Fábio José Justo dos Santos

SISTEMA DE GERENCIAMENTO DE REDES BASEADO EM CONHECIMENTO

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Pós-Graduação *Lato Sensu* Administração em Redes Linux, para obtenção de título de especialista.

Orientador

Prof. Dr. Gustavo Guimarães Parma

LAVRAS

MINAS GERAIS – BRASIL

2004

Fábio José Justo dos Santos

SISTEMA DE GERENCIAMENTO DE REDES BASEADO EM CONHECIMENTO

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Pós-Graduação *Lato Sensu* Administração em Redes Linux, para obtenção de título de especialista.

Aprovada em 12 de Dezembro de 2004.

_____

Prof. MSc. Joaquim Quinteiro Uchôa

_____

Prof. MSc. Fernando Cortez Sica

_____

Prof. Dr. Gustavo Guimarães Parma
(Orientador)

LAVRAS
MINAS GERAIS - BRASIL

Dedico este trabalho a todos que me ajudaram, especialmente a meu orientador prof. Dr. Gustavo Parma, e a todos que compreenderam a importância deste passo, em especial aos meus pais.

Agradeço à Deus por permitir
a realização deste trabalho,
dando-me forças e sabedoria
para concluí-lo.

"Rerum omnium magister usus."
A experiência é mestra de todas as coisas.
CÉSAR (101-44 a.C.) A guerra Civil, II.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

Resumo

A gerência de redes é a ciência responsável por manter o controle, de forma integrada, de todos os serviços e recursos que há compõe [MELCHIORS, 1999]. Tal atividade compreende o monitoramento, a análise e a resolução de problemas. Dentro desse conceito, a ISO[1] apresentou o modelo OSI[2] de gerência de redes, cuja arquitetura funcional está dividida em cinco áreas funcionais que serão tratadas nesse documento: gerência de configuração, gerência de segurança, gerência de desempenho, gerência de contabilização e gerência de falhas.

Com base nos sistemas de registro de problemas *(trouble ticket system)* que tem como objetivo armazenar o histórico dos incidentes ocorridos em uma rede, torna-se possível a criação de um sistema que realize a monitoração e a análise dos problemas apresentados. Uma abordagem da Inteligência Computacional que tem atraído muito a atenção nos últimos anos é a do raciocínio baseado em casos *(case-based reasoning)* [MELCHIORS, 1999]. Através dos sistemas especialistas que utilizam-se da heurística e que fazem uso desta abordagem para tomar suas decisões, podemos desenvolver uma ferramenta de auxílio na tarefa de gerência de redes.

O objetivo deste trabalho é apresentar os conceitos que norteiam a gerência de redes de computadores e os sistemas especialistas baseados em casos. Utilizando-se de uma abordagem teórica dos conceitos, técnicas e práticas das duas ciências, serão estudados os conceitos deste domínio, a literatura existente e por último o estudo de um protótipo já desenvolvido.

Palavras-Chave: *gerência de falhas; gerência de redes; sistema de registro de problemas; sistemas especialistas baseados em conhecimento.*

1International Organization for Standardization

2 Open System Interconection

1. Introdução

Com o crescimento da heterogeneidade dos tipos e aplicações de redes, manter um tempo de resposta satisfatório e alta disponibilidade com custos compatíveis com a realidade das empresas torna-se uma tarefa cada vez mais crítica.

Os benefícios ofertados pelo surgimento das redes de computadores são inúmeros. Certamente, se hoje vivemos na era da informação, um dos recursos que permitiram tal evolução foram as redes de computadores. Atualmente o uso deste recurso como uma plataforma que ofereça suporte à computação distribuída, cooperativa e tolerante a falhas tem crescido muito. Com surgimento de novas aplicações e funcionalidades para as redes de computadores, qualquer problema nesta estrutura pode acarretar queda no desempenho, redução ou mesmo perda da comunicação, dentre outras circunstâncias anômalas, como falhas no sistema.

Associada às diversas possibilidades de aplicabilidade das redes de computadores, temos o crescimento constante do número de heterogeneidade e de equipamentos presentes nessas rede, o que torna necessária a utilização de um técnico especialista para controlar e manter a disponibilidade e qualidade dos serviços da rede, através do gerenciamento das mesmas.

Por essas e outras razões, dentre as cinco áreas funcionais de gerência de redes apresentadas pela ISO no modelo de gerência de redes OSI, a gerência de falhas é a que tem ganho mais atenção nos últimos tempos, juntamente com gerência de segurança. Com o objetivo de oferecer suporte ao gerenciamento de falhas, de modo a tornar os índices de disponibilidade e confiabilidade de uma rede em limites satisfatórios para a administração, surge o conceito dos

"Sistemas de Gerenciamento de Redes Baseados em Conhecimento" (SGRBC). Tais sistemas trazem consigo o paradigma de sistemas especialistas baseado em casos.

Alguns trabalhos realizados, tratam exclusivamente do gerenciamento de falhas. O CRITTER [MELCHIOR, 1999] é um exemplo de sistema especialista para gerenciamento de redes desenvolvido para o contexto de gerencimento de falhas e que faz uso do paradigma de raciocínio baseado em casos sobre um sistema de registro de problemas. Outro exemplo é o SAGRES, que será objeto de estudo no capítulo 4.

Os Sistemas de Registros de Problemas (*Trouble Ticket System – TTS*) são ferramentas que auxiliam no armazenamento dos incidentes ocorridos em uma rede, acumulando o histórico dos fatos e o conhecimento aplicado nas resoluções de tais problemas. Com o auxílio destes registros, torna-se possível verificar se o nível do serviço apresentado corresponde ao desejado e obter os índices de funcionalidade e de performance em tempo real. O sistema FNMS (*Free Network Management System*), desenvolvido pela PUCRS, apresenta um módulo TTS integrado à um SGRBC, que tem como objetivo axuxiliar o administrador de redes na tomada de suas decisões. Com uma base de casos dentro do sistema, o FNMS avalia o problema apresentado e, de acordo com os históricos, propõe uma solução ao administrador. Parte das informações obtidas por meio do monitoramento são armazenadas em um Banco de Dados (BD) para realização de análises estatísticas, e parte é utilizada para comparar o status real da rede com o desejado, descobrindo assim qualquer comportamento anômalo à rede.

O diagnóstico apresentado pelo sistema ao gerente da NOC (*Network Operational Control*) é gerado pelo mecanismo de inferência, que manipula a

base de casos em busca de ocorrências anteriores que apresentam semelhanças que possam contribuir para a resolução do problema corrente. Uma grande vantagem dos sistemas baseados em casos é a capacidade de aprendizado com as novas ocorrências. Situações até então desconhecidas, tornam-se "fonte de conhecimento" que alimentam a base de casos do sistema e ficam disponíveis para utilização de consultas futuras em casos iguais ou parecidos.

O domínio apresentado pela área de gerência de redes, incorpora nova tecnologias de redes muito rapidamente, gerando uma evolução e mudança nas redes muito acentuada. Conseqüentemente a variedade e complexidade dos problemas apresentados são crescentes. Com base no contexto apresentado, o uso dos sistemas especialistas baseados em casos traz inúmeras vantagens se comparado com outros recursos disponíveis, como por exemplo os sistemas especialistas baseados em regras. Dentre as principais vantagens, podemos destacar a diminuição da fragilidade do motor de inferência, dada sua capacidade de incorporar novos casos, suportando, assim, mais facilmente as mudanças do domínio; e a possibilidade de um processo de aquisição de conhecimento mais natural.

Além da funcionalidade de auto-aprendizagem apresentada pelos sistemas especialistas baseados em casos, outro fator importante que os diferenciam dos sistemas especialistas baseados em regras é a forma como o mecanismo de inferência trabalha. As decisões tomadas nos sistemas baseados em casos baseiam-se em experiências anteriores, ou seja, quando uma nova solução é requerida para um novo problema inserido no sistema, sua resolução é realizada com base nos problemas anteriores, isto é, um caso semelhante ao analisado será recuperado da base de casos para que a nova solução seja apresentada. Após a resolução do novo caso, este torna-se disponível para

futuras consultas, resultando assim em um novo conhecimento para o sistema. Já os sistemas baseados em regras não apresentam tal flexibilidade. As decisões por eles apresentadas são tomadas com base em regras pré existentes no sistema, de maneira que estas regras são elaboradas na concepção do sistema e, para uma adaptação ou novo aprendizado, requer a atualização de tal base.

2. - Sistemas especialistas

Inteligência Computacional (IC) é uma área interdisciplinar, sendo atualmente aplicada, dentre outros campos do saber, na Ciência da Computação, nas Engenharias, na biologia,... sendo baseada em um grau muito grande na inteligência humana, ou seja, com a IC espera-se apresentar sistemas que apresentam capacidade de aprendizado, capacidade de resolução de problemas e mesmo de racionalização, mesmo que em um nível muito baixo se comparado com a capacidade humana. No ramo da IC, sistemas especialistas é uma das tecnologias que mais se destacam por apresentar resultados práticos de sucesso em diversas áreas de conhecimento [AZEVEDO, 1999]. Estes sistemas podem utilizar das mais diversas técnicas para raciocínio e representação do conhecimento como, por exemplo, RBC, regras, lógica nebulosa ou ainda redes neurais.

A engenharia do conhecimento, disciplina da IC que suporta o desenvolvimento desses sistemas, permite modelar a perícia dos especialistas humanos e armazená-la em sistemas computacionais não convencionais denominados sistemas especialistas. Sua tecnologia pode ser empregada para a conservação, organização e manutenção do conhecimento. Ela possibilita o emprego de computadores para auxiliar na tomada de decisões em situações nas quais, até alguns anos atrás, não era possível a utilização de modelos computacionais, como é o caso de um sistema para gerenciamento de redes de computadores. Além disso, essa tecnologia permite formalizar a sistematização do conhecimento prático existente em um domínio específico do conhecimento [AZEVEDO, 1999]. Normalmente os sistemas especialistas são sistemas que armazenam a perícia humana, muitas vezes dificilmente encontrada na literatura

técnica. De acordo com FERNANDES [apud WATERMAN, 1986], a tecnologia utilizada de um sistema especialista (SE), está contida no conceito dos sistemas baseados em conhecimento, conforme Figura 1.

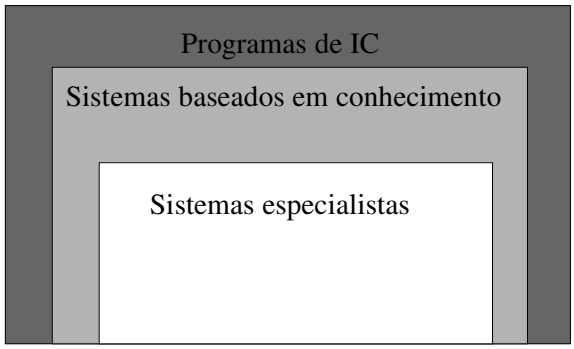


Figura 1 – Estrutura de contextualização dos sistemas especialistas

O âmago de um sistema especialista é o conhecimento sobre um domínio específico acumulado durante a construção do sistema. O conhecimento é explícito e organizado de forma a simplificar a tomada de decisões [CUNHA, 1998]. O SE pode explicar em detalhes como uma nova situação conduz a mudanças. Ele permite ao usuário avaliar o efeito de novos fatos ou dados e entender o relacionamento deles com a solução; avaliar os efeitos de novas estratégias ou procedimentos aplicados à solução.

Os SEs foram desenvolvidos para executar diversas tarefas em diferentes domínios [CAMARGO, 1999]. Na Tabela 1 estão representados os tipos de sistemas especialistas segundo as tarefas desenvolvidas e principais áreas de aplicação.

Dentre as apresentadas na Tabela 1, as tarefas de diagnóstico, projeto, observação e manutenção são utilizadas em um SGRBC de forma integrada, buscando fornecer ao administrador de redes um diagnóstico rápido e confiável.

Tabela 1 – Tipos de sistemas especialistas segundo a tarefa e a aplicação [CAMARGO, 1999]

T A R E F A	**A P L I C A Ç Ã O**
Diagnóstico	Deduz possíveis problemas a partir de observações ou sintomas: diagnósticos médicos, mecânicos.
Interpretação	Descreve a partir de observações: compreensão de fala, análise de imagens.
Predição	Deduz conseqüências a partir de situações: predição de tempo, de clima, de tráfego.
Projeto	Desenvolve configurações de objetos que satisfazem determinados requisitos ou restrições: projeto de circuitos digitais, projetos arquitetônicos.
Planejamento	Desenvolvem planos, cursos de ação: movimento de robôs, estratégia militar ou comercial.
Observação	Comparam observações de comportamento de sistemas, com características consideradas necessárias para alcançar objetivos: observação de rede de distribuição elétrica, controle de tráfego aéreo.
Debugging	Prescreve correções para defeitos
Instrução	Diagnostica e ajusta o desempenho de estudantes: toda a área de "computer-aided instruction".
Controle	Comanda de forma adaptativa o comportamento de um sistema: robôs, gerência de produção.
Manutenção	Desenvolvem e aplicam plano para consertar problema diagnosticado:manutenção de redes de comunicação, manutenção de sistemas de Computação.

2.1 Estrutura de um Sistema Especialista

Segundo [FERNANDES, 1996], um sistema especialista apresenta cinco componentes básicos: (i) base de conhecimento; (ii) máquina de inferência, (iii) subsistema de aquisição de conhecimento, (iv) subsistema de explicações e (v) interface do usuário, conforme mostrado na Figura 2.

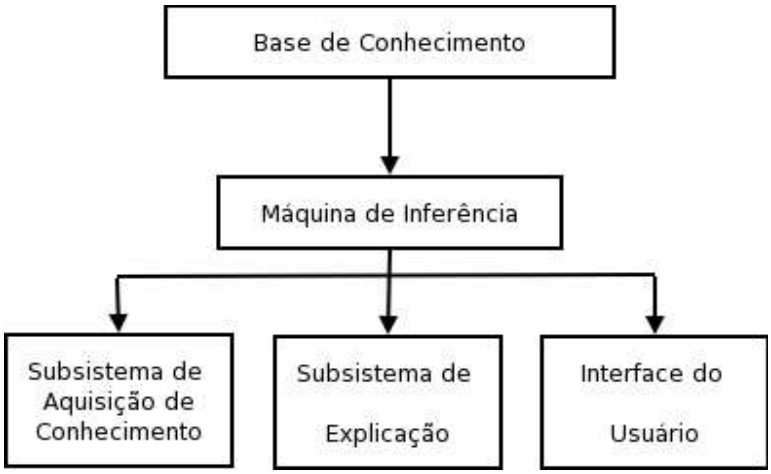


Figura 2 – Componentes básicos de um sistema especialista

2.1.1 – Base de conhecimento

A base de conhecimento é o local onde os fatos e regras que representam as regras de inferência do especialista humano residem. Boa porcentagem dos sistemas especialistas existentes utilizam regras como base para suas operações, por isso muitos são chamados de sistemas baseados em regras [CAMARGO, 1999]. Podemos ainda mencionar a representação de conhecimento através de redes semânticas ou ainda através de casos, sendo este último o objeto de estudos deste documento.

A base de conhecimento é formada pelas regras e procedimentos que o

especialista humano usa na solução de problemas. Quando o conhecimento de um especialista humano é capturado, denominamos este processo como a eliciação do conhecimento [AZEVEDO, 1999].

Pelo fato da base de conhecimento ser separada da máquina de inferência, o conhecimento contido na base é fácil de ser modificado. Quando alguma mudança na base de conhecimento for necessária, basta remover ou atualizar as regras existentes na base, ou ainda adicionar novas regras à mesma.

Representar o conhecimento por regras de produção ou simplesmente regras, é uma maneira bastante utilizada nos diversos sistemas especialistas existentes no mercado mundial, como por exemplo o ORPHEUS [SOUZA, 1999], que é um sistema especialista para análise de disfonias da voz, e o MYCIM [apud FERNANDES, 2003], sistema pioneiro, cujo objetivo era a realização de diagnósticos médicos. Da forma como estes sistemas foram desenvolvidos, os conhecimentos são representados através de pares condição - ação [PASSOS, 1989]. As regras são estruturas do tipo:

> Se < condição > então < ação >, onde:
>
> < condição > estabelece um teste cujo resultado depende do estado atual da base de conhecimento. Tipicamente o teste verifica a presença ou não de certas informações na base.
>
> < ação > executa automaticamente a ação necesária para a resolução do problema, ou então, propõe a solução do problema através da interação com o usuário do sistema por meio da *interface* opercional.

Existem duas maneiras de se validar as regras: através do Encadeamento para Frente (*Forward Chaining*), onde se partindo de um ponto inicial, chega-se a uma conclusão; e com o Encadeamento para Trás (*Backward Chaining*), que começa com uma previsão para a solução do problema

(hipótese) e procura-se valores para confirmá-la.

2.1.2 – Mecanismo de Inferência

O mecanismo de inferência trabalha, ou "raciocína", a partir do conhecimento contido na base e gera informações para o usuário, representando a estratégia que o perito emprega para resolver um problema específico. Após uma busca e ordenação das regras a serem avaliadas, o processo de inferência é direcionado. Funciona como um "supervisor", que dirige a operação sobre o conhecimento contido no sistema especialista. Uma máquina de inferência toma decisões e julgamentos baseados em dados simbólicos contidos na base de conhecimento [FERNANDES, 2003].

As funções básicas da máquina de inferência são a inferência e controle. Depois de iniciado o sistema, a máquina de inferência busca na base de conhecimento casos e regras, comparando-os com as informações fornecidas pelo usuário. Sua operação consiste em uma busca específica e a combinação dos fatos e regras fornecidos através do usuário e comparados com o conteúdo existente na base de conhecimento.

2.1.3 – Subsistema de aquisição de conhecimento

O subsistema de aquisição de conhecimento é utilizado para alimentação da base de conhecimento. Através dele é possível introduzir novos conhecimento, remover ou alterar os antigos.

2.1.4– Subsistema de explicações

O subsistema de explicações é utilizado como uma "ferramenta" de argumentação do "Por quê?" ou "Como?" uma determinada conclusão foi a sugerida. Podemos dizer que este módulo é o responsável por explicar ao usuário a linha de raciocínio utilizada pelo sistema especialista para chegar à solução proposta.

2.1.5 – Interface com o usuário

A interface com o usuário é a parte do sistema especialista utilizada para estabelecer um meio de comunicação entre o usuário e o sistema. Sua interface pode ser na forma de menus, perguntas e representações gráficas que são exibidas na tela do computador. A interface do usuário também exibe todas as perguntas, respostas e resultados de consultas, além de ser o meio utilizado para a inserção dos dados e informações requeridos pelo sistema especialista para o usuário.

2.2– Raciocínio Baseado em Casos (RBC)

O Raciocínio Baseado em Casos é uma técnica de Inteligência Computacional que resolve novos problemas através da recuperação e adaptação de soluções anteriores, ou seja, os sistemas especialistas que têm o conhecimento humanístico e literário representado através de casos, buscam as soluções para os problemas atuais comparando-os com as experiências passadas. O caso que mais se assemelhar com o problema atual será utilizado como referência para a resolução do problema. Segundo [FERNANDES, 2003],

este procedimento pode ser detalhado em alguns passos:

- Identificação de um problema a ser resolvido (problema de entrada);
- Definição das principais características que identificam este problema;
- Busca e recuperação na memória de casos com características similares;
- Seleção de um ou mais dentre os casos recuperados;
- Revisão deste(s) casos(s) para determinar a necessidade de adaptação;
- Reutilização do caso adaptado para resolver o problema de entrada;
- Avaliação da solução do problema de entrada;
- Inclusão do caso adaptado na memória de casos (aprendizagem);

Ainda de acordo com [FERNANDES, 2003], podemos citar como principais características do RBC:

- Aquisição do conhecimento: o conhecimento presente em um sistema de RBC fica armazenado na própria base de dados;
- Manutenção do conhecimento: um usuário do sistema pode ser habilitado a adicionar novos casos na base de casos sem a intervenção do especialista;
- Eficiência crescente na resolução de problemas: a reutilização de soluções anteriores ajuda a incrementar a eficiência na resolução de novos problemas. O RBC armazena as soluções que não obtiveram sucesso, assim como aquelas bem sucedidas. Isto faz com que eventuais insucessos sejam evitados.
- Qualidade crescente nas soluções: quando os princípios de um domínio não são bem conhecidos, regras não são a melhor solução. Nesta situação, as soluções sugeridas pelos casos refletem o que realmente aconteceu em uma determinada circunstância.

- Aceitação do usuário: um dos pontos chaves para o sucesso de um sistema de IC é a aceitação do usuário. Os sistemas de RBC podem comprovar o seu raciocínio, apresentando ao usuário os casos armazenados na base.

O entendimento da técncia de RBC, esta implícito em assumir alguns princípios da natureza [KOSLOSKY, 1999]:

- Regularidade: como o mundo, na grande parte das vezes, é regular, as ações executadas nas mesmas condições tendem a ter resultados similares ou iguais. Dessa forma, soluções para problemas similares são utilizadas para iniciar a resolução de um outro.
- Tipicalidade: alguns tipos de problemas têm a tendência de tornarem-se repetitivos, principalmente dentro de um único domínio específico. As razões para experiências serão, provavelmente, as mesmas para futuras ocorrências.
- Consistência: mudanças pequenas ocorridas no mundo pedem apenas pequenas mudanças na forma de interpretá-lo. Conseqüentemente, exigem pequenas mudanças nas soluções de novos problemas.
- Facilidade: as coisas não se repetem da mesma forma. As diferenças possuem a tendência de serem pequenas, e pequenas diferenças são fáceis de serem compensadas.

Dentro do contexto apresentado podemos perceber que, a implementação de um *software* de gerência de redes através de RBC é um dos meios possíveis mais adequado, pois a capacidade de auto-aprendizagem do sistema e a facilidade de similaridade que os sistemas especialistas baseados em casos possuem são fatores determinantes.

2.2.1– Casos

Antes de tratarmos especificamente do ciclo de desenvolvimento de um sistema de RBC é preciso entender o que é um caso. Caso é forma de conhecimento contextualizado representando uma experiência que ensina uma lição útil. Lições úteis são aquelas que têm o potencial para ajudar o raciocinador a alcançar uma meta ou um conjunto de metas ou advertem sobre a possibilidade de uma falha ou apontam para um problema futuro [KOLODNER, 1993].

Segundo [BUTA, 1997], um caso é uma abstração de uma experiência, que deve estar descrita em termos de conteúdo e contexto. Estas experiências precisam ser organizadas em unidades bem definidas, formando a base de raciocínio ou memória de casos. Os casos representam o próprio conhecimento presente no sistema.

Um caso pode ser representado de diversas maneiras, entretanto, a descrição através de atributos que identifiquem um caso é a mais usual. A dificuldade de definir quais atributos são mais importantes e a necessidade de um conhecimento profundo do especialista no domínio, tornam a fase de representação dos casos uma das mais delicadas e, sem dúvida alguma, a mais importante de todas.

Um caso possui dois componentes: descrição do problema e descrição da solução. [KOLODNER, 1993] insere um terceiro componente: o resultado da aplicação da solução do problema. Este componente é responsável pela percepção do sistema de RBC em determinar em quais situações o sistema obteve sucesso ou fracasso, servindo como referência para que erros cometidos

anteriormente não se repitam.

Nas palavras de [KOLODNER, 1993], um caso possui três partes assim descritas:

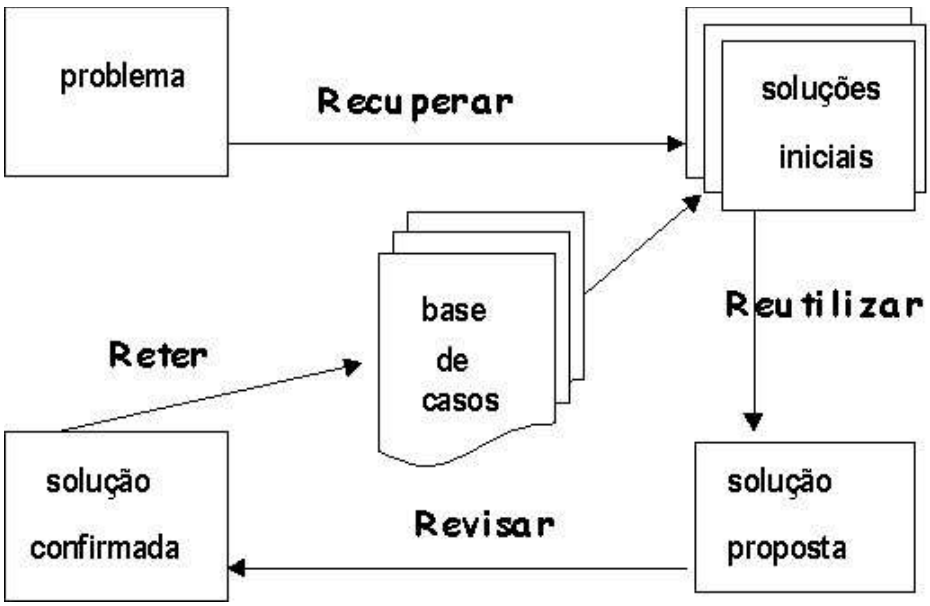
1. Descrição do problema, que mostra o estado da "situação" de quando o caso aconteceu com a representação do problema. Engloba:
   - Metas a serem alcançadas na solução do problema;
   - Restrições contidas nestas metas;
   - Características e relações entre as partes da situação.
2. Solução do problema identificado na descrição do mesmo, sendo os componentes das soluções:
   - A solução propriamente dita;
   - Passos necessários para a resolução do problema;
   - Justificativas do porquê das decisões tomadas na resolução do problema;
   - Soluções possíveis que não foram escolhidas (soluções alternativas) ou soluções que foram excluídas por não serem aceitáveis;
   - Expectativas dos resultados.
3. Resultado da aplicação, sendo seus componentes:
   - O próprio resultado;
   - Se o resultado preencheu ou violou as expectativas;
   - O resultado foi sucesso ou falha;
   - Explicação da violação da expectativa ou da falha;
   - Estratégia para reparo que poderia evitar um problema;
   - Apontar para uma provável próxima solução.

A grande dificuldade dos sistemas de Raciocínio Baseado em Casos é

decidir o que será armazenado e encontrar a estrutura ideal para descrever o conteúdo do caso, de modo que a estrutura definida consiga abranger a totalidade dos casos possíveis em um determinado domínio.

2.3 - Ciclo RBC

O processo de desenvolvimento de um sistema de RBC em qualquer domínio é uma tarafa interativa e não se encaixa em uma metodologia genérica. Portanto, serão descritas, a seguir, algumas destas etapas, consideradas mais importantes, tais como: Representação dos Casos; Indexação; Recuperação; Ajuste da Situação; Aprendizagem [LEE, 1998]. A Figura 3 mostra o ciclo do RBC.



Fonte: Adaptação de [LEE, 1998].

Figura 3 – Ciclo RBC.

2.3.1– Representação dos casos

Um caso é uma parte contextualizada de um problema que representa uma valiosa experiência de onde pode-se tirar boas lições no futuro. Dessa forma, um caso pode ser visto sob dois aspectos: o que ele pode ensinar e o contexto no qual ele se insere. Determinar o que é um caso, é o primeiro problema na modelagem do RBC [LEE, 1997]. São os casos que contêm elementos para que a solução do problema proposto seja alcançado.

Em [LEE, 1996] é mencionado que a representação dos casos é a representação do conhecimento. Há momentos em que algum conhecimento do especialista é representado em um sistema de RBC, entretanto, nos casos é que está contido o conhecimento que servirá para sugerir uma solução para o problema.

O problema na representação de RBC é, essencialmente, o que se deve guardar de um caso, a estrutura apropriada para a descrição do mesmo e como a memória de casos deve ser organizada e indexada para efetuar-se satisfatoriamente a recuperação e reutilização.

2.3.1.1– Modelagem dos casos

De acordo com [LEE, 1996], a representação de casos é realizada através de uma lista de atributos com valor, sendo que as características que identificam um caso referem-se ao par atributo-valor. Existem três componentes básicos na representação dos casos: a descrição do problema, a descrição da solução e o resultado.

A descrição do problema, nada mais é do que a atribuição de características que irão descrever o problema de entrada, podendo ter forma de nomes, números, funções, textos, restrições e as relações existentes entre suas partes, que juntos irão determinar a similaridade com outro caso. [LAGEMANN, 1998] coloca que se pode utilizar arquivos e Banco de Dados para armazenar informações sobre os casos. Este assunto será tratado com maior ênfase mais adiante.

[KOLODNER, 1993] apresenta duas diretrizes para decidir a inclusão, ou não, de determinada descrição:

- Inclusão de todos os aspectos que tenham servido para atingir o objetivo representado pelo caso;
- Inclusão das características normalmente utilizadas para descrever casos do tipo em estudo.

Não há definição por parte dos pesquisadores, sobre quais informações deveriam fazer parte da descrição de um caso. Entretanto, para definir um caso, é muito importante levar-se em consideração a funcionalidade da informação e a facilidade de sua aquisição. As características de descrição do caso devem conduzir o raciocínio a solução do problema inicial e informar qual o resultado da aplicação desta solução no problema inicial.

A descrição da solução é composta da solução sugerida pelo RBC propriamente dita, a metodologia utilizada para chegar a resolução do problema, soluções aceitáveis que não foram escolhidas ou soluções não aceitáveis e as expectativas sobre o resultado, além de conter a justificativa do porquê da escolha de determinada resolução.

O resultado da aplicação da solução no problema inicial, apresenta se as expectativas foram atendidas ou violadas de alguma forma, se houve sucesso ou

falha, qual o motivo da violação das expectativas ou da falha quando houver, uma estratégia de reparo, como evitar que o mesmo problema ocorra novamente e, por último, deve conter uma nova solução possível, no caso de insucesso com a solução proposta inicialmente.

2.3.1.2 - Modelagem da memória

A base de casos nada mais é do que o conjunto de casos que representa o conhecimento de um sistema de RBC. Os modelos de memória nada mais são do que estruturas de organização dos casos. A memória irá compreender a base de casos e os mecanismos de acesso da base a outros módulos da arquitetura do sistema.

"A filosofia de modelagem de memória refere-se ao tipo de modelo de memória utilizado na representação do conheciemnto."

[LEE, 1996]

Questões como eficiência e tempo computacional estão diretamente ligadas ao tipo de memória escolhido para organizar os casos.

[LEE, 1996] coloca que existem dois aspectos que devem ser enfocados separadamente, quando do tratamento de modelagem de memória:

- O tipo de filosofia de representação que simula um determinado sistema, podendo ser uma rede semântica, memória episódica ou mesmo memória dinâmica.
- Em segundo, Quando se trata de um enfoque de implementação, a modelagem irá tratar do estilo de organização adotada para os casos, como por exemplo, a forma de estruturação da memória de casos.

[DELPIZZO, 1997] opina que, apesar do modelo de representação através de redes semânticas ter sido um dos primeiros, o mesmo não consegue representar o conhecimento totalmente. Desta forma, as redes semânticas não serão abordadas neste trabalho.

A representação do conhecimento através de redes semânticas é uma tentativa de simular o modelo psicológico de memória associativa humana. Ela modela o conhecimento como um conjunto de pontos chamados nós ou nodos, conectados por ligações chamados arcos que descrevem as relações entre os nós [PASSOS, 1989].

Em 1983, Tulving apresentou o modelo de memória episódica com o intuito de complementar o modelo de redes semânticas. Então, Roger Schank desenvolveu o modelo de Memória Conceitual, dos *scripts,* conseguindo assim melhores resultados que Tulving, evoluindo posteriormente para os MOPs (*Memory Organization Packets*), pacotes de organização de memória do modelo de Memória Dinâmica.

Os *scripts* são estruturas de informação que auxiliam a compreensão de situações do comportamento padronizado. Foram propostos por MELCHIORS [apud SCHANK e ABELSON, 1977] e inspiraram o estudo de sistemas de Raciocínio Baseado em Casos. Os *scripts* são úteis porque, no mundo real há padrões para a ocorrência de eventos. Contudo, o conceito de um *script* não é compartilhado por todos [RIESBECK & SCHANK, 1989] já que cada memória compreende um *script* sobre uma experiência a partir do próprio ponto de vista. Portanto a teoria dos *scripts* não é uma teoria completa. Os *scripts* contém o conhecimento normativo, mas não o conhecimento da experiência. MOPs serão tratados dentro de memória dinâmica.

2.3.1.2.1– Memória Dinâmica

Esta forma de estruturação da memória, tem a qualidade de se transformar ao longo do uso. Sua dinâmica está principalmente no fato da geração automática dos MOPs. Neste tipo de estrutura, os casos são caracterizados pelos episódios aos quais estão associados e, seus atributos não são apenas os próprios, mas os atributos de suas abstrações e ligações que modelam o contexto do caso.

Segundo [LEE, 1996], a memória dinâmica usa uma estrutura hierárquica de pacotes de organização de memória. Os MOPs são caracterizados pelos seguintes objetos: Normas, são as características comuns aos casos indexados ao MOP; os índices, que diferenciam os casos indexados ao mesmo MOP, sendo representado por um nome e um valor; e por último os próprios casos.

Os MOPs têm como objetivo representar eventos padronizados e surgiram de uma evolução dos *scripts*. Os MOPs são organizados em estruturas que reúnem eventos similares através de abstrações e hierarquias do tipo "todo-parte". Quanto ao conteúdo, os MOPs são estruturas de conhecimento que representam experiências. MOPs representam eventos através de cenas que abrangem situações e são representadas por informações normativas e descritivas. As cenas são suposições associadas a situações de uma experiência e, consequentemente, estão sujeitas a mudar com a experiência [SCHANK, KASS & RIESBECK, 1994]. Eles diminuem a redundância e permitem a percepção das informações sob vários pontos de vista, traduzindo as

expectativas dos diversos participantes de uma determinada situação. Dessa forma, a entidade básica da Memória Dinâmica são os Pacotes de Organização de Memórias. A existência do modelo de Memória Dinâmica permite representar computacionalmente um modelo de organização de memória que compreende em recordar, entender, experimentar e aprender. Os MOPs permitem representar o conhecimento sobre classes de eventos de duas formas:

- Instâncias, que representam casos, eventos ou objetos;
- Abstrações, que representam versões generalizadas de instâncias ou outras abstrações.

O desenvolvimento da teoria mais geral de Schank conduziu-o aos pacotes de organização de memória episódios (E-MOPs), implementado no sistema CYRUS [KOLODNER, 1993]. A idéia básica é organizar casos específicos, que compartilham propriedades similares sob uma estrutura mais geral, ou seja, E-MOP. Uma E-MOP, contém casos, as propriedades comuns entre eles e as feições que os diferenciam.

Considerando a memória de casos como sendo uma árvore de discriminação, cujos nodos[3] são esses objetos, O papel de uma E-MOP é indexar a estrutura de forma a armazenar, buscar e recuperar os casos. O modelo é dinâmico, pois novas E-MOPs são criadas a medida que novos casos são inseridos, para poder discriminá-los em relação aos anteriormente armazenados. O processo que permite a indexação automática de novos casos, tende a levar a uma explosão do número de índices à medida que cresce o número de casos.

3 Representam objetos individuais, categorias de objetos, conceitos ou eventos.

2.3.1.2.2 - Modelo de categoria de exemplares

Neste modelo de memória considera-se que os casos do mundo real podem ser vistos como exemplares de acontecimentos. Aqui, uma memória de casos é uma rede semântica de categorias e casos ligados por relações semânticas de hierarquia, de semelhanças ou diferenças. Cada caso é associado com uma categoria de acordo com as características, que definem se este deve ou não ser enquadrado na categoria. Feições similares de um caso apontam para as de outro caso ou categoria, assim como categorias de pequenas diferenças também são ligadas. Essa rede compõe uma estrutura de conhecimento genérico do domínio que permite alguma recuperação do raciocínio do sistema para gerar explicações. Ao armazenar um novo caso, um caso anterior semelhante ao atual é buscado no banco de casos. Se houver pequenas diferenças entre os dois somente um é retido, ou então, uma combinação entre os dois casos é criada e armazenada.

Um exemplo de uma estrutura para armazenamento de um problema através de casos, segundo o modelo de categoria de exemplares, é mostrado na Figura 4, extraída do sistema DUMBO [MELCHIORS, 1999].

Descrições de casos tendem a ter dezenas, ou até centenas, de atributos para descrever cada objeto. Entretanto, os problemas da escolha da representação vão além do tamanho e da complexidade intrínsica do problema representado. As decisões de projetos, listadas abaixo, são feitas com o apoio das informações eliciadas do especialista através da engenharia do conhecimento.

- Que estrutura de representação é facilmente compreendida pelo usuário e, ao mesmo tempo, permite um gerenciamento eficiente no

computador?

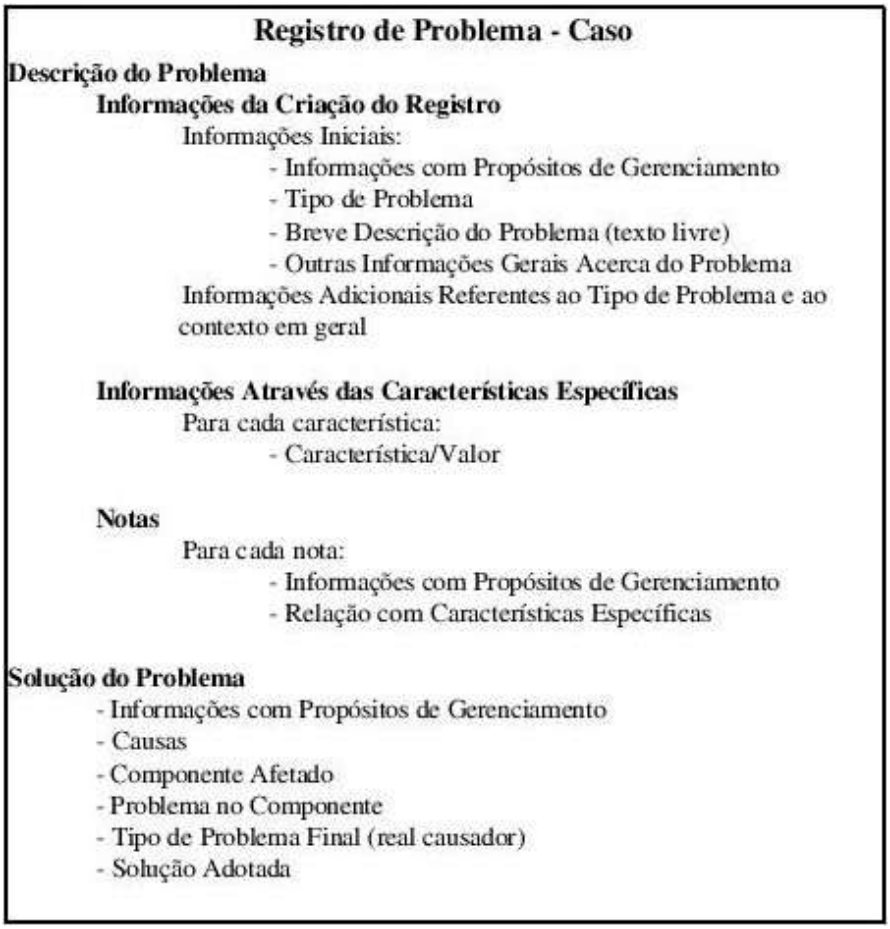


Figura 4 – Componentes de um caso para um SGRBC

Casos podem ser armazenados no sistema nos mais diferentes formatos, no entanto, muitos deles são excessivamente complexos para serem manipulados pelo desenvolvedor do sistema ou para serem compreendidos pelos usuários do sistema. A representação deve considerar formatos que tenham uma correspondência natural com a

forma como a informação costuma estar disponível para evitar o desperdício de processamento em traduções de uma estrutura para outra. Como exemplo, se os casos se constituem em formulários para controle de acesso físico, uma boa solução é utilizar, dentro do sistema, um modelo semelhante aos formulários já utilizados.

- Quais os casos que devam ser representados?

  Os casos registram experiências concretas que ajudam a auxiliar a alcançar um determinado objetivo, porém, nem todos os casos devem ser selecionados para serem inclusos no sistema. Apenas aqueles que contém uma lição útil [KOLODNER, 1993] em relação aos demais devem ser armazenados. Isso significa que casos repetem uma situação anterior já representada, ou com pequenas modificações não deveriam ser incluídos, uma vez que essas diferenças podem ser compensadas pelos algoritmos de adaptação do sistema. Ao mesmo tempo, os casos representados não devem divergir excessivamente dos problemas à serem resolvidos pelo sistema sob o risco do sistema conter casos que nunca serão utilizados. O equilibrio entre as duas restrições é difícil de ser alcançado, uma vez que as medidas de diferença e alcance do domínio são subjetivas, dependendo de uma avaliação particular da aplicação.

- Qual granularidade da informação a ser representada?

  Por exemplo, um projeto arquitetônico de uma casa constitui um caso, contém as necessidades do cliente e a solução proposta baseando-se nos detalhes do projeto. Porém, considerar todo o projeto como um

caso, não seria de muita utilidade, pois dificilmente dois clientes teriam as mesmas necessidades para a casa inteira. Mesmo com um grande número de casos armazenados, a possibilidade de recuperação de um caso anterior seria muito pequena. Nesse exemplo, a melhor solução seria particionar o projeto em um número maior de casos, cada um enfocando uma necessidade do cliente. Embora mais casos devessem ser recuperados, o sistema apresentaria melhores soluções em cima de um número menor de projetos armazenados.

O problema de aquisição e representação de casos em RBC, inicia, portanto, com uma análise para definir o grau de disponibilidade dos casos e quanta informação adicional deve ser eliciada através do especialista. O próximo passo é determinar a melhor forma de representação de um caso, tendo em conta a facilidade de compreensão do usuário e a eficiência no armazenamento por computador. Outras decisões incluem a decisão de quais, entre os casos disponíveis, são os que realmente devem armazenados e qual o grau de granularidade do conhecimento que compõe o caso. O banco assim construído deve contar ainda com uma forma de indexação eficiente para possibilitar ao sistema atingir uma boa performance na resolução dos problemas.

Alguns estilos de organização de casos, como cita [LEE, 1996], são: memória plana, memória hieráquica, banco de dados relacional, redes semânticas, redes discriminatórias, redes de características compartilhadas e árvores de decisão.

2.3.2- Indexação dos casos

A indexação se faz necessária para que os casos possam ser recuperados, sendo que ela determina quais os atributos que devem ser comparados para se avaliar a similaridade entre o caso de entrada e os casos da base.

Para [KOLODNER, 1993], indexação de casos é a associação de rótulos em casos, de maneira a caracterizá-los para posteriormente recuperá-los em uma base de casos. Esta não é uma tarefa simples. Para construir uma boa coleção de índices para um conjunto de casos é necessário ser ter em mente a importância de um bom índice e como escolhê-lo.

Já para [LEE, 1998], a indexação é a essência do RBC, pois orienta a avaliação de similaridade. A similaridade refere-se à comparação entre o caso de entrada e os casos da base para determinar quão semelhantes eles são. Portanto, os índices são utilizados para determinar o grau de similaridade entre um caso e outro.

A escolha dos índices deve ser realizada cuidadosamente. Características superficiais são facilmente extraídas de um caso, entretanto, estas características podem ser menos úteis do que índices mais complexos obtidos pela combinação e composição das características que distinguem os casos de cada lição que ele pode ensinar.

A escolha de bons índices pode requer uma interpretação ou processo de elaboração durante o qual as características funcionais podem ser feitas de forma adequada [KOLODNER, 1993].

1. Deve-se procurar formas de descrever ou representar o caso, isto é, tarefas e domínios devem ser analisados para obter descritores funcionalmente relevantes que deveriam ser utilizados como índices para o caso.

2. Para qualquer caso particular, é preciso designar quais partes da descrição ou quais características atuarão como índices (seleção de índices).

Um bom índice deve ser preditivo e abstrato o suficiente para fornecer cobertura, no entanto, concreto suficiente para ser reconhecível.

Características preditivas são combinações de descritores de um caso responsáveis pela solução, que influenciam no resultado ou caracterizam o problema. Índices devem ser mais abstratos do que os detalhes de uma caso particular. Embora casos sejam específicos, índices para casos precisam ser escolhidos para que o caso possa ser usado amplamente em uma coleção de situações apropriadamente. Enquanto índices precisam ser geralmente aplicáveis, eles também precisam ser suficientemente concretos, para que possam ser reconhecidos com pouca inferência. Este índices devem ser escolhidos para fazer os tipos de predição que seriam úteis em futuros raciocínios. Índices utéis são aqueles que rotulam um caso como sendo capazes de dar guias sobre as decisões que o motor de inferência irá tratar.

Os índices podem ser selecionados tanto manualmente quanto automaticamente. A seleção manual analisa caso a caso para determinar quais características descritas que determinam as variações sobre as conclusões. Os métodos automáticos buscam quantificar as diferenças entre os casos e os relacionamentos entre feições dos problema e soluções adotadas. Algumas formas de selecionar índices são descritas a seguir [KOLODNER, 1993]:

- Técnicas baseadas em explicação.

  Os casos são analisados individualmente para determinar os elementos do problema que são utilizados para construir a solução. Esses

elementos são utilizados como índices.

- Índices baseados em conhecimento do domínio.

  Utilizando protocolos retrospectivos sobre os casos são extraídas as correlações entre elementos e conclusões nos casos particulares e no domínio como um todo (processos abstratos). Esses elementos e processos são utilizados como índices.

- Análise matemática.

  Todos os elementos do domínio e suas dimensões são analisados numericamente para identificar quais as feições que determinam ou influenciam as conclusões. Os elementos e valores computados são utilizados para construir os índices. São os métodos utilizados nos sistemas MEDIATOR ABEL [apud SIMPSON, 1985] e CHEF [ABEL, 1996].

- Índices baseados nas diferenças entre os casos.

  O sistema analisa casos similares e os indexa especificamente nas características que os diferenciam, como no sistema CYRUS [ABEL, 1996].

- Métodos de generalização.

  O método utiliza a definição de casos abstratos a partir dos elementos compartilhados entre diversos casos armazenados. Esses elementos são utilizados para a indexação dos casos abstratos, enquanto que as funções que os diferenciam indexam os casos individuais.

- Métodos de aprendizado indutivo.

  Identificam os elementos que determinam as conclusões para serem utilizados como índices. Esses métodos são muito difundidos especialmente pela utilização do sistema *ReMind* [ABEL, 1996] e variações do algoritmo para indução de regras *ID3* [ABEL, 1996].

Apesar de os métodos automatizados auxiliarem na escolha de bons índices, na prática, os sistemas cujos índices foram definidos manualmente tendem a ter melhor desempenho do que aqueles puramente processados.

2.3.3- Recuperação dos casos e grau de similaridade.

[KOSLOSKY, 1999] coloca que o objetivo desta etapa é a recuperação dos casos que possam auxiliar o raciocínio que se produz nos passos seguintes. A recuperação é feita usando as características do novo caso que são relevantes na solução de um problema.

Um algoritmo de recuperação de casos é o responsável por encontrar, a partir da descrição de um problema ou situação, um pequeno conjunto de casos similares ao problema corrente que seja útil para a identificação de sua solução. A busca pelos casos similares não deve considerar, porém, apenas a descoberta de algumas dimensões da descrição do problema similares à situação. Na identificação da similaridade entre os casos, alguns atributos são mais importantes que outros e esta valorização pode variar de acordo com os objetivos almejados pelo sistema. Assim, a recuperação de casos similares

envolve considerar que os casos similares ao problema corrente são aqueles que são similares *nas dimensões que auxiliam o sistema a realizar suas tarefas ou atingir os objetivos desejados* [KOLOSKY, 1993].

A etapa de recuperação dos casos é iniciada com uma descrição do problema e encerrará quando for encontrado o melhor caso. Segundo [FERNANDES, 2003], esta tarefa divide-se em:

- Identificação das características: informa ao sistema quais as características do caso atual.
- Unificação inicial: recupera um conjunto de candidatos possíveis.
- Busca: processo mais elaborado que irá selecionar qual o melhor candidato entre os casos recuperados no casamento inicial.
- Seleção: os casos são ordenados conforme algum critério ou de acordo com a métrica de classificação, sendo o caso escolhido aquele que possui a mais forte similaridade com o novo problema.

Ainda segundo [FERNANDES, 2003], três fatores são fundamentais na etapa de recuperação:

- Eficiência: velocidade de recuperação dos casos pelo sistema.
- Precisão: grau de casos recuperados que podem ser utilizados para alcançar o objetivo proposto.
- Flexibilidade: grau de recuperação de casos para raciocínios inesperados.

A similaridade é o ponto crucial do RBC, pois todo o raciocínio que dá fundamento a esta técnica encontra-se aqui. Através dela avalia-se a similaridade do caso a ser solucionado (problema de entrada) com os casos candidatos. O que faz um caso ser similar ou não a outro é a semelhança das características que realmente representam o conteúdo e o contexto da

experiência [DELPIZZO, 1997].

Um caso será similar a outro quando as características que representam realmente o conteúdo e o contexto do mesmo forem semelhantes. As características semelhantes combinadas entre si, determinam a sua solução [LEE, 1996].

De acordo com FERNANDES, [apud AAMODT e PLAZA, 1999], a avaliação da similaridade subdivide-se em dois gurpos:

- Similaridade sintática: é a mais superficial, sendo os atributos comparados por sua semelhança sintática. Whitaker propôs avaliar por categorias, tais como sinônimos, categorias ordinais, análise de perfil, clusterização e qualificadores. A avaliação dos atributos pode ser feita de diversas maneiras, dependendo somente da natureza das dimensões.
- Similaridade semântica: é uma avaliação mais profunda, que abrange o significado dos casos e compara um com o outro. Thagard e Holyoak [apud LEE, 1996] expõem uma abordagem para comparar semanticamente os casos, com base em um sistema de referência léxica com uma estrutura de 30 mil palavras hierarquicamente separadas.

Já os métodos de recuperação dividem-se em:

- Métodos numéricos: utilizam-se de uma função numérica para medir o grau de similaridade entre dois casos, sendo conhecidos Métrica de Similaridade.
- Métodos eliminatórios: são aqueles que utilizam restrições para reduzir o espaço solução da busca por casos similares na memória de casos. Podem ser combinados também com outros métodos, sendo

também uma boa escolha para sistemas que se propõe a resolver tarefas distintas, possuindo memórias formadas com várias bases de casos.

- Métodos de classificação: é apropriada para sistemas em que a memória é classificada em categorias. Pode ser implementado através da clusterização de todos os casos da memória, ou pode-se buscar os casos somente dentro de uma categoria.

Em seqüência aos métodos de recuperação, podemos classificar os tipos de busca em três:

- Busca direta: é aquela que avalia diretamente as características indexadas do caso.
- Busca numa estrutura de índices: é aquela realizada numa estrutura de índices gerada a partir dos casos.
- Busca além do domínio: transcede o conhecimento da base de casos e procura casos similares em outra base de casos que proporcione conhecimento maior.

[KOSLOSKY, 1999], define as tarefas na recuperação de casos como:

- Avaliação e métrica de similaridade.
- Recuperação.
- Seleção.

A avaliação é o resultado do valor numérico dado para avaliar a similaridade entre os dois casos, número este que representa o conhecimento do especialista.

Uma das maneiras de se fazer a aquisição do conhecimento com objetivo de saber o peso dos índices, é solicitar que o especialista faça uma lista em ordem de importância [LEE, 1997].

O estabelecimento de métricas de similaridade em um RBC é uma das etapas mais importantes e cruciais para a eficiência da metodologia como um todo, pois são elas que definem o quão semelhante e útil pode ser um caso armazenado para a resolução do novo problema.

A determinação da medida de similaridade é um importante componente para identificar a utilidade do caso. Deve-se considerar também, que o grau de utilidade de um caso depende dos propósitos a que ele se destina e quais dos seus aspectos foram relevantes no passado.

2.3.3.1– Método *matching* e *ranking*

De acordo com [KOLODNER, 1993], a procura por casos similares na base de casos é realizada por heurísticas de *match* e *ranking*, que irão escolher os casos mais úteis do conjunto.

Match, é um processo que compara dois casos entre si e determina o grau de similaridade entre os mesmos. Já *ranking* trata de ordenar os casos *partialy-matching* conforme sua utilidade, ou seja, a determinação de qual é o melhor que os outros.

Kolodner informa que as entradas de dados para os processos de *Match Ranking* são as seguintes:

- Novo problema que o sistema está tentando resolver.
- O objetivo de uso para os casos recuperados.
- O conjunto de casos recuperados
- Índices associados com cada caso recuperado.
- Critério de *match* razoável, indicando quando os procedimentos de *match* e *ranking* devem parar.

2.3.3.2 – Método "O vizinho mais próximo"

A técnica do vizinho mais próximo (*nearest neighbour*) é talvez a mais usada para o estabelecimento da similaridade [WATSON, 1997]. Os aspectos de definição e identificação dos índices é fator fundamental para uma recuperação de sucesso. Garantidos estes aspectos, a técnica de busca indica em qual região do espaço o problema em questão está inserido. É a técnica mais indicada para problemas com bases de casos pequenas e com poucos atributos indexados, devido ao volume de cálculos necessários para determinar cada um dos atributos indexados e cada um dos casos.

A similaridade entre o novo caso e um caso na base de casos é determinada para cada atributo. Esta medida deve ser multiplicada por um fator peso. A somatória de todos os atributos é calculada e permite estabelecer a medida de similaridade entre os casos da biblioteca e o alvo. Isto permite estabelecer a medida de similaridade entre os casos da base de casos e o novo caso.

A fórmula da similariadade pelo vizinho mais próximo é apresentada na Equação 1:

$$Similaridade\ (N, F) = \sum_{i=1}^{n} f\ (N_i, F_i) \times w_i$$

Equação 1 – Fórmula da similaridade do Nearest Neigbour Retrieval

Onde:

N = Novo caso.

F = Casos existentes na memória de casos.

n = Número de atributos.

i = Atributo individual variando de 1 a n.

f = Função de similaridade para o atributo i nos casos N e F.

w = Peso do atributo i.

Este cálculo é repetido para cada caso da biblioteca para obter-se o *ranking* dos mesmos. As similaridades são usualmente normalizadas para um intervalo entre zero e um[4]. A grande dificuldade deste método é a determinação dos pesos relativos das características. A limitação desta abordagem é a convergência para a solução correta e o número de recuperações. Em geral o tempo de recuperação aumenta linearmente com o número de casos.

Outras técnicas de recuperação podem ser encontradas na literatura, tais como Recuperação Indutiva (*Iductive Retrieval*), *Redundant Discrimination Networks (RDN)* e *Shared Feature Networks (SFN)*, apresentadas em [FERNANDES, 2003].

2.3.4 – Reutilização e técnicas de adaptação

[KOLODNER, 1993] coloca que, pelo fato de nenhum problema do passado ser exatamente igual a um problema atual, soluções passadas usualmente são adaptadas para solucionar novos problemas. A adaptação pode ser uma simples substituição de um atributo da solução por outro ou, então, uma complexa e total modificação na estrutura da solução.

De acordo com [LAGERMANN, 1998], o processo de adaptação pode ser realizado de diversas formas:

- Inclusão de um novo comportamento à solução recuperada.

4Zero quando sem similaridade, um quando a similaridade for exata.

- Eliminação de um comportamento da solução recuperada.
- Substituição de parte de um comportamento.

FERNANDES, [apud VERGARA, 1995] afirma em seu trabalho que existem dois tipos de adaptação generalizados descritos na literatura:

- Adaptação estrutural: neste processo a adaptação de regras é aplicada diretamente à solução armazenada no caso;
- Adaptação derivacional: neste processo as regras geradas para a solução original são rodadas novamente para gerar uma solução nova. Quando um caso é recuperado, o sistema verifica se as diferenças entre o caso proposto e o caso passado afetam algumas decisões básicas à solução armazenada no caso. A solução armazenada é adaptada pela re-execução das partes do processo da solução original e não mudando-a diretamente.

Quanto às técnicas de adaptação de um caso, [KOSLOSKY, 1999] apresenta:

- Adaptação nula: esta técnica indica que não é necessária nenhuma modificação, ou seja, a ação é simplesmente aplicar a solução de um caso recuperado à nova situação. Pode ser aplicada quando o raciocínio para uma solução é complexa, mas a solução é simples;
- Soluções parametrizadas: quando um caso é recuperado para aplicá-lo a uma situação determinada, as descrições dos problemas, passado e novo, são comparados por parâmetros específicos e as diferenças são usadas para modificar as soluções dos parâmetros na solução apropriada. As soluções parametrizadas são de valor, porque modificam uma solução existente não criando uma nova solução única;

- Abstração e especialização: é uma técnica estrutural que pode ser usada para realizar simples adaptações de uma forma complexa e gerar novas soluções. Quando um traço de uma solução não pode ser aplicada a um problema, dada a sua dificuldade, o sistema deve procurar abstrações deste traço na solução que não apresentem a mesma dificuldade;
- Reinstalação: esta técnica é um método de adaptação derivacional. Não opera na solução original, mas sim nos métodos que foram usados para gerar uma solução. Os meios da reinstalação substituem um passo em uma solução, selecionando e aplicando um plano de ação que gera este passo no contexto de uma situação comum.

Segundo [LEAKE, 1996], a grande dificuldade do RBC é a fase de adaptação. [WATSON, 1997] afirma que "*a menos que a adaptação possa ser feita facilmente e utilizando parâmetros bem compreendidos, caso contrário meu conselho é que deve ser evitada"* e coloca também "*que a adaptação, em muitos casos pode ser considerada o calcanhar de aquiles de RBC"*. No SGRBC em discussão, optar-se-á em utilizar uma etapa de interação do especialista com o sistema para que ele possa avaliar as reais necessidades de adaptação da solução armazenada com a necessidade de solução do caso em questão.

2.3.5 - Revisão

Após a recuperação do caso na base de casos e sua adaptação, é necessário que a solução proposta seja avaliada. Caso a solução proposta não

venha produzir um resultado satisfatório, então esta solução deve ser reparada para que uma nova solução seja gerada. Após encontrar a solução correta para o caso, a experiência obtida deve ser aprendida, sendo armazenada para uso futuro. Estes processos podem ser vistos como obtenção da experiência e os processos de recuperação e adaptação podem ser vistos como a aplicação da experiência adquirida.

A avaliação da solução é resultado da aplicação da solução proposta em um ambiente e avaliação dos resultados ocorridos. O ambiente pode ser representado por um ambiente de simulação apto a gerar os resultados corretos a uma solução. Um exemplo é o sistema CHEF [ABEL, 1996], em que a solução proposta (no caso um prato culinário) é aplicada a um modelo interno, que é considerado eficiente para fornecer a avaliação necessária à solução fornecida [AAMODT, 1994]. De modo geral, porém, essa etapa é avaliada no ambiente real, para o problema real. Os resultados da execução da solução podem variar desde alguns segundos, quando aplicada, por exemplo, para corrigir automaticamente problemas de configuração em uma rede, até vários meses, quando aplicada, por exemplo, a um tratamento médico. Durante o período de execução, portanto, um caso pode ser aprendido, sendo já mantido armazenado na base, entretanto, deve-se sinalizar na base que o caso não foi, ainda, avaliado [AAMODT, 1994]. A Figura 5 mostra o esquema dos processos da revisão.

De acordo com Lundy Lewis em [LEWIS, 1995], existem, de modo geral, três modelos de execução para uma solução: execução manual, execução sem supervisão e execução supervisionda. Na execução manual, o usuário do sistema é responsável por interpretar a solução proposta e decidir se ela deve ou não ser executada. Ocorre na maior parte dos sistemas de RBC, em que o

sistema somente sugere, com base na experiência, no processo de recuperação e no processo de adaptação, uma boa solução para o problema, que é executada pelo usuário.

Figura 5 – Esquema do processo de revisão

Em alguns domínios, porém, quando uma solução é expressa em um programa de computador, um sistema de RBC pode ter a capacidade de executar a solução que ele propõe, realizando-a automaticamente sem intervenção ou controle humano. Quando isso ocorre, é formado um ciclo fechado de solução de problemas sem interveção humana, em que o problema é submetido ao sistema, um caso similar é recuperado e sua solução é adaptada para a situação corrente, a solução é executada pelo sistema e os resultados são inseridos na base de casos. Esse tipo de execução envolve, contudo, um alto

risco, pois delega muita responsabilidade ao sistema [LEWIS, 1995]. Um modo intermediário é a execução da solução proposta de modo automático com o controle do usuário. Nessa modalidade, o usuário pode permitir ou proibir a execução de uma solução que é sugerida pelo sistema, que a executa automaticamente se for autorizado.

Após uma solução ter sido avaliada, se o seu resultado for satisfatório, a experiência que foi obtida durante o processo de resolução do problema corrente deve ser armazenada no sistema. Isso é feito através do processo de aprendizado, que será tratado mais adiante. Se, entretanto, o resultado da avaliação mostrou que a solução proposta não produziu um resultado adequado, o caso deve ser reparado. A reparação do caso envolve a detecção dos erros da solução corrente e a recuperação ou geração da explicação para a ocorrência destes erros. Um segundo passo no processo de reparação utiliza, então, as explicações das falhas para modificar a solução corrente de modo que os erros não mais ocorram. A reparação de falhas pode ser executada diretamente ou pode ser avaliada e reparada novamente se necessário.

2.3.6 - Armazenagem para aprendizado

Um sistema baseado em casos deve também ser um sistema de aprendizagem, pois ele deve reutilizar suas próprias experiências obtidas. Os sistemas baseados em casos exigem uma aprendizagem baseada em conhecimento, que faça com que a compreensão do planejador do mundo determine o que deva ser aprendido e, quando deveria ser aprendido, tais como os planos de aprendizagem, expectativas de aprendizagem e críticas de

aprendizagem. Assim sendo, a aprendizagem feita pelo sistema é realizada através das lembranças dos casos passados.

Pode-se distinguir três tipos básicos de aprendizagem: aprendizagem de planos, aprendizagem por expectativas e aprendizagem baseado em casos.

Aprendizagem de planos é a criação e armazenagem de novos planos, como resultado do planejamento de situações que o planejador nunca encontrou antes. O planejador, dessa forma, tem que elaborar um novo plano e decidir quais características são melhores para indexá-lo na memória.

A aprendizagem por expectativas está ligada a indexação de planos na memória. Ela envolve a aprendizagem de características dentro de um domínio que é preventivo contra interações negativas entre os passos dos planos. Esta habilidade preventiva é utilizada para antecipar certos problemas e, então, procurar por planos na memória projetados para evitá-los. Uma vez que um destes prognósticos estão ativados, problemas podem ser evitados procurando na memória por um plano que os leve em consideração.

O planejamento baseado em casos envolve, por sua vez, três tipos de aprendizagem:

- aprender novos planos, que evitem problemas;
- aprender as características, que previnam o problema;
- aprender os reparos, que tem que ser feitos, se aqueles problemas surgirem novamente em circunstâncias diferentes.

Todos os 3 tipos de aprendizagem são apoiados por um vocabulário de planejamento, que descreve os planos relacionados aos objetivos diretos que eles satisfazem e suas respectivas interações.

COSTA [apud KOSLOSKY, 1999] coloca que um dos métodos utilizados para a educação é o aprendizado baseado em casos, onde os alunos

adquirem novos conhecimentos a partir da exploração de situações em uma grande biblioteca de experiências passadas. O propósito é tentar aplicar soluções já testadas no problema a ser resolvido. O enfoque é fazer com que os alunos não sejam meros aplicadores de regras pré-estabelecidas, mas buscar analogias, aplicá-las e tentar explicar suas próprias regras de decisão.

Em seu trabalho [KHAN e YIP, 1996] são citados, e apresentam 14 princípios pedagógicos que têm contribuído para o desenvolvimento de sistemas de ensino baseado em casos:

- Ensino baseado em estórias: explora o interesse inerente dos estudantes de aprender através de estórias e o desejo básico de professores e especialistas por contar estórias que encapsulam suas experiências;
- Ensino auto direcionado: os estudantes são motivados a refinar seus modelos cognitivos de um domínio por auto exploração de um ambiente. Este ambiente deve permitir que os modelos possam ser testados;
- Instrução significante: estórias são melhor apresentadas em um contexto que habilite o estudante determinar onde ele está no conteúdo e como ele poderia se conectar a outras estórias;
- Ensino dirigido ao impasse: estórias poderiam ser utilizadas para ilustrar pontos pedagógicos somente quando o estudante necessita saber a informação;
- Instrução centrada na tarefa: habilidades devem ser ensinadas em tarefas onde o conhecimento é normalmente aplicado;
- Ensino dirigido a falha: estudantes deveriam ser motivados a aprender a partir de situações de falha durante a execução de uma tarefa;

- Ensino dedutivo: pessoas aprendem sobre um domínio deduzindo regras generalizadas a partir de casos dados. Portanto, o ensino pode ocorrer através da apresentação de exemplos bem escolhidos;
- Congruência instrucional: uma seleção conduzida de exemplos assegura a realização de metas instrucionais pretendidas e evita erros de entendimento;
- Raciocínio analógico: estudantes utilizam a lembrança de soluções passadas para resolver novos problemas. Estas soluções podem ainda ser generalizadas para serem aplicadas em outros domínios. Esta análise pode ser considerada o processo de entendimento por parte do estudante;
- Estratégias de elaboração: estudantes podem aprender a criar suas próprias explicações se eles aprendem boas estratégias para elaborar o conteúdo dos exemplos trabalhados;
- Auto explicação: estudantes aprendem através da construção de explicações que os ajudam a entender o conteúdo. Desta forma, o entendimento do estudante pode ser testado pela análise de suas explicações;
- Perguntas explicativas: estudantes aprendem através de respostas dadas a um conjunto de perguntas investigativas;
- Explicações derivativas: professores incorporam suas explicações passadas na derivação de novas explicações;
- Auxílio à memória: humanos funcionam melhor quando são assistidos por uma memória externa que os auxilia com raciocínio analógico.

2.4 – Considerações finais

O conteúdo apresentado neste capítulo visa familiarizar o leitor com a teoria da representação do conhecimento através de casos, técnica escolhida para o desenvolvimento de um sistema de gerencimento de redes.

O próximo capítulo apresenta os conceitos de gerência de redes que deverão ser representados dentro do sistema. É evidente que além do conhecimento literário descrito neste material, o conhecimento heurístico do especialista também é de extrema importância, sendo peça fundamental na elaboração do sistema.

O conhecimento heurístico provém da experiência do especialista, ou seja, é o conhecimento que emana da vivência que o especialista tem em determinado assunto. A busca de uma solução por intermédio de uma informação denomina-se "Estratégia de Busca com Informação", ou simplesmente heurística, que usa o conhecimento do especialista para auxiliar todo o processo de tomada de decisão.

3. Gerência de Redes

Por menor e mais simples que seja, uma rede de computadores precisa ser gerenciada a fim de garantir aos seus usuários a disponibilidade dos serviços, a um nível de desempenho aceitável.

À medida que a rede cresce aumenta a complexidade de seu gerenciamento, forçando a adoção de ferramentas automatizadas para a sua monitoração e controle.

Podemos definir gerência de redes como o conjunto de atividades voltadas para o planejamento, monitoramento e controle dos serviços prestados pela infra-estrutura de rede e pelas aplicações que dependem dessa infra-estrutura.

De acordo com [SAYDAM, 1996], o significado de gerenciamento de redes é:

> "*Gerenciamento de rede inclui a disponibilização, a integração e a coordenação de elementos de hardware, software e humanos, para monitorar, testar, consultar, configurar, analisar, avaliar e controlar os recursos da rede, e de elementos, para satisfazer às exigências operacionais, de desempenho e de qualidade de serviço em tempo real a um custo razoável.*"

O objetivo da gerência de redes é procurar maximizar o desempenho, aprovisionar recursos diante de alterações de demanda, minimizar falhas, documentar e manter configurações e zelar pela segurança dos elementos que as compõem.

O bom gerenciamento da rede faz com que os recursos sejam aproveitados da melhor forma possível, garantindo o retorno esperado para o

investimento em Tecnologia da Informação (TI), de forma que prejuízos causados por problemas em seu funcionamento não ocorram.

3.1 - Ferramenta de gerenciamento

O protocolo SNMP (*Simple Network Management Protocol*) [Stallings 1993] [Stevens 1994] vem sendo ao longo do tempo sendo definido como ferramenta padrão pela IETF (*Internet Engineering Task Force*) no gerenciamento de equipamentos de rede. Este protocolo tem obtido tanto sucesso, que seu uso não se restringe mais somente para a gerência dos equipamentos tradicionais de redes [AZAMBUJA, 2001].

Segundo [STALLINGS, 1996], no gerencimento de uma rede, o modelo de gerência através do protocolo SNMP consiste nos componentes, mostrados na Figura 6:

Figura 6 – Principais componentes do Modelo de Gerência OSI e Internet

O Gerente é uma espécie de *software* que permite a obtenção e o envio de informações de gerencimento junto aos Objetos Gerenciados, mediante a

comunicação com um ou mais Agentes [AZAMBUJA, 2001]. As informações devem ser obtidas através de requisições realizadas pelos Gerentes aos Agentes do sistema, ou então, através de envio automático disparado pelo Agente a um determinado Gerente (mensagens denominadas *traps*). Normalmente, um Gerente está presente em uma estação de gerenciamento de rede [MEIRELLES, 1997].

O Agente nada mais é do que um *software* presente nos dispositivos gerenciados. A principal função de um Agente é o atendimento às requisições efetuadas pelo *software* Gerente e o envio automático de informações de gerenciamento ao Gerente, indicando a existência de um evento previamente programado. Também é de responsabilidade do Agente efetuar a interface entre diferentes mecanismos utilizados na instrumentação das funcionalidades de gerenciamento inseridas em um determinado dispositivo [MEIRELLES, 1997].

Ao conjunto de variáveis utilizadas para representar informações estáticas ou dinâmicas vinculadas a um determinado Objeto Gerenciado, denominamos MIB (*Management Information Base)*. Grande parte das funcionalidades de um Gerente/Agente, destina-se à troca de dados existentes na MIB [MEIRELLES, 1997].

O Protocolo de Gerenciamento define o conjunto de regras e o formato das mensagens. Os mecanismos de comunicação entre Gerentes e Agentes são elaborados com base nas especificações do protocolo utilizado [AZAMBUJA, 2001].

Ainda segundo definição de [AZAMBUJA, 2001] quanto aos elementos

que compõe o gerenciamento de redes no modelo SNMP, estes elementos estão representados na Figura 7.

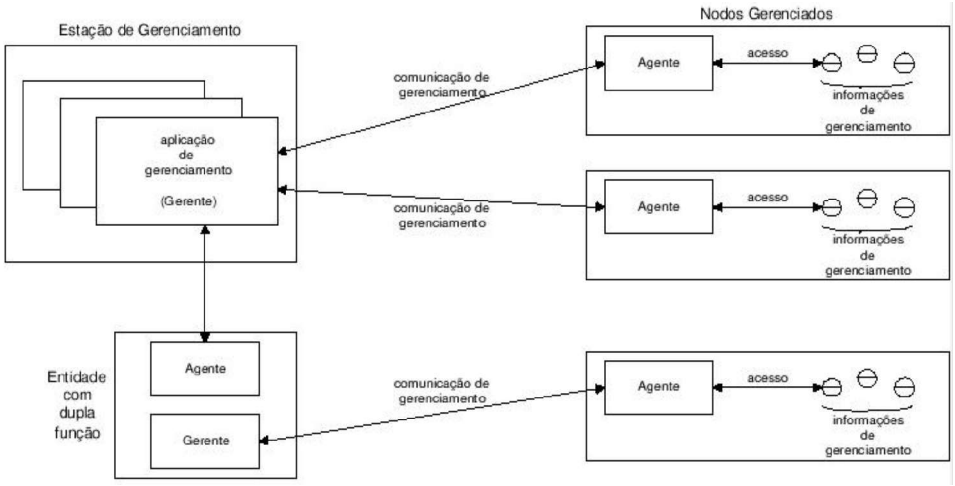


Figura 7 - Elementos do modelo de gerenciamento SNMP

- Deve existir no mínimo uma estação de gerenciamento, contendo uma ou mais entidades SNMP denominadas Gerente;
- Um ou mais nodos gerenciados, cada um com uma entidade SNMP denominada Agente;
- Opcionalmente, entidades SNMP com dupla função, capazes de desempenhas o papel de Gerente-Agente;
- Um protocolo de comunicação utilizado pelos Gerentes e Agentes durante a troca de mensagens de gerenciamento.

Então, uma ferramenta de gerenciamento é um programa que transforma os dados coletados pelo SNMP em informações usadas pelos usuários da Aplicação Gerente (administradores de rede, usuários dos serviços de rede etc). As aplicações de gerenciamento compreendem uma variedade de

programas que interagem com os Agentes, via operações de gerenciamento (`Get / Set`), manipulam e formatam as mensagens de *trap* ou acessam o Banco de Dados. As aplicações de gerenciamento devem auxiliar no processamento e análise dos dados obtidos junto aos equipamentos gerenciados.

Estas aplicações de gerenciamento obtêm dados destinados a apoiarem os administradores e usuários dos serviços de rede na tomada de decisões inteligentes, fornecendo gerenciamento pró-ativo e estendendo as funcionalidades das plataformas de gerenciamento em muitas direções.

3.1.1– SMI (S*tructure of Management Information*)

A SMI (estrutura de informações de gerenciamento) é a linguagem utilizada para definir as informações de gerenciamento que residem em um determinado objeto gerenciado por intermédio do SNMP. Note que a SMI define a linguagem na qual a informação está especificada e não um exemplar específico para os dados em que uma entidade da rede é gerenciada.

A RFC 2578 define os tipos de dados básicos na linguagem para a definição de módulos da SMI MIB. Os 11 tipos de dados definidos na RFC 2578 são mostrados na Tabela 2. Dos dados apresentados, boa parte deve ser de conhecimento dos leitores. Trataremos com mais detalhes o OBJECT IDENTIFIER, que é utilizado para dar nome a um objeto gerenciado.

A construção OBJECT-TYPE é usada para especificar o tipo de dado, o status e a semântica de um objeto gerenciado e possui quatro cláusulas [KUROSE, 2003]. A cláusula SYNTAX de uma definição OBJECT-TYPE especifica os dados básicos associados ao objeto gerenciado. A cláusula MAX-

ACCESS especifica se o objeto pode ser lido, escrito, criado ou ter seu valor incluso em uma notificação. A cláusula STATUS indica se a identificação do objeto é atual e válida, obsoleta ou depriciada. A cláusula DESCRIPTION contém uma definição textual do objeto. Como exemplo de uma construção OBJECT-TYPE, considere o `ipInDelivers` da RFC 2011, apresentado na Figura 8.

Tabela 2 – Tipos de dados básicos da SMI. Fonte [KUROSE, 2003]

Tipo de dado	**Descrição**
INTEGER	Número inteiro de 32 bits, como definido em ASN.1 (*Abstract Syntax Notation One*), com valor entre -2^{31} e $2^{31} - 1$, inclusive, ou um valor de uma lista de valores constantes possíveis, nomeados.
Integer32	Número inteiro de 32 bits, com valor entre -2^{31} e $2^{31} - 1$, inclusive.
Unsigned32	Número inteiro de 32 bits sem sinal na faixa de 0 a $2^{32} - 1$, inclusive.
OCTER STRING	Uma cadeia de bytes de formato ASN.1 que representa dados binários arbitrários ou texto de até 65535 bytes de comprimento.
OBJECT IDENTIFIER	Formato ASN.1 atribuído administrativamente (nome estruturado).
Endereço IP	Endereço Internet de 32 bits, na ordem de bytes da rede.
Counter32	Contador de 32 bits que cresce de 0 a $2^{32} - 1$ e volta a 0.
Counter64	Contador de 64 bits.
Gauge32	Número inteiro de 32 bits que não faz contagens além de $2^{32} - 1$ nem diminui para menos que zero.
TimeTicks	Tempo, medido em centésimos de segundo, transcorrido a partir de algum evento.
Opaque	Cadeia ASM.1 não interpretada, necessária para compatibilidade com as versões anteriores.

```
ipInDelivers OBJECT TYPE
  SYNTAX       Counter32
  MAX-ACCESS   read-only
  STATUS   current
  DESCRIPTION
     "The total number of input
      datagrams successfully
      delivered to IP user-
      protocols (including ICMP)"
::= { ip   9}
```

Figura 8 – Exemplo de construção OBJETC-TYPE

O objeto apresentado acima, define um contador de 32bits que tem o objetivo de armazenar a quantidade de datagramas IP que foram recebidos e entregues com sucesso à camada superior. A última linha desse objeto refere-se ao seu nome, um tópico que será tratado mais adiante.

A construção MODULE-IDENTITY permite que objetos relacionados entre si sejam agrupados. Além das definições OBJECT-TYPE, a construção MODULE-IDENTITY contém informações de contato com o autor do módulo, a data da última atualização, um histórico de revisões e uma descrição do módulo. Como exemplo de sua construção, considere a Figura 9.

```
ipMIB MODULE-IDENTITY
      LAST-UPDATED "941101000Z"
      ORGANIZATION "IETF SNMPv2 Working Group"
      CONTACT-INFO
           " Keith McCloghrie, …"
      DESCRIPTION
           "The MIB module for managing IP
           and ICMP implementations, but
           excluding the management of
           IP routes."
      REVISION "019331000Z"
      DESCRIPTION
           "The initial revision of this MIB
           module was part of MIB-II."
::= {mib-2 48}
```

Figura 9 – Exemplo de construção MODULE-IDENTITY

Existem ainda as construções NOTIFICATION-TYPE, MODULE-COMPLIANCE e AGENT-CAPABILITIES. A primeira especifica informações referentes a mensagens *trap* e *Information Request* geradas pelos agentes ou gerentes. A segunda construção define o conjunto de objetos gerenciados em um módulo implementado por um agente. A última, especifica as capacidades dos agentes relativas às definições de notificação de objetos e eventos. Maiores detalhes serão tratados em trabalho futuro.

3.1.2 - MIB

Como vimos anteriormente, a MIB nada mais é do que a base de informações de gerenciamento. A MIB pode ser pensada como um banco virtual, onde são armazenadas informações dos objetos gerenciados cujos valores, coletivamente, refletem o "estado" atual da rede [KUROSE, 2003]. Estes valores podem ser consultados pelas entidades gerentes através de mensagens SNMP ao agente. Os objetos gerenciados são especificados com o uso da construção OBJECT-TYPE da SMI e agrupados em módulos MIB que utilizam a construção MODULE-IDENTITY.

3.1.3 - Operações do protocolo SNMP e mapeamentos de transporte

O SNMP é utilizado para transportar informações MIB entre gerentes e agentes. Seu uso mais comum é no modo comando-resposta, no qual um gerente envia uma requisição a um agente que à recebe e realiza alguma ação para o envio da resposta. Em geral, uma requisição MIB é utilizada para consultar ou

modificar valores dos objetos. Um segundo uso do SNMP é para o envio automático de mensagens *trap*, disparado pelo agente a um determinado gerente, como vimos anteriormente. As mensagens do tipo *trap* são utilizadas para notificar um gerente de uma situação excepcional que resultou em mudança nos valores dos objetos da MIB. O uso de *traps* pode relatar problemas como queda de uma interface, um alto nível de congestionamento ou quando ocorre qualquer outro evento notável. São definidos pelo SNMP[5], sete tipos de mensagens, conhecidas genéricamente como PDUs, como é mostrado na Tabela 3.

Tabela 3 – Tipos de SNMPv2-PDU

Tipo de SNMPv2-PDU	Remetente-Receptor	Descrição
`GetRequest`	Gerente a Agente	Pega o valor de uma ou mais instâncias de objetos MIB.
`GetNextRequest`	Gerente a Agente	Pega valor da próxima instância de objeto MIB na lista ou tabela.
`GetBulkRequest`	Gerente a Agente	Pega valores em grandes blocos de dados, por exemplo os valores de uma grande tabela.
`InformRequest`	Gerente a Agente	Informa à entidade gerenciadora remota os valores remotos da MIB para seus acessos.
`SetRequest`	Gerente a Agente	Define valores de uma ou mais instâncias de objetos MIB.

5 SNMPv2

Tipo de SNMPv2-PDU	Remetente-Receptor	Descrição
`Response`	Agente a Gerente ou Gerente a Agente	Gerada em resposta a `GetRequest`, `GetNextRequest`, `GetBulkRequest`, `SetRequest PDU` ou `InformRequest`
`SNMPv2-Trap`	Agente a Gerente	Informa ao administrador um evento excepcional.

Tabela 3 – Tipos de SNMPv2-PDU (Continuação)

Dada a natureza requisição/resposta do SNMP, é importante mencionar que as SNMP-PDUs podem ser transportadas por muitos protocolos de transporte diferentes, entretanto, a RFC 1906 estabelece o UDP (*User Datagram Protocol*) como o "mapeamento de transporte ideal" sendo, portanto, transportadas na carga útil de um datagrama UDP [KUROSE, 2003]. Uma vez que o UDP é um protocolo de transporte não confiável, não há garantia de que uma requisição ou resposta será recebida no destino pretendido.

Neste trabalho não será abordada a implementação da gerência SNMP associada à sistemas especialitas, bem como o detalhamento de implementação técnica deste serviço.

3.2 – Áreas funcionais de gerenciamento

Como parte da especificação de gerenciamento de sistemas OSI, a ISO fez uma separação funcional das necessidades no processo de gerenciamento de redes com o objetivo de segmentar as áreas de gerenciamento, facilitando assim sua administração. Esta divisão foi adotada pela maioria dos fornecedores de

sistemas de gerenciamento de redes para descrever as necessidades de gerenciamento. São cinco as áreas de gerência apresentadas:

- Gerenciamento de Falhas.
- Gerenciamento de Desempenho.
- Gerenciamento de Configuração.
- Gerenciamento de Contabilização.
- Gerenciamento de Segurança.

3.2.1– Gerênciamento de Falhas.

O gerenciamento de falhas é a área funcional responsável pela detecção de eventos anormais e pelo diagnóstico de problemas que levaram a esses eventos, permitindo assim o isolamento e a correção das operações anormais na rede. Um *software* que visa oferecer suporte ao gerenciamento de falhas, deve mostrar ao administrador de rede o número, os tipos, a hora da ocorrência e a localização dos erros na rede. Quando ocorrem falhas em uma rede é importante que os seguintes procedimentos sejam seguidos:

- Determinar exatamente a localização da falha;
- Se possível, isolar o resto da rede da falha, assim a rede pode continuar a funcionar sem interferências;
- Reconfigurar ou modificar a rede de tal maneira a minimizar o impacto da operação mesmo sem o(s) equipamento(s) com falha;
- Corrigir a(s) falha(s) existente(s), para que o funcionamento da rede possa retornar ao seu estado inicial.

Uma observação importante que faz-se necessária é que as informações

obtidas através das gerências de configuração e de desempenho, devem ser utilizadas de forma pró-ativa para evitar falhas previsíveis. O controle das informações existentes nas áreas funcionais de configuração e desempenho, permite que "furos" nestas áreas sejam corrigidos, não causando erros a serem gerenciados pela área funcional de gerência de falhas.

O monitoramento de falhas tem como objetivo determinar a ocorrência de falhas da forma mais rápida possível e identificar as causas dessas falhas. Identificada sua(s) causa(s), ações devem ser tomadas para solucionar o problema. 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 à 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 alguns problemas podem ocorrer. 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. 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 qualquer um desses subsistemas;

- Várias observações relacionadas: uma falha na linha de comunicação pode afetar toda a comunicação entre as estações conectadas em uma rede padrão IEEE *(Institute of Electrical and Electronics Engineers)* 802.5[6] e as estações conectadas à uma rede padrão IEEE 802.3[7], 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. Por exemplo, uma falha na linha de comunicação 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;
- Interferência de procedimentos de reparação local em diagnóstico: procedimentos de correção local podem destruir importantes evidências referente à natureza da falha, impossibilitando o diagnóstico;

6 O IEEE 802.5 é o padrão para redes em anel e utiliza o método de acesso *Token Ring* a 4 ou 16 Mbps, através de cabeamento STP.

7 O IEEE 802.3 usa o acesso CSMA/CD em várias velocidades e em vários meios físicos. Extensões ao padrão do IEEE 802.3 especificam implementações da *Fast Ethernet*. As variações físicas da especificação original do IEEE 802.3 incluem 10Base2, 10Base5, 10BaseF, 10BaseT e 10Broad36. Variações físicas da *Fast Ethernet* incluem 100BaseT, 100BaseT4 e 100BaseX.

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

3.2.1.1– Funções da Gerência de Falhas

A primeira exigência em um sistema de gerência de falhas é que ele detecte e informe a ocorrência das falhas ocorridas. No mínimo, um agente de monitoramento de falhas deve manter um arquivo de *log* com os eventos e erros mais significativos. Tipicamente, um agente de monitoramento de falhas tem a capacidade para, de forma independente, 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 (referências como latência, perda de pacotes e disponibilidade da rede).

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 e uma vez atingidos estes limites o sistema deve emitir um alarme.

Um sistema de gerência de falhas deve também permitir um diagnóstico da falha apresentada e os procedimentos para recuperação, através dos seguintes testes:

- 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 amigável.

3.2.1.2 – Diagnósticos de Falhas

A maneira de detecção de falhas consiste, em geral, na comparação entre um comportamento esperado (normal) e o comportamento apresentado. Discrepâncias entre estes comportamentos indicam que o sistema está com problemas. Confirmada a discrepância, deve-se determinar as causas do problema, ou seja, um diagnóstico inicial. Assim, o objetivo deste diagnóstico é determinar os elementos responsáveis pelo mal funcionamento do sistema.

A discrepância entre o comportamento esperado e o comportamento observado é utilizada para guiar a pesquisa pelo diagnóstico. Existem dois tipos de diagnósticos:

1. Diagnóstico Baseado em Modelo:

O princípio básico da abordagem baseada em modelo para diagnósticos, 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;

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 fator fundamental nos diagnósticos baseados em modelo é que os modelos devem ser completamente corretos. Entretanto, tal como em quanquer modelagem matemática, o modelo adotado para ser utilizado como referência trabalha com uma quantidade de hipóteses simplificadas e aproximações que são incapazes de reproduzir a situação real do sistema implementado com precisão. 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 diagnósticos baseados em modelos, tem trabalhado em dois segmentos:

- Modelo de comportamento correto: define como o sistema normalmente trabalha;
- Modelo de comportamento falho: especifica como ele trabalha se determinadas falhas ocorrem.

Resumidamente, o diagnóstico baseado 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 da discrepância ou falha);
- Determinação dos componentes do sistema que em hipótese de falha explicam tal discrepância (diagnóstico).

O diagnóstico basedo em modelo utiliza um formalismo apropriado para determinar o comportamento esperado do sistema de interesse. As formas de se fazer diagnósticos baseados em modelo, depende do conhecimento que se obtém do comportamento do sistema, e se classificam em:

- Diagnóstico baseado em consistência: é baseada em um modelo que descreve o comportamento esperado do sistema;
- Diagnóstico baseado em abdução: é baseada em um modelo que descreve o comportamento falho do sistema.

2. Diagnóstico heurístico

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 oferecem 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 meta-heurística e algumas meta-heurísticas têm sido propostas e especialmente projetadas para evitar que o procedimento fique preso em armadilhas de ótimos locais.

O diagnóstico heurístico usa o conhecimento de especialistas e o conhecimento obtido através da observação de uma quantidade significativa de dados. Tipicamente, este conhecimento pode ser expresso através de regras, associando sintomas com as falhas observadas.

Vários problemas têm sido identificados quando se faz uso da abordagem heurística:

- A aquisição do conhecimento de especialistas humanos é uma tarefa difícil e consome 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;

- Apenas o conhecimento sobre o comportamento do sistema até a data atual pode ser utilizado e, portanto, alguns tipos de falhas raras podem não ser diagnosticados.

3.2.1.3– 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 de registrar alarmes, leva em consideração as necessidades de serviços dos usuários, os protocolos necessários 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.

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.

3.2.1.4– Controle de *Log*

O controle de *logs*, diz respeito a requerimentos de usuário, serviços oferecidos, e o protocolo necessário para oferecer os serviços de registro de *logs*. 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,* os quais são enviados para um ou mais arquivos de *log* após terem sido devidamente filtrados. Os filtros têm um conjunto de regras que determinam que registros de *log* serão gravados.

3.2.2- Gerenciamento de Desempenho

O gerenciamento do desempenho de uma rede consiste na monitoração das atividades da rede e no controle dos recursos através de ajustes e trocas, com o objetivo de assegurar que a rede tenha capacidade para suportar e acomodar uma certa quantidade de usuários. Algumas das questões relativas ao

gerenciamento do desempenho, são:

- Qual é o nível de capacidade de utilização?
- O tráfego é excessivo?
- O *throughput* está em um nível aceitável?
- Existem gargalos?
- O tempo de resposta está aumentando?
- Qual a latência?

Para tratar estas questões, o gerente deve focalizar um conjunto inicial de recursos a serem monitorados a fim de estabelecer níveis de desempenho. Isto inclui associar métricas e valores apropriados aos recursos de rede que possam fornecer indicadores de diferentes níveis de desempenho. Muitos recursos devem ser monitorados para se obter informações sobre o nível de operação da rede. Colecionando e analisando estas informações, o gerente da rede pode ficar mais capacitado no reconhecimento de situações indicativas de degradação de desempenho.

Estatísticas de desempenho podem ajudar no planejamento, administração e manutenção de grandes redes. Estas informações podem ser utilizadas para reconhecer situações de gargalo antes que elas causem problemas ao usuário final. Ações corretivas podem ser executadas como, por exemplo, a troca de tabelas de roteamento para balancear ou redistribuir a carga de tráfego durante horários de pico, ou ainda, em longo prazo, indicar a necessidade do aumento de banda.

Para atingir estes objetivos, deve-se monitorar a taxa de utilização dos recursos, a taxa em que estes recursos são pedidos e a taxa em que os pedidos a um recurso são rejeitados. Para cada tipo de monitoração definimos um valor máximo aceitável (*threshold*), um valor de alerta e um valor em que se remove

a situação de alerta. Definem-se três modelos para atender aos requisitos de monitoração do uso dos recursos do sistema:

- Modelo de Utilização: Provê a monitoração do uso instantâneo de um recurso.
- Modelo de Taxa de Rejeição: Provê a monitoração da rejeição de um pedido de um serviço.
- Modelo de Taxa de Pedido de Recursos: Provê a monitoração dos pedidos do uso de recursos.

Desta forma, a gerência de desempenho deve utilizar-se de alguns indicadores de desempenho, tais como:

- Vazão: em um sistema de filas que são formadas quando da chegada de pacotes em um servidor, serão considerados o tempo de chegada de pacotes, e o tempo de serviço para atender os "fregueses", ou seja, a capacidade de atendimento dos serviços (saídas).
- Disponibilidade: é a probabilidade do serviço estar em funcionamento durante um período de tempo. Esta medição é muito importante em sistemas críticos, ou seja, aqueles que se necessita de *downtime* muito reduzido.
- *Throughtput:* é também conhecido como banda passante ou largura de banda sendo medido em função do número de bits que podem ser transmitidos sobre a rede em um certo período de tempo. Normalmente, a unidade utilizada para quantificar a largura da banda é a *bps,* utilizada para expressar a quantidade de bits que são transmitidos durante um segundo, sendo também utilizada a notação bits/segundo.

Também são importantes indicadores de desempenho, os atrasos[8] na entrega dos pacotes e os erros que ocorrem tanto na entrada quando na saída dos dados.

Com referência aos problemas na gerência de desempenho, em sua maioria estão relacionados à "situação" que o especialista definiu como ideal dentro do sistema. Deve-se ficar atento a alguns itens:

- Monitorar os indicadores de desempenho: ficar atento aos pontos estabelecidos como indicadores de desempenho e providenciar alterações quando eles indicam redução da eficiência.
- *Baseline:* isto significa descrever o que deve ser considerado normal no desempenho da rede, ou seja, são escolhidas várias medidas para retratar o desempenho da rede.
- Definição de limiares: é a definição de patamares, os quais serão responsáveis por gerar eventos ou alarmes de desempenho. Ex: Ajustam-se *thresholds* acima dos valores normais para gerar eventos, quando o desempenho cruzar os limiares, tem-se duas situações (patamares), de advertência e crítica.

Alguns dados estatísticos devem ser levantados. A análise de desempenho e planejamento de capacidade alterando o modo de operação, fazem parte da gerência de desempenho. Ex: de um segmento compartilhado para uma rede comutada.

Assim, algumas estatísticas de desempenho são importantes, permitindo a formação de uma base de dados referente á tendências e eventos que servirão para planejamento de expansões, balanceamento da carga de rede entre recursos, etc.: Destas estatísticas de desempenho pode-se ressaltar estatísticas:

8 Também conhecidos como *delay*

- Para interfaces:
  - ✔ Utilização dos enlaces, em intervalos de 10 minutos. Pode incluir distribuição de freqüência com histograma (0-20%, 20%-60%, 60%-100%).
  - ✔ Utilização das interfaces por protocolo.
  - ✔ Qualidade dos enlaces por hora.
  - ✔ Fração de erros na entrada e na saída.
  - ✔ Número total de erros.
  - ✔ Disponibilidade.
  - ✔ Taxa de descarte.
  - ✔ Taxa total de pacotes chaveados por segundo.
- Para Roteadores
  - ✔ Utilização de CPU.
  - ✔ Utilização de Memória.
  - ✔ Disponibilidade.
  - ✔ Taxa de descarte.
  - ✔ Taxa total de chaveados por segundo.
- Para *LANs*[9] *Ethernet*
  - ✔ Colisões devem ficar em, no máximo, 3%.
  - ✔ Alguns administradores toleram até 5%.
- Para *hosts* (no que diz respeito à rede)
  - ✔ Taxa de retransmissão TCP.

3.2.3- Gerenciamento de Configuração

9 *Local Area Network* – Redes Locais

O gerenciamento de configuração permite que o administrador da rede saiba quais dispositivos fazem parte da rede administrada e quais são suas configurações de *hardware* e *software* [KUROSE, 2003].

O objetivo da gerência de configuração é permitir a preparação, a iniciação, a partida, a operação contínua, e a posterior suspensão dos serviços de interconexão entre os sistemas, tendo então, a função de manutenção e monitoração da estrutura física e lógica de uma rede, incluindo a verificação da existência dos componentes, e a verificação da interconectividade entre estes componentes [KUROSE, 2003].

O serviço de gerência de configuração contempla a realização de uma série de atividades dentro desta área funcional, desenvolvendo ações para materialização de resultados. As atividades desenvolvidas são classificadas por funcionalidade, responsabilidade e resultados, e devem oferecer total visibilidade para o administrador de redes no desenvolvimento de suas funções. Dentre as principais, pode-se destacar:

- Identificação dos elementos funcionais: processo de descoberta, classificação e identificação dos dispositivos da rede;
- Construção de mapas de topologia: apresentação e construção de mapas com a topologia dos elementos[10] e representação da estrutura de interconexão física ou lógica destes elementos;
- Inventário de *hardware* e *software*: mecanismos para inventário de *hardware* e *software* de diferentes dispositivos e ambientes, fornecendo informações sobre configuração e disponibilidade de recursos em cada elemento;
- Construção de bases de dados de configuração: Devem conter as

10Dispositivos IP

aplicações que auxiliam no processo de configuração de elementos de rede:

- ✔ Implementação em larga escala: mecanismos para a reaplicação de configuração em larga escala, facilitando o processo de implementação e o fornecimento de recursos.
- ✔ Verificação de integridade: análise crítica das configurações de todo o ambiente da rede.

- Gestão de alteração na configuração dos dispositivos: mecanismos de sinalização e acompanhamento de mudanças. O objetivo é desenvolver processos capazes de acompanhar as modificações implementadas por um usuário na infra-estrutura.

3.2.4 - Gerenciamento de Contabilização

Gerência de contabilização tem como objetivo descobrir a forma como os usuários utilizam os recursos da rede. Com essas informações podemos garantir que os recursos sempre estejam disponíveis quando necessários aos sistemas de gerenciamento. Assim, torna-se possível a realização de um melhor planejamento do crescimento da rede, detectar abusos no uso dos recursos e até cobrar, de alguma forma, pelos serviços utilizados.

Mesmo que nenhuma cobrança interna seja feita pela utilização dos recursos da rede, o administrador da rede deve estar habilitado para controlar o uso dos recursos por usuário ou grupo de usuários, com o objetivo de:

- Evitar que um usuário ou grupo de usuários abuse de seus privilégios de acesso e monopolize a rede, em detrimento de outros usuários;

- Evitar que usuários façam uso ineficiente da rede, assistindo-os na troca de procedimentos e garantindo o desempenho da rede;
- Conhecer as atividades dos usuários com detalhes suficientes para planejar o crescimento da rede.

Portanto, podemos definir gerenciamento de contabilização como a a área que trata da coleta de dados sobre o consumo de recursos para propósitos de análises de capacidade e tendências, alocação de custos, auditoria e cobrança. O objetivo da alocação de custos é alocar um custo conhecido dividindo-o entre diversas entidades (entidades podem representar por exemplo, empresas parceiras ou departamentos de uma única firma). A auditoria é o ato de verificação da correção de um procedimento, comumente relacionado com dados de contabilização. Tarefas de auditoria incluem a verificação da correção de uma fatura submetida por um provedor de serviços a um cliente, ou a verificação da conformidade de políticas de utilização, acordos de nível de serviço ou ainda metas de segurança. O gerenciamento de contabilização requer que o consumo de recursos seja medido, tarifado, designado e comunicado entre as partes apropriadas [ABOBA, 2000].

Na literatura, o termo gerenciamento de contabilização define uma gama variada de etapas ou processos. Apesar das diferenças no número de processos e características dos mesmos, o conjunto final de processos deve atender aos objetivos do gerenciamento de contabilização, conforme a Figura 10.

Conforme [EVLOGIMENOU, 2002] e [STILLER, 2001], os processos utilizados no gerenciamento da contabilização, podem ser descritos como mostrado a seguir:

- *Metering* (medição): Envolve o monitoramento e a medição de

recursos de rede. A granularidade da medição varia de acordo com a camada de rede em que está sendo aplicada e também conforme acordos de nível de serviços, quando for o caso. O processo de medição só é requerido para esquemas baseados no uso. Esquemas de taxa fixa dispensam este processo.

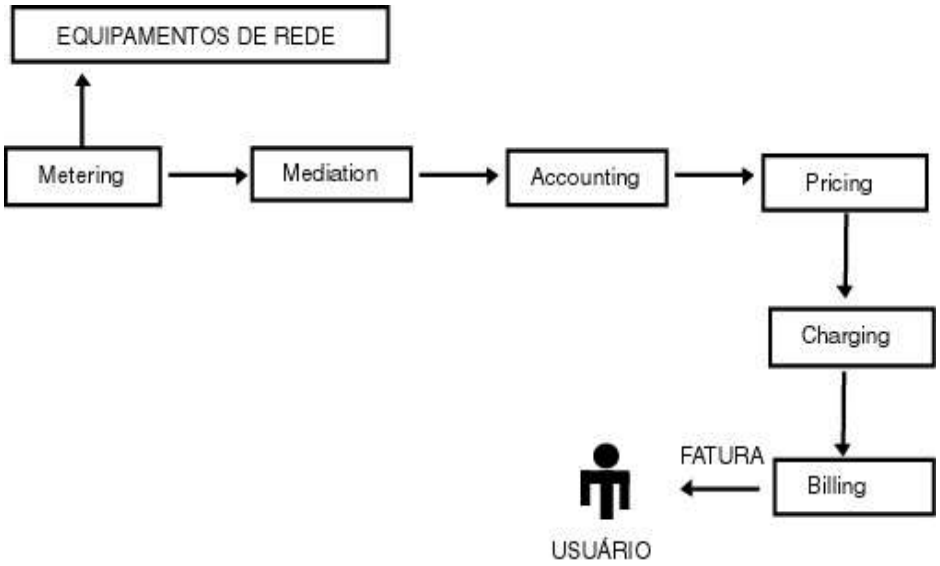


Figura 10 – Processos no gerenciamento de contabilização

- *Mediation* (mediação): Os dados colhidos pelo processo de medição, na maioria das vezes são dados muito técnicos. Podem conter, por exemplo, informações sobre pacotes e tamanhos de filas, mas nunca relacionam estes dados a um cliente específico. O processo de mediação os transforma em um formato que pode ser armazenado e utilizado para processamentos posteriores. Isto é feito coletando e agregando os dados oriundos de diversas unidades de medição e de outras fontes. Tarefas comuns ao processo de mediação incluem

filtragens, agregações e correlações.

- *Accounting* (contabilização): Define a sumarização de informações relacionadas com a utilização de um serviço por um determinado cliente. O registro de contabilização resultante deste processo é expresso em medidas de consumo de recursos, por exemplo, por sistemas finais, aplicações, ou qualquer tipo de conexão.
- *Pricing* (determinação de preços): Cobre a especificação e determinação de preços considerando os recursos de rede e os serviços em um cenário de mercado aberto. Este processo deve combinar considerações técnicas (por exemplo, consumo de recursos) e econômicas (técnicas de tarifação e práticas de mercado). A determinação de preços deve ser com base no custo/benefício ou na situação atual do mercado.
- *Charging* (tarifação) Durante este processo as tarifas e preços são combinados com os dados oriundos do processo de contabilização para a geração de registros de cobrança (quando couber). Neste processo informações técnicas são transformadas em unidades monetárias. As tarefas comuns a esse processo podem ser relacionadas como:
  - ✔ Receber registros de contabilização.
  - ✔ Receber preços oriundos do processo de precificação.
  - ✔ Receber informações sobre os serviços (SLA's).
  - ✔ Calcular os encargos (combinando as informações anteriores).
  - ✔ Coletar registros de tarifação para um determinado período de cobrança.
  - ✔ Enviar registros de tarifação para o processo de cobrança.

- *Billing* (cobrança): Coleta informações de tarifações para um cliente em um determinado período de tempo (por exemplo, mensalmente) e as combina em uma fatura (quando couber), num formato previamente acordado, para envio aos destinatários apropriados. Esta fatura pode conter uma listagem detalhada acerca da utilização de recursos e tarifas aplicadas.

As propriedades desejáveis de um esquema de contabilização se definem basicamente sob dois distintos pontos de vista : o ponto de vista do gerente da rede ou provedor dos serviços e o ponto de vista dos clientes [FERRARI, 1999]. Do ponto de vista do provedor de serviços, os objetivos mais importantes relacionam-se com a grande probabilidade de recuperação de custos, o estabelecimento de preços competitivos, o encorajamento (ou desencorajamento) de comportamentos dos clientes que irão acentuar (ou degradar) a eficiência da rede e a redução de custos de implementação e utilização. Sob o ponto de vista dos clientes, as principais propriedades se definem com a compreensibilidade, controlabilidade, previsibilidade, estabilidade e justiça. Para a satisfação destes pontos de vista, algumas características devem se fazer presentes em um esquema de contabilização no nível de serviços e essas características representam os principais desafios desta área de pesquisa:

- Contabilização baseada no uso.
- Aplicação de acordos de nível de serviço e qualidade de serviço.
- Contabilização intra e inter-domínios.
- Sensível ao *feed-back* dos usuários.
- Flexibilidade.
- Segurança, escalabilidade e baixa geração de tráfego de

gerenciamento.

A contabilização no nível de serviços deve acompanhar as evoluções das tecnologias de comunicação e de provimento dos serviços. Para que os recursos de rede disponíveis sejam utilizados de forma eficiente e para que os custos de implantação da infra-estrutura e sua operação possam ser recuperados, é essencial que existam mecanismos de cobrança eficientes em termos econômicos. Esses novos mecanismos de contabilização também devem ser capazes de incentivar o uso das tecnologias e serviços oferecidos. Para tal é necessário que o mesmo seja transparente, previsível e controlável pelos clientes. A contabilização baseada na utilização de recursos é um passo importante para o alcance de todas essas metas.

3.2.5 - Gerenciamento de segurança

O objetivo do gerenciamento de segurança é o de dar subsídios a aplicação de políticas de segurança, que são os aspectos essencias para que uma rede baseada no modelo OSI seja operada corretamente, protegendo os objetos gerenciados e o sistema de acessos indevidos por intrusos.

O gerenciamento de segurança deve, pois, providenciar um alarme ao gerente da rede sempre que se detectarem eventos relativos à segurança do sistema. Estes alarmes podem ser gerados com o auxílio de ferramentas *IDS (Instrusion Detection System)* ou *firewalls.*

São distinguidos dois conceitos no modelo OSI em relação a segurança:

- Arquitetura de segurança do modelo OSI;
- Funções de gerenciamento de segurança, sendo que estas formam a área funcional de gerência de segurança.

O objetivo da arquitetura de segurança do modelo OSI é o de dar uma descrição geral dos serviços de segurança e dos mecanismos associados a este, e de definir em que posição do modelo de referência situam-se os serviços de segurança e os seus mecanismos associados. A norma de referência da arquitetura de segurança trata exclusivamente da segurança dos canais de comunicação, através de mecanismos como a criptografia e a assinatura numérica, que permitem aos sistemas que usam este canal se comunicarem de forma segura. Para isso, define-se os seguintes serviços:

- Autenticação tanto de entidades pares quanto da origem dos dados (*authentication*);
- Controle de acesso aos recursos da rede (*access control*);
- Confidencialidade dos dados (*confidenciality*);
- Integridade dos dados (*integrity*);
- A não-rejeição ou não-repudiação (*non-repudiation*);

Os mecanismos a serem adotados dependem do uso de uma política de segurança, que é feita pelo uso das funções de segurança do gerenciamento de redes OSI. Estas funções, que compõem o gerenciamento de segurança, tratam do controle dos serviços de segurança do modelo OSI, e dos mecanismos e informações necessárias para se prestar estes serviços. Os objetivos do gerenciamento de segurança são, portanto:

- O fornecimento de relatórios de eventos relativos à segurança e o fornecimento de informações estatísticas;
- A manutenção e análise dos registros de histórico relativos à segurança;
- A seleção dos parâmetros dos serviços de segurança;
- A alteração, em relação à segurança, do modo de operação do sistema

aberto, pela ativação e desativação dos serviços de segurança.

Para que estes objetivos sejam atingidos, deve-se olhar as diferentes políticas de segurança a serem adotadas no sistema aberto. Todas as entidades que seguem uma mesma política de segurança pertencem ao mesmo domínio de segurança.

Devido ao gerenciamento do sistema necessitar distribuir as informações de gerenciamento de segurança entre todas as atividades que se relacionam com a segurança, os protocolos de gerenciamento, assim como os canais de comunicação, devem ser protegidos usando os mecanismos previstos na arquitetura de segurança.

As informações de gerenciamento de segurança são armazenadas numa MIB especial que deve dar apoio às três categorias de atividades de gerenciamento de segurança existentes. Esta MIB é chamada de SMIB (*Security MIB).*

Conforme a proposta de gerenciamento de segurança OSI, vários são os mecanismos de proteção que podem ser usados para garantir segurança a um ambiente de comunicação em sistemas abertos.

3.2.5.1 – Mecanismos de autenticação

Em sistemas distribuídos, os parceiros da comunicação (usuário origem e usuário destino) estão interconectados através de uma rede aberta na qual vários outros usuários também podem ter acesso. Assim sendo, toda troca de informações realizada entre os dois pontos, deverá ser encaminhada através da rede, sendo, portanto, possível a interceptação destas informações por um outro

usuário que poderá modificar ou destruir as mensagens enviadas bem como inserir mensagens falsas nesta comunicação. Particularmente o processo de autenticação de usuários é bastante prejudicado por essas possibilidades de interceptação, decorrentes do fato de que as informações confidenciais que são necessárias para autenticar o cliente (usuário origem) junto ao servidor (usuário destino), poderão ser manipuladas por outros usuários ao trafegarem pela rede.

Devido a esses problemas de vulnerabilidade da comunicação possibilitando ameaças à segurança do sistema, um sistema de autenticação propício a esse ambiente deverá requerer:

- Uma autenticação forte (*strong authentication*): seu objetivo é fazer com que as informações necessárias para autenticar um usuário não sejam divulgadas durante a comunicação;
- Uma autenticação mútua: a autenticação deve ocorrer nos dois sentidos, ou seja, tanto a origem deve ser autenticada no destino para que este tenha a garantia de onde a mensagem foi originada, como o destino deve ser autenticado na origem para garantir que realmente é ele que irá receber e interpretar a mensagem enviada;
- Uma autenticação contínua: a freqüência do processo de autenticação deverá ser periódica; apenas uma autenticação inicial não é suficiente pois um intruso poderá se fazer passar por um usuário já autenticado e deturpar o intercâmbio.

As ameaças de segurança em um ambiente distribuído poderão ser controladas usando criptografia para fornecer uma autenticação forte, mútua e contínua. Independentemente de qual técnica de criptografia se irá utilizar é necessário que os parceiros da comunicação tenham conhecimento da chave criptográfica que irá ser utilizada na segurança do seu processo; se a chave de

criptografia é compartilhada apenas entre os dois parceiros, denominamos esse processo de protocolo de autenticação *two-party*.

Na criptografia simétrica, cada dupla de parceiros deverá compartilhar entre si uma única informação confidencial; a cada novo parceiro de comunicação, uma nova informação confidencial deverá ser intercambiada entre eles. Dependendo do número de parceiros que estão envolvidos no processo de comunicação, poderá ficar bastante complexo a etapa de autenticação devido a quantidade de chaves confidenciais que irão ser necessárias para autenticar cada parceiro isoladamente.

Na criptografia assimétrica, teremos um problema semelhante com o gerenciamento das chaves públicas que, apesar de não ser uma informação confidencial como no caso anterior, continuará existindo uma diversidade de chaves[11].

Baseado na complexidade do processo de autenticação em sistemas distribuídos, idealizou-se um serviço de autenticação utilizando um servidor dedicado: Servidor de Autenticação (SA). Nesse protocolo o usuário precisará ter conhecimento apenas da chave relacionada com o SA. Por sua vez, o SA deverá ter conhecimento de todas as chaves dos usuários relacionados com o seu domínio de atuação. Esse tipo de protocolo é denominado protocolo de autenticação *third-party* por existir um terceiro parceiro que será encarregado de, com segurança, administrar os usuários e suas senhas bem como autenticar usuários autorizados por sessão de comunicação.

Os modernos sistemas criptográficos podem ser classificados em duas categorias de acordo com o tipo de chave que utiliza:

- Criptografia de chave secreta ou simétrica - utiliza apenas uma única

11Cada parceiro possui a sua própria chave pública

chave que deverá ser mantida sobre sigilo (KS - chave secreta) e será usada tanto para criptografar como para decriptografar;

- Criptografia de chave pública ou assimétrica - utiliza duas chaves: uma chave privada e uma chave pública; o usuário deverá divulgar sua chave pública e manter em sigilo sua chave privada. As chaves são complementares no processo criptográfico, assim sendo uma delas será usada para criptografar a mensagem e a outra para decriptografar.

Como a criptografia simétrica está baseada no compartilhamento de uma mesma chave entre o emissor e o destinatário da mensagem, o principal problema deste método é o gerenciamento das chaves: a chave secreta tem que ser gerada, transmitida e armazenada de uma maneira segura e confidencial para garantir que apenas o usuário origem e o usuário destino tenham acesso a essa informação; ela deve ser protegida da ação dos intrusos que poderão tentar capturá-la para decifrar as mensagens secretas intercambiadas entre os parceiros de uma comunicação.

A principal vantagem da criptografia assimétrica é não necessitar desse tipo de gerenciamento de chaves: ela possibilita uma maior segurança por não precisar compartilhar uma mesma chave criptográfica. A chave privada deve ser conhecida apenas pelo usuário proprietário e a chave pública correspondente (necessária para fazer o processo criptográfico inverso) poderá ser conhecida por todos. Apesar da criptografia assimétrica ser bem mais segura que a simétrica, ela possui uma grande desvantagem: para permitir a propriedade de utilizar duas chaves distintas, a sua execução está baseada em protocolos complexos que exigem mais recursos computacionais. Assim sendo os algoritmos simétricos são bem mais rápidos que os assimétricos.

A solução ideal seria combinar as duas técnicas de tal forma que

usufruíssemos a vantagem de segurança do algoritmo assimétrico e da vantagem de rapidez de execução do algoritmo simétrico. Um método híbrido bastante utilizado atualmente baseia-se em:

- Usar criptografia assimétrica para a troca de uma chave secreta temporária - como essa etapa será única e fundamental para a confidencialidade total do intercâmbio, o tempo adicional gasto na criptografia será compensado pela segurança do sigilo oferecido;
- Usar criptografia simétrica para proteger as outras mensagens - essa etapa será repetida inúmeras vezes devendo necessariamente usar um método rápido de criptografia; como a chave criptográfica foi intercambiada de uma maneira sigilosa e é temporária, o algoritmo simétrico fornecerá uma boa segurança.

As mensagens intercambiadas entre os parceiros devem ser protegidas contra modificações que possam a vir ocorrer durante a fase da comunicação. Na realidade, não existem maneiras de evitar que a mensagem seja corrompida mas caso isso venha a ocorrer, os parceiros obrigatoriamente devem identificar essa adulteração.

Dentre os mecanismos de integridade, podemos destacar os mecanismos à prova de colisão (*collision proof*) - que compreendem as funções *hash one-way* ou *message digest*: são funções que a partir de uma mensagem de tamanho variável e aplicando certos cálculos matemáticos, é gerado um valor resumo da mensagem (*hash* ou *checksum*)[12] de tamanho fixo. As principais características dessas funções é que deve ser computacionalmente impossível descobrir a mensagem original a partir do seu valor *hash* e de preferência não deverá existir duas mensagens que gerem o mesmo valor *hash*.

12 Este assunto não será tratado neste trabalho, pois foge de sua proposta inicial.

A assinatura digital é uma aplicação especializada da criptografia utilizada para assegurar a origem da mensagem e a identidade do emissor. Esse esquema está baseado na utilização de algoritmos criptográficos de chave pública: o emissor irá assinar a mensagem usando sua chave privada e o destinatário irá decodificar a assinatura usando a chave pública do emissor. Assim sendo no processo de assinatura digital, a chave privada é usada apenas para assinar uma mensagem e a chave pública é usada apenas para autenticar assinaturas. Uma assinatura digital segura não pode ser repudiada, isto é, um assinante de um documento não poderá rejeitar posteriormente que sua assinatura foi forjada.

Será apresentado a seguir, uma proposta de mecanismo de autenticação baseado na existência de um servidor de autenticação. O *software* Kerberos.

3.2.5.1.1 - Kerberos

Kerberos é um serviço de autenticação distribuída desenvolvida no MIT (*Massachusetts Institute of Technology*) que permite um parceiro provar sua identidade perante um outro parceiro sem enviar dados confidenciais pela rede. Esse processo é realizado como um serviço de autenticação com um terceiro parceiro confiável, utilizando criptografia convencional. Opcionalmente ele também fornece integridade e confidencialidade das mensagens trocadas.

A origem do nome desse serviço é proveniente da mitologia grega, onde KERBEROS era o nome do cachorro de três cabeças que vigiava os portões de Hades e sua principal missão era evitar a entrada de pessoas ou coisas indesejáveis. Sendo assim, esse foi o nome dado ao serviço de autenticação do projeto Athena, por ele estar baseado em três servidores: Servidor de

Autenticação (SA), Servidor de Concessão de *Ticket* (TGS) e Servidor de Administração (KADM).

O Kerberos utiliza criptografia para provar a identidade do usuário manipulando dois tipos de chave:

- Chave secreta do usuário: chave conhecida apenas pelo usuário e pelo Kerberos com a finalidade de autenticar o usuário ao Kerberos; deverá existir uma etapa anterior em que serão cadastrados os clientes e suas chaves secretas ficando armazenadas na base de dados do Kerberos;
- Chave de sessão: chave gerada pelo Kerberos após ter autenticado o usuário e tem por objetivo autenticar o intercâmbio realizado por um determinado par de usuários que definem uma sessão; a chave é gerada atendendo a uma solicitação feita por um dos usuários, sendo válida por um tempo pré-determinado e conhecida apenas pelos dois parceiros para os quais ela foi originalmente gerada.

Existem dois tipos de credenciais usadas neste modelo, o *ticket* e o autenticador:

- O *ticket* é um certificado distribuído pelo Kerberos criptografado com a chave secreta do usuário destino, cuja finalidade será informar com segurança, a identidade do usuário para quem o *ticket* foi originalmente concedido e uma chave de sessão a ser utilizada no intercâmbio; ele também contém dados que garantirão que o usuário que está utilizando esse *ticket* é o mesmo que solicitou a sua concessão.
- O autenticador é uma credencial gerada pelo cliente contendo informações adicionais que, quando comparada com as informações do *ticket*, garante que o cliente que o está apresentando é o mesmo

para o qual o *ticket* foi concedido; ele também é utilizado para evitar *replay* de mensagens devido a sua particularidade de ser único por conter o horário em que ele foi criado.

O protocolo de autenticação básico do Kerberos permite ao usuário com conhecimento da chave secreta do cliente, obter um *ticket* e uma chave de sessão. Normalmente a chave secreta do cliente deveria estar presente cada vez que o cliente realizasse autenticação com um novo servidor. Isso acaba por gerar problemas, pois o cliente precisaria estar constantemente redigitando sua chave secreta. Uma alternativa seria armazenar a chave secreta em uma área da estação, a qual poderia ser capturada por um intruso. Dessa forma o Kerberos armazena apenas os *tickets* e chaves de sessão que possuem validade por um período limitado[13].

O funcionamento do *software* Kerberos pode ser dividido em quatro etapas:

1. Conexão ao sistema: o usuário se autentica perante o Kerberos recebendo um *ticket* e uma chave de sessão correspondente ao intercâmbio cliente X Kerberos, criptografados com a sua chave secreta;
2. Cliente solicita *ticket* ao Kerberos para acessar o servidor: a cada novo serviço a ser utilizado pelo cliente, deve ser solicitado um *ticket* e uma chave de sessão para esse novo intercâmbio: cliente X servidor; essas informações são enviadas para o cliente, criptografadas com a chave de sessão cliente X Kerberos;
3. Cliente acessando servidor: o cliente inicialmente envia o *ticket* recebido para o intercâmbio com o servidor. Como esse *ticket* está

13Normalmente 8 horas

criptografado com a chave secreta do servidor, o mesmo deverá decriptografá-lo e assim terá acesso a chave de sessão cliente X servidor que será válida por um período limitado. Após esse procedimento, a comunicação entre eles poderá ser efetuada de uma maneira confidencial;

4. Manutenção da base de dados do Kerberos: compreende o cadastramento das chaves secretas do cliente e do servidor.

A criptografia tem por objetivo transformar uma informação legível em uma informação inelegível, através de um algoritmo criptográfico e uma chave.

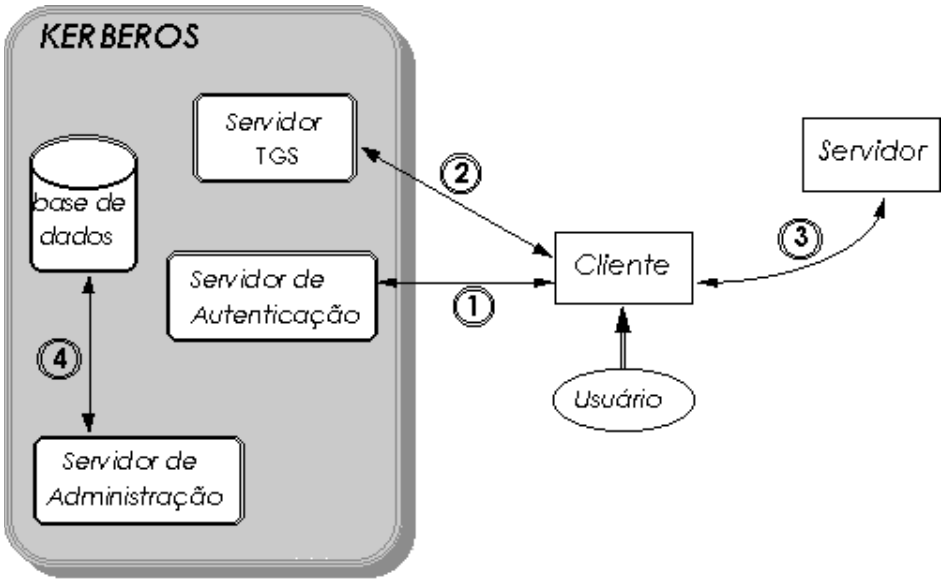


Figura 11 – Modelo de funcionamento do Kerberos

3.3 - Sistemas de Registro de Problemas

O *Trouble Ticket System* (TTS – Sistema de Registro de Problemas), é

utilizado para monitorar os problemas em uma rede, mantendo o rastro do ciclo de vida de um problema. Os TTSs devem manter um histórico completo dos problemas ocorridos, de forma que qualquer operador da rede possa tomar alguma iniciativa sem que para isso tenha de consultar outro operador ou o administrador do sistema.

O propósito do estudo do TTS neste trabalho é apresentá-lo como uma ferramenta de referência para a busca de soluções pelo sistema especialista, para o gerênciamento das falhas ocorridas em uma rede. Entretanto, ele também pode ser utilizado para manter o registro de outros aspectos da rede, tais como modificações de configuração, modificações de segurança, requisições e melhoramento de performance. [LEE, 1996]

São atribuídos a estes sistemas, muitas funções e características, das quais pode-se citar:

- Deve permitir um melhor escalonamento de problemas atribuindo prioridades aos mesmos. Os supervisores e operadores poderão tomar decisões acerca da necessidade ou não de mais pessoal pela carga corrente do NOC (*Network Operation Center)*. Seria interessante permitir que a prioridade dos registros mudassem de acordo com a hora do dia ou em resposta a alarmes de tempo;
- Se o TTS for suficientemente integrado ao sistema de correio eletrônico então alguns registros podem ser despachados diretamente ao responsável;
- atribuir temporizadores[14] para cada registro de problema. Caso o problema não seja resolvido em tempo, automaticamente é acionado um alarme lembrando sobre o problema. A fim de se evitar

14 Timeout

postergação indefinida, pode-se adotar um escalonamento baseado no tempo de espera, no tipo de rede e na severidade do problema;

- Caso a empresa opere em mais de um NOC, enviar relatórios eletronicamente para os representantes de cada rede controlada pelo domínio de gerência, com resumos dos problemas associados a essa rede, de modo a informar sobre o estado corrente de cada ocorrência ainda não solucionada ;
- Fornecer mecanismos para a obtenção de estatísticas tais como Tempo Médio entre Falhas e Tempo Médio de Conserto. Uma coleta e análise apropriada de tais estatísticas permite que se tome medidas preventivas a eventuais falhas em dispositivos do sistema;
- Atuar como filtro dos alertas que sejam relacionados a um registro de problema em aberto;
- Permitir aos usuário e administradores da rede a visualização das atividades desenvolvidas pelo centro de operações de gerência para a resolução de falhas, indicando assim os esforços empregados para a resolução destes.

Um outro propósito dos sistemas de registro de problemas é permitir uma interação entre os diversos domínios envolvidos em um problema. É importante ressaltar que a gerência de redes em um ambiente de processamento distribuído, admite o surgimento de ilhas de gerência, nas quais a responsabilidade pela administração da rede é alocada ao pessoal local. Essa modalidade de segmentação, referida na bibliografia especializada como domínio de gerência, costuma surgir de forma quase natural em redes complexas, heterogêneas e com múltiplos locais de concentração [MAD, 1994]. Se por um lado, a divisão da responsabilidade facilita o diagnóstico, uma vez

que os administradores locais possuem grande conhecimento daquele segmento da rede, por outro lado, a possibilidade de problemas em sub-rede surgirem em função de anomalias de outra sub-rede leva à necessidade de estabelecer algum mecanismo de apoio à interação e cooperação entre os responsáveis pelas diversas sub-redes. Desta forma, sistemas de registro de problemas podem ser usados para compartilhar informações a respeito das soluções adotadas para a resolução dos mais diversos problemas, permitindo a colaboração dos especialistas dos diversos domínios envolvidos no diagnóstico dos problemas.

Um TTS cria para cada problema informado um novo registro, atribuindo a este um número identificador, e registra os dados sobre o problema e ações realizadas ao longo deste, desde a sua criação até o seu encerramento. Os registros podem ser criados automaticamente, a partir de alarmes, ou manualmente, por usuários ou gerentes da rede. Uma vez registrado o problema, o sistema interage com sua base de dados de modo a preencher automaticamente as informações solicitadas pelo registro que ele tem condições de responder.

Todo problema registrado deve ser associado à uma categoria de problemas automaticamente ou manualmente pelo gerente responsável pelo tratamento do problema, o que pode auxiliar no futuro a identificar os problemas que ocorrem mais freqüentemente. Algumas classificações comuns, apontadas em [LEE, 1996], são: falha no enlace, falha em equipamento da rede, brecha na segurança, erro de configuração, problema de performance e questão de contabilização. Podem existir diferentes tipos de registro para os diferentes problemas encontrados em uma rede, variando o formato dos registros principalmente nos campos fixos.

O histórico dos problemas ocorridos pode ser armazenado através de campos fixos ou de texto de forma livre [JOH, 1992]. Os campos fixos têm a vantagem de serem utilizados mais facilmente para busca e ter sua consistência verificada com mais exatidão. Este tipo de armazenamento é apropriado para dados que são fornecidos automaticamente pelo sistema. Embora tendam a tornar os dados mais consistentes e confiáveis e seu uso seja aconselhado para ambientes de resolução de problemas bem compreendidos e específicos, os campos fixos têm a desvantagem de forçar os usuários a escolherem entre valores preparados e permitidos que nem sempre representam a situação com precisão.

A estrutura de um registro de problema para redes de computadores, sugerida por [JOH, 1992], consiste de três partes: cabeçalho, atualizações e dados da resolução. O cabeçalho é responsável pelas informações de abertura do problema, que incluem [JOH, 1992]:

- Hora e data do início do problema;
- Identificação do usuário que abriu o registro;
- Severidade do problema;
- Descrição do problema;
- Quem relatou o problema;
- Quais os equipamentos envolvidos;
- Qual a rede envolvida (quando o NOC é responsável por várias redes);
- Endereço da máquina do usuário;
- Endereço da máquina destino;
- Próxima ação;
- Hora e data para o alarme associado ao problema;
- Para quem enviar o registro;

- Responsável pelo registro.

Neste cabeçalho, os quatro primeiros itens apresentados são sugeridos para todos os sistemas. Os demais são específicos para o armazenamento de informações associadas aos diferentes tipos de problemas. Para permitir uma maior flexibilidade no sistema, pode ser desenvolvido um TTS que apresente características chaves em forma de campo fixo e, em determinados campos, permitir uma maior flexibilidade ao usuário que está registrando o problema dando a ele a possibilidade de redigir a respeito do ocorrido.

Segundo [MELCHIORS, 1999], as informações de atualização representam as ações e diagnósticos realizados ao longo do ciclo de vida do problema. A primeira atualização pode representar uma descrição do problema, já que quando o problema é aberto geralmente sua natureza exata é desconhecida e a descrição fornecida pode ser imprecisa e demasiado complexa. É sugerido exista ao menos um campo de texto livre nesse estágio do problema, para esse tipo de informação. Os demais campos podem ser bastante simples, tais como exemplificado em [JOH, 1992] "*Site chamado; sem resposta"*, e podem ser armazenados tanto em campos fixos como em campos de texto livre. É sugerido ainda que haja sempre uma indicação da próxima ação associada ao registro, que, mais uma vez, pode ser implementada como um campo fixo especial ou como um texto livre.

Finalizando, os dados da resolução dos problemas representam as informações que resumem o problema para futuras análise estatísticas e, também, um guia de referência para resolução em problemas similares futuros. Os campos indicados em [JOH, 1992] que são definidos como úteis para esta etapa são:

- Hora e data da resolução do problema;

- Duração;
- Uma linha descrevendo o ocorrido (para registro no relatório);
- Descrição da resolução do problema;
- Componentes afetados;
- Quem verificou o problema depois que este foi resolvido;
- Quem foi consultado para auxílio na resolução do problema;
- Campo temporário para armazenar informações temporárias utilizadas investigações estatísticas;
- Estado corrente do problema;
- Usuários afetados; e
- Prováveis causas do problema.

Os potenciais usuários de um TTS dependerão de quão sofisticado será o sistema de registro de problemas. Se este sistema tiver um mecanismo de ajuda orientado por um sistema especialista, que é a proposta desse projeto, boa parte do registro poderá ser feita automaticamente. Dessa forma, qualquer usuário, incluindo o usuário final, poderá usufruir do sistema.

Deve-se ressaltar também, que mecanismos de segurança são fundamentais (prover *logs* e *passwords*) para um bom e correto funcionamento de um TTS. Caso o TTS não seja tão amigável, este provavelmente será utilizado somente pelo pessoal que detenha um conhecimento mais aprofundado do sistema. É importante ainda, que o TTS esteja disponível ao usuário final porque diminui a burocracia na solução de qualquer problema.

3.4– Sistemas Especialistas para gerência de redes

Compreendendo como funciona a gerência de redes de computadores, tem-se noção da complexidade e da funcionalidade de um sistema especialista para automação da gerência de redes. Segundo MELCHIORS [apud OLIVEIRA, 2000] o sistema especialista para a gerencia de redes deve implementar dois módulos que compoem a mesma: a monitoração e o controle.

O trabalho de monitoração requer uma atenção especial, pois é através dele que os problemas serão previstos ou detectados. A monitoração consiste na coleta em tempo-real e avaliação dos dados coletados. Estes dados podem ser de vários tipos, cada qual com seu propósito. O monitor da rede é responsável por coletar e armazenar os dados através do monitoramento. Esses dados são todas as informações possíveis de se coletar de um determinado componente da rede. Ao exemplo de um roteador, o monitor poderia inquerí-lo sobre tabela de roteamento, trafego nas linhas, estatística do próprio roteador e tudo o que se relaciona a roteadores.

Normalmente existem três formas dos dados serem obtidos. A primeira delas é a forma mais comum, onde os dados precisam ser constantemente analisados. Nesse caso, o dispositivo da rede envia periodicamente informações para o monitor da rede. Uma segunda forma utilizada é quando não há necessidade de grandes informações de controle, apenas quando ocorre alguma excessão. Para estes casos, o componente da rede somente envia dados ao monitor em situações ocasionais, normalmente quando precisa ser tomada alguma medida de urgência - preventiva ou reparadora. A última forma de obtenção de informações é utilizada em situações especiais, onde o monitor requere ao dispositivo o dado.

O controle deve ser exercido por várias razões. Entre elas podemos citar:

- reparar componentes falhos;
- reconfigurar a rede;
- executar rotinas de manutenção de *software*;
- fazer experimentos com novos *software* e *hardware*.

Quando uma funcionalidade monitorada relata um problema ou quando o monitor alerta para um potencial problema, o gerente da rede precisa localizar a falha específica antes de efetuar os reparos apropriados. O gerenciador da rede precisa ser capaz de efetuar as reconfigurações, se necessárias. Isto inclui insersão e remoção de módulos de programas e habilitação e desabilitação de interfaces. Em suma, o gerenciador da rede deve estar habilitado a efetuar a manutenção da maior gama possível de problemas.

Entretanto, é interessante que o sistema permita que o gerenciamento possa, ser feito da forma convencional, ou seja, que o gerente da rede possa através do teclado ou mouse, selecionar itens de algum menu ou clicar em algum objeto gerenciado para poder inquerí-lo sobre suas condições. As informações colhidas anteriormente devem poder ser integradas com as novas e o sistema deve poder focar as atenções aos itens que o gerente está gerenciando.

Com a automação do processo de gerência de rede, temos todo o ganho que um sistema especialista pode fornecer: velocidade de processamento de uma máquina com o tipo de raciocício de um humano [OLIVEIRA, 2000].

Uma rede de computadores livre de erros é provida por flexibilidade de *hardware*, redundância e funções de diagnósticos inteligentes. O sistema de gerenciamento da rede deve, continuamente, monitorar a demanda de tráfego e o ajuste dos componentes da rede. Para que haja um controle automatizado da rede deve haver uma monitoração adequada.

A arquitetura de um SGRBC foi concebida de maneira a permitir alta

modularidade [MUR, 1994]. Os sistemas baseados em conhecimento podem ser desenvolvidos independentemente e cada um deles conter conhecimento próprio, permitindo melhorias nos níveis de desempenho e disponibilidade. Uma arquitetura típica de um SGRBC é apresentada na Figura 12.

Os seguintes elementos de informação, conceituados abaixo, são necessários para o entendimento das funcionalidades de cada módulo da arquitetura [HOLANDA, 1998]:

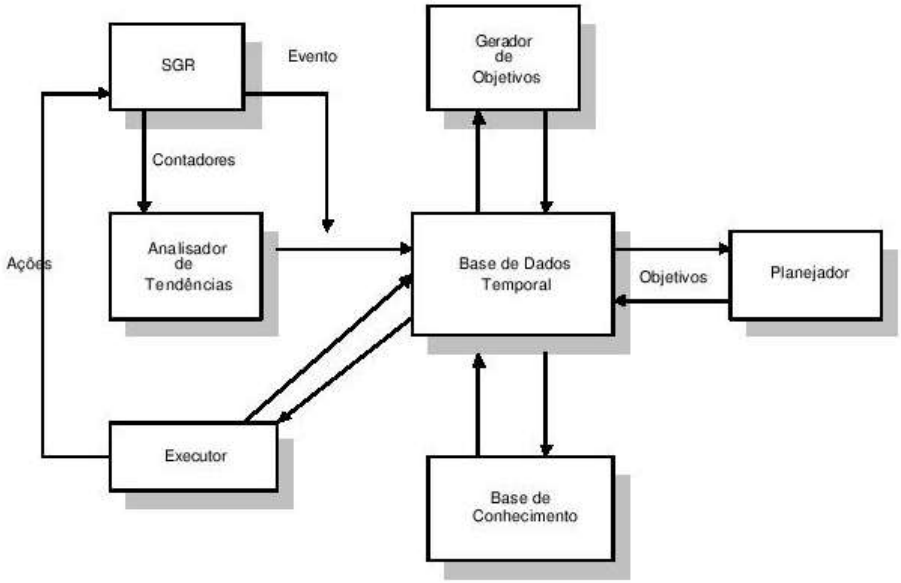


Figura 12 – Arquitetura SGRBC [Fonte HOLANDA, 1998]

- 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 temportal;
- 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*).

Segundo [HOLANDA, 1998] os principais componentes da arquitetura dos Sistemas de Gerenciamento de Redes Baseados em Conhecimento podem ser descritos como:

- 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 armazenados 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;
- 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.

Os sistemas especialistas podem ser utilizados na gerência de redes para diversos fins. No gerenciamento de configuração eles podem auxiliar no planejamento de redes. Para que isso ocorra, esses sistemas devem possuir informações sobre a topologia, física e lógica da rede, bem como os mapas de roteamento.

Na área de gerenciamento de falhas é que tivemos os primeiros sistemas especialistas. Nesta área, eles podem ser utilizados para o diagnóstico e manutenção das redes. As ferramentas de diagnósticos efetuam a coleta de dados e análise das falhas da rede e seus impactos, a fim de determinar as prováveis causas e definir os reparos e manutenção necessários para resolver o problema. Os benefícios dos sistemas especialistas de diagnósticos incluem: diminuir o tempo para detectar as causas do problema, sugerir aos gerentes de redes ações para resolver o problema; e automatizar a resolução de problemas pela intervenção direta, resultando em comandos corretivos para uma rede inteligente.

O controle da rede também é uma outra aplicação de sistemas especialistas para gerenciamento de falhas, sendo utilizado para estender as capacidades dos operadores de rede, e não para substituí-los. Os benefícios de tais sistemas são o aumento da precisão e eficiência da intervenção do operador, facilitação no processo de tomada de decisão e redução na quantidade de tempo necessária para restaurar ou alterar a rede [ERI, 1989]. Além do diagnóstico e

controle, sistemas especialistas podem ser aplicados também para a interpretação de eventos, fornecendo mensagens de acordo com a ordem e os códigos de propriedade associados.

As áreas de gerenciamento de performance, gerenciamento de contabilização e gerenciamento de segurança também podem ser abordadas pelos sistemas especialistas. Uma aplicação a área de gerenciamento de segurança, pode combinar o conhecimento sobre o sistema alvo, o perfil da história das atividades passadas dos usuários e heurísticas de detecção de intrusão, a fim de detectar violações específicas que ocorrem no computador alvo [ERI, 1989].

3.5– Considerações Finais

Neste capítulo foi apresentado todo o conceito necessário para o gerenciamento de redes de acordo com as áreas funcionais definidas pela ISO. Através do SNMP torna-se possível o monitoramento da rede e, por intermédio dele, detecta-se possíveis anomalias nesta rede. Com o auxílio de um TTS a avaliação do problema detectado pode ser realizada e, conseqüentemente, ter uma solução proposta pelo sistema.

No próximo capítulo será realizada uma análise básica de uma ferramenta com propósito semelhante ao proposto neste documento.

4. Estudo de caso – SAGRES

O SAGRES é um sistema de gerenciamento de redes focado exclusivamente na área funcional de gerência de falhas, desenvolvido pelo aluno de mestrado da UFC (Universidade Federal do Ceará), Ramir Holanda Filho. A Figura 13 apresenta o modelo de concepção do SAGRES, acrescido do refinamento do elemento Arquitetura Funcional.

Segundo o autor [HOLANDA, 1998], dois motivos foram determinantes na decisão de focar o SAGRES na área de gerencia de falhas:

- Falha é uma área funcional de destaque no que se refere a gerência de redes de computadores;
- Existe todo um formalismo no tratamento de falhas que facilita a concepção de uma arquitetura [DAV, 1994].

A arquitetura funcional do SAGRES é derivada de sua infraestrutura conceitual. O SAGRES faz uso da metodologia DAG (Desenvolvimento de Aplicações de Gerenciamento) aplicada à gerência de falhas e dos conceitos de SGRBC.

A metodologia DAG especifica quais atividades os administradores de redes devem contemplar para desenvolverem aplicações de gerenciamento de redes. Como disposto na Figura 14, ela está dividida em três fases:

1. Levantamento da necessidade e do ambiente de gerenciamento;
2. Tratamento das informações de gerenciamento;
3. Geração da aplicação de gerenciamento.

O diagnósito de falhas realizado pelo SAGRES é baseado em heurística e para realizar a detecção de falhas é realizada a comparação entre o comportamento normal da rede e o comportamento apresentado por ela.

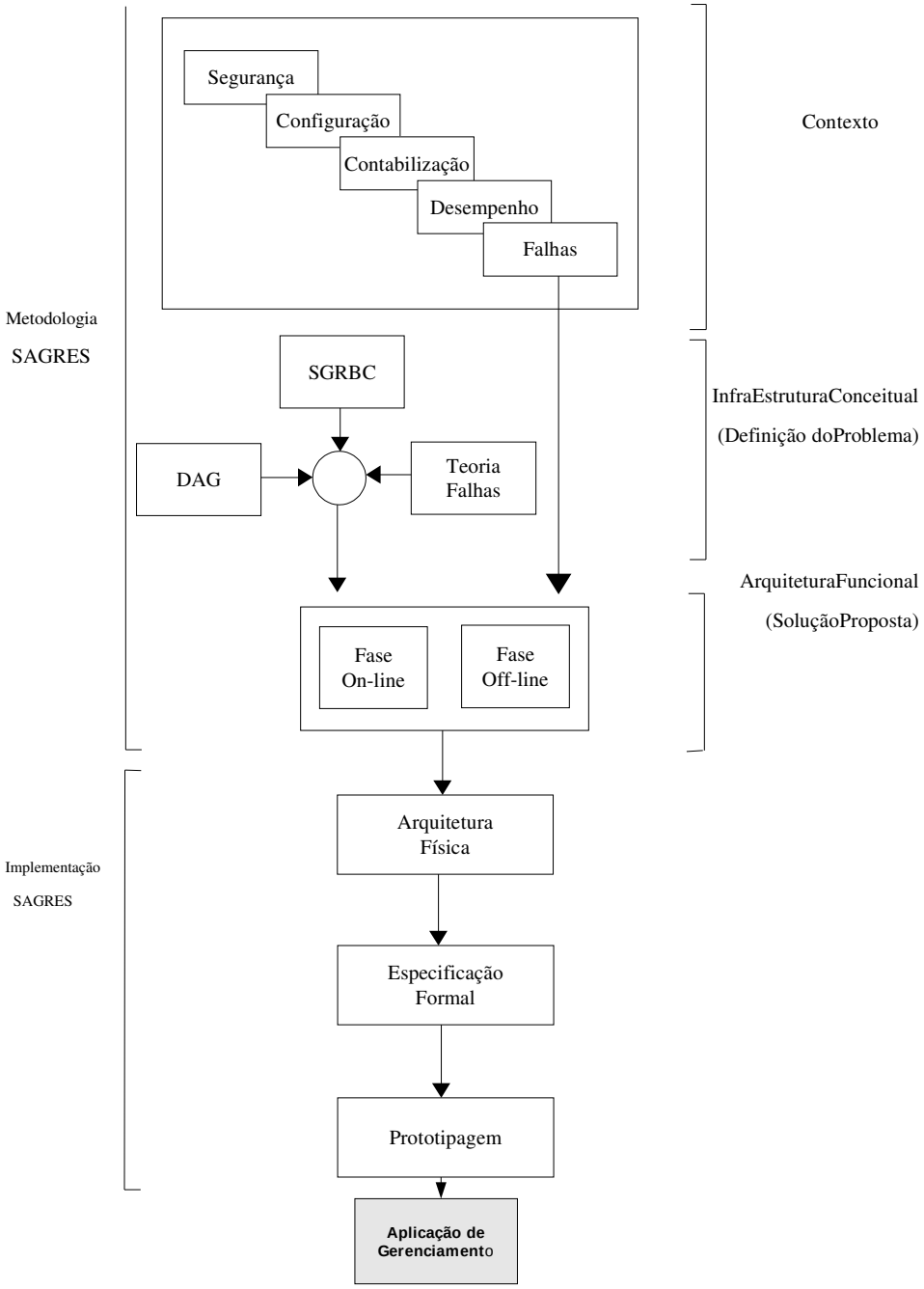


Figura 13 – Modelo de concepção do SAGRES – Arquitetura funcional

A arquitetura funcional do SAGRES é definida em sua infraestrutura conceitual (Teoria de Falhas, Metodologia DAG e SGRBC). Seu funcionamento é caracterizado pela existência de duas fases, a serem executas de forma seqüêncial, conforme ordem apresentada:

- Fase *Off-Line*: Esta fase consiste na formulação das regras que irão auxiliar o administrador de redes a detectar os possíveis problemas. Antes que o sistema esteja em operação propriamente dito (Fase *On-Line*) é necessário dispor das regras que irão auxiliar o administrador. A fase *Off-Line* é composta por três processos:

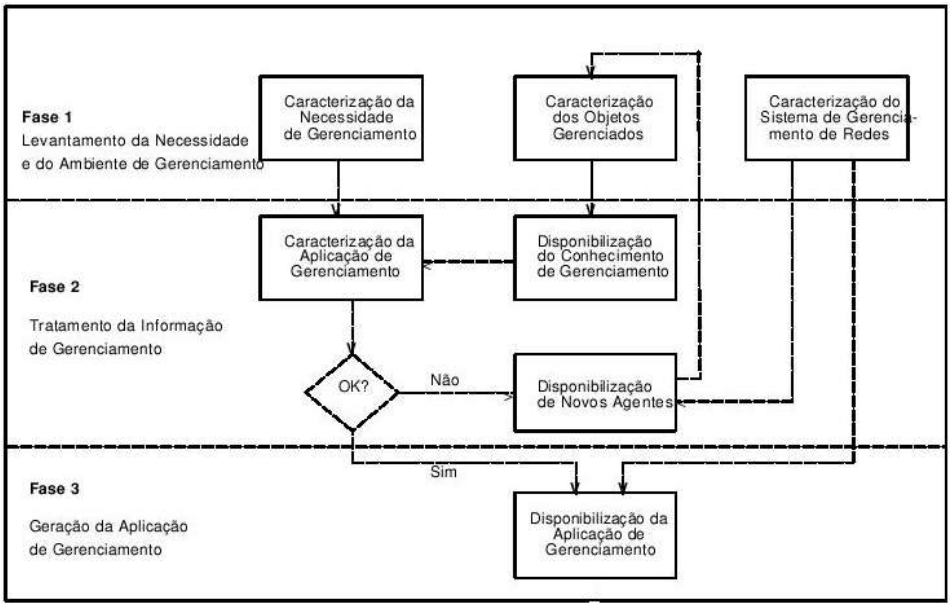


Figura 14 – Arquitetura DAG

1. Levantamento e seleção dos objetos que irão compor a MIB: A escolha dos objetos a serem gerenciados possui influência direta nas regras que compõem o sistema. A metodologia DAG

descreve que cada objeto gerenciado deve ser catalogado com sua respectiva descrição. Ainda nesta fase, inicia-se a coleta dos valores apresentados pelos objetos gerenciados através de consecutivas ações de *GET,* para posteriormente serem armazenadas em um arquivo de log.

2. Geração do *baseline:* Após a geração dos arquivos de *logs,* uma *baseline* da rede a ser gerenciada foi gerada, onde definiu-se os valores ou faixa de valores que são considerados dentro da normalidade.
3. Consturção de regras: Este é o passo final da fase *Off-Line*. A construção das regras que irão monitorar a rede pode ocorrer por intermédio de uma análise da *baseline*, onde são definidas as regras de acordo com o comportamento observado e tido como normal, ou então, as regras podem ser definidas através da literatura existente e da opinião dos especialistas na área. Estas regras são conhecidas como regras heurísticas. Em alguns momentos é possível que ocorra conflito entre as regras definidas pelo comportamento da rede e as regras heurísticas. Para evitar este problema as regras foram cadastradas de maneira a oferecerem diagnósticos baseados em heurística e em comportamento.

A fase *Off-Line* é concluída com a construção da base de conhecimento, resultado da execução do módulo "Construção de Regras".

- Fase *On-Line*: A elaboração da base de conhecimento é fator condicional para que o sistema possa entrar em operação. As atividades desta fase podem ser agrupadas em três processos:

1. Coleta de dados dos objetos gerenciados: Este processo corresponde ao comportamento observado nos objetos gerenciados. Para que isso ocorra, o administrador interage com o sistema fornecendo os seguintes parâmetros: IP do elemento de rede a ser gerenciado, relação dos objetos, período de monitoramento e o intervalo entre estes. Os dados coletados são armazenados em uma base denominada "Dados Coletados".
2. Inferência: Este processo é responsável pela análise e diagnóstico do estado da rede. Para isso, ele se utiliza das regras existentes na Base de Conhecimento e das informações contidas na base Dados Coletados. Para qualquer problema encontrado, o sistema deve apresentar as correções necessárias.
3. Alteração parâmetros da MIB: Os procedimentos de correção do SAGRES podem ocorrer de duas maneiras. Na primeira ele interage com o administrador da rede expondo-o a origem do problema propondo a solução adequada para o mesmo. Na segunda, além de interagir com o administrador, ele pode iniciar o processo de alteração dos parâmetros da MIB de forma automática, através do comando SNMP SET.

4.1– Considerações Finais

Como constatado neste capítulo, um SGRBC não é uma ferramenta de simples desenvolvimento. A abrangência de diversas áreas da Ciência da Computação em sua concepção exige do desenvolvedor, ou equipe de desenvolvimento, conhecimentos específicos em todas as áreas envolvidas.

Certamente, a melhor meneira de desenvolvimento de um SGRBC é estruturá-lo em módulos, conforme apresentado neste documento.

No próximo capítulo as últimas considerações a cerca do tema serão realizadas.

5. - Conclusão

O propósito do desenvolvimento deste trabalho foi apresentar os conceitos teóricos inerentes às áreas de IC e de Gerência de Redes de Computadores para o desenvolvimento de um SGRBC. Com este intuito, o trabalho foi estruturado em cinco partes: Introdução, Sistemas Especialistas, Gerência de Redes, Estudo de Casos e Conclusão.

Os benefícios que um sistema especialista baseado em casos associado às técnicas de monitoramento de redes pode fornecer são extremamente importantes para um administrador de redes. O tempo necessário para o descobrimento e correção de anomalias em uma estrutura de redes é reduzido significativamente além de existir também a contribuição direta para manter o desempenho e estabilidade da rede.

Sem a integração do sistema especialista com as técnicas disponíveis para o gerenciamento de redes, este trabalho torna-se impraticável. O papel de ferramentas como o SNMP e TTS é fundamental para o sucesso do sistema. Com o uso do protocolo SNMP o monitoramento é realizado constantemente através dos Gerentes e Agentes presentes na estrutura, onde a presença de qualquer anomalia deve ser informada o quanto antes ao Gerente para que a mesma seja tratada pelo sistema. O TTS é o local onde todo o conhecimento do especialista deverá ser armazenado através de casos. Com base nos parâmetros detectados no problema, um caso semelhante deverá ser resgatado e sua solução adaptada, ou não, apresentada.

O protótipo estudado (SAGRES) realiza uma comparação do comportamento apresentado pela rede e do comportamento definido como ideal. Quando houver algum problema este será detectado e tratado. Este protótipo utiliza os conceitos de MIB, Gerentes e Agentes e é em muitos pontos semelhante ao proposto neste trabalho.

A manipulação da base de casos (TTS) pelo mecanismo de inferência é o ponto chave na implantação do projeto. A definição dos índices que servirão de referência para pesquisa no TTS será realizada de forma manual, pois como visto neste trabalho o resultado obtido é mais rápido e preciso. A princípio, a técnica para recuperação dos casos utilizada será a do "Vizinho Mais Próximo", pois a comparação individual dos parâmetros considerando o peso que cada atributo possui na definição final do problema, é um fator que contribui em muito para o sucesso desta técnica.

A organização dos casos no TTS deverá ocorrer através da técnica de Memória Dinâmica com o uso dos MOPs. Desta forma será possível organizar a base de conhecimento em classes de casos, facilitando assim a busca por um casos já existentes. Em cada classe existente, os casos estarão indexados através dos atributo definidos a eles como índices.

Certamente, o uso do SNMP integrado às técnicas de representação e busca do conhecimento apresentadas neste trabalho é uma das características principais deste projeto. Conseqüentemente merecerá atenção especial nos projetos futuros.

5.1– Trabalhos futuros.

O objetivo principal deste trabalho foi fornecer os conceitos necessários para o desenvolvimento de um SGRBC. A estrutura e os conceitos necessários e para desenvolvimento do sistema foram estudados, ficando como próximo objetivo a implementação deste projeto.

Para os dois anos seguintes, o objetivo é realizar estudos de implementação de RBC de forma integrada às ferramentas de gerência de redes no curso de mestrado. Dessa integração surgirá então o SGRBC denominado SYNEMA[15], uma espécie de acrônico para *NEtwork MAnagement SYstem*

15 Nome provisório

6 – Bibliografia

[ABEL, 1996] ABEL, Mara. *Um estudo sobre raciocínio Baseado em Casos.* Porto Alegre: UFRGS. Trabalho indivudual.

[AZAMBUJA, 2001] AZAMBUJA, Marcelo Cunha. *PSWeM: Desenvolvimento e Implementação de Uma Ferramenta Baseada na Web para Gerênciamento de Redes ao Nível de Serviço.* Porto Alegre: PUCRS. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

[AZEVEDO, 1999] AZEVEDO, Sérgio Lund. *Desenvolvimento de um protótipo de sistema especialista para escolha do tipo de fundações.* Porto Alegre: UFRGS. Tese de Doutorado apresentada ao departamento de Engenharia Civil.

[BELLO, 1995] BELLO, Roberto. *Automação de Gerência de Redes de Computadores.* [on-line]. Disponível em: <http://penta.ufrgs.br/gr952/trab1/geren.html>. Documento capturado em setembro de 2004.

[BRANCO, 1999] BRANCO, Moisés A. C. *Um Algoritmo para Diagnóstico Distribuído de Falhas em Redes de Computadores.* Fortaleza: UFC. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

[CAMARGO, 1999] CAMARGO, Katia Gavranich. *Inteligência Artificial Aplicada à Nutrição a Prescrição de Planos Alimentares.* Florianópolis: UFSC. Dissertação de Mestrado apresentada ao departamento de Engenharia de Produção.

[COSTA, 1999] COSTA, Marcello Thiry Comicholi da. *Uma Arquitetura Baseada em Agentes para Suporte ao Ensino à Distância.* Florianópolis: UFSC. Dissertação de Mestrado apresentada ao departamento de Engenharia de Produção.

[DAZZI, 1999] DAZZI, Rudimar Luís Scaranto. *Sistemas Especialistas Conexionistas: Implementação por Redes Diretas e Bidimensionais.* Florianópolis: UFSC. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

[DELPIZZO, 1997] DELPIZZO, Vanessa Lins Francalacci. *Prescrição de Atividades Físicas através do Uso da Inteligência Artificial.* Florianópolis: UFSC. Dissertação de Mestrado apresentada ao departamento de Engenharia de Produção.

[FERNANDES, 1996] FERNANDES, A. M. R. *Sistema Especialista Difuso Aplicado ao Processo de Análise Química Qualitativa de Amostras de Minerais.* Florianópolis: UFSC. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

[FERNANDES, 2003] FERNANDES, Anita Maria da Rocha. *Inteligência Artificial – noções gerais.* Florianópolis: Editora Visual Books, 2003.

[FreeNMS, 2003] Free Network Management System. [on-line]. *FreeNMS – Documentação e Descrição do Projeto.* Disponível em: <http://www.freenms.org>. Documento capturado em outubro de 2004.

[HOLANDA, 1998] HOLANDA, Ramir F. *SAGRES: Um Sistema Baseado em Conhecimento para Apoio à Gerência de Falhas em Redes de Computadores.* Fortaleza: UFC. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

[JOH, 1992] JOHNSON, D. *Internal Integrated Trouble Ticket System: Funcional Specification Whishlist.* RFC 1297, 1992.

[KOLODNER, 1993] KOLODNER, J. *Case-Based Rasoning.* San Mateo CA. Morgan Kaufmann Publishers, 1993.

[KOSLOSKY, 1999] KOSLOSKY, Marco Antônio Neiva. *Aprendizagem Baseada em Casos: Um ambiente para Ensino de Lógica de Programação*. Florianópolis: UFSC. Dissertação de Mestrado apresentada ao departamento de Engenharia de Produção.

[KUROSE, 2003] KUROSE, James F. *Redes de Computadores e a Internet:*

uma nova abordagem. São Paulo: Addison Wesley, 2003.

[LAGEMANN, 1998] LAGEMANN, Gerson Volney. *RBC para o Problema de Suporte ao Cliente nas Empresas de Prestação de Serviços de Software: O Caso Datasul.* Florianópolis: UFSC. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

[LEE, 1998] LEE, Rosina Weber. *Pesquisa Jurisprudencial Inteligente.* Florianópolis: UFSC. Tese de Doutorado apresentada ao departamento de Engenharia de Produção.

[LEWIS, 1995] LEWIS, Lundy. *AI and Intelligent Networks in the 1990s and into the 21st Century.* In: Approaches to Telecommunications and Network Management. Amsterdan: IOS Press, 1995.

[MAÑAS, 1999] MAÑAS, Antonio Vico. *Administração de Sistemas de Informação.* São Paulo: Érica, 1999.

[MEIRELLES, 1997] MEIRELLES, L. F. T. *Uma Proposta para o Gerenciamento de Aplicações em Rede.* Florianópolis: UFSC. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

[MELCHIORS, 1999] MELCHIORS, Cristina. *Um Estudo sobre Raciocínio Baseado em Casos.* Relatório técnico. Porto Alegre, PPGC da

UFRGS, 1999.

[MELCHIORS, 1999] MELCHIORS, Cristina. *DUMBO: Uma Abordagem para Gerenciamento de Falhas Utilizando Raciocínio Baseado em Casos*. Porto Alegre: UFRGS. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

[NASCIMENTO, 1999] NASCIMENTO, Adriano S. *Desenvolvendo Agentes Inteligentes para a Gerência Pró-Ativa de Redes ATM*. Fortaleza: UFC. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

[RAMOS, 1994] RAMOS, Suzana. *Uma Metodologia para Análise e Desenvolvimento de Aplicações de Gerenciamento de Redes de Computadores*. Recife: UFPE. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação.

RFC 1297. *NOC Internal Integrated Trouble Ticket System – Functional Specification Whishlist.* [on-line]. Disponível em: <http://www.ietf.org/rfc/rfc1297.txt>. Documento capturado em agosto de 2004.

RFC 1906. *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)* [on-line]. Disponível em: <http://www.ietf.org/rfc/rfc1906.txt>. Documento capturado em agosto de 2004.

RFC 2011. *SNMPv2 Management Information Base for the Internet Protocol using SMIv2* [on-line]. Disponível em: <http://www.ietf.org/rfc/rfc2011.txt>. Documento capturado em agosto de 2004.

RFC 2578. *Structure of Management Information Version 2 (SMIv2)* [on-line]. Disponível em: <http://www.ietf.org/rfc/rfc2578.txt>. Documento capturado em agosto de 2004.

[RUSSEL, 2002] RUSSEL, Ryan. *Rede Segura – Network.* Rio de Janeiro: Alta Books, 2002.

[RUSSEL, 2004] RUSSEL, Stuart J. *Inteligência Artificial 2ªed.* Rio de Janeiro: Elsevier, 2004.

[SHORTLIFFE, 1975] SHORTLIFFE, E. H. *A computer program providing antimicrobial therapy recommendations* [on-line]. Disponível em: <http://www.dcc.ufmg.br/pos/html/spg99/anais/helderss/helderss.html>. Documento capturado em dezembro de 2004.

[SOARES, 1995] SOARES, Luiz Fernando G. *Redes de computadores: das LANs, MANs e WANs às redes ATM.* Rio de Janeiro: Campus, 1995.

[SOUZA, 1999] SOUZA, Hélder Soares. *ORPHEUS – Sistema Especialista Para Avaliação de Disfonias em Voz Profissional* [on-line].

Disponível em: <http://www.dcc.ufmg.br/pos/html/spg99/anais/helderss/helderss.html>. Documento capturado em dezembro de 2004.

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

[STEVENS, 1994] STEVENS, W. R. TCP/IP Illustrated,Volume 1 - The Protocols. Addison Wesley. 1994.

[TANENBAUM, 1994] TANENBAUM, Andrew S. *Redes de computadores 4ª ed.* Rio de Janeiro: Elsevier, 2003

[UCHÔA, 1998] UCHÔA, Joaquim Quinteiro. *Representação e indução do conhecimento usando teoria dos conjuntos aproximados.* São Carlos: UFSCar, 1998.

[WATSON, 1997] WATSON, Ian. *Applying Case-Based Reasoning: Techiniques for Enterprise Systems.* San Francisco: Morgan Kaufmann, 1997.