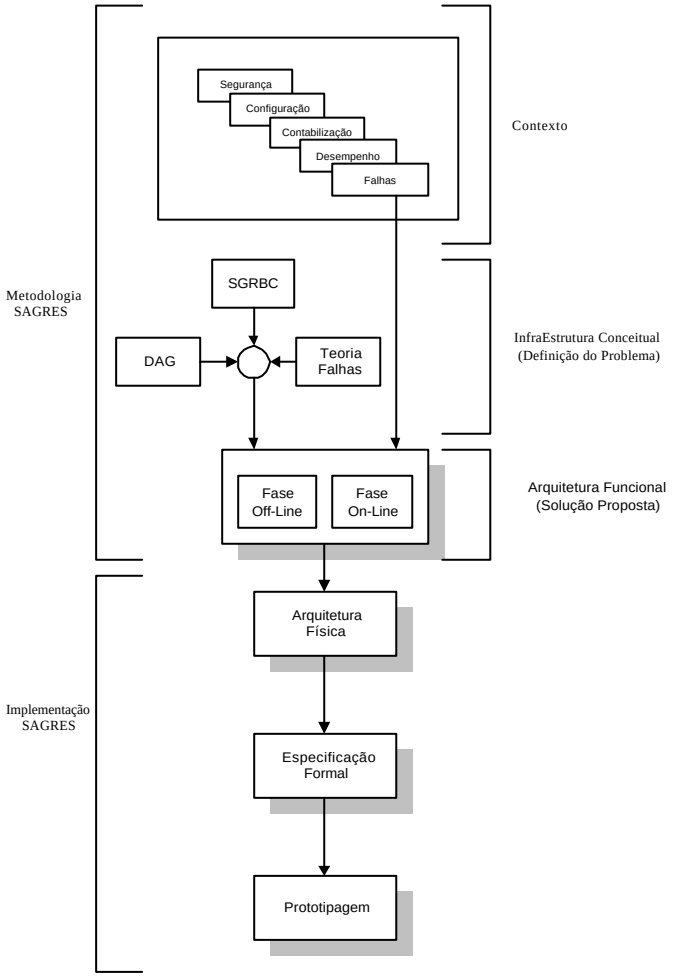


Figura 6.1: Modelo de Concepção do SAGRES – Arquitetura Funcional

6.1 SAGRES e Áreas Funcionais

Conforme visto no capítulo 2, as diversas atividades de gerenciamento de redes foram divididas pela OSI em cinco áreas funcionais de gerenciamento, a saber: falhas, desempenho, configuração, contabilização e segurança.

Dentre as cinco áreas funcionais, o SAGRES trata exclusivamente a gerência de falhas. A gerência de segurança não foi explorada nesta primeira versão do SAGRES devido ao nível de complexidade inerente a esta área. Gerências de desempenho, contabilização e configuração apresentam uma complexidade relativamente menor e podem ser tratadas por sistemas de gerenciamento de redes convencionais.

Portanto, pode-se dizer que a concepção da arquitetura do SAGRES é orientada a gerência de falhas [1]. Dois fatos foram determinantes nesta decisão, além do argumento acima:

- Falha é uma área funcional de destaque no que se refere a gerência de uma rede de computadores;
- Existe todo um formalismo no tratamento de falhas que facilita a concepção de uma arquitetura [Dav84].

6.2 SAGRES e a Infraestrutura Conceitual

O SAGRES é um sistema baseado em conhecimento para apoio à gerência de falhas em redes de computadores. A arquitetura funcional do SAGRES foi derivada a partir de sua infraestrutura conceitual.

O SAGRES faz uso da metodologia DAG aplicada à gerência de falhas e dos conceitos de SGRBC's para definir sua arquitetura funcional.

6.2.1 SAGRES e a Metodologia DAG

O SAGRES é um sistema baseado em conhecimento que tem como objetivo apoiar a atividade de gerência de falhas em redes de computadores. Na concepção de sua arquitetura funcional foi utilizado a metodologia DAG (Capítulo 4) devido a duas razões. Primeiro por esta metodologia congregar um conjunto de atividades que facilita

[1] Além disso, as demais áreas funcionais poderão fazer uso desta arquitetura

a análise e o desenvolvimento de aplicações de gerenciamento de redes de computadores, em especial a gerência de falhas. Segundo porque esta metodologia foi desenvolvida no contexto de um grupo de pesquisadores (LAR), que tem por princípio privilegiar os esforços resultante a partir desse grupo.

Do modelo proposto pela metodologia DAG, o SAGRES privilegia a implementação do módulo "Disponibilização do Conhecimento de Gerenciamento", conforme mostrado na Figura 4.2. Este módulo detém o conhecimento ("expertise") de um profissional em gerenciamento de redes, tendo sido implementado no SAGRES por um Sistema Especialista.

A tabela Objetos de Gerenciamento do DAG (Tabela 4.1)**,** elaborada no módulo Caracterização dos Objetos Gerenciados, foi também utilizada, pois contribui para a formação do conhecimento.

6.2.2 SAGRES e os SGRBC's

Na definição do SAGRES, foram também empregados conceitos de SGRBC's (Capítulo 5), os quais são utilizados no sistema por intermédio da definição de um conjunto de regras armazenadas em uma base de conhecimento para análise dos objetos a serem gerenciados.

O SAGRES portanto, pode auxiliar na detecção de eventos críticos em uma rede, e fornecer possíveis soluções para os mesmos. Seu principal objetivo é reduzir o nível de experiência (*expertise*) exigida de administradores para que se possa diagnosticar e corrigir problemas de maneira rápida e confiável.

6.2.3 SAGRES e a Teoria de Falhas

No SAGRES a diagnose de falhas utilizada é baseada em heurística. A figura 6.2 representa esta diagnose aplicada ao SAGRES.

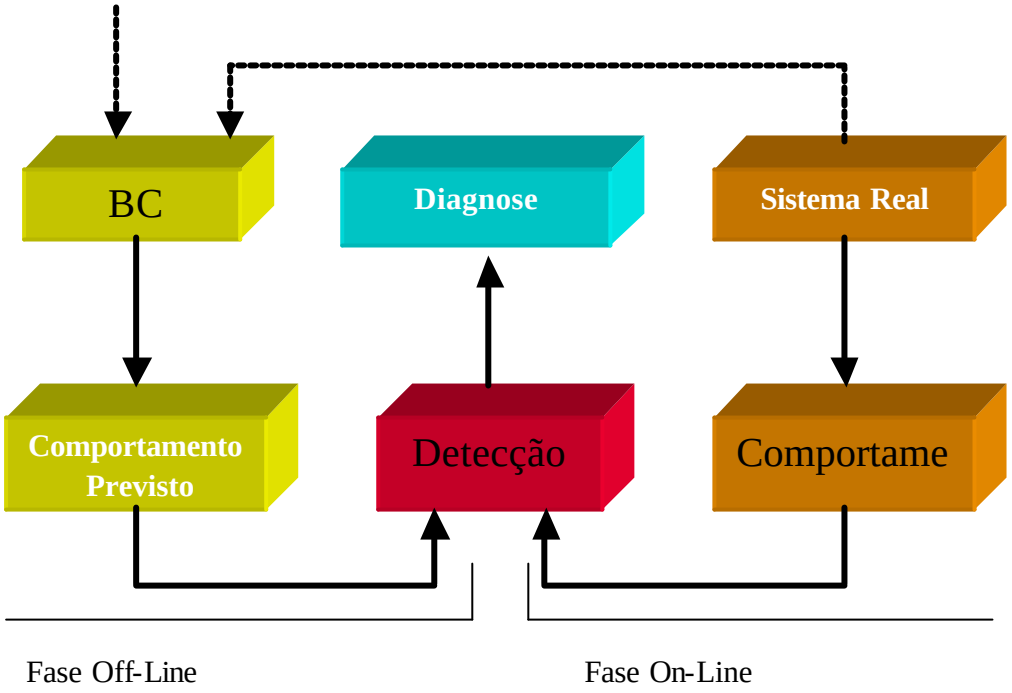


Figura 6.2: Diagnose Baseada em Heurística no SAGRES

Para realizar a detecção de falhas, é realizada uma comparação entre um comportamento esperado (normal) e um comportamento apresentado. Discrepâncias entre estes comportamentos indicam que o sistema está com problemas. Confirmada a discrepância, deve-se determinar as causas do problema ou diagnose.

Para tanto, deve-se inicialmente definir o comportamento esperado para a rede a ser gerenciada. A definição deste comportamento é realizado a partir dos seguintes elementos:

- observação do funcionamento real de uma determinada rede (*baseline*);
- obtenção de informações de especialistas em gerência de redes;
- consulta a bibliografia especializada.

O passo seguinte consiste na determinação de falha ou discrepância de comportamento na rede. Isto é realizado através da comparação entre o comportamento esperado e a observação do comportamento atual da rede. Encontrada alguma falha, deve-se determinar a diagnose.

6.3 Arquitetura Funcional

A arquitetura funcional do SAGRES utiliza os conceitos dos elementos que constituem sua infraestrutura conceitual (Teoria de Falhas, Metodologia DAG e SGRBCs).

A partir destes elementos, uma arquitetura funcional foi proposta, visando solucionar o problema de gerência de falhas em redes de computadores.

A arquitetura funcional é mostrada utilizando diagramas DFD (Diagramas de Fluxos de Dados) [Apêndice D] para detalhar o funcionamento do sistema SAGRES. Seu funcionamento é caracterizado pela existência de duas fases: Off-Line e On-Line. Essas fases são de natureza obrigatória e devem ser executadas de forma sequencial, nesta ordem.

6.3.1 Fase Off-Line

O SAGRES utiliza um conjunto de regras para auxiliar o administrador da rede a detectar possíveis problemas. Porém, para dispor dessas regras, algumas atividades devem ser efetuadas numa fase anterior (Fase Off-Line) à operação do sistema propriamente dito (Fase On-Line). Estas atividades são bem representadas por três processos do sistema (Figura 6.3):

- levantamento e seleção de objetos da MIB;
- geração do *baseline*; e
- construção de regras.

Processo 1: Levantamento e seleção de objetos da MIB

O funcionamento do SAGRES na Fase Off-Line (Figura 6.3) consiste, inicialmente, em realizar um levantamento dos objetos disponíveis na MIB que representam os elementos de rede a serem gerenciados.

De posse deste levantamento, o administrador da rede seleciona que objetos deseja monitorar. Esta seleção é de fundamental importância, pois ajudará posteriormente na definição das regras do sistema. Conforme definido na

metodologia DAG (capítulo 4) deve-se elaborar a tabela "Objetos de Gerenciamento" tendo-se para cada objeto uma descrição sucinta e a classificação quanto aos critérios comportamental, modificação e funcional.

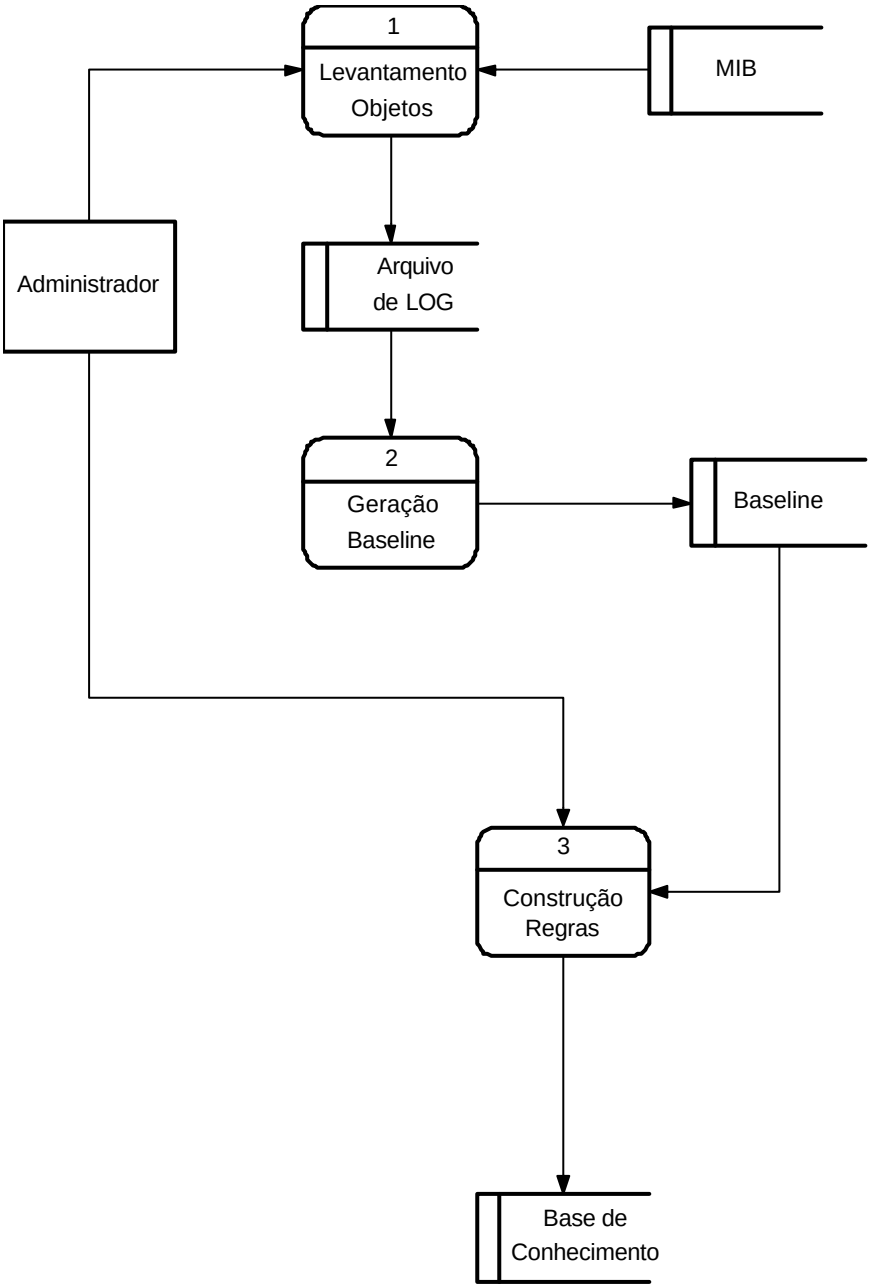


Figura 6.3: DFD Fase Off-Line

Selecionados os objetos, inicia-se o processo de coleta de seus valores durante um intervalo de tempo. Nesta etapa, o sistema efetua consecutivas ações *GET* sobre os Agentes dos elementos de rede, com prazos e intervalos definidos, gerando um arquivo de *log* conforme mostrado na Figura 6.3.

Assim, o processo "Levantamento de Objetos" (Processo 1) engloba tanto a atividade de levantamento de objetos propriamente dita quanto a seleção de objetos e a geração do arquivo de Log.

Processo 2: Geração do Baseline

Após a geração do arquivo de Log, o administrador deve estabelecer o *baseline* da rede a ser gerenciada, ou seja, para cada objeto monitorado qual é o valor ou faixa de valores que deverá ser considerado dentro da normalidade. Este processo é representado pelo processo "Geração do Baseline".

Processo 3 Construção de Regras

O passo final da fase Off-Line do SAGRES é a execução do processo "Construção de Regras". Este processo corresponde na Figura 6.2 às linhas tracejadas direcionadas para a base de conhecimento e provenientes do Sistema Real e do conhecimento de especialistas. O primeiro conjunto de regras do módulo "Construção de Regras" é obtido a partir da observação do comportamento da rede. Este comportamento é expressado por um baseline. As regras geradas a partir das informações deste baseline são consideradas triviais por envolverem pouco conhecimento, permitindo no entanto ao sistema entrar em operação.

Um segundo conjunto de regras é obtido a partir de consulta à bibliografia especializada em gerência de redes ou a partir de informações fornecidas por especialistas. Estas regras são definidas como regras heurísticas e em algumas situações podem vir a conflitar com o conjunto de regras definido a partir do baseline da rede.

Para evitar estes conflitos, as regras podem ser cadastradas de maneira a oferecer diagnósticos baseados no baseline separados dos diagnósticos gerados a partir das regras heurísticas.

A fase Off-Line termina com a montagem de uma Base de Conhecimento resultado da execução do módulo "Construção de Regras". Por intermédio das regras cadastradas na base de conhecimento, é possível ao sistema determinar o comportamento previsto para a rede, conforme definido na Figura 6.2.

6.3.2 Fase On-Line

A construção da base de conhecimento é condição para o sistema entrar em operação. Somente desta forma, o SAGRES poderá identificar a ocorrência de fatores anormais presentes na rede.

A Figura 6.4 apresenta o diagrama de fluxo de dados do SAGRES em sua fase On-Line. A interação do administrador da rede com o sistema, ocorre no decorrer de diversos processos nesta fase.

As atividades desta fase podem ser agrupadas em três processos:

- Coleta de dados dos objetos selecionados;
- Inferência; e
- Alteração parâmetros da MIB.

Processo 1: Coleta de dados dos objetos selecionados

Inicialmente o processo "Coleta de dados dos objetos selecionados" extrai os dados referentes ao estado atual da rede. Para tanto ele envia comandos *GET* aos Agentes do elemento de rede gerenciado. Este Processo recebe do administrador da rede os seguintes parâmetros: endereço IP do elemento de rede monitorado, relação de objetos, período de monitoramento e intervalo de monitoramento.

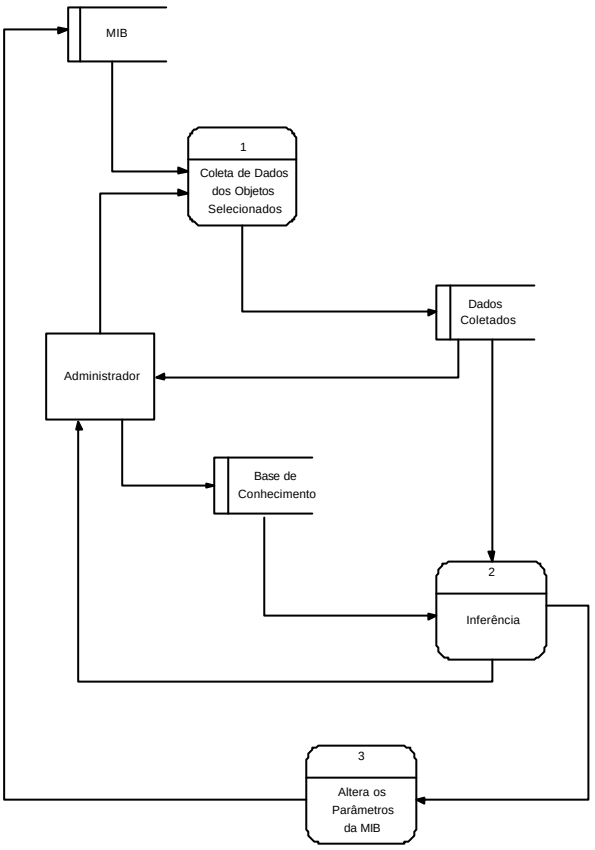


Figura 6.4: DFD Fase On-Line

Os dados coletados são armazenados em uma base de dados denominada “Dados Coletados” e servirá de entrada para o processo “Inferência”. Este processo corresponde ao estabelecimento do comportamento observado da Figura 6.2

Processo 2: Inferência

O processo Inferência é responsável pelas atividades de análise e diagnóstico do estado da rede. Para realizar estas atividades, são utilizadas as regras contidas na Base de Conhecimento e as informações sobre os objetos coletados da rede que ficam armazenadas na base Dados Coletados. O processo inferência é executado por um sistema especialista que, após diagnosticar a existência de algum problema na rede, deve encaminhar as correções. As funcionalidades implementadas aqui por um sistema especialista, correspondem ao bloco detecção de discrepâncias da Figura 6.2.

Processo 3: Alteração parâmetros da MIB.

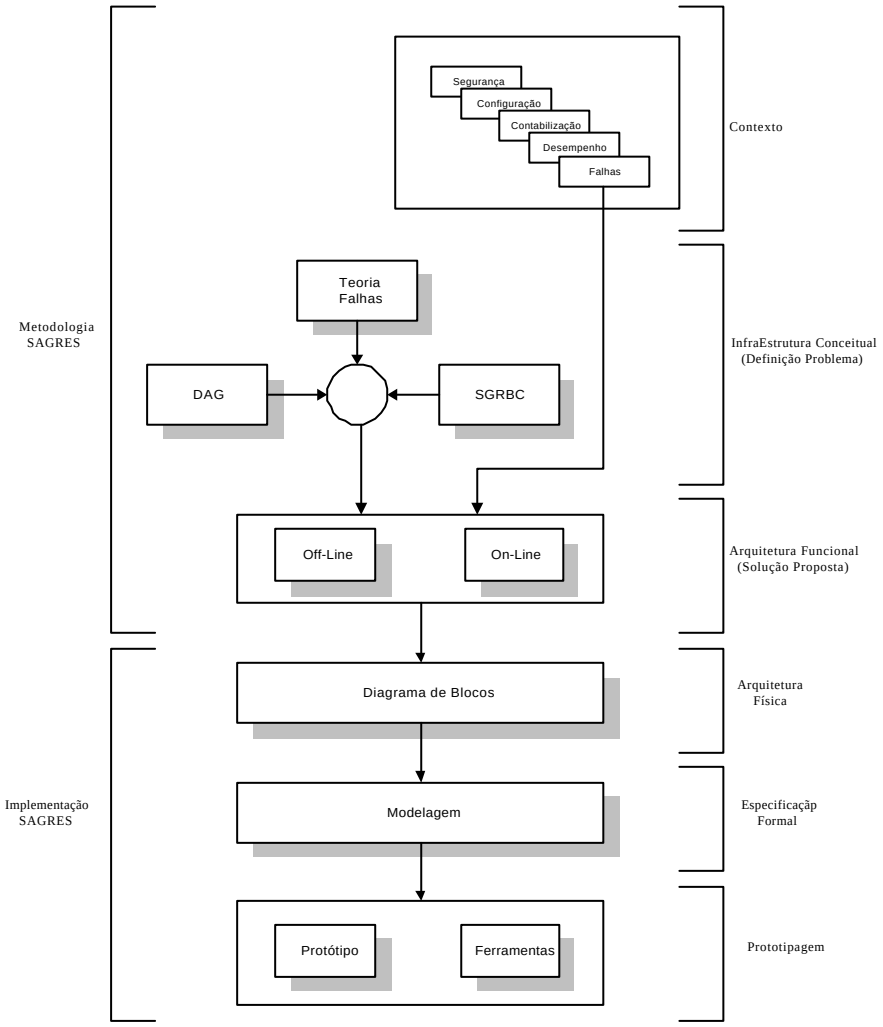
Os procedimentos de correções do SAGRES ocorrem de duas maneiras. Na primeira, o sistema simplesmente interage com o administrador expondo-o a origem do problema e sugere as atitudes que devem ser tomadas. Na segunda, o SAGRES além de interagir com o administrador, pode disparar automaticamente o processo 3 (Altera os Parâmetros da MIB) para que sejam feitas as devidas correções. O processo 3 consiste na execução do comando SNMP *SET*. A execução deste processo é consequência da determinação da diagnose conforme descrito na Figura 6.2.

Capítulo 7

Implementação do SAGRES

Este capítulo trata da implementação do SAGRES. A figura 7.1 mostra o modelo de concepção do SAGRES – Arquitetura Funcional (Figura 6.1) adicionado do detalhamento dos seguintes elementos: arquitetura física, especificação formal e prototipagem.

Figura 7.1: Modelo de Concepção SAGRES - Implementação



7.1 Arquitetura Física

A arquitetura física do SAGRES (Figura 7.2) integra as duas fases (Off-Line e On-Line) descritas na arquitetura funcional. O SAGRES possibilita a coleta, a análise e a geração de ações relacionadas ao comportamento da rede.

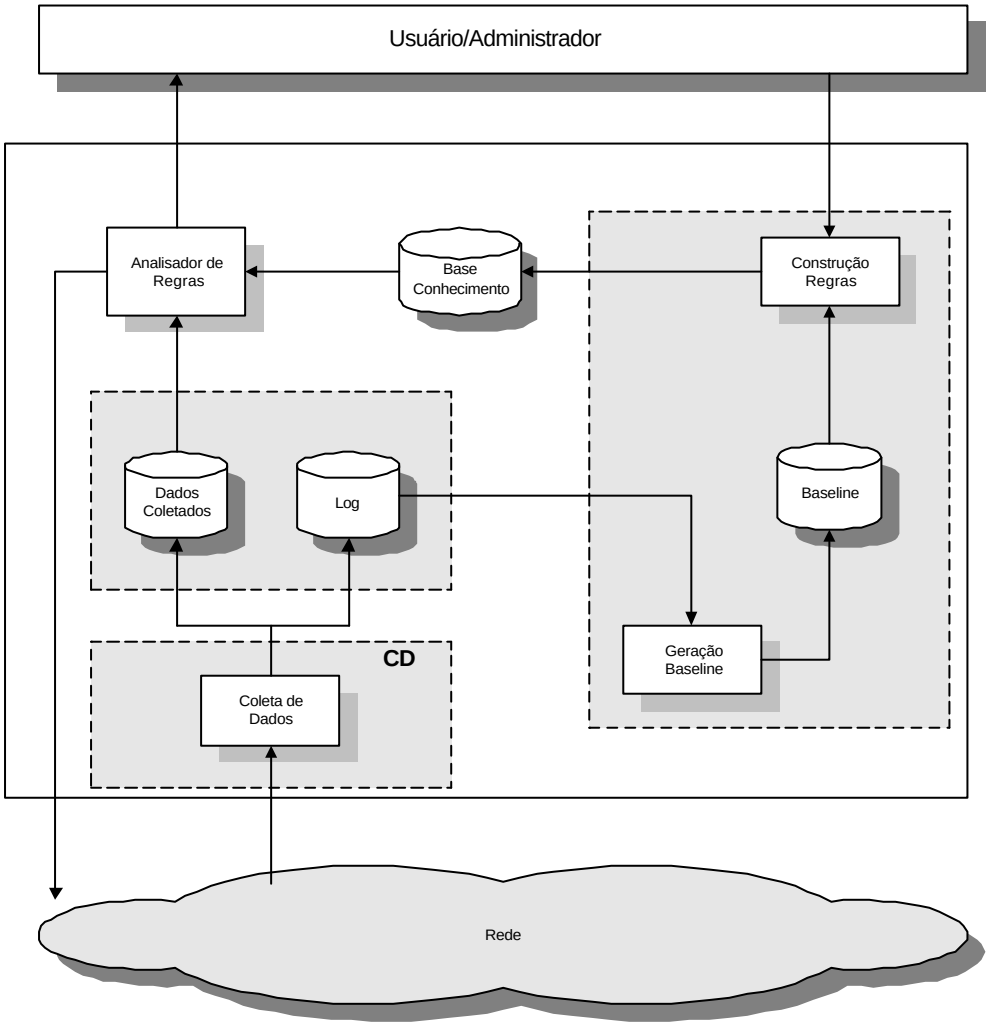


Figura 7.2: Arquitetura Física do SAGRES

O módulo "Coleta de Dados" da arquitetura, coleta informações sobre os componentes da rede. Estas informações são extraídas da MIB de um determinado elemento de rede gerenciado, consistindo no envio de comandos *GET* aos

agentes SNMP deste elemento. As informações coletadas são utilizadas para a geração do *baseline* da rede durante a fase Off-Line e abastecer o processo 2 (“Inferência”) da fase On-Line.

As informações geradas pelo módulo “Coleta de Dados” tem duas finalidades, tendo conseqüentemente informações sendo armazenadas em bases distintas. Essas bases de dados são o arquivo *Log* e arquivo “Dados Coletados”. O arquivo de *Log* gerado, irá subsidiar a construção de regras através da montagem do *baseline* durante a fase Off-Line enquanto os dados do arquivo “Dados Coletados” servirão para avaliar momentaneamente o estado da rede durante a fase On-line do sistema.

O módulo “Construção Regras” da arquitetura é responsável pela geração de regras na base de conhecimento. Este módulo corresponde ao processo de mesmo nome descrito na Fase Off-Line. As informações necessárias à construção destas regras são provenientes de duas fontes. A primeira deve-se à elaboração de um *baseline* da rede, extraindo-se os valores a serem considerados normais para cada objeto a ser gerenciado. A segunda fonte diz respeito à extração do conhecimento de um perito, podendo este perito ser o próprio administrador da rede.

A Base de Conhecimento é o repositório de dados onde estão armazenados o conhecimento de um especialista em gerência de redes e os valores que traduzem o comportamento normal de uma rede. A estrutura base de conhecimento dependerá do tipo de conhecimento representado. Para uma situação específica em que se aplica conhecimento dedutivo, a base será, normalmente, composta por regras. A base de conhecimento é a estrutura de ligação entre as Fases Off-Line e On-Line.
As funcionalidades do módulo “Analisador de Regras” definido na arquitetura do sistema correspondem às funcionalidades do processo “Inferência” (Processo 2) da fase On-Line. Esse módulo acessa as bases “Dados Coletados” e “Base de Conhecimento”. O “Analisador de Regras” consiste basicamente em uma máquina de inferência, tendo portanto como função selecionar e aplicar a regra apropriada, manipulando a Base de Conhecimento.

Uma máquina de inferência pode partir de premissas ou peças elementares de informação para atingir seu objetivo através da combinação delas. Nesse caso, diz-se que realiza caminhamento para a frente, ou pode partir de um objetivo e verificar as premissas necessárias dos fatos envolvidos chegando a uma conclusão, assim neste diz-se

que realiza caminhamento para trás. O módulo de inferência é utilizado para detectar problemas na rede, identificar a origem do problema e fornecer uma solução.

Como resultado da execução do módulo "Analisador de Regras" (Figura 7.2), temos uma interação do sistema com o administrador da rede, e/ou uma interação com a MIB do elemento de rede gerenciado. Estas ações estão representadas pelos processos 2 (Inferência) e 3 (Alteração Parâmetros da MIB) da fase On-Line, conforme definição da arquitetura funcional do SAGRES.

7.2 Modelagem

Foi realizada uma modelagem baseada em objetos para especificação, visualização e documentação do sistema. A linguagem utilizada foi a UML (*Unified Modeling Language*) [Apêndice A].

A ferramenta escolhida para realizar a modelagem foi a *Rational Rose*. As razões que justificaram a escolha foi o fato de esta ferramenta possuir um ambiente visual, apresentar inúmeras funcionalidades, além de ser dominante entre os produtos de análise e projeto orientado a objeto. A *Rational Rose* segue o padrão criado para linguagem de modelagem visual que é o UML (Unified Modeling Language). Sua principal característica é a existência do *Rose Extensibility Interface* (REI), uma potente e flexível interface para estender a funcionalidade do *Rational Rose*.

Devido ao enorme grau de detalhamento que é gerada nas visões da arquitetura proposta pela UML, apresentamos apenas a **visão lógica.** Nesta visão, a documentação dos *package* (pacotes), o modelo de classes e o modelo dinâmico (relacionamento entre as classes) são mostrados.

O Ambiente Sagres, conforme Figura 7.3 está divido em três camadas básicas (*packages*):

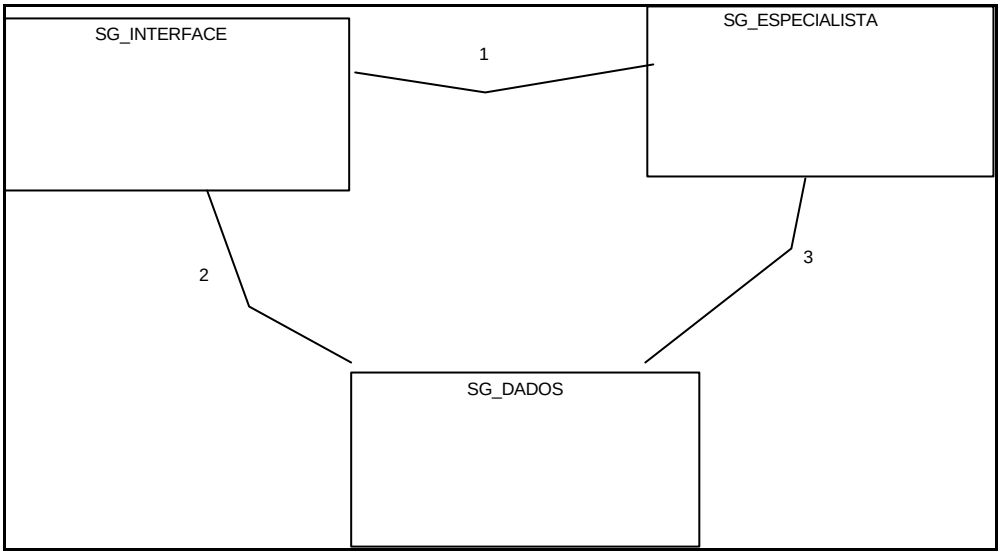


Figura 7.3: Modelagem - Pacotes

- SG-INTERFACE: pacote responsável pelas funcionalidades existentes entre a ferramenta e o usuário, ou seja, todo o tratamento da interface visual entre a ferramenta e o usuário;
- SG-ESPECIALISTA: pacote responsável pelo tratamento do conhecimento em gerência de redes. Suas funcionalidades estão relacionadas ao processamento inteligente que a ferramenta realiza; e
- SG-DADOS: pacote que disponibiliza as informações da rede, e funcionalidades em relação aos dados de gerência.

O pacote SG-INTERFACE representado graficamente na Figura 7.4 tem as seguintes classes:

- SG-I_REGRAS: Esta classe faz o tratamento sobre coleção de regras que está disponibilizada na classe SG_BASE_CONHECIMENTO, e tem associado os seguintes métodos*: RegistraRegra, AlteraRegra, ExcluiRegra, VisualizaTodasRegras, VisualizaUnicaRegra*;
- SG_I_MONITORCOLETA: Esta classe configura os parâmetros que definem a forma de coleta dos dados da classe SG_D_DADOMIB. Ele apresenta o método *Configura* onde são definidos: a freqüência e o intervalo de monitoramento; e o endereço do elemento de rede a ser gerenciado;
- SG_I_DADOS: Esta classe genérica disponibiliza os dados retornados pela classe SG_D_TRATADOREDE. Esta classe apresenta o método *ApresentaValoresRede;*
- SG_I_DTEXTUAL: Esta classe apresenta os dados em forma textual;

- SG_I_GRAFICO: Esta classe apresenta os dados de forma gráfica; e
- SG_E_MOTORINFERÊNCIA: Esta classe apresenta o resultado da análise do sistema especialista.

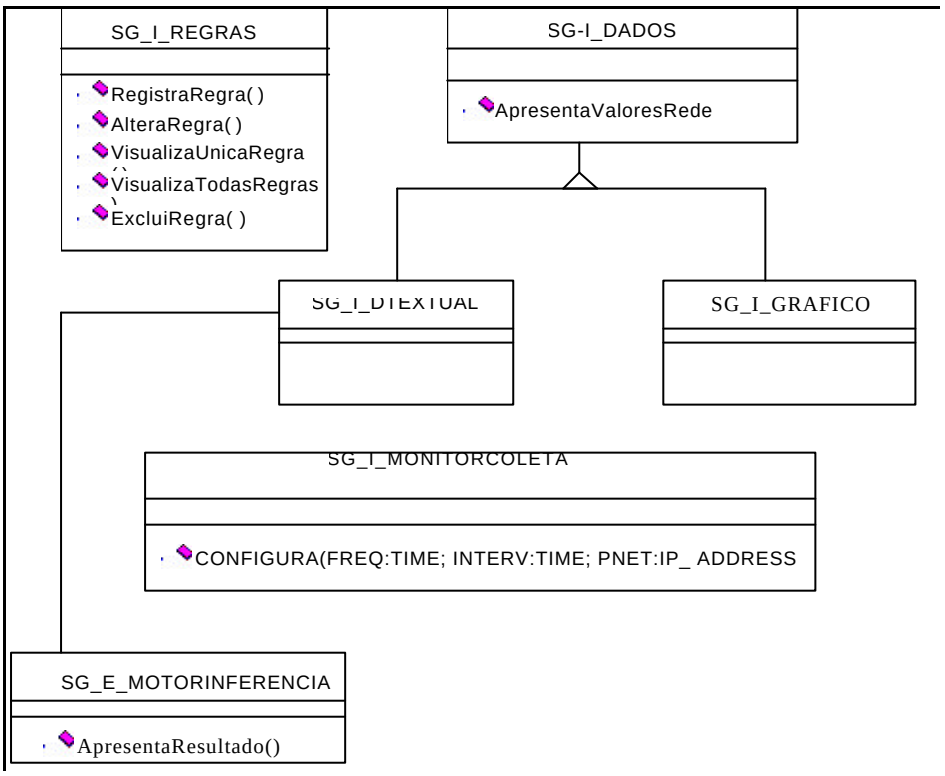


Figura 7.4: Pacote SG-Interface

O pacote SG_ESPECIALISTA tem as seguintes classes, conforme figura 7.5:

- SG_E_MOTORINFERÊNCIA: Esta classe realiza a inferência na classe SG_BASE_CONHECIMENTO, dada uma regra para a realização do processamento. Ela apresenta os seguintes métodos: *Iniciar, SelecionaRegra, AplicaRegra, ApresentaResultado;*
- SG_BASE_CONHECIMENTO: Esta classe contém as regras de comportamento do sistema, que são inicialmente fornecidas e que são disponibilizadas pela classe SG_E_CREGRAS;
- SG_E_CREGRAS: Esta classe contem uma coleção de regras. e tem associado os seguintes métodos: *InsereRegra, AlteraRegra, ExluiRegra, ApresentaRegra*; e

- SG_E_REGRAS: Esta classe disponibiliza as regras.

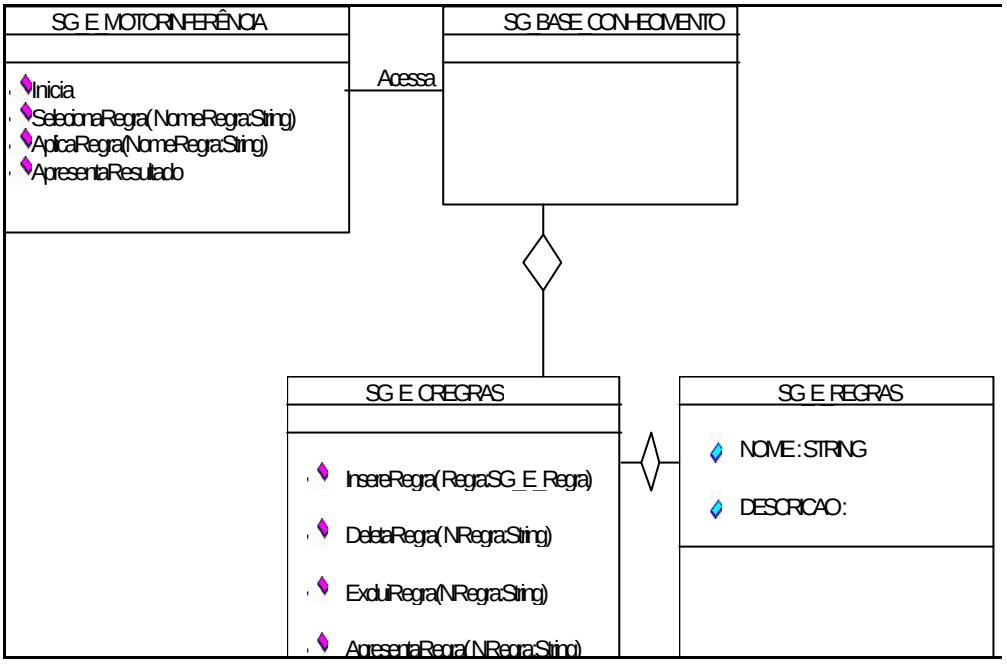


Figura 7.5: Pacote SG-Especialista

O pacote SG_DADOS tem as seguintes classes, conforme representado na figura 7.6:

- SG_D_DADOGENÉRICO: Esta classe genérica realiza as funcionalidades de um objeto de dados;
- SG_D_DADOMIB: Esta classe representa os dados contidos na MIB dos elementos de rede;
- SG_D_DADOCOLETAMIB: Esta classe realiza a tarefa de extração dos dados da Mib e armazena-os; e
- SG_D_DADOTRATADOREDE: Esta classe faz o tratamento dos dados retornados pela classe SG_D_COLETAMIB, realizando os cálculos estáticos necessários e disponibilizando os dados as demais classes do sistema.

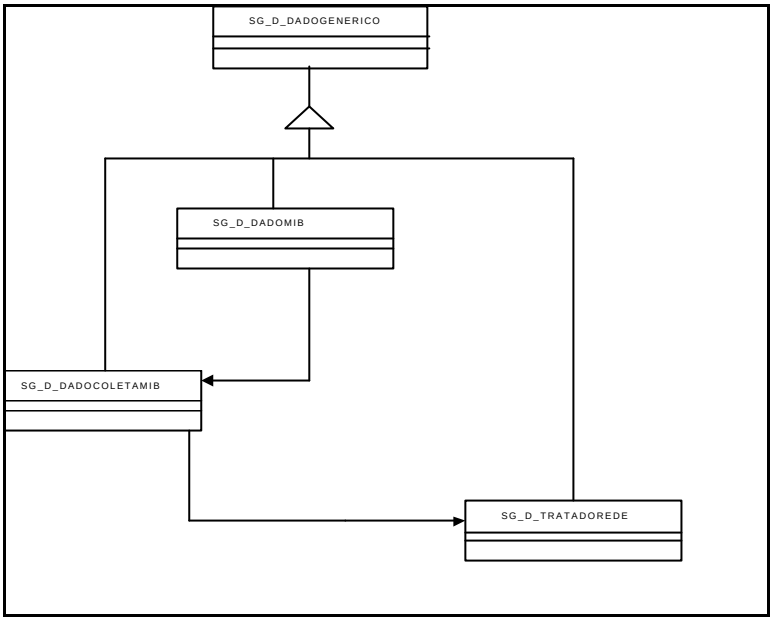


Figura 7.6: Pacote SG-Dados

O diagrama de eventos da Figura 7.7, representa o processamento realizado quando o SAGRES está verificando se a rede está com algum problema.

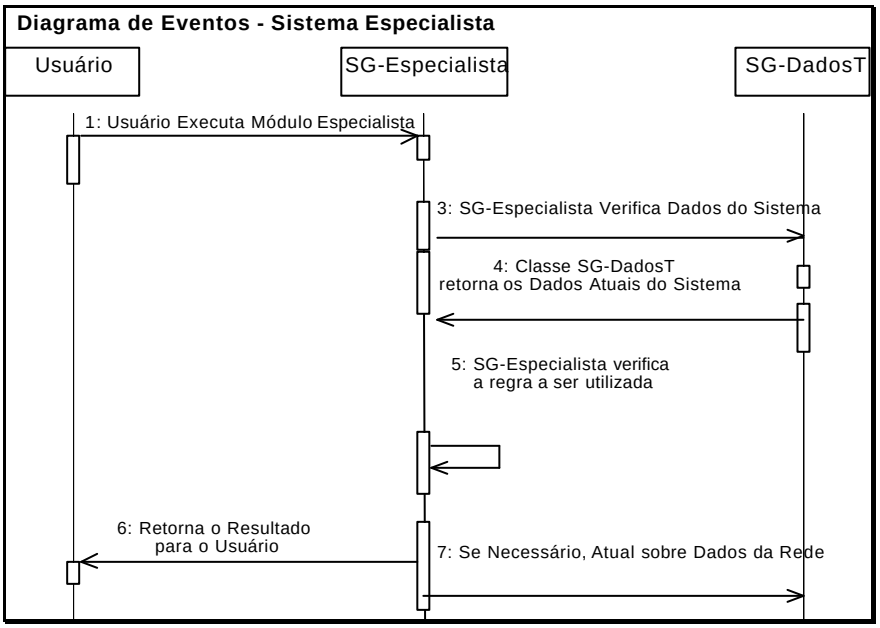


Figura 7.7: Diagrama de Eventos

7.3 Protótipo

7.3.1 Ambiente de Implementação

O sistema desenvolvido apresenta em sua primeira versão, uma heterogeneidade de ambientes, sendo parte da aplicação implementada em ambiente Unix, e a outra parte em ambiente PC. Os módulos do ambiente Unix são os referentes à coleta de dados e modificação dos parâmetros da rede. Em sua segunda versão, o sistema encontra-se totalmente desenvolvido para executar no ambiente PC.

7.3.1.1 Linguagem Delphi

Nos módulos da aplicação que são executados no ambiente PC foi utilizado o Delphi 3. O Delphi é uma ferramenta de desenvolvimento de alta performance voltada para a criação de aplicações visuais distribuídas utilizando a tecnologia Cliente-Servidor. Suas principais características são :

- compilador otimizado de 32 bits com geração de código nativo;
- permite criação de Dlls reutilizáveis;
- criação de arquivos executáveis (.EXE) independentes;
- suporte a Windows 95/NT;
- repositório de objetos;
- implementação de herança; e
- disponibilização de uma biblioteca de componentes visuais (VCL).

7.3.1.2 Tcl/Tk

Os módulos do ambiente Unix, utilizam a linguagem Tcl/Tk juntamente com o Scotty. O Tcl (*Tool Command Language*) foi desenvolvido por John K. Ousterhout e estudantes da Universidade de Berkeley, na Califórnia,

motivados pela criação de uma linguagem que facilitasse o desenvolvimento de aplicações e permitisse a reutilização de funções já desenvolvidas.

O Tcl é uma linguagem interpretada, baseada em *scripts*, que possui uma série de facilidades que auxiliam na programação. Seu interpretador é formado por um conjunto de procedimentos em linguagem C que podem ser facilmente incorporados às aplicações. Suas aplicações também podem ser usadas, como uma biblioteca de funções, por outras aplicações. Já o Tk é uma extensão do Tcl que fornece funções para geração de interfaces gráficas para interação com o usuário.

Os fatores decisivos para a escolha do Tcl/Tk foram: a existência de uma API (*Application Program Interface*) voltada para gerência SNMP; uma vasta biblioteca de funções disponíveis na Internet; e a possibilidade de uma implementação portátil, independente da plataforma, de maneira a tornar o sistema aberto. O sistema foi desenvolvido com as versões 7.6 do Tcl, 4.2 do Tk.

Maiores detalhes sobre o Tcl/Tk, assim como seu FAQ (*Frequently Asked Questions*) e as várias versões do Tcl e do Tk podem ser encontrados em vários *sites* espalhados na Internet, nos seguintes endereços :

http://www.sunlabs.com:80/research/tcl/
http://www.sunlabs.com:80/research/tk/
ftp://sunsite.doc.ic.ac.uk/packages/tcl/
http://www.sco.com/Technology/tcl/Tcl.html
ftp://ftp.uu.net/languages/tcl/
ftp://ftp.smli.com/pub/tcl/
ftp://ftp.aud.alcatel.com/tcl/

7.3.1.3 Scotty

O Scotty é uma extensão para a linguagem Tcl, desenvolvida pelo Departamento de Computação Científica da Universidade Técnica de Braunschweig, na Alemanha, para permitir a implementação de aplicações voltadas para a gerência de redes. Com esta extensão, é possível o uso de vários recursos das redes TCP/IP, como os seguintes :

- enviar e receber pacotes ICMP;

- consultar DNS (*Domain Name System*);
- acessar *sockets* UDP;
- recuperar ou servir documento HTTP;
- enviar e receber mensagens SNMP;
- desenvolver agentes SNMP;
- analisar e acessar as definições das MIB's; e
- agendar job's para serem executados periodicamente.

O conjunto de funções fornecidas pelo Scotty são muito úteis na construção de ferramentas para o gerenciamento em redes TCP/IP e permitem uma implementação fácil e rápida, além delas serem totalmente integradas com o ambiente do Tcl.

O Scotty foi divulgado no terceiro workshop sobre Tcl/Tk em 1995 e o paper apresentado nesse workshop, contendo mais detalhes sobre as ferramentas disponíveis e sobre o desenvolvimento destas está disponível para cópia em:

ftp://ftp.ibr.cs.tu-bs.de/pub/local/papers/tcltk-95.ps.gz

Existem também na Internet alguns sites onde é possível copiar todo o pacote de Scotty para instalação, páginas do manual, *papers* e apresentações sobre o Scotty, além de uma lista onde é possível reportar *bugs* ou tirar dúvidas de instalação e implementação. Essas informações podem ser encontradas nos seguintes endereços:

http://wwwsnmp.cs.utwente.nl/~schoenw/scotty/

http://wwwsnmp.cs.utwente.nl/~schoenw/ietf-nm/

http://www.cs.tu-be.de/ibr/projects/nm/welcome.html

ftp://ftp.ibr.cs.tu-bs.de/pub/local/papers/

As razões para escolha do Scotty foram várias. Dentre elas podemos destacar: a quantidade de funções que ele fornece para a gerência de redes; a fácil manipulação de objetos da MIB e uma interface simples.

7.3.1.4 Sistemas Especialistas e o Shell Expert Sinta

No sistema SAGRES, as atividades de análise e detecção de problemas na rede são realizadas por um sistema especialista. Sistemas Especialistas são programas 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. Esses recursos podem então ser utilizados sem a presença do(s) especialista(s). Todos os sistemas especialistas são projetados para solucionar problemas de um determinado domínio. Portanto, todo o conhecimento desses sistemas deve ser fornecido por especialistas nesse domínio. Um sistema especialista pode tratar dados inexatos, incompletos e complexos, podendo proporcionar explicações de suas conclusões, bem como aprender pela experiência.

Um dos elementos mais importantes num sistema especialista é o conhecimento. Portanto, é necessário saber como representar e adquirir as informações que irão fazer parte da base de conhecimentos desse sistema especialista.

A representação de conhecimento envolve decisões sobre como o conhecimento pode ser estruturado explicitamente, como pode ser manipulado para inferir os dados adicionais e como é feita a aquisição requerida pelo sistema. A escolha de uma representação do conhecimento para uma tarefa particular depende do tipo de problema a ser resolvido e da forma na qual o conhecimento pode ser mais facilmente especificado e utilizado. A aquisição de conhecimento requer duas ações básicas: primeiro, é necessário obter o conhecimento requerido inicialmente para criar a base de conhecimentos; depois, deve haver a manutenção e o aperfeiçoamento do conhecimento.

Um sistema especialista é formado por três componentes principais, descritos a seguir [Cro 88].
a) Base de conhecimento : A base de conhecimento é um tipo de armazenamento que contém todos os fatos, idéias, relacionamentos e interações de um domínio limitado. A informação mantida em uma base de conhecimentos é uma informação organizada detalhadamente em relação a uma evidente estratégia de processamento;

b) Máquina de Inferência : A máquina de inferência é a que analisa o conhecimento e deriva conclusões. As máquinas de inferência imitam o tipo de pensamento que um especialista humano utiliza quando tenta solucionar um problema; isso é, começa com uma hipótese e tenta encontrar evidências que apóiem essa hipótese, ou pode começar com as evidências disponíveis e tentar determinar que conclusões são deriváveis. Em sistemas especialistas, esses dois métodos são denominados **Encadeamento Reverso** (*backward Chaining*) e **Encadeamento para Adiante** (*Forward Chaining*) respectivamente. Uma inferência é uma conclusão baseada em

fatos ou premissas. Um motor de inferência é um programa que determina como aplicar o conhecimento contido em uma base de conhecimentos para atualizar fatos e premissas descritas em uma memória de trabalho, com o objetivo de inferir novos dados que poderão ser usados para futuras inferências. Em um sistema baseado em regras, o motor de inferência determina quais regras são aplicáveis e qual dessas regras candidatas poderia ser usada numa dada situação. O motor de inferência de um sistema especialista baseado em regras de produção implementa conceitos como a escolha de uma ação, a consideração de todas as possíveis alternativas e a decisão de qual alternativa é a melhor. Em sistemas de produção, o mecanismo de controle de inferência é um ciclo de reconhecimento de ações que possui três etapas:

- **comparação:** encontra todas as regras que satisfaçam o conteúdo atual da memória de trabalho; essas comparações são chamadas de conjunto de conflito;
- **seleção ou resolução de conflito:** determina quais dessas comparações no conjunto de conflito é a melhor para ser invocada;
- **invocação de regras (execução):** processo de aplicação das comparações especificadas pela regra escolhida; são ações que tipicamente trocam dados assim que novos padrões são formados; em alguns sistemas, as regras são somadas, removidas ou modificadas.

c) Interface ao usuário : A interface do sistema especialista permite que um novo conhecimento seja agregado ao sistema e implementa a comunicação com o usuário. A interface pode também apresentar o processo pelo qual a conclusão foi alcançada, permitindo ao usuário acompanhar a lógica envolvida. Os 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 isso sugere que sistemas baseados em regras deveriam ser o método escolhido para construir sistemas especialistas com vasto conhecimento [Hay 85]. Os sistemas especialistas estão classificados em quatro categorias: **Consultores** (sistemas orientadores baseados em diálogos), **Monitores** (sistemas orientadores baseados em monitoração), **Servidores** (sistemas controladores baseados em diálogos) e **Agentes** (sistemas controladores baseados em sensores). Os termos Controladores e Orientadores referem-se a uma outra subclassificação dos sistemas especialistas: **Controladores** são aqueles que dirigem suas saídas diretamente sobre o mundo físico na forma de sinais de controle a outros dispositivos e **Orientadores** são aqueles que oferecem suas saídas a uma audiência humana.

Para facilitar a implementação de sistemas especialistas, foram criadas ferramentas, *shells*, aptas a realizar muito do trabalho necessário para transpor um sistema especialista para um computador. Essas ferramentas permitem que o criador do sistema preocupe-se somente com a representação do conhecimento do especialista, deixando para o *shell* a tarefa de interpretar o conhecimento representado e executá-lo em uma máquina. Permite ainda que sejam realizadas depurações e explicações de como o computador chegou àquelas conclusões. O *shell* utilizado neste trabalho foi o Expert SINTA [LIA96]. Suas principais características são a utilização do encadeamento para trás, de fatores de confiança e serviços de depuração.

A seguir, ilustramos na Figura 7.8 a interface principal da shell Expert Sinta:

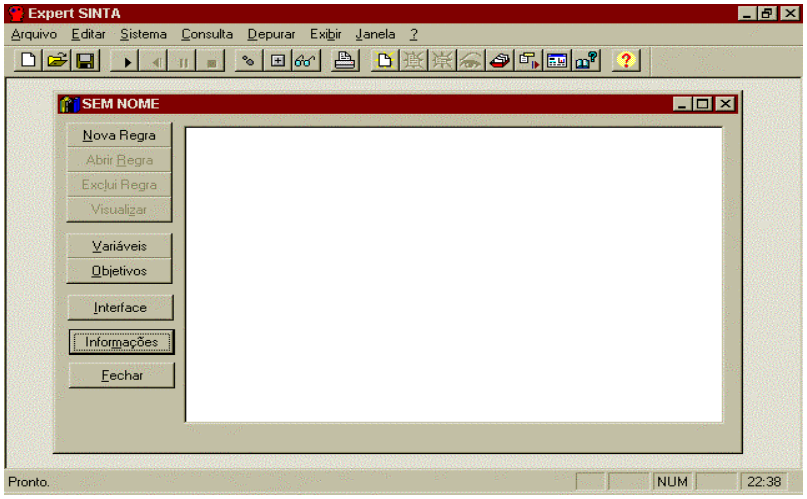


Figura 7.8: Interface Principal Expert Sinta

7.3.2 Implementação – Fase Off-line

O SAGRES foi construído com o objetivo de atender os seguintes requisitos:

- parametrização: quando da construção do *baseline*, por exemplo, o período e intervalo de coleta devem ser totalmente parametrizados, permitindo ao administrador definir os parâmetros de seu interesse;

- extensibilidade: o sistema após sua implementação deve permitir a incorporação de novas informações, principalmente regras, uma vez que não se sabe, a priori, todos os tipos de informações que serão manipulados;
- interfaces gráficas de alta qualidade: o sistema, uma vez implementado, deve possuir interfaces gráficas amigáveis voltadas para o usuário final.

Conforme o diagrama de fluxo de dados da Fase Off-Line do SAGRES (Figura 6.3) o primeiro módulo a ser executado é o "Levantamento dos Dados". Neste módulo, um arquivo de log é gerado a partir de informações da MIB do elemento de rede gerenciado. O sistema atualmente está preparado para coletar informações de todos os objetos da MIB2. Para tanto, foi construído um módulo para interagir com os agentes SNMP de um dispositivo qualquer. A ferramenta utilizada na primeira versão do sistema foi o Scotty juntamente com o Tcl/Tk em uma plataforma Unix. Para iniciar o processo de coleta, devem ser informados o endereço IP do dispositivo a ser gerenciado, o período de monitoramento e o intervalo de monitoramento. Como visto, a periodicidade da coleta é determinada pelo administrador, permitindo que ele defina os parâmetros de coleta que considerar mais confiável. Sua execução consiste de consecutivas ações "Get", armazenando estes dados em um arquivo de log conforme Figura 7.9.

Apresentamos a seguir um trecho do programa escrito em Scotty. Esta rotina coleta o número de pacotes descartados em uma interface.

```
#
# Variavel ifInDiscards
#
$s get ifInDiscards.2 {
  if {"%E" == "noError"} {
     set d [lindex [lindex "%V" 0] 2]
     set fileid [open /root/scotty-2.1.5/unix/logsc {RDWR APPEND}]
     puts $fileid "ifInDiscards"
     puts $fileid $d
     close $fileid
  }
}
```

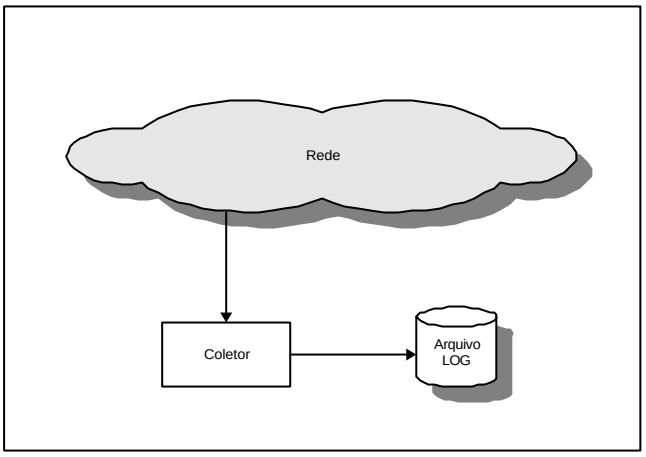


Figura 7.9: Representação Coletor de Dados

A modificação ocorrida neste módulo na segunda versão do SAGRES, foi a substituição do Scotty por bibliotecas Delphi SNMP. Estas bibliotecas foram desenvolvidas pela empresa Dart Communications. A seguir é mostrado um trecho do programa de coleta desenvolvido en Delphi e a interface com o usuário (Figura 7.10).

```
procedure TForm1.Bt_enviarClick(Sender: TObject);
var
   pp : pchar;
   len1 : word;
   tObjectID : array [0..1] of Longint;

begin
   len1 := length(Edit_variavel.text);
   GetMem(pp,Len1+1);
   strPCopy( pp,Edit_variavel.text);
   tObjectID[0] := longint(pp);
   SNMP_D1.SendGetRequest(Combo_IP.TEXT,161,'public',1,1,@tObjectID);
   FreeMem(pp,Len1+1);
```

end;

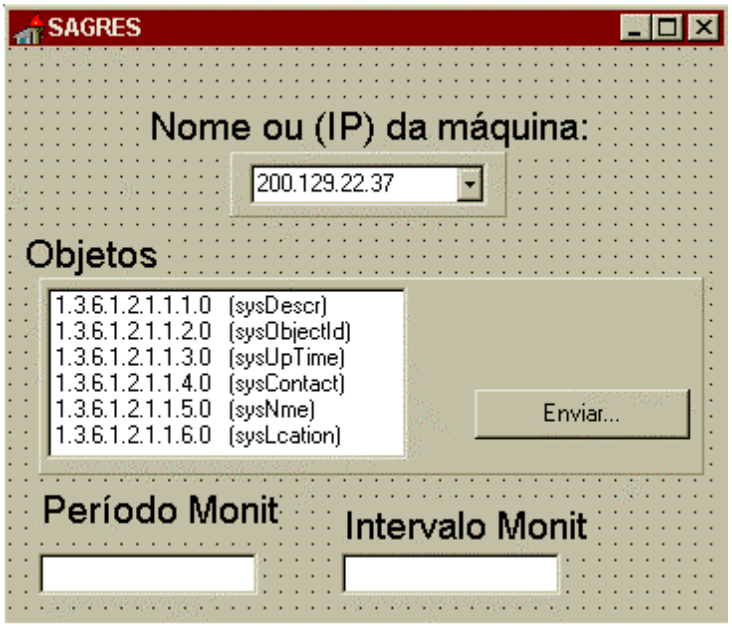


Figura 7.10: Interface Coletor de Dados

Uma vez disponível o arquivo de log, o administrador deve selecionar os objetos de seu interesse. As informações do arquivo de log devem ser tabulados e submetidos a um tratamento estatístico para calcular o comportamento de cada objeto. Este comportamento irá definir o *baseline*, e deverá ser utilizado na definição das primeiras regras do sistema.

O módulo "Construção de Regras" do DFD Fase Off-Line do SAGRES (Figura 6.3) consiste no cadastramento das regras em uma base de conhecimento. A Figura 7.11 mostra a interface utilizada para interagir com a base de conhecimento.

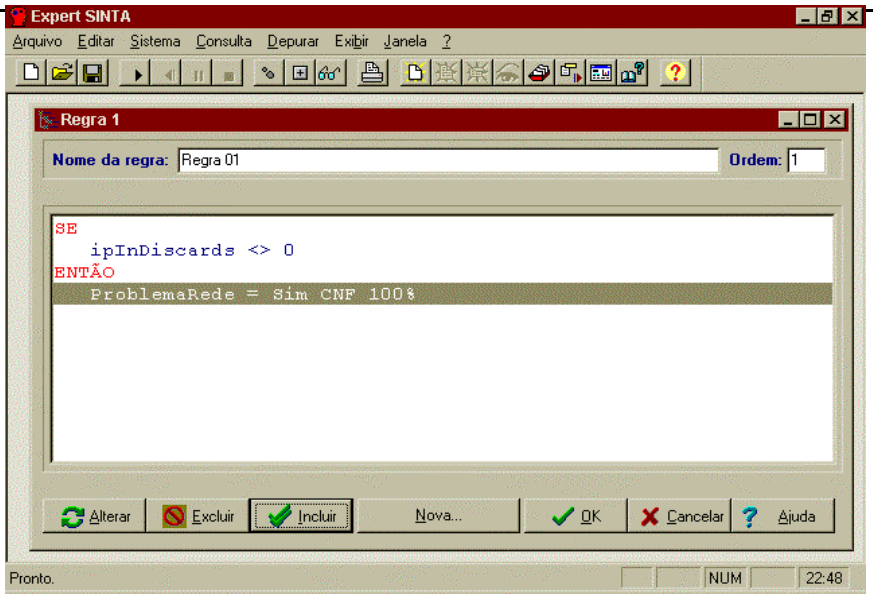


Figura 7.11: Interface Cadastro de Regras

Concluimos assim, a fase Off-Line do SAGRES, com a construção da base de conhecimento.

7.3.3 Implementação – Fase On-Line

O módulo implementado para realizar as funções do processo “Coleta de Dados dos Objetos Selecionados” da fase On-Line, é semelhante ao módulo “Levantamento Objetos” da fase Off-line. Porém, na fase On-Line, os valores coletados são utilizados para avaliar o estado da rede e não mais para construir um *baseline*. Este módulo foi implementado em Scotty e TCL/TK em uma máquina com sistema operacional Unix e atualmente está sendo modificado para Delphi.

Os dados, após serem coletados, passam por uma conversão de ambiente e de base de dados. Os dados coletados são armazenados inicialmente em formato texto no ambiente Unix e, após o tratamento, estes dados ficam disponíveis em PC numa base do tipo Paradox. A opção pelo Paradox decorreu de sua fácil disponibilidade além de oferecer um desempenho razoável nas aplicações desenvolvidas em Delphi. Após esta conversão, os dados podem abastecer o módulo “Inferência”.

A interface mostrada na Figura 7.12. disponibiliza para o administrador as informações coletadas.

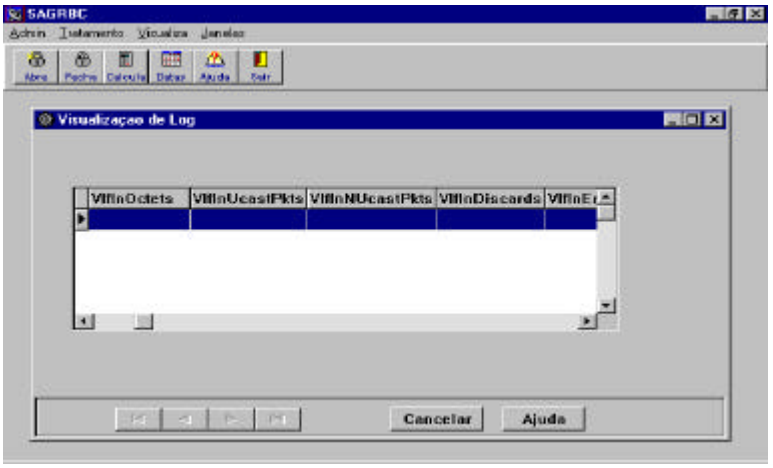


Figura 7.12: Interface Consulta Informações Coletadas

O administrador pode interagir com a base de conhecimento realizando inclusões, alterações e remoções de regras. Estas operações estão representadas na interface mostrada na Figura 7.13

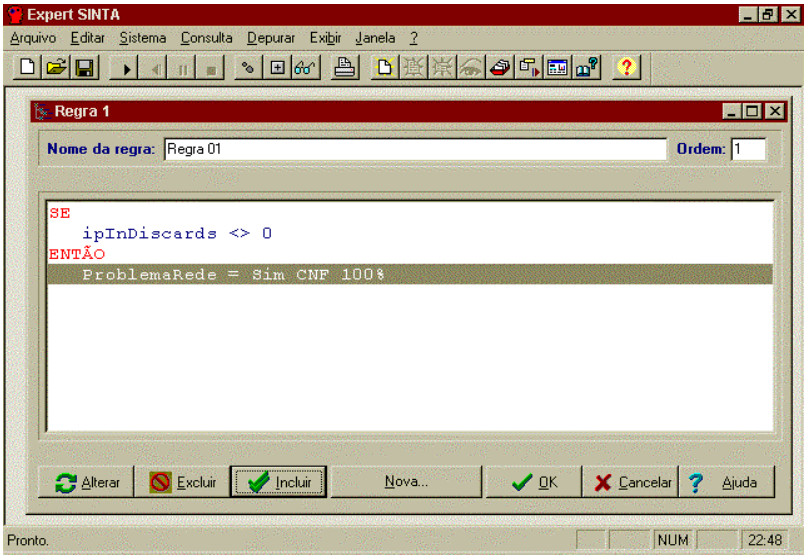


Figura 7.13: Interface Administração de Regras

As regras após cadastradas, podem ser visualizadas de forma individual ou coletiva, conforme mostra a Figura 7.14

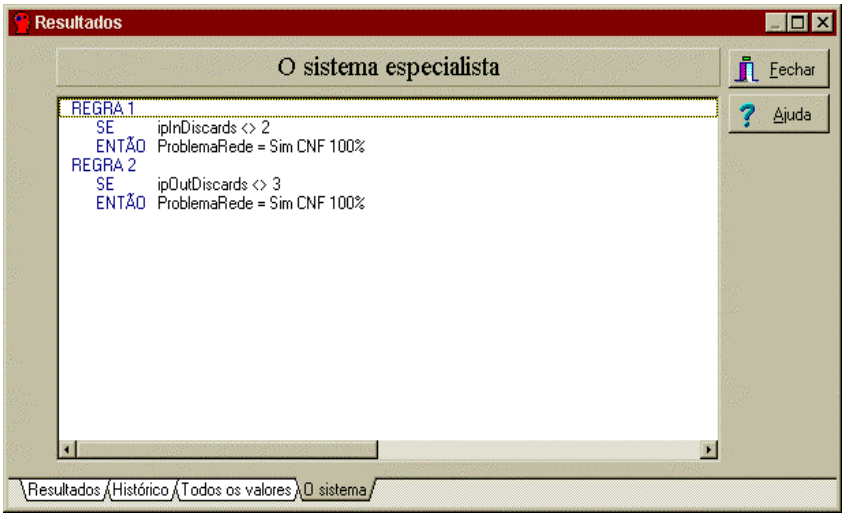


Figura 7.14: Interface Visualização de Regras

A implementação do módulo "Inferência" (Figura 7.15) utiliza as informações coletadas da rede e as regras cadastradas em uma base de conhecimento para detectar discrepâncias entre o comportamento observado e o comportamento previsto para a rede. Ao se confirmar a discrepância, o administrador deve ser informado da origem do problema e dos procedimentos para correção. Computacionalmente, este módulo representa o sistema especialista.

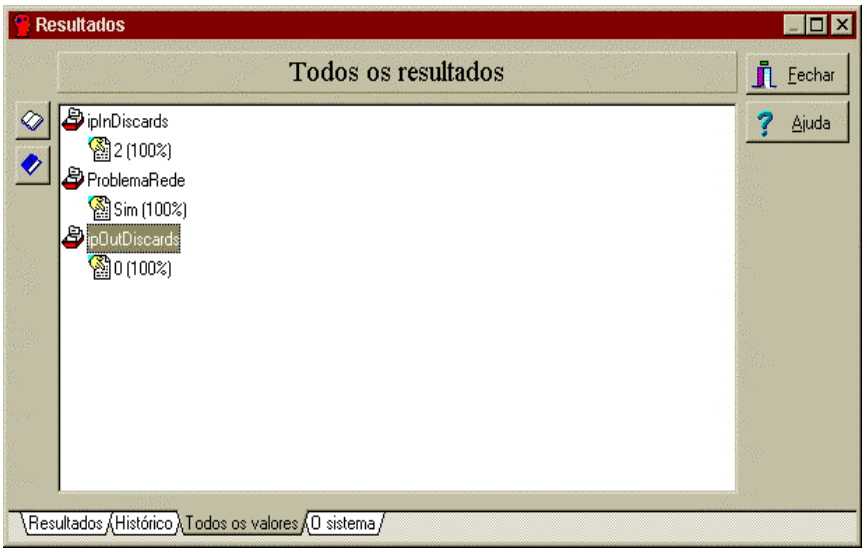


Figura 7.15: Interface Resultados da Inferência

O administrador faz uso da interface mostrada na Figura 7.16 para atuar sobre o elemento de rede gerenciado. Este módulo foi implementado em Delphi tendo um trecho se sua implementação mostrada a seguir:

```
procedure TSetValue.Button1Click(Sender: TObject);
var
   len1, len2 : Integer;
   pp, pp1, pp2 : pointer;
   hType : integer;
   ss : string;
begin
   len1 := length('1.3.6.1.2.1.1.1.0');
   GetMem(pp,Len1+1);
   hType := 4;
   if SetValue_item.Text = 'sysDescr' then
      strPCopy(pp,'1.3.6.1.2.1.1.1.0')
   else if SetValue_item.Text = 'sysObjectID' then
      strPCopy(pp,'1.3.6.1.2.1.1. 2.0')
   else if SetValue_item.Text = 'sysContact' then
      strPCopy(pp,'1.3.6.1.2.1.1.4.0')
   else if SetValue_item.Text = 'sysName' then
      strPCopy(pp,'1.3.6.1.2.1.1.5.0')
   else if SetValue_item.Text = 'sysLocation' then
      strPCopy( pp,'1.3.6.1.2.1.1.6.0');
   len2 := length(SetValue_editbox.text);
   GetMem(pp1,Len2+3);
   pp2 := pp1;
   word(pp1^) := len2;
   longint(pp2) := longint(pp1) + 2;
   StrPCopy(pp2,SetValue_editbox.text);
```

```
   tObjectID[0]    := longint(pp);
   tObjectType[0]  := hType;
   tObjectValue[0] := longint(pp1);
   Form1.SNMP_D1.SendSetRequest  (       form1.hosts.items[form1.Hosts.ItemIndex], 161, 'public', rim+1, 1,
@tObjectID, @tObjectValue, @tObjectType );
   freemem(pp,len1+1);
   freemem(pp1,len2+3);          { .tve. }
   Close;
end;
```

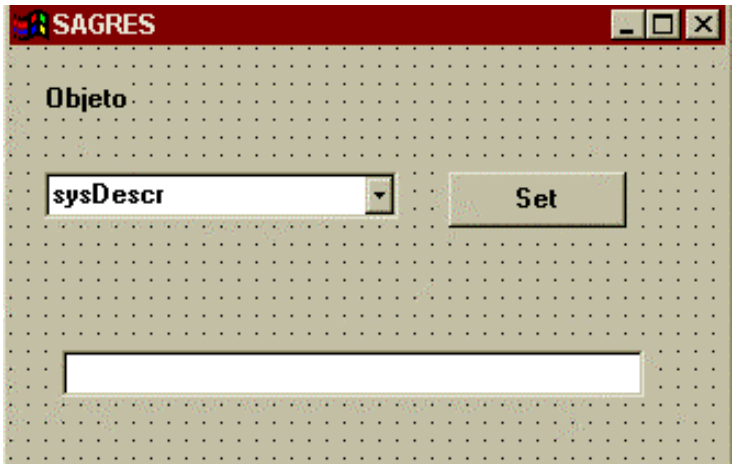


Figura 7.16: Interface Comando SET

O módulo “Altera Parâmetros da MIB” é disparado a partir do módulo “Inferência”, para automaticamente atuar sobre Agentes do elemento de rede. Este módulo foi inicialmente implement ado em Scotty, encontra-se atualmente disponível em Delphi. Foram utilizadas bibliotecas SNMP Delphi que implementam a operação SNMP “SET”.

7.4 Estudo de Caso

Um estudo de caso foi realizado utilizando o protótipo desenvolvido neste trabalho. Utilizamos para tanto, as instalações do laboratório do Mestrado em Ciência da Computação da UFC. O elemento de rede escolhido para ser gerenciado foi o roteador deste laboratório que detém o endereço IP 200.19.179.65.

A necessidade de gerenciamento a ser atendida neste estudo de caso está relacionada com a ocorrência de falhas em uma das interfaces do roteador.

A seguir, apresentamos a execução das fases Off-Line e On-Line ocorridas durante a realização do estudo de caso

7.4.1 Execução da Fase Off-Line

O primeiro módulo denominado "Levantamento Objetos", fez a coleta dos valores dos objetos da MIB do roteador para gerar um arquivo de log. O período de coleta foi realizado entre os dias 14 e 18 de setembro de 1997 nos horários descritos abaixo:

1a. Coleta : 8:00 hs
2a. Coleta : 9:00 hs
3a. Coleta : 10:00 hs
4a. Coleta : 11:00 hs
5a. Coleta : 14:00 hs
6a. Coleta : 15:00 hs
7a. Coleta : 16:00 hs
8a. Coleta : 17:00hs

As tabelas do Apêndice B, demonstram os valores coletados dos objetos. Essa coleta ficou armazenada em uma base denominada "Arquivo de Log".

Foram monitorados os seguintes objetos:

- IfOperStatus
- IfInDiscards

- IfOutDiscards
- IfInUnknownProtos
- IfInOctets
- IfInNUcastPkts
- IfOutOctets
- ifOutNUcastPkts
- IpInHdrErrors
- IpInAddrErrors
- IpReasmFails
- IpOutNoRoutes
- TcpAttemptFails
- TcpEstabResets
- TcpRetransSegs
- IcmpInErrors
- IcmpOutErrors
- UdpInErrors
- UdpinDatagrams
- UdpOutDatagrams

No módulo denominado Seleção dos Dados da Fase Off-Line, resolvemos utilizar todos os objetos monitorados. Após esta seleção, o *baseline* foi construído. Determinou-se assim, o comportamento esperado para cada objeto. Definimos que o *baseline* refletiria o comportamento de cada objeto para um intervalo de monitoramento de 1 (uma) hora. Para tanto determinou-se a média diária e em seguida a média semanal para chegar aos valores apresentados a seguir:

Objeto	Variação Média	Variação Máxima
IfOperStatus	Up	Up
IfInDiscards	0	0
IfOutDiscards	0	0
IfInUnknownProtos	16	313

IfInOctets	2586139	48573566
IfInNUcastPkts	79	1811
IfOutOctets	5006405	22961125
IfOutNUcastPkts	20	364
IpInHdrErrors	0	0
IpInAddrErrors	0	0
IpReasmFails	0	0
IPOutNoRoutes	0	0
TcpAttemptFails	9	45
TcpEstabResets	7	82
TcpRetransSegs	21	91
IcmpInErrors	0	2
IcmpOutErrors	358	2813
UdpInErrors	0	0
UdpInDatagrams	6244	151520
UdpOutDatagrams	1693	37192

Vale ressaltar que os valores acima definidos, foram baseados em um *baseline* definido durante apenas uma semana. A construção do *baseline*, é de responsabilidade do administrador. Ele deve definir o período (uma semana, um mês, um ano, etc.) que julgar mais adequado.

O comportamento de cada objeto no *baseline* construído foi utilizado na definição das primeiras regras do sistema. A descrição destas regras estão no apêndice C.

7.4.2 Execução da Fase On-Line

A fase On-line do SAGRES consiste inicialmente na coleta de informações do elemento gerenciado. Foi determinado que o intervalo de coleta seria de uma hora. A execução deste passo está relacionada ao processo "Coleta de Dados dos Objetos Selecionados" do DFD da Fase On-Line.

Estas informações, após serem extraídas, ficam armazenadas em um repositório denominado “Dados Coletados”. Para exemplificar, o quadro a seguir apresenta duas leituras realizadas sobre o elemento gerenciado para um intervalo de monitoramento de uma hora.

Objeto IfOperStatus	1ª Leitura up	2ª Leitura Up	Diferença up
IfInDiscards	0	0	0
IfOutDiscards	0	0	0
IfInUnknownProtos	1830	1844	14
IfInOctets	252901733	261999295	9097562
IfInNUcastPkts	123495	124276	781
IfOutOctets	716564947	735995541	19430594
IfOutNUcastPkts	1848	1866	18
IpInHdrErrors	0	0	0
IpInAddrErrors	0	0	0
IpReasmFails	2	2	0
IPOutNoRoutes	0	0	0
TcpAttemptFails	771	774	3
TcpEstabResets	192	195	3
TcpRetransSegs	3428	3429	1
IcmpInErrors	32	32	0
IcmpOutErrors	30221	30396	175
UdpInErrors	0	0	0

UdpInDatagrams	801968	812722	10754
UdpOutDatagrams	501406	505369	3963

Após ter realizado a coleta, o sistema determina as discrepâncias, ou seja, realiza uma comparação entre o comportamento observado e o comportamento esperado.

Neste caso, o sistema detectou discrepância para os seguintes objetos:

Com relação à variação média:

ifInOctets, inInNUcastPkts, ifOutOctets, UdpInDatagrams,UdpOutDatagrams

Com relação à variação máxima:

Nenhuma discrepância detectada

Baseado nas regras cadastradas na base de conhecimento, o sistema chegou à seguinte diagnose:

Alguma aplicação fazendo uso de protocolo desconhecido com algumas variáveis apresentando comportamento acima da média, porém em todos os casos encontrando-se abaixo do máximo.

Capítulo 8

Conclusões

Neste capítulo é apresentado um novo conceito: Gerência de Sistemas. Uma avaliação do sistema SAGRES é realizada sendo sugerido os futuros trabalhos que podem ser desenvolvidos.

8.1 Da Administração de Redes à Gerência de Sistemas

Nos tradicionais ambientes de *mainframes*, a utilização de ferramentas, normalmente proprietárias, tornava relativamente fácil o trabalho de manter em operação toda uma estrutura de terminais, bem como os recursos do mainframe (processos e arquivos). A administração de sistemas se restringia a manter esta máquina funcionando.

Com a evolução da computação centralizada para a computação distribuída, várias novas atividades relativas à manutenção e integração de computadores e elementos de rede tornaram-se relevantes. A distribuição de recursos criou a necessidade de se produzir mecanismos que pudessem monitorar e resolver problemas dispersos na rede.

Surge, assim, o gerenciamento de redes. Criado para permitir a integração de equipamentos distintos em uma mesma rede, e produzir um ambiente onde aspectos como desempenho, segurança, configuração, falhas pudessem ser avaliados, o gerenciamento de redes tornou-se tão importante quanto a administração de sistemas centralizados.

8.1.1 Administração de Sistemas

A administração de sistemas se ocupa da tarefa de tornar e manter disponíveis para o usuário, serviços e equipamentos. Vários problemas de administração de sistemas nos tradicionais ambientes de "mainframes" são bem especificados e resolvidos com a existência de ferramentas próprias.

Dentre as tarefas principais de um administrador de sistemas, podemos citar:

- Criar novos usuários;
- Realizar backups dos sistemas; e
- Aumentar o tamanho de partições de um disco.

Algumas outras tarefas, não existentes em ambientes centralizados, surgiram para permitir a integração de diversos equipamentos de forma harmoniosa em uma rede. O compartilhamento de arquivos importantes (como o arquivo que contém as senhas dos usuários), a utilização de partições de disco de estações remotas, a transformação dos endereços físicos das estações em nomes de mais fácil memorização, entre outros, são serviços típicos de ambientes de rede. Conceitualmente padronizados, mas implementados de forma diferente nos diversos tipos de sistemas operacionais, esses serviços são também mantidos pelo administrador. Administradores que tomam conta desses tipos de serviços são normalmente conhecidos como administradores de rede, visto que tratam de serviços específicos de ambientes de rede como: NIS, NFS e DNS.

8.1.2 Gerência de Redes

"Gerenciar uma rede é realizar tarefas com objetivo de monitorar e controlar seus recursos computacionais tais como roteadores, protocolos associados, etc". Essa definição é bastante utilizada e aceita no meio acadêmico, muito embora seja uma forma muito simplista de definir toda a complexidade por trás da tarefa de gerenciar uma rede.

A complexidade trazida pela heterogeneidade das redes, tornou necessário a criação de mecanismos que pudessem realizar, de forma uniforme, o controle de todas as atividades desses ambientes. Esse controle se estendia desde os mecanismos de conexão e comunicação entre os recursos até às aplicações que eram executadas nos computadores.

O gerenciamento de redes surge então para solucionar problemas que antes não existiam ou para criar mecanismos de integração desses ambientes heterogêneos.

Dentre os objetivos principais do gerenciamento de redes, os seguintes se destacam:

- Aumentar o disponibilidade da rede: essa tarefa surge da necessidade de se fornecer um ambiente rápido é seguro para o usuário. Nesse sentido, quaisquer problemas na rede devem ser resolvidos da forma mais rápida possível.
- Diminuir os custos de operação da rede: com o aumento da heterogeneidade das LAN's, aumentou também a necessidade de se fornecer um ambiente de gerenciamento heterogêneo que desse suporte a manutenção de equipementos de vários fornecedores;
- Permitir flexibilidade e facilidade de integração: nas LAN's, deve ser possível se adicionar novos equipamentos sem muita dificuldade;

Para criar um ambiente que desse suporte às atividade citadas acima, vários modelos foram sugeridos. Inicialmente foi imaginado a abertura de ferramentas de gerenciamento proprietárias, de forma a incorporar os requisitos de outros ambientes. Essa solução não obteve sucesso, visto que essa integração desejada, nem sempre poderia ser realizada de forma genérica o suficiente para incorporar os requisitos funcionais de vários fornecedores diferentes.

Trabalhando em paralelo, administração e gerenciamento são conceitos que algumas vezes causam confusão, mas cada uma dessas atividades tem um escopo muito bem definido. Tendo como principal diferencial a padronização, o gerenciamento de redes se caracteriza por possuir objetos claros de atuação, que são mapeados nas redes pelo conceito abstrato de objetos gerenciados [Sta93]. Manipulados por protocolos consolidados no ambiente comercial, como o SNMP (*Simple Network Management Protocol*), esses objetos se encontram armazenados em repositórios de informações distribuídos, denominados MIB's (Management Information Base), onde agentes podem atuar, realizando diversas tarefas, afetando dinamicamente o estado da rede. Na administração de sistemas os conceitos manipulados são relativos aos sistemas operacionais das máquinas a serem administradas. Sem possuir modelos padronizados de atuação, os administradores de sistemas têm a tarefa árdua de conhecer comandos e definições diferentes, que realizam a mesma tarefa, nos diferentes tipos de sistemas operacionais.

8.1.3 Gerência de Sistemas

Como pôde ser observado nas sessões anteriores, existem duas tarefas distintas: administrar e gerenciar a rede. Ambas trazem consigo, conceitos e ferramentas distintas, muito embora seja desejável que o administrador tenha que tratá-las de forma integrada. Propomos um novo conceito para aglutinar as tarefas de administrar sistemas e gerenciar redes heterogêneas: a gerência de sistemas.

Assim, o gerente de sistemas é o profissional capaz de integrar os conceitos e ferramentas de administração e gerenciamento. Essa integração permite uma visão mais ampla e o domínio maior de grandes redes.

Várias tarefas devem ser desempenhadas pelo gerente de sistemas, além das atividades de administração de sistemas citadas no ítem 8.1.1

- Configurar e gerenciar os recursos da rede: roteadores, bridges, modems, , etc.;
- Instalar e manter os serviços de rede: NIS, NFS, DNS, etc.;
- Monitorar os recursos da redes através de ferramentas de gerenciamento;
- Dispor das informações de gerenciamento para analisar e resolver problemas de segurança, desempenho e falhas;

8.2 Avaliação do SAGRES

Como contribuição inicial desta dissertação, temos a validação de parte da metodologia DAG, que foi proposta em uma outra dissertação de mestrado. Da metodologia DAG, executamos a seqüência de atividades propostas para o desenvolvimento de aplicações de gerenciamento de redes.

O SAGRES é um sistema que dentro das funcionalidades de gerência de redes permite as seguintes atividades:

- coleta de dados;
- tratamento dos dados;
- análise dos dados;
- identificação de problemas; e
- correção dos problemas.

O protótipo que se encontra operacional foi testado nas instalações do laboratório do mestrado em ciência da computação da UFC e pode ser utilizado para a formação de administradores de redes, bem como para auxiliar diretamente na atividade diária de gerência em inúmeras outras instalações.

Em comparação com outros sistemas baseados em conhecimento desenvolvidos anteriormente (Olho Vivo, Agente 6) o SAGRES oferece além de uma análise do estado da rede, a possibilidade de uma atuação automática sobre os agentes dos elementos de rede gerenciados. Os dois outros sistemas citados, utilizam o SunNet Manager na função de Gerente enquanto que o SAGRES é um sistema que trabalha independente, realizando as operações Get e Set.

Com relação aos softwares de gerenciamento comerciais (SunNet Manager, HP Open View, Netview) o SAGRES disponibiliza ao administrador, uma análise sofisticada do comportamento da rede. Para tal, são utilizadas técnicas de inteligência artificial (sistema especialista). Nos sistemas comerciais, em geral, o administrador deve possuir um vasto conhecimento em gerência de redes para poder extrair informações do comportamento da rede a partir dos dados fornecidos por estes sistemas.

O SAGRES, entretanto, não tem implementado alguns interessantes recursos que estão disponíveis em ferramentas comerciais, tais como: descoberta automática da rede e janelas de visualização da rede.

Uma outra limitação do SAGRES consiste no fato de não tratar problemas advindos da área de administração de sistemas, uma vez que estas duas áreas (administração e gerência) estão altamente interligadas.

Avaliamos também como uma limitação do SAGRES, o fato do sistema estar preso a uma estação de gerenciamento, não podendo ser utilizado remotamente. Esta limitação pode ser superada com a implementação de interfaces WWW.

8.3 Trabalhos Futuros

Como futuros trabalhos, sugerimos algumas alterações no ambiente SAGRES de maneira a atender às funcionalidades definidas no novo conceito de "Gerência de Sistemas", introduzido no início deste capítulo.

Estas alterações consistem na introdução de mecanismos de administração de sistemas ao ambiente existente, visando atender os seguintes requisitos:

- Realizar monitoração de recursos das estações da rede;
- Tratar os dados obtidos na monitoração e tomar decisões baseados nestes;
- Realizar ações nas estações, permitindo a correção de problemas detectados.

De maneira a atender os novos requisitos, as seguintes implementações devem ser realizadas:

- Criação de agentes coletores de dados das estações da rede (ObsAgentes);
- Introdução de novas regras na Base de Conhecimento que tratassem de administração de sistemas;
- Criação de agentes que atuarão na resolução de problemas detectados nas estações (Agentes de Atuação).

Para atender estes novos requisitos, uma nova arquitetura do SAGRES é proposta (Figura 8.1) sendo caracterizada pelo Subsistema de Decisão (SD) e pelo Subsistema de Coleta e tratamento de Dados.

Subsistema de Decisão (SD)

Os agentes de atuação, fornecem meios do SAGRES realizar correções em problemas detectados. Atuando diretamente na estação, esses agentes liberam o administrador de realizar tarefas repetitivas como, por exemplo, aumentar o tamanho de um filesystem.

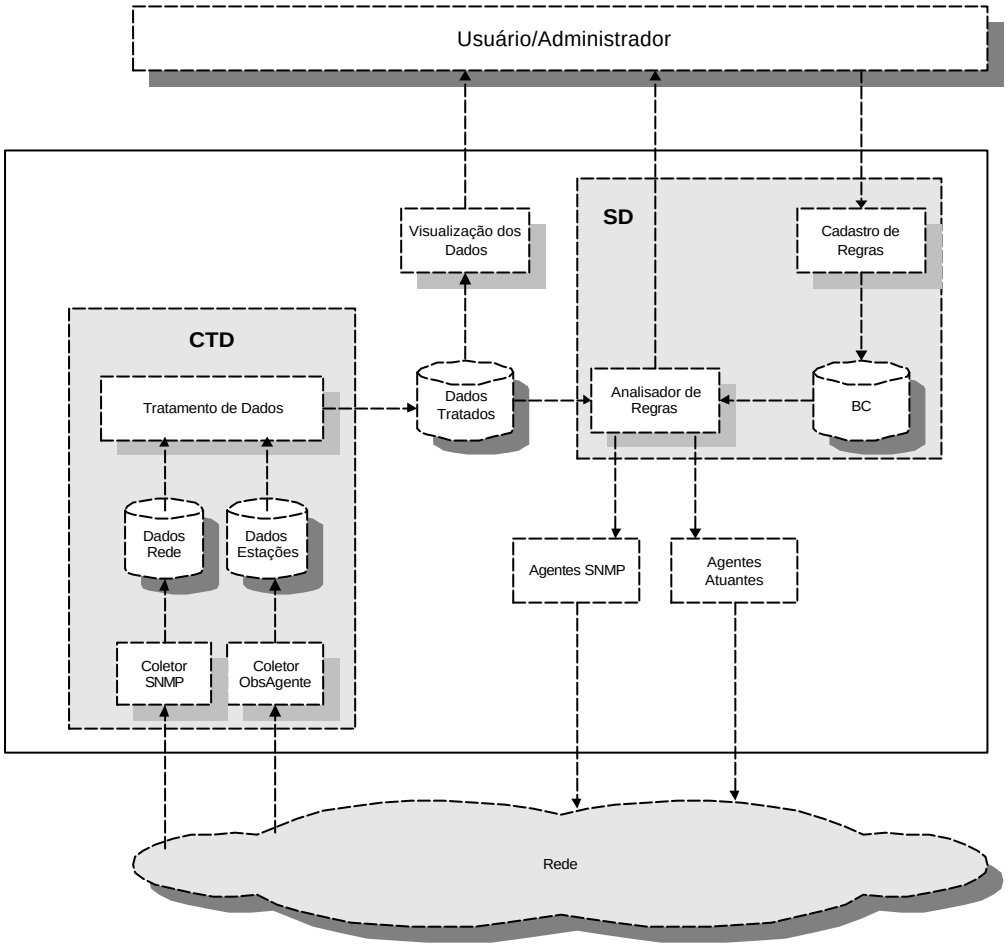


Figura 8.1

Subsistema de Coleta e Tratamento de Dados (CTD)

Os dados relativos à administração de sistemas, como taxa de ocupação de filesystems, quota de usuários, utilização de memória e etc, são coletados por agentes especiais denominados **ObsAgentes**. Esses agentes estão espalhados pelas diversas estações da rede, e armazenam, periodicamente, os dados capturados em arquivos de log locais.

O módulo tratamento de dados por sua vez, captura as informações armazenadas nos logs das estações, que são então formatados e enviados para o arquivo “Dados Tratados”.

8.4 Considerações Finais

Esta dissertação é resultante de um conjunto de esforços realizados pelo LAR (Laboratório Multiinstitucional de Redes e Sistemas Distribuídos). Criado em meados de 1994, o LAR congrega as principais atividades de pesquisa e desenvolvimento na área de redes de computadores no Estado do Ceará. Unindo pesquisadores de instituições como a UFC (Universidade Federal do Ceará), ETFCE (Escola Técnica Federal do Ceará) e UECE (Universidade Estadual do Ceará), o LAR tem proporcionado uma substancial produção científica na área de redes e sistemas distribuídos.

O LAR participa atualmente de 2(dois) projetos do Programa Temático do CNPq (PROTEM), o FLASH e o GERENTE e de 1(um) projeto internacional (NEURALTEL). No FLASH (Formalizações da Administração de Sistemas Heterogêneos), o LAR tem a parceria do Departamento de Informática da Universidade Federal de Pernambuco (DI-UFPE).

O FLASH é um projeto cooperativo e multidisciplinar, integrando esforços de grupos de pesquisa acadêmicos e empresariais das áreas de administração e integração de sistemas, gerência de redes e bancos de dados, visando contribuir para a criação de um estoque social de recursos humanos e gerar teorias, modelos, técnicas e ferramentas que sejam competitivos internacionalmente e ao mesmo tempo aplicáveis a problemas nacionais em administração de grandes sistemas de computação com plataformas heterogêneas de hardware, software básico e SGBDs distribuídos.

Na área de redes de computadores, são pesquisados os seguintes temas:

- interconexão de sub-redes locais dos laboratórios heterogêneos e análise da eficiência e desempenho das várias alternativas de interconexão;
- utilização da Internet para experimentos de interconexão das redes locais, com ênfase na utilização de padrões internacionais para os protocolos de gerência de redes;
- estudo das ferramentas de auxílio ao desenvolvimento de aplicações de gerência;

- implementação um protótipo de uma plataforma para estudos de heterogeneidade e interoperabilidade de sistemas de gerência de redes;
- investigação da introdução de comportamento inteligente nos agentes de gerência: alarmes, diagnóstico e recuperação de falhas, etc.

O sistemas SAGRES tem sua contribuição ligada à área de gerência de redes do projeto FLASH. Dentre vários trabalhos desenvolvidos no contexto do projeto FLASH temos além do SAGRES (descrito neste trabalho), o I-DREAM (descrito na seção 5.3.3 – Estado Atual dos Sistemas baseados em conhecimento). Ambos fazem uso de sistemas inteligentes para a resolução de problemas complexos, sendo o primeiro no âmbito de gerência e o segundo no de administração. Estes dois sistemas então hoje disponíveis permitem ao FLASH trabalhar na possibilidade de construção de um *framework* que venha a integrar estes sistemas e conseqüentemente, disponibilizar uma ferramenta com funcionalidades de gerência e administração.

O trabalho desenvolvido nesta dissertação contribuiu para o enriquecimento do projeto FLASH, e abre uma perspectiva para a execução de novos trabalhos. Como resultado dos enforços aqui realizados tivemos um artigo intitulado “SAGRES, um Sistema Baseado em Conhecimento para Apoio à Gerência de Redes de Computadores, publicado no 2º Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos realizado em Fortaleza no perído de 3 a 7 de Novembro de 1997. Este artigo apresenta de forma resumida o trabalho desenvolvido nesta dissertação de mestrado.

Apêndice A

Modelagem UML

Modelagem baseado em objeto é um novo modo de estudar problemas com utilização de modelos fundamentados em conceitos do mundo real [Rum94]. A estrutura básica é o objeto, que combina a estrutura e o comportamento dos dados em uma única entidade. Os modelos baseados em objetos são úteis para a compreensão de problemas, para a comunicação com os peritos em aplicações, para modelar empresas, preparar documentação e projetar programas e bancos de dados.

Ao se fazer uma modelagem, primeiro, prepara-se um modelo que sumarize os aspectos essenciais do domínio da aplicação, sem preocupações com uma eventual implementação. Esse modelo contém objetos encontrados no domínio da aplicação, incluindo uma descrição das propriedades dos objetos e de seu comportamento. Em seguida, são tomadas decisões acerca do projeto e acrescentam-se detalhes ao modelo para se descrever e otimizar a implementação. Os objetos do domínio da aplicação compõem a estrutura do modelo projetado, mas são implementados em termos de objetos do domínio do computador. No final, o modelo projetado é implementado em uma linguagem de programação, em um banco de dados ou em hardware.

Um modelo é uma abstração de alguma coisa, cujo propósito é permitir que se conheça essa coisa antes de construí-la. Como um modelo omite os detalhes não essenciais, sua manipulação é mais fácil do que a da entidade original. Para construir sistemas complexos, o desenvolvedor deve abstrair diferentes visões do sistema, construir modelos com utilização de uma notação precisa, verificar se os modelos satisfazem os requisitos do sistema e acrescentar detalhes gradativamente para transformar os modelos em uma implementação.

A técnica de modelagem que será utilizada neste trabalho para especificação, visualização, e documentação do sistema é a UML (*Unified Modeling Language*). A UML foi definida pela "Rational Software" através dos seguintes metodologistas : Grady Booch, Ivar Jacobson e Jim Rumbaugh. Seus principais benefícios são:

- oferece uma abordagem passo-a-passo para o desenvolvimento de software de qualidade;

- define um mapeamento da análise ao projeto e à implementação;
- define uma notação expressiva e consistente;
- facilita a comunicação entre as pessoas;
- ajuda a apontar inconsistências e omissões; e
- suporta análise e projeto de pequeno e grande porte.

A figura a seguir mostra as contribuições recebidas para a definição da UML:

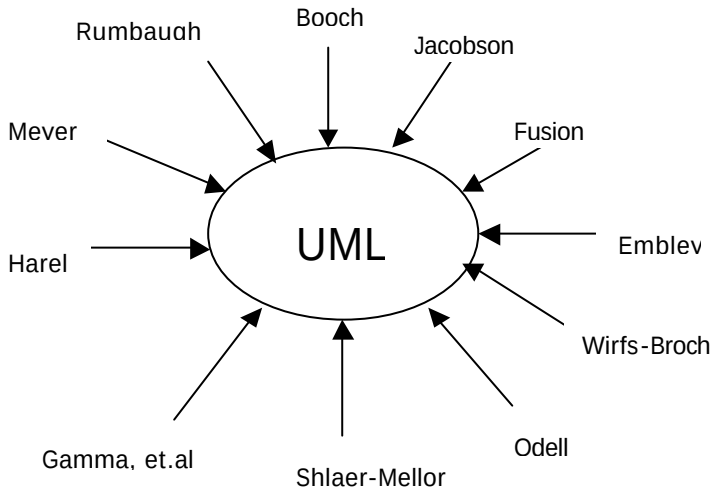


Figura A.1: UML - Contribuições

O modelo proposto pela UML (Figura A.2) é composto pelas seguintes visões: visão lógica, visão de desenvolvimento, visão de processo, visão física e visão dos casos de uso/cenários.

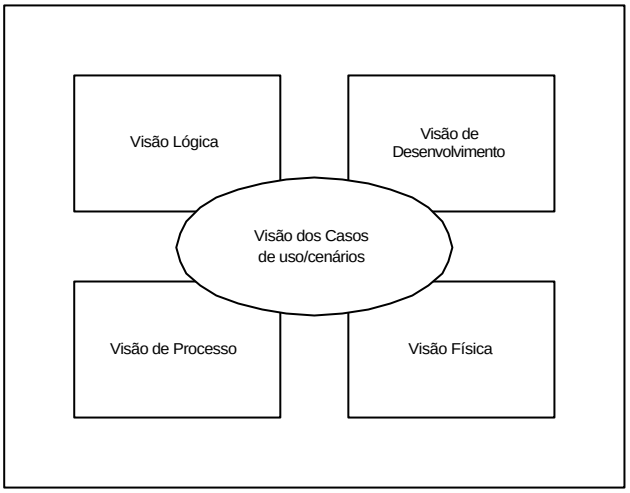


Figura A.2: UML – Modelo Proposto

A visão lógica é composta pelos diagramas de classe e diagramas de sequência; a visão de desenvolvimento pelos diagramas de componentes e a visões física e de processo são compostas pelos diagramas de desenvolvimento.

Um diagrama de seqüência representa as mensagens trocadas por um conjunto de objetos durante um cenário. Um diagrama de seqüência contém:

- objetos com suas linhas de vida;
- mensagens trocadas entre objetos em uma seqüência ordenada; e
- foco de controle (opcional).

Em um diagrama de seqüência, os objetos são desenhados como retângulos com nomes sublinhados e as *linhas de vida* dos objetos são representadas por linhas tracejadas descendentes (Figura A.3). As interações de objetos são mostradas como setas horizontais, direcionadas da linha vertical representando o objetocliente para a linha representando o objetoservidor (Figura A.4). O foco de controle representa o tempo relativo em que o fluxo de controle está focalizado em um objeto.

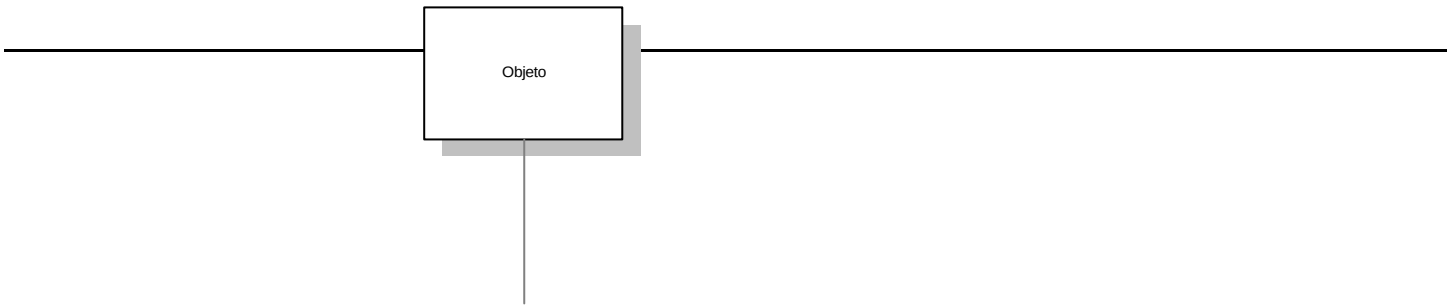
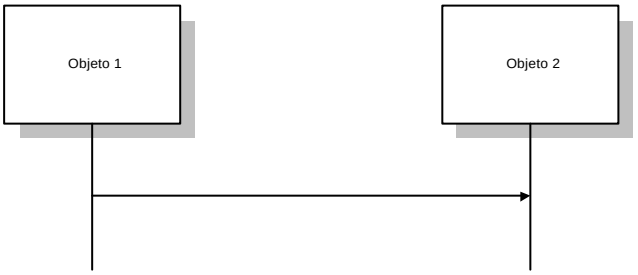


Figura A.3: Representação Objetos

Um diagrama de classe é uma perspectiva de alguns (ou todos) dos pacotes e classes de uma visão lógica. Um pacote é uma coleção lógica de classes e/ou outros pacotes (Figura A5). O diagrama de classes principal é tipicamente uma visão dos pacotes de mais alto nível em uma perspectiva lógica.

Figura A.4: Interação Objetos



Diagramas de classes (Figura A6) adicionais são acrescentados se for preciso:

- Visualizar classes participantes de um cenário;
- Visualizar as classe “privadas” de um pacote;
- Visualizar uma classe e seus atributos e operações; e
- Visualizar uma hierarquia de herança.

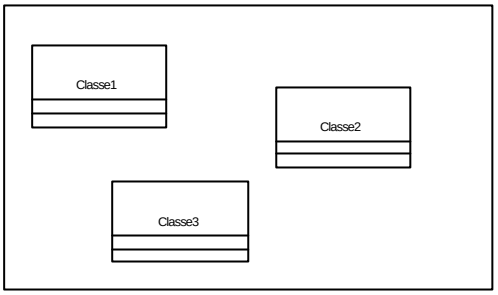


Figura A.5: Representação Pacotes

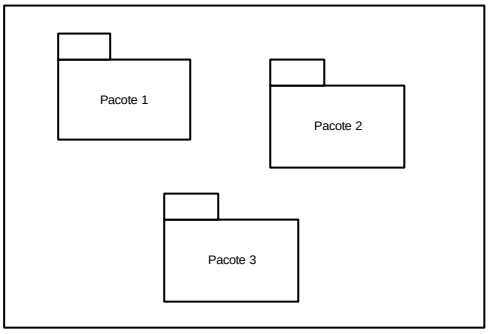


Figura A.6: Diagrama de Classes

A visão dos casos de uso/cenários é o que mantém as quatro visões juntas. Eles demonstram e validam as visões lógica, de processo, de desenvolvimento e física da arquitetura. Caso de uso é uma sequência de transações realizadas por um sistema, as quais resultam em valores mensuráveis para um ator em particular. Um ator representa qualquer coisa que interaja com o sistema. Um cenário é uma instância de um caso de uso, ou seja, é um esboço dos eventos que ocorrem durante a execução do sistema.

Herança define um relacionamento entre classes onde uma classe compartilha a estrutura e/ou comportamento de uma ou mais classes. Através da herança, defini-se uma hierarquia de abstrações na qual uma subclasse herda de uma ou mais superclasses.

Os tipos de herança säo:

- herança simples: a subclasse herda de apenas uma superclasse; e
- herança múltipla: a subclasse herda de mais de uma superclasse.

A Figura A7 mostra a representação da heranca.

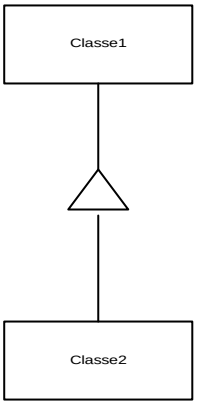


Figura A7: Representação Herança

Agregação é uma forma especializada de associação na qula o todo está relacionado a suas partes. A agregação é conhecida como “parte de” ou relacionamento de conteúdo. Sua representação é mostrada na Figura A8.

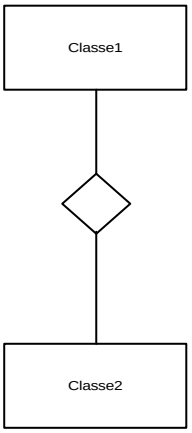


Figura A8: Representação Agregação

Apêndice B

Apêndice C

Regras

O comportamento de cada objeto no *baseline* construído, foi utilizado na definição das primeiras regras do sistema. Foram definidos então as seguintes regras para atender à necessidade de gerenciamento especificada:

Regra 1

Se OperStatus = 2
Então Problema = Interface Desligada ou Defeituosa

Regra 2
Se OperStatus = 1
E ifInDiscards > 0
Então Problema = Memória Insuficiente

Regra 3
Se OperStatus = 1
E ifOutDiscards > 0
Então Problema = Memória Insuficiente

Regra 4
Se OperStatus = 1
E ifInUnknownProtosErrors > 0
Então Problema = Uso de protocolo não suportado

Regra 5
Se OperStatus = 1
E ifInNUcastPkts > media
E ifInNUcastPkts < maximo

Então Alerta = Número de pacotes não Unicast recebidos acima da média.

Regra 6
Se OperStatus = 1
E ifInNUcastPkts > maximo
Então Alerta = Número de pacotes não Unicast recebidos acima do Máximo

Regra 7
Se OperStatus = 1
E ifOutOctets > media
E ifOutOctets < maximo
Então Alerta = Número de pacotes enviados acima da média

Regra 8
Se OperStatus = 1
E ifOutOctets > maximo
Então Alerta = Número de pacotes enviados acima do máximo

Regra 9
Se OperStatus = 1
E ifOutNUcastPkts > media
E ifOutNUcastPkts < maximo
Então Alerta = Número de pacotes não Unicast enviados acima da média.

Regra 10
Se OperStatus = 1
E ifInNUcastPkts > maximo
Então Alerta = Número de pacotes não Unicast enviados acima do Máximo

Regra 11
Se OperStatus = 1

E ipOutNoRoutes > 0
Então Problema = Rotas desconhecidas, verificar tabela roteamento

Regra 12
Se OperStatus = 1
E ipReasmFails > 0

Então Problema = Falha no algoritmo de remontagem dos pacotes IP

Regra 13
Se OperStatus = 1
E ipInHdrErrors > 0

Então Problema = Header do protocolo IP

Regra 14
Se operStatus = 1
E ipInAddrEors > 0
Então Problema = Endereço IP destino inválido

Regra 15
Se operStatus = 1
E icmpInErrors > 0
Então Problema = Protocolo ICMP

Regra 16
Se operStatus = 1
E icmpOutError > 0
Então Problema = Protocolo ICMP

Regra 17

Se OperStatus = 1

E udpInDatagrams > 6244

E udpInDatagrams < 151520

Então Alerta = Número de pacotes Udp recebidos acima da média

Regra 18

Se OperStatus = 1

E udpInDatagrams > 151520

Então Alerta = Número de pacotes recebidos acima do máximo

Regra 19

Se Operstatus = 1

E udpOutDatagrams < 1693

E udpOutDatagrams < 37192

Então Alerta = Número de pacotes udp enviados acima da média

Regra 20

Se Operstatus = 1

E udpOutDatagrams < 37192

Então Alerta = Número de pacotes udp enviados acima do máximo

Regra 21

Se OperStatus = 1

E udpInErrors > 0

Então Problemas = Pacotes udp estão deixando de ser enviados

Regra 22

Se OperStatus = 2

E tcpAttemptFails > 0

Então Problema = Falhas no estabelecimento de conexões

Regra 22

Se OperStatus = 1

E tcpRetransSegs > 0

Então Problema = Número de pacotes retransmitidos superior ao normal

Regra 23

Se OperStatus = 1

E ipReasmFails > 0
E ipInHdrErrors > 0
E ipInAddrEors > 0
E icmpInErrors > 0
E icmpOutError > 0
E udpInErrors > 0

E tcpAttemptFails > 0

E tcpRetransSegs > 0

Então Desativar_Interface = Sim

Apêndice D

Diagrama de Fluxo de Dados

Para a descrição do funcionamento das fases aqui descritas, fez-se uso do recurso do diagrama DFD (Diagrama de Fluxo de Dados). Um diagrama de fluxo de dados é constituído de processos, fluxo de dados, depósitos de dados e entidades externas, conforme Figura D.1.

- processos: é um componente procedural do sistema e opera sobre os dados. São representados por retângulos com os vértices arredondados;
- fluxo de dados: conduz o fluxo de informações através dos processos de um sistema. É representado por uma seta, com a ponta indicando a direção do fluxo;
- depósito de dados: representa um arquivo lógico. São representados por um par de linhas paralelas horizontais, ligadas em uma das extremidades; e
- entidades externas: representa as fontes de dados externas ao sistema. Uma entidade externa é representada por um quadrado.

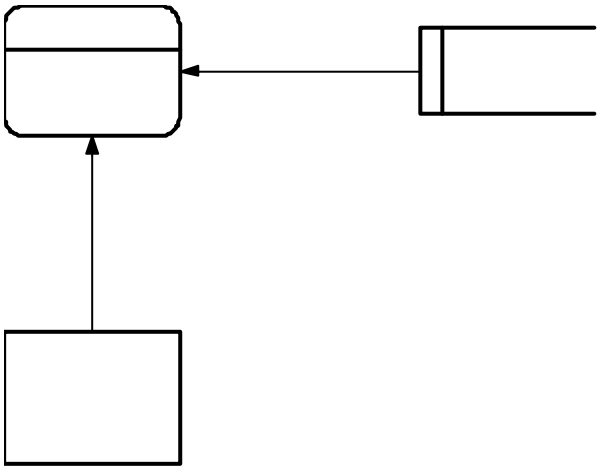


Figura D.1

Referências Bibliográficas

[Aid94] S. Aidorous and T. Plevyak. Telecommunications Network Management into the 21st Century: Techniques, Standards, Technologies, and Applications. IEEE Press, New York, 1994.

[Art96] Esmilda Artola. Um Sistema Especialista para Gerência Pró-Ativa Remota. Anais do XIV Simpósio Brasileiro de Redes de Computadores. Fortaleza, Ceará, 1996.

[Boo94] G. Booch. Object-Oriented Analysis and Design with Aplications, Benjamin/Cummings, Redwood City, Ca., 1994.

[Buc84] B. G. Buchanan.and E. H. Shortliffe. Rule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison Wesley, 1984.

[Cas90] J. Case, M. Fedor, M. Schoffstall and J. Davin. A Simple Network Management Protocol SNMP. *Request For Comments* 1157, May 1990.

[Com88] D. E. Comer. Internetworking with TCP/IP – Principles, Protocols, and Architecture. Prentice Hall, 1988.

[Cro 88] Robert Cronk, Paul Callahan and Lawrence Bernstein. Rule Based Expert System for Network Management and Operations: An Introduction. IEEE Network, New York, v.2, n.5, p.7-21, Sep. 1988.

[Dav84] R. Davis. Diagnostic reasoning based on structure and behavior. Artificial Intelligence, 1984.

[Der92] F. J. Derfler. Guide to Conectivity. Ziff-Davis Press, 1992.

[Fra97] Daniela Matos Franklin. I-DREAM – Um Sistema de Monitoração de Recursos e Aplicações Baseado em Intranet. Dissertação de Mestrado, Universidade Federal de Pernambuco,1997.

[Gas96] A. F. Gasparini., F. Barrela. TCP/IP Solução para Conectividade. Érica, 1996.

[Glo86] F. Glover. Future paths for integer programming and links to artificial intelligence, Computers and Oper. Res. 533-549, 1986.

[Glo87] F. Glover. Tabu search methods in artificial intelligence and operations research, ORSA Artificial Intelligence Newsletter 1, 1987.

[Goy94] S. K. Goyal. Artificial Intelligence in Support of Distributed Network Management. Chapter 21, pages 539-577. Volume 1 of Sloman, 1994.

[HAY 85] Frederick Hayes-Roth. Rule Based Systems. Communications of the ACM, New York, Sep. 1985.

[Her90] A. Hertz and D. Werra. The Tabu Search Metaheuristic: How We Used It. Annals of Mathematics and Artificial Intelligence, 1990

[ISO/IECIPS-89] International Standards Organization / International Electrotechnical Commission. Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework. ISO/IEC 7498-4, November, 1989.

[ISO/IEC9596-90] International Standards Organization / International Electrotechnical Commission. Information Technology - Open Systems Interconnection – Common Management Information Protocol Specification. ISO/IEC 9596, May 1990.

[ISO/IEC DIS10040-91] International Standards Organization / International Electrotechnical Commission. Information Technology - Open Systems Interconnection – Systems Management Overview. ISO/IEC DIS 10040, May 1991.

[ISO/IEC DIS101651-91] International Standards Organization / International Electrotechnical Commission. Information Technology - Open Systems Interconnection – Structure of Management Information – Part 1: Management Information Model. ISO/IEC DIS 10165-1, March 1991.

[ISO/IEC DIS101654-91] International Standards Organization / International Electrotechnical Commission. Information Technology - Open Systems Interconnection – Structure of Management Information – Part 4: Guidelines for the Definition of Managed Objects. ISO/IEC DIS 10165-4, March 1991.

[ISO/IEC9595-91] International Standards Organization / International Electrotechnical Commission. Information Technology - Open Systems Interconnection – Common Management Information Service Definition. ISO/IEC 9595, April 1991.

[Jack] Peter Jackson. Introduction to Expert Systems. Addison-Wesley, 1990.

[Jac92] I. Jacobson. Object-Oriented Software Engineering. Addison-Wesley, 1992.

[Jos90] C. A. Joseph and K. Muralidhar. Integrated Network Management in an Enterprise Environment. IEEE Network Magazine, July 1990.

[Kau92] F. Kauffels. Network Management - Problems, Standards and Strategies. Addison Wesley, 1992.

[Kle88] S. M. Klerer. The OSI Management Architecture: an Overview. IEEE Network Magazine, March 1988.

[LIA96] Laboratório de Inteligência Artificia-UFC : Expert Sinta – Uma ferramenta visual para criação de sistemas especialistas, 1996.

[Lor93] M. Lorenz. Object-Oriented Software Development. Prentice–Hall, 1993.

[Man93] M. S. Mansouri and M. Sloman. Monitoring Distributed Systems. IEEE Network, November 1993.

[Mur94] K. H. Muralidhar. Knowledge-based Network Management. Aydorous & Plevyak, 1994.

[Orf96] R. Orfali, D. Harkey and J. Edwards. The Essential Client/Server Survival Guide. John Wiley, 1996.

[Ous94] J. H. Ousterhout. Tcl and the Tk Toolkit. Addison Wesley, 1994.

[Ram94] Suzana Ramos. Uma Metodologia para Análise e Desenvolvimento de Aplicações de Gerenciamento de Redes de Computadores. Dissertação de Mestrado, Universidade Federal de Pernambuco, Dezembro, 1994.

[Ram96] Suzana Ramos, Paulo Cunha e Mauro Oliveira. Disponibilização de Conhecimento no Gerenciamento de Redes de Computadores. Belo Horizonte, Minas Gerais, Semish, 1996.

[Rei87] Raymond Reiter. A Theory of Diagnosis from First Principles. Artificial Intelligence, 1987.

[Roc96] M. A. Rocha. Gerência Pró-Ativa de Redes de Computadores usando Agentes e Técnicas de Inteligência Artificial. Anais do XIV Simpósio Brasileiro de Redes de Computadores. Fortaleza, Ceará, 1996.

[Ros91] Marshall T. Rose. The Simple Book - An Introduction to Management of TCP/IP based internets. Prentice Hall, 1991.

[Rum94] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. Modelagem e Projetos Baseados em Objetos. Campus, 1994.

[Sch95] J. Schoenwaelder, and L. Langendorfer. Tcl Extensions for Network Management Applications. In Proc. 3rd Tcl/Tk Workshop, Toronto, Canada, July 1995.

[Slo94] M. Sloman. Network and Distributed Systems Management. Addison-Wesley, 1994.

[Soa95] Luiz Fernando Soares, G. Lemos, e S. Colcher. Redes de Computadores: das LANs, MANs e WANs às Redes ATM. Campus, 1995.

[Sri97] Sriram Srinivasan. Advanced Perl Programming. O'Reilly & Associates, 1997.

[Sta93] William Stallings. SNMP, SNMPv2, and CMIP - The Practical Guide to Network Management Standards. Addison Wesley, 1993.

[Sun91] Sun Microsystems Inc. SunNet Manager 1.1 – Installation and User's Guide, 1991.

[Sur94] José Augusto Suruagy. Rede Digital de Serviços Integrados de Faixa Larga. IX Escola de Computação – Recife – Pe, 1994.

[Tan89] Andrew S. Tanenbaum. Computer Networks. Prentice Hall, 1989.

[Tay97] D. Taylor. High Performance Delphi3 Programming. Coriolis Group,1997.

[Uda96] D. K. Udapa. Network Management Systems Essentials. McGraw-Hill, 1996.

[Ull88] J. Ullman. Principles of Database and Knowledge Base Systems. Computer Science Press, 1988.

[Wal96] Lary Wall, Tom Christiansen and Randal Schwartz. Programming Perl. O'Reilly & Associates, 1996.

[Wat] Donald A. Waterman. A Guide to Expert Systems. Addison-Wesley, 1985.

[Wir90] R. Wirfs-Brock. Designing Object-Oriented Software. Prentice-Hall, 1990.