

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Raciocínio Baseado em Casos
Aplicado ao Gerenciamento de Falhas
em Redes de Computadores**

por

CRISTINA MELCHORS

Dissertação submetida à avaliação, como requisito parcial para
a obtenção do grau de Mestre em
Ciência da Computação

Profa. Liane Margarida Rockenbach Tarouco
Orientadora

Porto Alegre, agosto de 1999

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Melchiors, Cristina

Raciocínio Baseado em Casos Aplicado ao Gerenciamento de Falhas em Redes de Computadores / por Cristina Melchiors. — Porto Alegre: PPGC da UFRGS, 1999.

151f. : il.

Dissertação (mestrado) — Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 1999. Orientadora: Tarouco, Liane Margarida Rockenbach.

1. Gerenciamento de falhas 2. Sistemas de registro de problemas
3. Raciocínio baseado em casos 4. Gerenciamento de redes I. Tarouco, Liane Margarida Rockenbach. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Pós-Graduação: Prof. Franz Rainer Semmelmann

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenadora do PPGC: Profa. Carla Maria Dal Sasso Freitas

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

Gostaria de agradecer, em primeiro lugar, à minha orientadora, profa. Liane Tarouco, pelos seus valiosos conselhos, ensinamentos e atenção recebidos desde os tempos de auxiliar de pesquisa.

Um enorme agradecimento aos meus pais, pelo inesgotável e constante apoio e incentivo recebidos durante todo este período, e pela incrível paciência e atenção que sempre tiveram. Agradeço, também, às minhas irmãs, Lúcia e Paula, e meus cunhados, pelo carinho e compreensão.

Um agradecimento, também enorme e muito especial, ao Mauro, que tanto me incentivou e apoiou durante todo este trabalho, pelo enorme e constante carinho, paciência, compreensão, companheirismo.

Aos gerentes de redes, meus agradecimentos pela paciência e auxílio prestados nas diversas entrevistas e consultas, em especial à Cristina Nunes, Fernando Krahe e Margareth Schaffer. Aos meus colegas do CPGCC, pelo apoio e sugestões.

Agradeço, também, aos funcionários do CPGCC, à sempre atenciosa equipe da biblioteca, à Eliane, à equipe da administração da rede.

Por fim, meu agradecimento ao CNPq pelo auxílio financeiro que possibilitou a realização deste trabalho, à UFRGS e aos professores do curso pelos conhecimentos, auxílio e incentivo recebidos.

Sumário

| | |
|--|-----------|
| Lista de Abreviaturas | 7 |
| Lista de Figuras..... | 8 |
| Lista de Tabelas | 10 |
| Resumo..... | 11 |
| Abstract | 12 |
| 1 Introdução | 13 |
| 2 Raciocínio Baseado em Casos | 15 |
| 2.1 Histórico..... | 16 |
| 2.2 O Ciclo CBR | 17 |
| 2.3 A Memória de Casos | 18 |
| 2.4 Base de Casos..... | 18 |
| 2.4.1 Representação dos Casos | 19 |
| 2.4.2 Indexação..... | 21 |
| 2.5 Recuperação | 22 |
| 2.5.1 Estruturas de organizações da base de casos e algoritmos de recuperação ... | 24 |
| 2.5.1.1 Memória linear com busca serial..... | 24 |
| 2.5.1.2 Rede de características compartilhadas | 25 |
| 2.5.1.3 Redes de discriminação..... | 25 |
| 2.5.1.4 Redes de discriminação redundante..... | 26 |
| 2.5.2 Casamento e Classificação | 26 |
| 2.5.3 Considerações sobre a indexação e a recuperação | 29 |
| 2.6 Reutilização (adaptação)..... | 29 |
| 2.7 Revisão..... | 32 |
| 2.8 Armazenamento (aprendizado)..... | 34 |
| 2.9 Exemplos de Aplicações CBR | 35 |
| 2.9.1 Sistema CASCADE..... | 35 |
| 2.9.2 Sistema Smart | 37 |
| 2.10 Considerações Finais | 39 |
| 3 Gerência de Redes..... | 40 |
| 3.1 Introdução | 40 |
| 3.2 Áreas Funcionais da Gerência de Redes..... | 41 |
| 3.3 Sistemas de Registro de Problemas | 43 |
| 3.4 Sistemas Especialistas para Gerência de Redes..... | 45 |

| | |
|---|------------|
| 3.5 Raciocínio Baseado em Casos em Gerência de Redes | 48 |
| 3.5.1 Sistema NETTRAC | 49 |
| 3.5.2 Sistema ExSim | 52 |
| 3.5.3 Sistema CRITTER | 54 |
| 3.6 Considerações Finais | 58 |
| 4 Raciocínio Baseado em Casos Aplicado a um Sistema de Registro de Problemas: Sistema DUMBO..... | 59 |
| 4.1 Motivação..... | 59 |
| 4.2 Aquisição do Conhecimento..... | 60 |
| 4.3 Problemas Típicos das Redes Pesquisadas | 61 |
| 4.3.1 Camada Interface de Rede..... | 63 |
| 4.3.1.1 Ethernet | 65 |
| 4.3.1.2 Linhas Seriais | 67 |
| 4.3.2 Camada Internet | 68 |
| 4.3.3 Camada de Transporte | 70 |
| 4.3.4 Camada de Aplicação | 71 |
| 4.4 Considerações sobre os Problemas do Domínio | 72 |
| 5 Sistema DUMBO: Modelagem | 74 |
| 5.1 Sistema DUMBO - Uma Visão Geral das Técnicas Adotadas..... | 74 |
| 5.2 Representação dos Tipos de Problemas..... | 78 |
| 5.3 Estrutura do Sistema | 81 |
| 5.4 Base de Conhecimento..... | 83 |
| 5.4.1 Representação dos Casos | 83 |
| 5.4.2 Base de Casos: Organização e Indexação..... | 88 |
| 5.4.3 Base de Conhecimento Geral do Domínio..... | 88 |
| 5.5 Modelagem dos Processos CBR | 89 |
| 5.5.1 Definição do Contexto..... | 90 |
| 5.5.1.1 Elaboração dos Prováveis Componentes com Falha Física | 91 |
| 5.5.1.2 Elaboração dos Tipos de Problema Relacionados..... | 93 |
| 5.5.2 Recuperação: Busca e Classificação | 95 |
| 5.5.2.1 Algoritmo de Busca..... | 95 |
| 5.5.2.2 Classificação: Similaridade entre Características | 96 |
| 5.5.2.3 Classificação: Relevância das Características | 99 |
| 5.5.2.4 Classificação e Ordenamento | 102 |
| 5.5.3 Reutilização e Revisão..... | 104 |
| 5.5.3.1 Redefinindo o Contexto..... | 105 |
| 5.5.4 Aprendizado | 107 |
| 5.6 Considerações Finais | 108 |
| 6 Protótipo do Sistema | 109 |
| 6.1 Implementação..... | 109 |
| 6.1.1 Arquitetura do sistema | 109 |
| 6.1.2 Interação com o sistema..... | 117 |
| 6.2 Avaliação do Protótipo | 120 |

| | |
|--|------------|
| 7 Considerações finais | 128 |
| 7.1 Trabalhos Futuros | 129 |
| Anexo 1 Rede Semântica dos Tipos de Problemas | 131 |
| Anexo 2 Características de um Caso | 135 |
| Anexo 3 Alguns Casos do Sistema | 141 |
| Anexo 4 Especificação SDL do Sistema..... | 143 |
| Bibliografia..... | 146 |

Lista de Abreviaturas

| | |
|-------|---|
| CBR | Case-Based Reasoning (raciocínio baseado em casos) |
| CMIP | Common Management Information Protocol |
| CMIS | Common Management Information Services |
| CNPq | Conselho Nacional de Desenvolvimento Científico e Tecnológico |
| CPGCC | Curso de Pós-Graduação em Ciência da Computação |
| CPU | Central Processing Unit |
| IA | Inteligência Artificial |
| IAD | Inteligência Artificial Distribuída |
| IP | Internet Protocol |
| ISO | International Organization for Standardization |
| MBR | Model-Based Reasoning (raciocínio baseado em modelos) |
| MOP | Memory Organization Packets |
| OSI | Open Systems Interconnection |
| RBR | Rule-Based Reasoning (raciocínio baseado em regras) |
| SDL | Specification and Description Language |
| SNMP | Simple Network Management Protocol |
| TCP | Transmission Control Protocol |
| UFRGS | Universidade Federal do Rio Grande do Sul |

Lista de Figuras

| | |
|---|-----|
| FIGURA 2.1 - Ciclo CBR | 17 |
| FIGURA 2.2 - Esquema do processo de recuperação na base de casos | 23 |
| FIGURA 2.3 - Esquema do processo de revisão | 33 |
| FIGURA 2.4 - Exemplo de um caso do sistema CASCADE | 36 |
| FIGURA 2.5 - Arquitetura do sistema CASCADE | 36 |
| FIGURA 2.6 - Esquema de uma consulta de um caso no sistema Smart..... | 38 |
| FIGURA 3.1 - Arquitetura do sistema NETTRAC | 51 |
| FIGURA 3.2 - Estrutura do ExSim | 52 |
| FIGURA 3.3 - Função de similaridade utilizada no sistema ExSim..... | 53 |
| FIGURA 3.4 - Exemplo de um registro de problema | 55 |
| FIGURA 3.5 - Exemplo de um determinador | 55 |
| FIGURA 3.6 - Exemplo de adaptação parametrizada..... | 56 |
| FIGURA 3.7 - Exemplo de adaptação por abstração | 57 |
| FIGURA 3.8 - Arquitetura do sistema CRITTER..... | 57 |
| FIGURA 5.1 - Rede semântica dos tipos de problemas..... | 79 |
| FIGURA 5.2 - Rede semântica dos tipos de problemas (parte 2) | 80 |
| FIGURA 5.3 - Rede semântica da localização dos problemas | 81 |
| FIGURA 5.4 - Estrutura do sistema proposto..... | 82 |
| FIGURA 5.5 - Componentes de um caso..... | 87 |
| FIGURA 5.6 - Hierarquia da base de casos | 88 |
| FIGURA 5.7 - Possíveis componentes envolvidos em falhas físicas e de configuração do HW | 92 |
| FIGURA 5.8 - Função para cálculo da probabilidade final de cada tipo de problema.... | 95 |
| FIGURA 5.9 - Algoritmo para recuperação na base de casos..... | 96 |
| FIGURA 5.10 - Função de avaliação numérica utilizada para a classificação dos casos recuperados | 103 |
| FIGURA 5.11 - Função para cálculo da confiabilidade do casamento entre dois casos | 103 |

| | |
|---|-----|
| FIGURA 5.12 - Função para cálculo do fator de ordenamento | 104 |
| FIGURA 5.13 - Esquema do processo de reutilização e revisão do sistema | 105 |
| FIGURA 6.1 - Diagrama do sistema Dumbo | 110 |
| FIGURA 6.2 - Diagrama do bloco cliente..... | 110 |
| FIGURA 6.3 - Diagrama do bloco servidor | 112 |
| FIGURA 6.4 - Definição do Servidor | 113 |
| FIGURA 6.5 - Definição do CBR..... | 115 |
| FIGURA 6.6 - Tela inicial no registro do caso corrente | 117 |
| FIGURA 6.7 - Tela de coleta de características adicionais para linhas seriais | 118 |
| FIGURA 6.8 - Tela com casos recuperados..... | 119 |
| FIGURA A1.1 - Rede semântica da localização dos problemas (parte 1) | 130 |
| FIGURA A1.2 - Rede semântica da localização dos problemas (parte 2) | 131 |
| FIGURA A1.3 - Rede semântica da localização dos problemas (parte 3) | 132 |
| FIGURA A1.4 - Rede semântica da localização dos problemas (parte 4) | 133 |
| FIGURA A4.1 - Definição do InterpretaDados..... | 142 |
| FIGURA A4.2 - Definição do MostraTela..... | 142 |
| FIGURA A4.3 - Definição do ProcessaOperações..... | 143 |
| FIGURA A4.4 - Definição do ComunicaçãoCliente..... | 143 |
| FIGURA A4.5 - Definição do InterfaceBD..... | 144 |
| FIGURA A4.6 - Definição do ComunicaçãoServidor | 144 |

Lista de Tabelas

| | |
|---|-----|
| TABELA 2.1 - Técnicas de adaptação | 32 |
| TABELA 4.1 - Alguns problemas da camada interface de rede | 64 |
| TABELA 4.2 - Alguns problemas típicos de redes Ethernet..... | 65 |
| TABELA 4.3 - Alguns problemas típicos de linhas seriais..... | 67 |
| TABELA 4.4 - Alguns problemas típicos da camada internet..... | 68 |
| TABELA 4.5 - Alguns exemplos de problemas na camada de transporte | 71 |
| TABELA 4.6 - Alguns exemplos de problemas na camada de aplicação..... | 72 |
| TABELA 5.1 - Tipos de similaridades das características do sistema | 99 |
| TABELA 5.2 - Relevância de algumas características..... | 101 |
| TABELA 6.1 - Tabelas do banco de dados do sistema DUMBO | 114 |
| TABELA 6.2 - Distribuição dos casos do protótipo por tipo de problema | 120 |
| TABELA A2.1 - Similaridade das características adicionais implementadas | 136 |
| TABELA A2.2 - Algumas características adicionais propostas | 138 |
| TABELA A2.3 - Grupo estados das interfaces | 139 |
| TABELA A2.4 - Grupo produtos..... | 138 |
| TABELA A2.5 - Algumas características específicas do protótipo..... | 139 |
| TABELA A3.1 - Lista dos casos presentes no protótipo..... | 140 |

Resumo

Com o crescimento do número e da heterogeneidade dos equipamentos presentes nas atuais redes de computadores, o gerenciamento eficaz destes recursos torna-se crítico. Esta atividade exige dos gerentes de redes a disponibilidade de uma grande quantidade de informações sobre os seus equipamentos, as tecnologias envolvidas e os problemas associados a elas. Sistemas de registro de problemas (*trouble ticket systems*) têm sido utilizados para armazenar os incidentes ocorridos, servindo como uma memória histórica da rede e acumulando o conhecimento derivado do processo de diagnose e resolução de problemas. Todavia, o crescente número de registros armazenados torna a busca manual nestes sistemas por situações similares ocorridas anteriormente muito morosa e imprecisa. Assim, uma solução apropriada para consolidar a memória histórica das redes é o desenvolvimento de um sistema especialista que utilize o conhecimento armazenado nos sistemas de registro de problemas para propor soluções para um problema corrente.

Uma abordagem da Inteligência Artificial que tem atraído enorme atenção nos últimos anos e que pode ser utilizada para tal fim é o raciocínio baseado em casos (*case-based reasoning*). Este paradigma de raciocínio visa propor soluções para novos problemas através da recuperação de um caso similar ocorrido no passado, cuja solução pode ser reutilizada na nova situação. Além disso, os benefícios deste paradigma incluem a capacidade de aprendizado com a experiência, permitindo que novos problemas sejam incorporados e se tornem disponíveis para uso em situações futuras, aumentando com isso o conhecimento presente no sistema.

Este trabalho apresenta um sistema que utiliza o paradigma de raciocínio baseado em casos aplicado a um sistema de registro de problemas para propor soluções para um novo problema. Esse sistema foi desenvolvido com o propósito de auxiliar no diagnóstico e resolução dos problemas em redes. Os problemas típicos deste domínio, a abordagem adotada e os resultados obtidos com o protótipo construído são descritos.

Palavras-Chave: gerenciamento de falhas; sistemas de registro de problemas; raciocínio baseado em casos; gerenciamento de redes.

TITLE: “CASE-BASED REASONING APPLIED TO FAULT MANAGEMENT IN COMPUTER NETWORKS”

Abstract

With the increasing number of computer equipments and their increasing heterogeneity, the efficient management of those resources has become a hard job. This activity demands from the network manager a big amount of expertise on network equipments, technologies involved, and eventual problems that may arise. So far, trouble ticket systems (TTS) have been used to store network problems, working like a network historical memory and accumulating the knowledge derived from the diagnosis and troubleshooting of such problems. However, the increasing number of stored tickets makes the manual search of similar situations very slow and inaccurate in these kind of systems. So, an adequate approach to consolidate the network historic memory is the development of an expert system that uses the knowledge stored in the trouble ticket systems to propose a solution for a current problem.

Case-based reasoning (CBR), an approach borrowed from Artificial Intelligence that recently had attracted many researchers attention, may be applied to help diagnosing and troubleshooting networking management problems. This reasoning paradigm proposes solution to new problems by retrieving a similar case occurred in the past, whose solution can be reused in the new situation. Furthermore, the benefits of this paradigm include the experience learning capability, allowing new problems being added and becoming available to use in future situations, expanding the knowledge of the system.

This work presents a system that uses case-based reasoning applied to a trouble ticket system to propose solutions for a new problem in the network. This system was developed with the aim of helping the diagnostic and troubleshooting of network problems. It describes the typical problems of this domain, the adopted approach and the results obtained with the prototype built.

Keywords: fault management; trouble ticket systems; case-based reasoning; network management.

1 Introdução

As redes de computadores manipulam de modo rápido e eficiente grandes quantidades de informações, sendo utilizadas naturalmente no dia a dia das pessoas. Em virtude de sua importância, a interrupção de seus serviços pode gerar diversas complicações para seus usuários, causando perda de rendimentos, atraso no recebimento de dados críticos, etc.

Entretanto, com o crescimento do número e heterogeneidade dos equipamentos e tecnologias envolvidas nas redes, o número de problemas possíveis nessas e a complexidade envolvida no seu diagnóstico tornam-se críticos. Assim, as redes são controladas usualmente por técnicos especialistas, que são encarregados de manter a disponibilidade e a qualidade dos seus serviços, efetuando o gerenciamento das mesmas.

A fim de auxiliar no gerenciamento das falhas ocorridas na rede, sistemas de registro de problemas têm sido utilizados. Tais sistemas auxiliam os gerentes na monitoração dos problemas correntes, mantendo um registro do ciclo de vida do problema e armazenando, com isso, a memória histórica das falhas de uma rede.

Em virtude do conhecimento acumulado nos sistemas de registro derivado dos processos de diagnóstico e resolução de incidentes anteriores, esses sistemas podem ser utilizados para auxiliar na investigação de um novo problema. Entretanto, se o número de registros crescer muito, a busca de soluções em situações anteriores para tentar as mesmas ações corretivas acaba por se tornar muito morosa e imprecisa quando feita com base em mera inspeção manual dos registros. Assim, uma solução apropriada para consolidar a memória histórica da rede é a criação de um sistema especialista que utilize o conhecimento acumulado nos sistemas de registro de problemas para auxiliar os gerentes no diagnóstico de uma nova situação similar, propondo soluções a partir dos registros armazenados.

Nesse contexto, uma abordagem da Inteligência Artificial que tem despertado grande atenção nos últimos anos e que pode ser utilizada nesses sistemas é o raciocínio baseado em casos. Sistemas com esse paradigma de raciocínio propõem soluções para uma ocorrência corrente pela recuperação de situações similares ocorridas no passado, denominadas casos, que podem contribuir para a resolução do problema corrente. Esses sistemas têm também como característica o aprendizado com a experiência, possuindo a capacidade de armazenar novas situações solucionadas, que se tornam disponíveis para futuras consultas e aumentam naturalmente o conhecimento presente no sistema.

O domínio de gerenciamento de redes envolve uma grande complexidade e variedade de problemas. Além disso, incorpora novas tecnologias muito rapidamente, gerando uma evolução e mudança nas redes muito acentuada. Por estas especificidades, o uso de raciocínio baseado em casos, em sistemas especialistas do domínio de gerenciamento de redes, traz inúmeras vantagens, entre as quais se destacam: a diminuição da fragilidade de outros paradigmas de raciocínio, pela sua capacidade de incorporar novos casos e, assim, suportar mais facilmente as mudanças do domínio; a possibilidade de um processo de aquisição de conhecimento mais natural.

Este trabalho apresenta um sistema que incorpora o paradigma de raciocínio baseado em casos em um sistema de registro de problemas tradicional, visando utilizar o conhecimento armazenado nestes registros para propor soluções para um novo problema

corrente. Esse sistema, que objetiva manter a memória histórica da rede, é denominado **sistema DUMBO**.

O capítulo 2, que dá seqüência a esta introdução, apresenta o paradigma de raciocínio baseado em casos, fornecendo uma visão geral do mesmo e como ele pode ser utilizado para desenvolvimento de um sistema. Inicialmente, serão abordadas as questões referentes à base de casos, comentando a representação dos casos e a sua indexação. Em seguida, serão explicitados os processos envolvidos no ciclo de raciocínio do sistema, que compreendem a recuperação das situações similares da base de casos, a reutilização da solução presente nesses casos, a revisão da solução quando necessário e, por fim, o armazenamento da experiência obtida no processo de modo que o caso corrente possa ser útil para a resolução de problemas futuros. Como último aspecto, são descritos dois sistemas que utilizam este paradigma e que contribuíram para o desenvolvimento de particularidades dos processos ou da base de conhecimento do sistema DUMBO.

O capítulo 3 discorre sobre a gerência de redes, enfocando o gerenciamento de falhas e o desenvolvimento de sistemas especialistas para o domínio. Conceitos e principais tópicos da área são comentados, e os sistemas de registro de problemas são apresentados. O uso de sistemas especialistas e suas abordagens são então discutidos, e, por fim, alguns sistemas do domínio de gerência de redes que fazem uso de raciocínio baseado em casos são citados.

Na seqüência, o capítulo 4 introduz o sistema DUMBO, apresentando a motivação para desenvolvimento de tal sistema. O estudo efetuado sobre os problemas do domínio é então comentado, sendo apresentados alguns dos problemas típicos das redes pesquisadas e algumas considerações resultantes deste estudo que foram utilizadas para a modelagem do sistema.

A modelagem do sistema é apresentada no capítulo 5. Após uma visão geral da abordagem utilizada, a representação dos problema usando o formalismo em redes semânticas é apresentada, de modo a fornecer uma visão da hierarquia e dos atributos dos tipos de problemas identificados na etapa da aquisição do conhecimento e utilizados no sistema. Em seguida, a estrutura do sistema e os aspectos da base de conhecimento são abordados, incluindo o modo como os casos foram representados no sistema, como a base de casos está organizada e que informações gerais sobre o domínio são mantidas pelo sistema. Por fim, é comentada a forma como cada um dos processos do ciclo de raciocínio apresentado no capítulo 2 foi desenvolvido no sistema.

O capítulo 6 apresenta, então, o protótipo implementado. Aspectos da sua implementação são abordados e a especificação do protótipo utilizando a linguagem de especificação SDL (*Specification and Description Language*) é apresentada. Por fim, a avaliação do protótipo é comentada.

Encerrando o texto, uma análise crítica do trabalho é apresentada no capítulo final, enfocando os principais pontos positivos e negativos do sistema modelado, e uma lista de metas é traçada como trabalho futuro.

2 Raciocínio Baseado em Casos

Sistemas especialistas são aplicações capazes de representar e raciocinar sobre algum domínio de conhecimento, a fim de auxiliar na resolução de problemas ou fornecer conselhos [JAC 86]. Esses sistemas capturam as estratégias de resolução de problemas dos especialistas do domínio e aplicam-nas em circunstâncias específicas, de modo seletivo. O conhecimento dos especialistas, entretanto, não pode ser obtido simplesmente a partir do conhecimento básico do domínio isoladamente — ele precisa representar a experiência obtida por estes especialistas, experiência que os capacita a concentrarem-se nas causas mais prováveis de um problema e adaptarem respostas para problemas específicos [GOY 91].

Os sistemas especialistas utilizam diversos paradigmas, sendo o baseado em regras o mais utilizado [HAR 89]. Outras importantes abordagens dentro da Inteligência Artificial (IA), que podem fornecer meios poderosos para tais sistemas, estão também sendo pesquisadas atualmente, tais como a Inteligência Artificial Distribuída e o Aprendizado Automático [GOY 91]. Mas além destas, nos últimos anos tem atraído atenção uma abordagem da IA que pode ser utilizada para o raciocínio na resolução de problemas: a abordagem *Case-based Reasoning* (CBR — Raciocínio Baseado em Casos).

Sistemas com esse paradigma de resolução de problemas propõem soluções para novos problemas pela recuperação de um caso similar ocorrido no passado, que é reutilizado na nova situação. Além disso, esses sistemas possuem a capacidade de aprendizado, uma vez que um novo problema pode ser armazenado após ter sido solucionado, tornado-se disponível para futuros problemas e aumentando com isso o conhecimento presente no sistema.

Conforme demonstram os resultados de pesquisas da psicologia cognitiva, o raciocínio através da reutilização de experiências passadas é um poderoso e freqüente meio para resolução de problemas por seres humanos [AAM 94]. Quando, por exemplo, um gerente de rede se depara com um problema a ser solucionado, ele utiliza primeiramente sua memória de incidentes passados (podendo fazer uso de arquivos de registros dos problemas anteriores da rede, por exemplo) para propor e justificar a sua decisão [CAU 95].

Assim, um caso representa na terminologia CBR uma situação ocorrida anteriormente — denominada caso anterior ou do passado — que é capturada e aprendida de modo a ser utilizada na solução de problemas futuros — denominados casos correntes ou novos [AAM 94]. Os casos, que podem ser representados em diferentes formatos e tamanhos, reproduzem o conhecimento específico para um determinando contexto, registrando o conhecimento a nível operacional. Um caso registra uma experiência que tem o potencial de ser útil para atingir um objetivo ou conjunto de objetivos mais facilmente no futuro ou de alertar para a possibilidade de falhas imprevistas em um problema [KOL 93].

Apresentaremos, neste capítulo, uma breve revisão da abordagem de raciocínio baseado em casos. Leituras adicionais sobre a abordagem podem ser obtidas em [AAM 94][KOL 93][ABE 96][WAT 94][WAT 97][MAR 94][MEL 99], além de

referências de aplicações desenvolvidas [ABE 95][ACO 92][SMI 92][BRA 91][DRE 95][LEW 93][CAU 95][STA 93][MAT 95][LEA 96], entre outras.

2.1 Histórico

Em 1977, Schank e Abelson *Apud* [WAT 97] propuseram que o conhecimento geral das pessoas sobre as situações está armazenado em *scripts*, permitindo que elas criem expectativas sobre o que ouvem, e assim, construam inferências sobre as relações entre as coisas das quais ouviram. Os *scripts* foram propostos como uma estrutura de memória conceitual, descrevendo informação sobre eventos estereotipados, como ir a um restaurante ou a uma consulta a um médico. Experimentos com *scripts* mostraram, entretanto, que eles não representam uma teoria completa de representação de memória, já que as pessoas confundem eventos que tem *scripts* similares [WAT 94]. Os *scripts* fornecem apenas um tipo de conhecimento que as pessoas utilizam para o entendimento: elas se valem também outros tipos de conhecimento, como o conhecimento sobre objetivos, planos, relações interpessoais e papéis efetuados pelas pessoas [KOL 93]. Representações sobre estes tipos de conhecimento têm sido propostas e sistemas de computadores que usam estes tipos de conhecimento para o entendimento têm sido desenvolvidos.

Em 1982, Roger Schank *Apud* [AAM 94] propôs a teoria da *Dinamic Memory* (memória dinâmica) e o papel central que a lembrança de situações anteriores (episódios, casos) e modelos de situações (*scripts*, MOPs — pacotes de organização de memória) desempenham na resolução de problemas. Este trabalho, segundo Aamodt e Plaza, apresenta as raízes do uso de raciocínio baseado em casos na Inteligência Artificial. Outras características do campo CBR têm sido originadas do estudo de raciocínio analógico proposto por Gentner *Apud* [AAM 94] e das teorias de formação de conceitos, resolução de problemas e aprendizado experimental dentro da filosofia e psicologia [AAM 94].

O primeiro sistema CBR, denominado CYRUS, foi desenvolvido por Janet Kolodner [AAM 94]. Este sistema, baseado no modelo de memória dinâmica de Schank e na teoria MOP de resolução de problemas e aprendizado, foi basicamente um sistema de questões e respostas com conhecimento de várias viagens e encontros do primeiro secretário do estado americano, Cyrus Vance. O modelo de memória de casos desenvolvido por este sistema serviu de base para outros diversos sistemas CBR [WAT 94], como MEDIATOR, PERSUADER, CHEF, JULIA e CASEY.

Outra base de CBR e conjunto de modelos foi desenvolvida por Bruce Porter e seu grupo na University of Texas, em Austin. Inicialmente trabalhando com o problema de aprendizado automático para classificação de tarefas, o grupo desenvolveu o sistema PROTOS [AAM 94]. Esse sistema enfatiza a integração do conhecimento geral do domínio e do conhecimento específico de casos em uma estrutura de representação unificada — um modelo de memória de casos. Além dessa, outra contribuição importante para a área foi o trabalho do grupo de Edwina Rissland na University of Massachusetts, em Amherst, que desenvolveram o sistema HYPO, aplicado para o domínio do Direito.

Atualmente, as pesquisas em CBR têm estendido-se rapidamente, sendo percebidos um crescente número de artigos envolvendo CBR em periódicos de IA e um

aumento de aplicações comerciais de sucesso [WAT 94], além de pesquisas em desenvolvimento em diversos países.

2.2 O Ciclo CBR

Além da base de conhecimento composta por casos que representam situações anteriores, os sistemas de raciocínio baseado em casos requerem mecanismos para recuperá-los da base de conhecimento, adaptá-los para o caso corrente, validar a solução proposta e armazenar o conhecimento obtido durante o processo. Conforme apresentado por Aamodt e Plaza [AAM 94] e Watson e Mahir [WAT 94], os mecanismos envolvidos no raciocínio baseado em casos podem ser representados de modo geral por um ciclo formado por quatro processos:

1. **Recuperação** do caso ou casos mais similares;
2. **Reutilização** da informação e conhecimento presente no caso para solucionar o problema corrente;
3. **Revisão** da solução proposta se necessário;
4. **Armazenamento** de parte da experiência obtida nos processos de modo a ser útil na solução de problemas futuros.

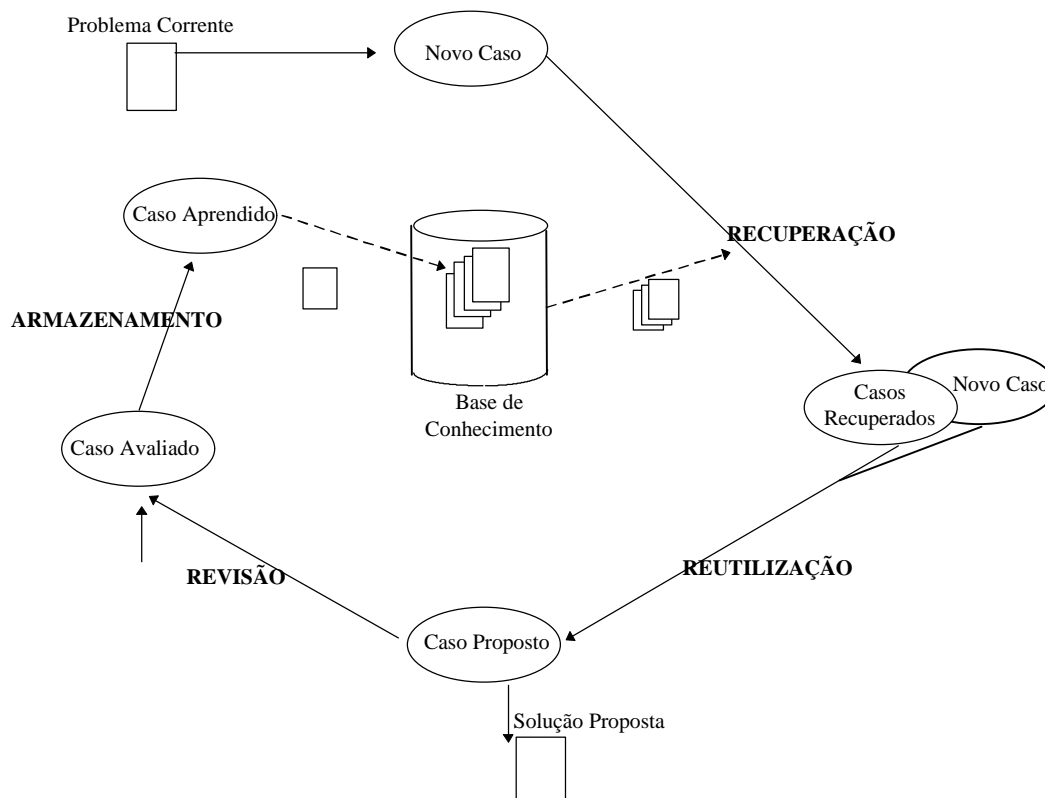


FIGURA 2.1 - Ciclo CBR

A figura 2.1, adaptada de [AAM 94] e [WAT 94], representa o ciclo de processos CBR. A descrição de um problema corrente gera um novo caso, que é usado para **recuperar** da base de casos um ou mais casos similares ao problema corrente. A solução apresentada pelos casos recuperados é então combinada com o novo caso através da **reutilização**, gerando uma solução proposta para o problema inicial. Com o

processo de **revisão**, esta solução é testada (sendo aplicada ao ambiente no mundo real ou avaliada por um especialista) e revisada se necessário, produzindo um novo caso, que será **armazenado** como um novo caso aprendido ou como uma modificação de alguns casos existentes. O uso de conhecimento geral do domínio geralmente faz parte do ciclo sendo suportado pelos processos CBR [AAM 94]. Este suporte varia nos diversos métodos utilizados pelos sistemas CBR existentes, sendo algumas vezes muito fraco ou mesmo nulo e em outras fornecendo um suporte muito forte.

O ciclo apresentado raramente ocorre sem intervenção humana [WAT 94]. Muitas aplicações CBR agem primordialmente como sistemas recuperadores de casos, sendo a adaptação freqüentemente realizada por gerentes da base de casos.

Além dos processos do ciclo apresentado, faz parte das pesquisas em raciocínio baseado em casos o estudo da representação do conhecimento na base de casos, envolvendo a representação dos casos e a escolha dos índices para a base [KOL 93]. Nas seções a seguir, enfocaremos a base de casos e cada um dos processos envolvidos no ciclo CBR.

2.3 A Memória de Casos

Um sistema de raciocínio baseado em casos é altamente dependente da estrutura de sua coleção de casos. Como a resolução de problemas é efetuada pela recuperação de uma experiência anterior aplicável para a solução do novo problema, os processos de busca e casamento precisam ser eficientes e ter um tempo de execução razoável. Além disso, como a experiência obtida durante a resolução de um problema é armazenada para uso futuro, são necessários mecanismos para integrar o novo conhecimento na base de casos. Assim, é necessário obter um balanço entre os métodos de armazenamento que preservem a riqueza semântica dos casos e índices e os métodos que efetuem e simplifiquem o acesso e recuperação de casos relevantes. Estes métodos são usualmente referenciados como modelos de memória de casos [WAT 94].

A memória de casos é formada pela base de casos, que serve como repositório dos casos, e pelos procedimentos de acesso à base [KOL 93]. Juntos, a base de casos e os procedimentos disponibilizam os casos para a consulta e incorporam novos casos, mantendo a acessibilidade dos itens já presentes na base de casos.

Os dois modelos de memória de casos considerados mais influentes nos estudos de CBR são [AAM 94][WAT 94][ABE 96] o **modelo de memória dinâmica**, proposto por Roger Schank e o **modelo de categoria e exemplar**, proposto por Ray Barreis e Bruce Porter. Informações sobre estes modelos podem ser obtidas em [AAM 94][KOL 93][ABE 96][MEL 99].

2.4 Base de Casos

O desenvolvimento de uma base de casos envolve o estudo sobre como os casos serão representados — abrangendo questões como o que armazenar em um caso e qual estrutura utilizar para descrever os seus conteúdos — e como serão indexados — de modo a serem recuperados no momento apropriado [AAM 94].

Na seção seguinte, abordaremos a representação e indexação de casos. Nas seções subseqüentes, apresentaremos os procedimentos de acesso a base de casos, anteriormente identificados no ciclo CBR.

2.4.1 Representação dos Casos

Conforme descrevemos no início deste capítulo, os casos representam um conhecimento específico relacionado a um contexto, armazenando conhecimento em nível operacional. Devem ser armazenados, entretanto, apenas os casos que possuem diferenças dos demais, de modo que possam acrescentar algum conhecimento útil para o sistema. Os casos podem apresentar diferentes tamanhos e modelos, assim como podem representar um pequeno ou grande intervalo de tempo. Janet Kolodner define a partir dessas observações que “*um caso é um pedaço de conhecimento contextualizado representando uma experiência que ensina uma lição fundamental para atingir os objetivos do raciocinador*” [KOL 93],p.13.

Os casos são formados por duas partes maiores: as lições que ele ensina — contidas no conteúdo dos casos — e o contexto em que pode ensinar tais lições — representado pelos índices. Tipicamente, o conteúdo de um caso é formado por três partes principais [WAT 94][KOL 93][ABE 96]:

- **descrição do problema ou situação**, apresentando as condições do ambiente no momento que o caso ocorreu e as restrições associadas;
- **solução**, que expressa a solução derivada para o problema, podendo ser formada por uma ação, um conjunto de procedimentos, um diagnóstico, uma classificação, um projeto, etc., variando conforme o tipo de problema que o sistema se propõe a resolver;
- **resultado**, que descreve o estado no ambiente real após o caso ter ocorrido e a solução ter sido, assim, aplicada.

A **descrição do problema ou situação** representa o problema que precisa ser solucionado ou a situação que precisar ser interpretada, classificada ou entendida. Como, de modo geral, um sistema CBR utiliza as similaridades entre a descrição do caso corrente e do caso armazenado para decidir se um caso armazenado deve ser recuperado, a **descrição** deve ser suficientemente detalhada para que o sistema seja capaz de julgar a aplicabilidade do caso na nova situação.

A **descrição do problema** é formada por três componentes maiores [KOL 93]: pelos **objetivos** a serem atingidos na resolução de um problema, pelas **restrições** relacionadas a estes objetivos e pelas **características** do problema e relações entre as partes.

Os **objetivos** na descrição do problema se referem as intenções ou propósitos a serem atingidos na situação. Estas intenções podem ser concretas ou abstratas, dependendo de que lições o caso almeja fornecer. Se o caso deseja diagnosticar um conjunto de sintomas, então o objetivo será “diagnóstico”. Já se o caso pretende criar uma solução para uma questão, seu objetivo será “criação”. As **restrições**, por sua vez, envolvem as condições relacionadas aos objetivos almejados que *precisam* ser consideradas no processo, como, por exemplo, não incluir um certo ingrediente na criação de uma receita culinária ou não causar determinada reação em um paciente que possui alergia a um medicamento.

Por fim, as **características** da situação envolvem todas as outras informações descritivas sobre a situação que são relevantes para a obtenção dos seus objetivos. Em uma aplicação de diagnóstico médico, por exemplo, sintomas e resultados de exames do paciente são descritores *necessários* para construir a solução, enquanto outras características *podem* também ser consideradas para a busca do objetivo. Do mesmo modo, alguns sintomas ou particularidades observados em uma rede de computadores *devem necessariamente* ser usados para solucionar um problema, enquanto outros *podem* ser usados para a solução desejada.

Dependendo da aplicação sendo desenvolvida, diferente atenção é exigida para cada um dos componentes da descrição do problema — algumas enfatizam os três componentes da descrição, outras enfatizam somente um ou dois. As situações que exigem uma solução explícita, como um projeto ou planejamento de tarefas, enfatizam, de modo geral, os objetivos e as restrições em suas representações do problema. Aquelas que requerem entendimento ou interpretação, por sua vez, tendem a ter um único objetivo simples, como diagnóstico ou entendimento, e enfatizam os descritores da situação problema. Assim, conforme sugere Janet Kolodner, há dois princípios gerais que devem ser seguidos na decisão de que informações devem pertencer à descrição do problema:

- incluir toda a informação descritiva que deve ser explicitamente levada em conta na busca do objetivo do caso;
- incluir todos os tipos de informação descritiva que são normalmente usados para descrever os casos desta espécie.

O primeiro princípio permite que o sistema ignore características que são relevantes, mas implícitas, por nunca variarem. O segundo, é necessário para a acessibilidade do caso. Ao menos algumas porções da descrição do caso são usadas em um sistema para a indexação e julgamento da similaridade, sendo melhores para a recuperação aqueles que são semelhantes nas dimensões que se sabe serem relevantes para a solução do problema. Porém, como existe muita informação que não é conhecida quando um novo caso é descrito, é necessário alternar o modo de determinar os melhores casamentos [KOL 93].

A parte componente **solução** de um caso representa os conceitos ou objetos que atingiram os objetivos descritos na descrição do problema, levando em consideração as restrições e as características especificadas. Existem diferentes tipos de soluções — em um problema de projeto a solução é o artefato elaborado, enquanto que em um problema de interpretação de uma situação ela é a interpretação apresentada para o caso. Além da solução de um caso recuperado ser usada para derivar uma solução para o caso corrente, ela pode também possuir componentes que acrescentem adaptação e revisão da solução proposta. Entre as partes componentes da solução que podem ser úteis para essas funções estão, além da solução propriamente dita, o conjunto de passos usados no raciocínio para resolver o problema, o conjunto de justificativas para as decisões que foram tomadas, as soluções aceitáveis que não foram escolhidas e suas justificativas, as soluções inaceitáveis que foram eliminadas e suas justificativas e as expectativas frente aos resultados após a solução ser efetuada.

Por fim, a parte componente **resultado** de um caso identifica o que ocorreu como consequência da solução efetuada e como ela foi realizada. Este componente inclui tanto o *feedback* do ambiente real como as interpretações para este *feedback*, de modo

que o sistema possa antecipar problemas em potencial e prever resultados de uma solução proposta. Além disso, o resultado pode também incluir a explicação do porquê uma solução ou expectativas frente a uma solução falharam e o que foi feito para reparar estas falhas.

Os casos podem ser representados em uma grande variedade de formas usando praticamente que todos os formalismos de representação da Inteligência Artificial [WAT 94], incluindo frames, objetos, predicados, redes semânticas e regras, além de estruturas menos ricas semanticamente, como os modelos de banco de dados comerciais. Muitos sistemas usam mais de um dos formalismos acima combinados [ABE 96].

2.4.2 Indexação

Os índices são utilizados para facilitar a recuperação dos casos armazenados na base, sendo os responsáveis por tornar um caso acessível no momento e condição apropriado — isto é, quando ele possuir um potencial para contribuir para a solução do problema corrente. São combinações dos descritores importantes de um caso, isto é, daqueles descritores que distinguem um caso dos outros.

O esquema de indexação envolve várias partes. Primeiramente, devem ser designados rótulos para os casos no momento em que estes são armazenados na base, de modo a garantir que eles possam ser recuperados no momento apropriado. Tais rótulos são usados no momento de recuperação para julgar se o caso armazenado deve ser selecionado. Uma segunda questão envolvendo a indexação é a definição da organização dos casos, de modo que a busca através da base de casos seja eficiente e precisa. Relacionada a estas questões está a definição dos algoritmos de recuperação a serem utilizados.

A atribuição de índices aos casos depende da compreensão do conteúdo e das informações que um caso pode fornecer. Ela deve permitir que sejam reconhecidas as similaridades entre a situação corrente e os casos armazenados que podem contribuir para atingir os objetivos do caso corrente. Assim, para a escolha de um bom índice, é necessário uma grande compreensão da situação problema, de forma que essas similaridades possam ser corretamente identificadas. Considere-se, por exemplo, a resolução de um problema ocorrido em uma rede de computadores. Informações como a aplicação utilizada e o usuário que identificou o problema não são importantes quando o problema envolve o acesso físico de um equipamento por problemas em sua placa de rede ou no cabeamento. Por outro lado, se o problema envolver a não autenticação do usuário por uma aplicação, a aplicação utilizada e o usuário são dados importantes. Como apontado em [ABE 96], a similaridade que se busca em sistemas de raciocínio baseado em casos é muitas vezes diferente daquelas semelhanças superficiais, obtidas pela comparação de dados descritivos. É um estilo de similaridade mais abstrata, que permite reconhecer em diferentes contextos soluções que possam ser aplicadas a novos casos.

Através da experiência com a construção de sistemas CBR, a comunidade CBR propõe alguns princípios que devem ser seguidos para a escolha de índices apropriados [KOL 93][ABE 96][WAT 94]:

- eles devem ser preditivos, isto é, devem prever a utilização da informação presente nos casos para diferentes situações problema;

- devem endereçar os propósitos em que o caso pode ser usado;
- devem ser abstratos o suficiente para permitir que um caso seja útil em uma variedade de diferentes situações;
- devem ser concretos o suficiente para que possam ser facilmente reconhecidos em situações futuras.

Os índices podem ser escolhidos através de métodos manuais, onde a escolha começa com a análise dos casos para a identificação da utilidade que poderia ter o caso, e sob que circunstâncias. Essas informações devem ser então *traduzidas* para representações que o sistema pode usar, definindo um conjunto de descritores, que são então trabalhados de modo a garantir que os índices sejam aplicáveis em âmbito geral e que possam ser reconhecidos no máximo de situações possíveis [KOL 93].

Além dos métodos manuais, existem, também, métodos de indexação automáticos, que são disponibilizados por muitas das ferramentas de CBR disponíveis no mercado. Entre os métodos automáticos referenciados na bibliografia, podemos citar o método baseado em uma *checklist*, o método baseado em diferenças e o método baseado em explicação [WAT 94][ABE 96] [KOL 93]. Entretanto, apesar do sucesso de muitos métodos automáticos [WAT 94], Janet Kolodner alerta que os índices tendem a ser melhor escolhidos pelo modo manual, e em aplicações práticas devem ser escolhidos desta forma.

2.5 Recuperação

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 da 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 no julgamento da mesma e esta valoração 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* [KOL 93].

A recuperação dos casos úteis à situação corrente envolve várias etapas, cada uma possuindo diferentes pontos que devem ser analisados. Inicialmente, precisam ser identificadas quais as características ou dimensões do caso corrente que devem ser utilizadas para julgar a similaridade dos casos armazenados. Isso é determinado levando-se em conta os propósitos para os quais os casos estão sendo recuperados e as dimensões que foram relevantes no passado para determinar o resultado do ambiente para as soluções aplicadas. Essas características serão então utilizadas pelos procedimentos de casamento e classificação para identificar quais dos casos armazenados têm o potencial de serem mais úteis.

Como os índices de um caso indicam quais das suas dimensões são mais importantes para julgar a similaridade, os algoritmos de casamento utilizam os índices para auxiliar no processo de identificação de que características devem ser consideradas para a similaridade entre os casos. Esses algoritmos devem, contudo, estar habilitados

para distinguir, dentre os vários índices existentes, quais focalizar para um dado momento quando os casos estiverem indexados em muitas formas [KOL 93].

Uma outra questão relevante no processo de recuperação são os algoritmos de busca propriamente ditos. Estes algoritmos são os processos que viabilizam encontrar na base aqueles casos que potencialmente podem ter um bom casamento com a situação corrente. A partir dos casos identificados pelos algoritmos de busca empregados, os procedimentos de casamento são aplicados.

Os algoritmos de busca são relacionados diretamente às estruturas utilizadas nas bases de casos para organizá-las. Uma estrutura em lista é, por exemplo, consultada pelo algoritmo de modo diferente de uma estrutura em árvore complexa. Assim, diferentes organizações dos casos na base ocasionam diferentes algoritmos para recuperar seus casos.

Um esquema dos processos envolvidos na recuperação de um caso pode ser visto na figura abaixo, adaptada de [KOL 93]. Conforme mostra o nível superior da figura, inicialmente é feita uma avaliação da situação e são identificadas as características que serão usadas para a busca na base de casos, sendo freqüentemente necessário que algumas dessas características sejam elaboradas pelo sistema. Os algoritmos de recuperação, através da descrição do problema e dos índices selecionados para a consulta na base, buscam os casos com potencial de serem similares e utilizam os mecanismos de casamento, seja para calcular o grau de casamento entre a situação corrente e os casos encontrados, seja para calcular o casamento de uma dimensão individual. Os algoritmos de recuperação retornam, então, uma lista dos casos parcialmente casados, cada qual com no mínimo algum potencial de ser útil para o sistema. Os casos são, por fim, analisados pelos procedimentos de classificação e aqueles que possuem mais potencial de serem úteis são retornados.

Recuperação

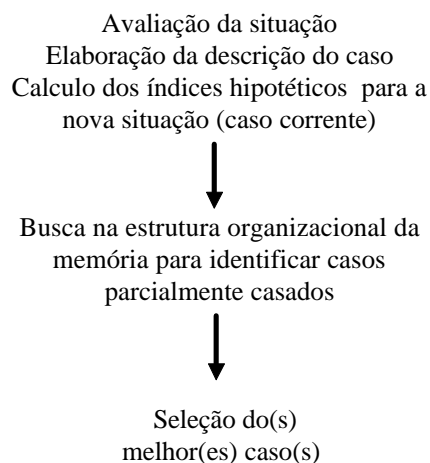


FIGURA 2.2 - Esquema do processo de recuperação na base de casos

Na seção a seguir, apresentaremos algumas possíveis estruturas de organização da base de casos e discutiremos os algoritmos de recuperação utilizados para a busca nessas organizações. Na seção subsequente, discutiremos alguns procedimentos para casamento e classificação e, por fim, apresentaremos, na seção 2.5.3, considerações

sobre o processo de avaliação da situação e sobre algumas relações entre os processos de indexação e recuperação.

2.5.1 Estruturas de organizações da base de casos e algoritmos de recuperação

Uma base de casos pode ser vista como um tipo especial de banco de dados. Ela deve ser, assim como os sistemas de banco de dados, percorrida em uma quantidade de tempo razoável pelos seus algoritmos de recuperação, mesmo quando manusear centenas ou milhares de casos.

Um dos grandes focos dos estudos em CBR tem sido a busca para tornar a recuperação eficiente, pois só serão possíveis abordar problemas em larga escala quando os algoritmos de recuperação forem suficientemente robustos e eficazes para manusear milhares de casos. Diferentes dos bancos de dados que recuperam registros a partir de campos idênticos, as bases de casos exigem a recuperação a partir de um casamento parcial, porém os algoritmos de recuperação parcial são dispendiosos. Assim, a recuperação de casos deve ser efetuada de modo que o casamento seja feito apenas nos casos com algum potencial de relevância para a nova situação.

Como o casamento parcial é importante para a recuperação dos casos, mas é dispendioso, devem ser encontrados meios de particionar o espaço de busca, de modo que o casamento parcial total seja realizado apenas em um pequeno número de casos. Assim, os algoritmos de recuperação devem fazer uma primeira seleção para distinguir quais os casos têm relevância para o caso corrente.

Os índices têm um importante papel nos algoritmos de busca e nas estruturas de organizações utilizadas nas bases de casos, sendo usados em muitos algoritmos para organizar os casos e direcionar a busca ao longo das estruturas organizacionais. Se, por exemplo, o casamento de uma característica é importante, os casos são indexados por essa característica em alguma estrutura organizacional e, no processo de busca, ela será considerada. Já se o casamento parcial da característica é importante, então o índice deve ser familiar com os valores que indicam similaridade para a característica, permitindo aos algoritmos de recuperação reconhecer dois itens como potenciais para casamento, mesmo que os valores de suas características não sejam exatamente os mesmos. De modo alternativo, os algoritmos de busca podem utilizar os procedimentos de casamento para determinar quão similares são dois valores diferentes em casos sendo comparados.

A seguir, apresentamos algumas diferentes organizações de casos e os algoritmos utilizados para recuperação e inserção de casos nestes estruturas. Informações adicionais sobre estas organizações, assim como sobre outras organizações aqui não comentadas, tais como o uso de memória linear com busca paralela e memória hierárquica com busca paralela, podem ser obtidos em [KOL 93][MEL 99].

2.5.1.1 Memória linear com busca serial

Numa memória horizontal, os casos são armazenados seqüencialmente em uma lista simples, um vetor ou um arquivo, e são recuperados pela aplicação de funções de casamento, em seqüência, para cada caso, calculando o grau de casamento de cada um e retornando os casos que melhor casaram. Não há organização no topo desses casos e o algoritmo de recuperação que é utilizado é bastante simples. São na verdade as heurísticas do procedimento de casamento as responsáveis pela recuperação.

O algoritmo de busca serial nesta estrutura é de fácil implementação e efetua uma busca completa na base de casos. Ele tem, porém, a desvantagem de se tornar muito lento quando o número de casos na base cresce.

2.5.1.2 Rede de características compartilhadas

Quando a base de casos trabalhada é grande, devem ser utilizados meios para particionar a base de casos, de modo que apenas um subconjunto de casos seja avaliado durante a recuperação, subconjunto este que deve conter os casos mais úteis para a situação corrente. Existem vários métodos de agrupamento indutivo que podem ser usados para a identificação desses subconjuntos, que procuram geralmente por similaridades sobre um grupo de instâncias e formam categorias baseadas nessas similaridades. Um modo de determinar os subconjuntos corretos é, por exemplo, agrupar os conjuntos de características que são compartilhadas em um grande número de itens. Outra forma é agrupar as características individuais que dividem o conjunto em agrupamentos de mesmo tamanho. Um terceiro modo seria ainda agrupar as características individuais que diferenciam um pequeno grupo de itens dos outros. Cada um destes métodos resulta em diferentes agrupamentos.

Nas redes de características compartilhadas, os casos que compartilhem muitas características são agrupados juntos. Cada nodo interno de uma estrutura deste tipo mantém as características partilhadas com casos abaixo dele, sendo os casos mantidos pelos nodos folha. Na recuperação de um caso, a situação corrente é relacionada e casada para cada um dos conteúdos dos nodos no mais alto nível da rede de modo a ser escolhido o nodo com melhor casamento. Se este nodo for um caso, ele é retornado. Do contrário, o processo é repetido para o próprio nodo interno retornado, até que seja retornado não um nodo interno, mas um caso.

A recuperação numa estrutura em rede de características compartilhadas é mais eficiente que em uma busca serial. Ela permite, entretanto, que casos com bom casamento sejam perdidos se a hierarquia dos nodos da rede não corresponder à importância das características do caso corrente. Além disso, a inserção de casos é uma operação complexa e, com o crescimento da base, é difícil manter a rede numa estrutura ótima, o que pode tornar a busca ineficiente.

2.5.1.3 Redes de discriminação

A estrutura organizacional rede de discriminação é similar às redes de características compartilhadas. Nessa estrutura, cada nodo interno é representado por uma questão ou pergunta que subdivide o conjunto de casos armazenados em hierarquia inferior a dele. Os nodos filhos representam as diferentes respostas para a questão formulada pelo nodo pai, e agrupam os casos que possuem a sua resposta. Assim como nas redes de características compartilhadas, as questões mais importantes devem ser colocadas numa hierarquia superior para evitar que casos com importantes características similares não sejam consultados.

As redes de discriminação são, assim como as redes de características compartilhadas, mais eficientes que organizações em lista para a busca dos casos, mas têm a desvantagem de requererem também um espaço adicional para representar a própria rede. Em ambos os esquemas, porém, a estrutura não fornece nenhum mecanismo para continuar a busca quando uma questão sendo solicitada não pode ser respondida. Opções para resolver o problema incluem finalizar a busca, continuar a busca

procurando em todos os nodos filhos e escolher o caminho mais provável, porém nenhuma das opções é amplamente satisfatória. Em relação às redes de características compartilhadas, elas são mais eficientes, pois a busca seguindo apenas a resposta à questão é mais eficiente que a comparação do conjunto das características do nodo.

2.5.1.4 Redes de discriminação redundante

Esta forma de organização dispõe os casos em várias redes de discriminação redundante, resolvendo o problema de como continuar uma busca quando uma questão que não pode ser respondida é encontrada. A busca é realizada em paralelo, seguindo as diversas redes existentes, e quando uma questão nessas condições é encontrada, a busca é suspensa naquela rede de discriminação, mas continua nas outras redes. Com a redundância existente, geralmente ao menos uma das redes irá encontrar um caso similar.

2.5.2 Casamento e Classificação

Conforme já foi visto, o processo de recuperação dos casos similares da base envolve a definição dos índices a serem utilizados, a busca dos casos com potencial de serem úteis para a situação corrente, o casamento desses casos com o caso corrente e a classificação dos casos recuperados, de modo a selecionar o melhor ou os melhores casos. Esses processos não são, porém, isolados uns dos outros: eles dependem e são algumas vezes altamente integrados entre si.

Muitos dos procedimentos de recuperação para as estruturas discutidas acima são exemplos disso, agregando funções de casamento na própria busca. Algoritmos para busca em memória linear, por exemplo, fazem o casamento de cada caso com o caso corrente, computam o valor de casamento e escolhem o caso de maior valor. Já os algoritmos de busca em redes de características compartilhadas agregam funções de casamento no momento de busca, utilizando relações que fazem o casamento parcial das características do nodo. Os procedimentos de casamento de casos, assim, podem ocorrer não apenas após a busca na base de casos, mas também serem utilizados para esta busca.

Para a definição dos procedimentos de casamento é necessário que algumas questões sejam consideradas. Uma dessas questões envolve a *identificação da correspondência entre os campos do caso corrente e os casos armazenados*. Isso pode ser feito por meio da associação de campos com nomes iguais, contudo nem sempre esta associação é tão simples: os casos podem possuir número de características diferentes, estar representados em diferentes pontos de vista, variando com os objetivos almejados pela situação descrita, etc. Nessas situações, o uso de heurísticas de senso comum pode contribuir para a identificação da correspondência. Em [KOL 93], cinco heurísticas que podem ser utilizadas nesse processo são apresentadas:

1. campos com mesmo nome devem ser mapeados diretamente, mesmo considerando que uma estrutura possua subdivisões e outra não;
2. preferir o mapeamento entre as características que possuem o mesmo papel funcional;
3. preferir o mapeamento das características que satisfazem as mesmas restrições;
4. preferir o mapeamento das características que possuem os mesmos valores ou valores similares;

5. múltiplas características do problema podem ser mapeadas para uma característica simples mais geral ou vice versa.

Quando um modelo causal para o domínio está disponível, isso pode também ser utilizado para a definição dessa correspondência. O sistema CASEY, desenvolvido por Koton *Apud* [KOL 93], aplicado ao domínio da medicina, é um exemplo de como um modelo causal pode ser usado para determinar a correspondência entre características. Neste modelo, são utilizadas regras de evidência para determinar como dois campos podem ou não serem considerados iguais. É utilizado um modelo que define que características e resultados correspondentes a outras características e outros resultados, que características levam a outras, que valores entre os possíveis para uma dimensão pertencem ao mesmo intervalo e são assim considerados similares.

Outra questão necessária para os procedimentos de casamento é a definição de *como será calculado o grau de similaridade entre as características correspondentes*. Entre os métodos disponíveis [KOL 93] estão:

- distância qualitativa e quantitativa. Este método utiliza várias formas, dependendo do tipo de valor da dimensão: (i) dividir em regiões, atribuindo casamento exato para aqueles valores que estão na mesma e os valores uma região, duas regiões, etc. para aqueles que não se encontrarem na mesma. Para resolver problemas gerados por valores muito próximos dos limites dos intervalos, podem ser criadas regiões que se sobrepõem, (ii) medir a diferença numérica e atribuir classificação qualitativa, (iii) fazer a enumeração de valores qualitativos quando não há números e medir como casamento exato, casamento com uma região distante, duas regiões distantes, etc. Pode também ser atribuído um valor numérico para cada valor qualitativo e compará-los, (iv) comparar os atributos de cada valor utilizando uma estrutura de comparação que aponta as similaridades e diferenças ou também utilizando representações com frames, (v) comparar numericamente os valores numéricos, atribuindo um casamento pior para valores mais distantes.
- julgar o grau de similaridade baseado na eficiência da função. Para isso pode ser utilizado um modelo causal. Se dois valores diferentes qualitativamente (como, por exemplo, uma cadeira e uma pedra utilizada para uma pessoa sentar-se) são mostrados correspondentes por um modelo causal, então elas casam. Neste método não é comparado o grau de casamento, apenas se eles casam ou não.
- hierarquia de abstração. Neste método, a similaridade entre dois valores é medida pela utilização da abstração com valor comum a estes valores mais específica, sendo que quanto mais específica é a abstração comum, maior a similaridade.

Por fim, é necessário *atribuir valores de importância para as dimensões da representação*, de modo a apontar as informações mais relevantes para a similaridade dos casos sendo comparados. Para isso, devem ser atribuídos pesos para cada campo, o que pode ser feito de forma global, para um grande número de casos, ou atribuído de modo mais local, apenas para um pequeno conjunto. Esta importância entre as dimensões pode ser identificada com a ajuda de um especialista ou através de uma avaliação estatística para determinar quais campos resultam em diferentes resultados ou

soluções. Pode também ser interessante atribuir diferentes valores de importância para cada objetivo da situação, o que pode ser feito de forma dinâmica [KOL 93].

Uma vez que alguns casos tenham sido recuperados e os procedimentos de casamento tenham sido efetuados sobre eles, estes devem passar por métodos de classificação a fim de sejam escolhidos os casos mais similares a situação corrente. Como já foi dito, muitas vezes esses procedimentos são feitos de modo integrado, em uma única etapa.

Existem vários métodos envolvendo os procedimentos de casamento e classificação que são utilizados no processo de recuperação dos casos. Entre estes apresentamos:

1. **algoritmo de vizinhança** (*nearest-neighbor matching*). Este algoritmo é um procedimento numérico em que é utilizada uma função de avaliação numérica para cada característica. Para cada característica do caso corrente é encontrada a característica correspondente no caso recuperado, os valores são comparados, o grau de casamento é calculado e é multiplicado pelo grau de importância da característica;
2. **exclusão** dos casos que não possuem características ou relações que os tornam diferentes ou não úteis em relação ao caso atual, como no caso de objetivos diferentes, por exemplo. Após este processo, é realizada uma avaliação numérica.

Existem também métodos que levam em conta o contexto da situação, que estão presentes em sistemas que tratam diferentes propósitos ou contextos e, assim, precisam utilizar meios de avaliar seus graus de casamentos parciais também diferentes. Isso pode ser feito:

1. utilizando-se **múltiplas associações de importância**, em uma classificação dinâmica. Neste método são utilizados vários conjuntos de critérios de importância, cada um associado com as condições relacionadas às circunstâncias sobre as quais eles deveriam ser usados. Isso pode ser feito através (i) do fracionamento da base de casos sendo relacionado um diferente conjunto de critérios de importância para cada partição, (ii) da associação de diferentes critérios de importância para cada diferente objetivo de raciocínio ou (iii) da associação de um ou mais conjuntos de critérios de importância para cada caso, cada um associado com uma diferente tarefa para qual o caso pode ser usado. Quando somente um objetivo de raciocínio será usado na base de casos, mas o que é importante varia com o tipo da nova situação, a biblioteca de casos pode ser seccionada de acordo com o tipo e para cada partição é indicado um conjunto de valores de importância para as dimensões dos casos (caso i). Quando múltiplos objetivos são suportados pela base de casos, entretanto, o particionamento por tipo não é suficiente e os casos em cada partição devem ser avaliados diferentemente dependendo do objetivo do raciocínio para o qual ele deve servir (caso ii). Para acomodar diferentes objetivos de raciocínio, diferentes funções de avaliação podem ser associadas para cada diferente objetivo de raciocínio. Por fim, quando critérios para julgar a similaridade variam tanto em relação ao objetivo sendo buscado como em relação ao tipo de caso, valores de importância devem ser assinalados, levando ambos em conta (caso iii). Podem ainda existir, porém, casos

altamente particulares. Nessa situação, devem ser associados um ou mais conjuntos de critérios de importância para cada caso. Cada conjunto de valores é designado para um objetivo e enfatiza as dimensões do caso que, se casarem bem, indicam que ele pode servir para o objetivo.

2. usando **preferências para implementar um esquema de classificação relativa**. É utilizado para fazer um corte após os casos terem sido selecionados, passando-os por filtros se existem muitos casos após o casamento. Esses filtros podem ser relacionados ao objetivo de raciocínio, ao grau de especificidade e facilidade de adaptação, e à frequência e ocorrência recente.

2.5.3 Considerações sobre a indexação e a recuperação

Para fazer a indexação de um caso, deve ser estabelecido um equilíbrio entre os parâmetros proféticos e os parâmetros observáveis. Existem alguns sistemas, como o modelo causal do CASEY, que derivam parâmetros proféticos a partir de outros observáveis. Mas esse processo, conhecido como elaboração, pode ter dois pesos: (i) alguns parâmetros são difíceis de serem elaborados, exigindo tempo e força computacional, e (ii) podem haver muitas possíveis elaborações a serem feitas, cada uma consumindo alguma força e tempo. Assim, quando a elaboração é cara, ou potencialmente cara, ela deve ser controlada de modo inteligente.

A decisão de quais elaborações devem ser realizadas é conhecida como o problema de controle da *avaliação da situação* [KOL 93]. Quando poucas elaborações são necessárias e estas são de baixo custo, todas podem ser efetuadas. Nos demais sistemas, porém, é necessário um controle sobre esse processo, que pode ocorrer em uma ou várias das três fases abaixo:

- estabelecimento do contexto: processo ocorrido antes de efetuar a busca, quando é elaborado o conjunto de dimensões que se sabe ser necessário para uma boa recuperação. As dimensões a serem elaboradas são geralmente definidas por uma catalogação;
- refinamento do contexto: processo ocorrido durante a busca, em que a elaboração é feita quando necessária, usando a base de casos para definir as dimensões que são indispensáveis elaborar para que o sistema possa discriminar entre vários itens na memória. O sistema CASCADE [SIM 92] e vários programas que usam rede de discriminação apresentam esse processo;
- redefinição do contexto: processo ocorrido após a busca, em que o conjunto de itens recuperados é examinado para verificar se eles são realmente tão úteis como deveriam ser.

2.6 Reutilização (adaptação)

Após os casos similares terem sido recuperados da base e o melhor caso ter sido selecionado, a solução deste caso deve ser reutilizada, podendo ser adaptada para as necessidades do problema corrente. Este processo envolve a identificação das diferenças entre o caso recuperado e o caso corrente e a análise das partes do caso recuperado que podem ser transferidas para o caso corrente, aplicando fórmulas e regras que consideram estas diferenças e sugerem uma solução [WAT 94].

Uma solução pode ser reutilizada de modo direto, sendo aplicada a solução recuperada sem ser adaptada. Essa técnica, denominada por alguns de **adaptação nula** [WAT 94] ou identificada simplesmente como **cópia** [AAM 94], é útil em problemas que envolvem um complexo raciocínio, mas uma solução simples, e é utilizada nos sistemas em que as diferenças não são consideradas relevantes, ao passo que as similaridades entre os casos o são. Entretanto, em sistemas que levam essas diferenças em conta, um processo de adaptação é necessário, podendo ser feito através da simples substituição de componentes da situação recuperada por componentes da situação corrente, por meio da transformação da antiga solução para uma solução que seja aplicada ao novo problema e por meio da derivação de uma nova solução, utilizando métodos para derivar a solução ou pedaços da solução do caso armazenado. Entre os métodos ou técnicas que têm sido usados para adaptação podem ser citados:

- **reinstanciação:** usada para instanciar características da antiga solução com características da nova. É utilizada, por exemplo, pelo sistema CHEF para propor a substituição de um prato por outro na elaboração de um cardápio de uma refeição [WAT 94][KOL 93].
- **adaptação parametrizada:** usada para o ajuste de parâmetros da antiga solução através do uso de heurísticas especializadas que identificam as relações entre as especificações de entrada e as diferentes soluções. Esta técnica é frequentemente usada quando as variáveis relevantes do caso são numéricas, sendo um exemplo de sua aplicação o cálculo da quantidade de ingredientes pela relação do número de pessoas. Uma das dificuldades em aplicar esta técnica é o conhecimento da função que relaciona as variáveis descritivas do problema e a solução [LEW 95].
- **busca local:** fornece um meio de buscar em uma estrutura de conhecimento auxiliar um substituto para um valor ou estrutura do caso recuperado que não é apropriado para a situação corrente. Um exemplo seria a busca em uma rede semântica por frutas que pudessem substituir a fruta laranja em um prato de sobremesa sendo elaborado [KOL 93].
- **consulta a memória:** procura em estruturas de conhecimento auxiliares ou na memória de casos por algo com uma dada descrição.
- **substituição baseada em casos:** utiliza, para a substituição de um parâmetro, casos da base.
- **transformação de senso comum:** utiliza heurísticas de senso comum para inserir, remover ou modificar componentes em uma solução.
- **reparo guiado ao modelo:** efetua transformações na solução do caso anterior guiadas por um modelo causal. É indicada especialmente em sistemas que envolvem raciocínio sobre equipamentos, seja envolvendo diagnóstico ou projeto.
- **adaptação ou reparo de propósitos especiais:** utiliza heurísticas de propósitos especiais para modificar estruturas ou adaptações específicas ao domínio não fornecidas pelos demais métodos acima. Essas heurísticas são indexadas pelas situações em que são aplicáveis. Podem ser utilizadas para, por exemplo, inserir um passo extra em um procedimento de solução, se determinado componente fizer parte do problema.

- **adaptação por derivação:** utiliza o método usado para encontrar a solução do caso recuperado para derivar a solução da situação corrente. Usando este raciocínio, um problema pode ser resolvido pela mesma seqüência de passos que foi previamente utilizada para solucionar o caso recuperado. Em manuais de resolução de problemas de gerência de redes, por exemplo, é possível encontrar procedimentos ou fluxos de tarefas que apontam como localizar a causa do problema e as possíveis soluções. Além disso, gerentes freqüentemente planejam seus próprios procedimentos que irão divulgar para os usuários de uma rede quando um problema ocorre. Estes procedimentos podem ser considerados como árvores de decisão ou *checklists* que guiam a investigação das causas e possíveis soluções para o problema, podendo ser freqüentemente implementados como *scripts* em Unix. Um exemplo de um procedimento geral para uma estação com mal funcionamento é: *Checar todos os cabos. Determinar se mudanças ocorreram no hardware. Determinar se mudanças foram feitas na configuração da rede. Contatar e conversar com o último usuário da estação.* [LEW 95] É importante observar que procedimentos não são soluções do problema, mas instruções que devem guiar para a causa do problema e soluções potenciais. Com essa forma de adaptação, o sistema oferece (ou mesmo executa) os procedimentos que funcionaram em casos passados para chegar à solução do problema corrente. Para utilizar este método, é necessário haver um repositório dos procedimentos e um modo de mapear tipos de problemas para procedimentos que mais regularmente levam a uma solução. Essa adaptação também é denominada **adaptação procedural**. [LEW 95]
- **adaptação baseada em crítica:** nesta técnica, o usuário observa a solução proposta no caso recuperado e manualmente adapta a solução para o caso corrente. É usada quando o sistema recupera soluções que o usuário sabe que não funcionará, sabendo porque não funcionará e como fazer funcionar. Nessas situações, o usuário deve adaptar o caso por si mesmo, já que o caso sugerido como uma possível solução pode solucionar a situação com uma pequena modificação. Deve ser, entretanto, documentado o modo que o caso foi adaptado para que ele possa ser usado no futuro [LEW 95].

As técnicas acima são alguns exemplos de métodos usados para efetuar adaptações na solução de um caso recuperado de modo que esta seja reutilizada para o caso corrente. Elas podem ser utilizadas combinadas ou de modo isolado. Um resumo das técnicas usadas para adaptação pode ser visto na tabela abaixo.

TABELA 2.1 - Técnicas de adaptação

| Técnica | Orientação | Tarefa | Opera com |
|---|---|--------------------------------|------------------------|
| Reinstanciação | estrutural | substituição | valores |
| Parametrizada | senso comum | substituição | valores |
| Busca local | especializada, senso comum | substituição | valores, estruturas |
| Consulta à memória | especializada | substituição | valores, estruturas |
| Substituição baseada em casos | casos | substituição | valores |
| Transformação de senso comum | senso comum | transformação | valores, estruturas |
| Reparo guiado ao modelo | modelo causal | transformação, substituição | valores, estruturas |
| Adaptação ou reparo de propósitos especiais | especializada, senso comum, modelo causal | transformação, substituição | valores, estruturas |
| Adaptação por derivação | casos | transformação, substituição | valores, estruturas |
| Adaptação baseada em crítica | senso comum | transformação, substituição | valores, estruturas |

2.7 Revisão

Após um caso ter sido recuperado da base e sua solução ter sido adaptada para o caso corrente, é necessário que (i) a **solução proposta seja avaliada**. Se esta solução não produzir resultado satisfatório, (ii) ela deve ser **reparada**, e uma nova solução deve ser gerada. Uma vez que a solução encontrada for correta, a experiência obtida deve ser aprendida, sendo armazenada para uso futuro. Esses processos (i, ii e armazenamento) podem ser vistos como obtenção de experiência, ao passo que o processo de recuperação e de adaptação é visto como uma aplicação da experiência armazenada.

A **avaliação da solução** (i) 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, em que a solução proposta (um prato culinário) é aplicada a um modelo interno, que é considerado eficiente para fornecer a avaliação necessária à solução fornecida [AAM 94]. 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 uma configuração em uma rede, até vários meses, quando aplicada, por exemplo, a um tratamento médico. Durante o período de execução, um caso pode ser aprendido, sendo

já mantido armazenado na base, porém a informação de que o caso não foi ainda avaliado deve ser marcada a este [AAM 94].

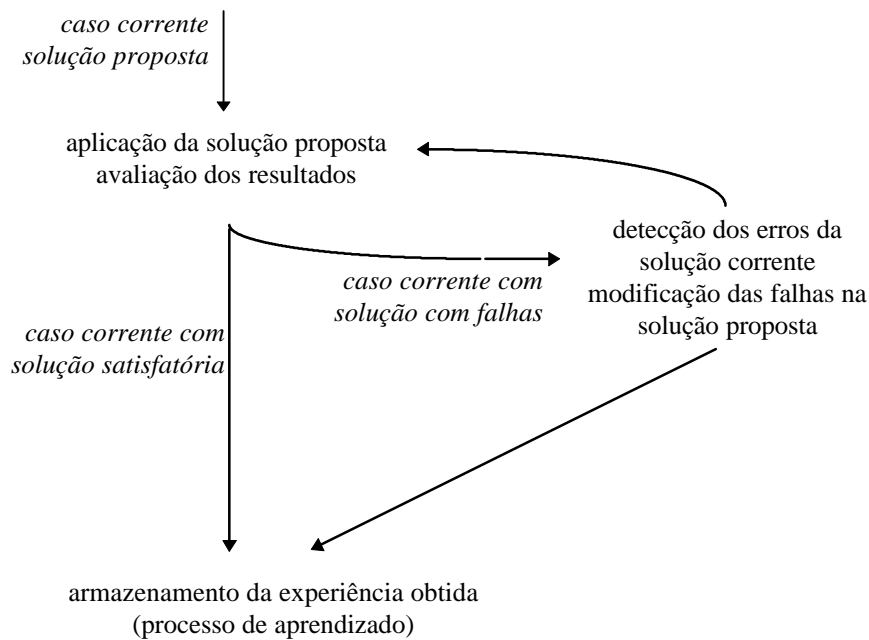


FIGURA 2.3 - Esquema do processo de revisão

Conforme apresentado por Lundy Lewis em [LEW 95], existem, de modo geral, três modelos de execução para uma solução: **execução manual**, **execução sem supervisão** e **execução supervisionada**. 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 CBR [LEW 95], em que o sistema somente sugere, com base na sua experiência, no processo de recuperação e no processo de adaptação, uma boa solução para o problema, que é executada pelo usuário.

Em alguns domínios, porém, quando uma solução é expressa em um programa de computador, um sistema CBR 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 intervenção 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 [LEW 95]. 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 visto na próxima seção. 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 exemplo de um sistema em que isso é realizado é o sistema CHEF, desenvolvido por Hammond *Apud* [AAM 94], que utiliza

um conhecimento causal para gerar explicações do porquê certos objetivos almejados no caso corrente não foram atendidos, e armazena as situações gerais que produziram falhas usando uma técnica de aprendizado baseada em explicação [AAM 94]. 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 (quando for garantido que está correta) ou pode ser avaliada e reparada novamente se necessário. O esquema do processo de revisão pode ser visto na figura 2.3.

2.8 Armazenamento (aprendizado)

O armazenamento dos casos é o processo de incorporação ao sistema da experiência obtida durante a resolução do problema. Como foi visto, esse aprendizado do sucesso ou falha da solução proposta é proporcionado pelo resultado da fase de avaliação e possível reparo da solução. O armazenamento envolve a seleção de que informações do caso devem ser retidas, em que formas estas informações devem ser retidas, como indexar o caso para posterior recuperação em problemas similares futuros e como integrar o novo caso na memória de casos [AAM 94].

Quando um caso foi resolvido pelo uso de um caso recuperado, um novo caso pode ser construído ou o caso recuperado pode ser generalizado para incluir o caso corrente. Já quando o problema foi resolvido por outros métodos, como por consultas ao usuário, um novo caso inteiro deve ser construído. Informações como descritores relevantes do problema e solução para a situação são dados geralmente utilizados para armazenamento do conhecimento, assim como explicações e outras formas de justificativa do porquê a solução utilizada foi proposta para o problema. No sistema CASEY, por exemplo, explicações são retidas nos casos e reutilizadas futuramente na modificação da solução. Informações obtidas da tarefa de reparo da solução podem também ser extraídas e retidas no sistema, seja como casos separados que falharam, seja dentro de casos problema totais.

Por fim, o sistema CBR deve integrar o conhecimento aprendido à base de casos. Isso pode ser feito com a modificação do esquema de indexação dos casos existentes, em que o sistema pode aprender o modo de ajustá-los para que a similaridade possa ser melhor identificada. Os pesos ou importância dos índices de um caso ou solução em particular podem ser ajustados pelo sucesso ou falha do uso do caso na solução do problema corrente. As características que foram julgadas relevantes na recuperação do caso com sucesso têm então sua importância de associações aumentada, ao passo que as associações das características que levaram a casos não similares serem recuperados são enfraquecidas. Com esse processo, a estrutura de índices tem a função de canalizar e adaptar a memória de casos para seu uso.

O aprendizado pode ocorrer também dentro de um modelo de conhecimento conceitual geral, por exemplo utilizando outros métodos de aprendizado automático ou através de interação com o usuário [AAM 94]. Assim, com uma interface apropriada ao usuário, o modelo de conhecimento geral do sistema pode ser incrementado e refinado naturalmente ao longo da resolução de problemas, do mesmo modo como é feito com a memória de casos.

2.9 Exemplos de Aplicações CBR

A seguir, apresentamos dois exemplos de aplicações que usam raciocínio baseado em casos. Estes exemplos foram escolhidos por terem contribuído para o desenvolvimento de algumas particularidades dos processos do sistema proposto, embora não sejam sistemas aplicados ao gerenciamento de redes.

2.9.1 Sistema CASCADE

O sistema CASCADE [SIM 92] foi desenvolvido para auxiliar engenheiros de suporte a resolver falhas ocorridas em *device drivers* que afetam o sistema operacional VMS da Digital. Esse sistema foi projetado utilizando um algoritmo de recuperação denominado *validated retrieval* (recuperação validada), que envolve duas fases distintas.

A primeira fase do algoritmo envolve a busca na base de todos os casos que parecem ser relevantes ao problema corrente, baseando-se nas características fornecidas inicialmente sobre o problema. Essas características, denominadas características superficiais, já são conhecidas pelos usuários — não gerando, assim, nenhum custo para serem obtidas — ou podem ser obtidas a baixo custo. Exemplos dessas características são o tipo de CPU e a versão do sistema operacional, que podem ser obtidas em um simples comando.

A segunda fase do processo é denominada *model-based validation* (validação baseada no modelo): nessa etapa, baseando-se no conteúdo dos casos recuperados, características adicionais são escolhidas. Essas características derivadas, que são mais discriminantes para a situação corrente, não são de fácil obtenção: elas são extraídas mediante um mecanismo complexo e dispendioso. As características derivadas ajudam a eliminar os casos recuperados que não são apropriados, a identificar os casos recuperados mais adequados e a recuperar casos relevantes adicionais que não foram identificados através do uso das características superficiais. Um caso é recuperado e fornecido ao usuário somente depois que todas as características superficiais e derivadas casaram com o problema corrente.

As características superficiais e derivadas são descritas através de pares atributo/valor. Os casos são indexados por um subconjunto das suas características superficiais e todas as suas características derivadas. As características derivadas são geradas e obtidas por meio de *probes* — objetos formados por ações que requerem dados sobre o problema e por um conhecimento específico ao domínio que gera as características derivadas a partir dos dados obtidos. Os casos armazenados no sistema são compostos de três partes:

- características superficiais com informações sobre a falha e o ambiente em que a falha ocorreu;
- características derivadas relevantes e ponteiros para as *probes* relacionadas;
- ação ou grupo de ações de reparo que soluciona o problema.

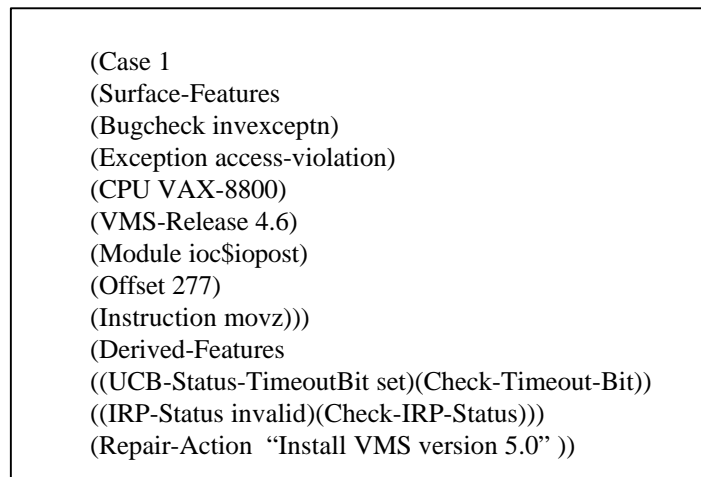


FIGURA 2.4 - Exemplo de um caso do sistema CASCADE

Como uma *probe* pode gerar diversas características derivadas de um particular atributo, elas podem ser utilizadas por muitos casos. Além disso, dependendo do domínio, as *probes* podem ser também inter-relacionadas: dados obtidos por uma *probe* pode ser condições para ações de outra, dados gerados por uma *probe* podem disparar outras *probes* relevantes, dados gerados por uma *probe* podem inibir outras *probes* que estavam sendo executadas.

O algoritmo de recuperação do sistema utiliza uma estrutura denominada *validation model* (modelo de validação), que contém as definições das *probes*, as inter-relações entre elas e outras informações tal como o seu custo de execução. O modelo baseado em validação considera o custo total para estabelecer as características derivadas do caso, e sempre leva em conta primeiro o caso com menor custo. A arquitetura do sistema pode ser visto na figura abaixo [SIM 92].

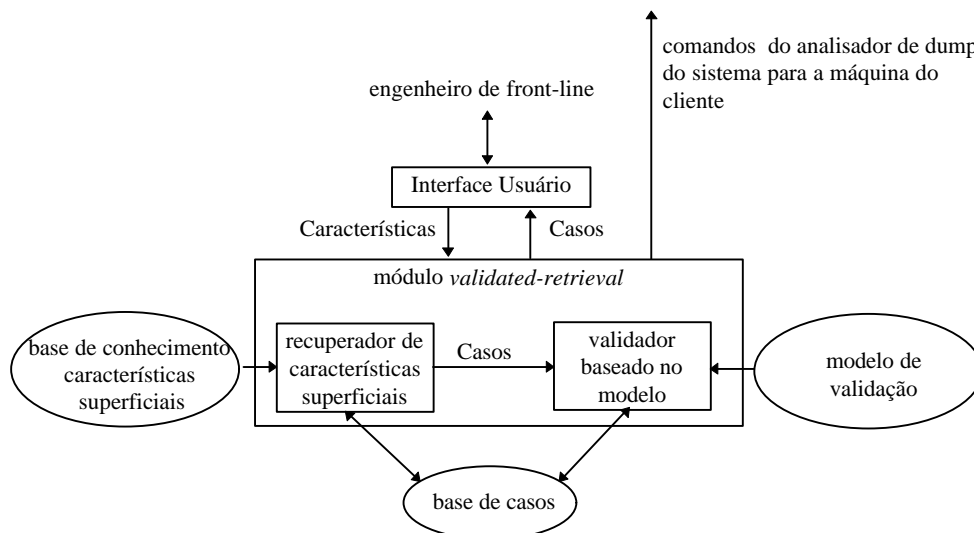


FIGURA 2.5 - Arquitetura do sistema CASCADE

O processo de raciocínio inicia quando o usuário descreve o problema, usando um conjunto das características superficiais. O sistema faz o casamento dessas características com os índices das características superficiais, utilizando o conhecimento armazenado em uma base de conhecimento para efetuar o casamento de valores não

idênticos. As *probes* associadas às características derivadas dos casos extraídos são acessadas, e é estabelecido o custo total para validar cada caso recuperado em relação ao custo de suas *probes*. Os casos são ordenados em relação ao seu custo e o caso com menor custo é considerado.

A *probe* correspondente à primeira característica derivada que ainda não tenha sido gerada é executada e a característica derivada é gerada. O sistema tenta fazer o casamento da característica correspondente do caso recuperado. Quando a característica não casa, ele rejeita o caso e utiliza os índices das características derivadas para também rejeitar os outros casos que possuem valores diferentes do obtido para o atributo da característica derivada. O sistema recupera então outros casos que incluem a característica obtida mas que não haviam sido inicialmente recuperados. Quando todas as características derivadas do caso são estabelecidas, o caso é selecionado e apresentado ao usuário.

A base de conhecimento das características superficiais utilizada pelo sistema contém dois tipos de conhecimento do domínio:

- conhecimento sobre similaridade entre valores das características superficiais. Indica, por exemplo, a similaridade entre duas CPU que pertencem a mesma família;
- conhecimento sobre inter-relações entre as diversas características de superfície, implementadas através de um conjunto de regras se-então. Identificam, por exemplo, que *os multiprocessadores VAX 8840 e 6230 são similares se a falha ocorreu no módulo SMP do sistema operacional e releases da versão 5.0 de VMS estavam sendo utilizadas.*

O sistema não utiliza conhecimento sobre similaridades entre características derivadas. As *probes* do sistema, implementadas através de funções, são executadas sobre um arquivo de *dump* da falha e geram as características derivadas sobre o estado das estruturas de dados, podendo ser executadas muitas *probes* sobre cada estrutura de dados.

O algoritmo *validated retrieval* é um dos aspectos desse sistema que influenciou o desenvolvimento da abordagem proposta no sistema DUMBO. No DUMBO, há também uma recuperação dividida em duas etapas distintas, uma responsável por uma recuperação inicial baseada em informações preliminares, outra responsável por uma avaliação da similaridade com o problema corrente mais específica, que faz uso de características extraídas dos casos recuperados na primeira etapa. Além disso, o uso de inter-relações entre as diversas características foi também aplicado no sistema DUMBO, sendo estas inter-relações utilizadas, contudo, para modificar a relevância de características, ao contrário de modificar o cálculo da similaridade.

2.9.2 Sistema Smart

O sistema Smart [ACO 92] é um sistema assistente para resolução de problemas integrado ao serviço telefônico de suporte ao cliente da empresa Compaq Computer Corporation, desenvolvido pela empresa Compaq em conjunção com a Inference Corporation. Este serviço oferece suporte técnico a consultas que vão desde requisição de informações sobre produtos da empresa até pedidos de resolução de problemas em

um complexo ambiente de rede com uma vasta quantidade de produtos envolvidos, não possuindo assim consultas típicas.

Inicialmente, quando o engenheiro de suporte recebe uma chamada, ele tenta resolvê-la baseada na descrição textual da requisição ou problema. Se, entretanto, ele não consegue solucionar o problema, ele aciona o sistema Smart, que recebe a descrição do problema fornecida pelo usuário. O sistema Smart realiza, então, uma busca inicial de casos que casem com a descrição do problema corrente. Essa busca é realizada utilizando para o casamento uma análise da descrição a nível de subpalavras que faz uso de um algoritmo de casamento denominado *trigram* [ACO 92], comparando a descrição do problema corrente com a descrição do problema dos casos armazenados.

Após a busca inicial, o sistema apresenta uma lista dos casos que apresentaram maior similaridade ao problema corrente e uma lista de questões associadas a estes casos, que visam obter informações adicionais sobre o problema, de modo a melhor definido. À medida que as respostas a essas questões vão sendo fornecidas, o sistema aciona uma nova busca, sendo que quanto mais informações forem fornecidas, mais o sistema aumenta sua acurácia do conjunto de casos relevantes ao problema corrente e das questões associadas.

Para cada caso recuperado pelo sistema, é informado um valor que representa seu grau de similaridade com relação ao caso corrente. Este valor é calculado com base no valor resultante do casamento das descrições, no peso das questões respondidas que casaram ou não com o caso recuperado e na quantidade de questões não respondidas. A figura 2.6 representa um esquema dos dados da consulta de um caso no sistema, extraída de uma imagem em [ACO 92].

| | |
|---|----------------------|
| Description: Experiencing spurious problems in a compaq file server resulting in a lockup situation, ethernet topology, high traffic contention. | |
| Questions about this Problem: | |
| How do you categorize the problem (pick the first that applies)?? | Networking proble... |
| What Network Operating system are you using?? | Banyan Vines |
| What major release of Banyan Vines are you using?? | 4.10 |
| Are you using Vines SMP?? | Yes |
| Wich family of Compaq maquine is it?? | Systempro |
| Are ther any interrupt conflicts??. | No |
| What is the Topology of the Network?? | Ethernet |
| Matching Cases: | |
| 73 [KG] Banyan & Systempro W/ multiple server panics & locks-up #2 | |
| 65 [KG] Banyan, Vines 4.10 Systempro server and PCs locking up. | |
| 50 [KG] Banyan, SMP Server sees 1 out of 3 NI3210 Ethernet Cards #2 | |
| 46 [KG] Banyan, SMP Server sees 1 out of 3 NI3210 Ethernet Cards #1 | |
| 42 [KG] Banyan, Server hanging when EtherLink II card cable is removed | |
| 40 [KG] Banyan, Vines PC program within PATH receives Command error | |

FIGURA 2.6 - Esquema de uma consulta de um caso no sistema Smart

Quando um registro de problema não é resolvido pelo sistema, ele é armazenado na base de casos com o formato de um caso e o seu estado de “caso não resolvido”. Quando esses casos são então resolvidos, eles são aprendidos pelo sistema e transformados em casos reais por engenheiros de suporte sênior treinados para a inserção de casos na base.

A base de casos está contida em um banco de dados relacional simples. Para garantir a consistência e alto uso entre os casos foram estabelecidas questões que são utilizadas em todos os casos em que se aplicam, tais como sistema operacional da rede, processador da máquina, família de produtos Compaq em uso, ambiente de operação. O sistema Smart tem se mostrado eficiente em 87% dos casos em que ele foi consultado [ACO 92].

Entre os aspectos desse sistema que influenciaram o desenvolvimento do sistema DUMBO, podemos citar o uso ações para aprimorar a recuperação, representadas por perguntas em texto livre, e o fornecimento ao usuário do grau de casamento de cada caso, a fim de permitir que este avalie o grau de segurança do sistema quanto à similaridade deste caso com o problema corrente.

2.10 Considerações Finais

Este capítulo procurou fornecer uma visão dos principais aspectos do paradigma de raciocínio baseado em casos, com enfoque nos tópicos relacionados aos propósitos desta dissertação. Este estudo foi realizado com o intuito de identificar as particularidades deste paradigma de modo a possibilitar sua correta utilização para a modelagem do sistema DUMBO.

O capítulo a seguir discorre sobre a gerência de redes. Neste capítulo são comentados seus principais conceitos, os sistemas de registro de problemas e o uso de sistemas especialistas para a área, com enfoque no uso de raciocínio baseado em casos.

3 Gerência de Redes

As redes de computadores, hoje em dia, manipulam de modo rápido e eficiente grandes quantidades de informação. Utilizadas normalmente no dia a dia das pessoas, podem gerar, com a interrupção de seu funcionamento, a suspensão de diversos serviços de empresas e da vida diária, provocando enormes complicações, tais como atraso no recebimento de dados críticos, perda de rendimentos, impossibilidade de obter fundos fora de horário bancário.

Em virtude da importância de seus serviços, as redes têm sido usualmente controladas por técnicos especialistas que possuem a responsabilidade de instalar, manter e resolver seus problemas. Esses problemas podem abranger desde uma simples resposta a uma dúvida de um usuário até a identificação e substituição de um equipamento com mal funcionamento ou a ativação de procedimentos de recuperação de falhas após a ocorrência de um evento catastrófico.

Com o crescimento do número e da heterogeneidade dos equipamentos envolvidos nas redes, o número de problemas potenciais e a complexidade envolvidas nestes problemas tornam-se críticos, e exigem que os gerentes de rede possuam uma vasta quantidade de informação sobre as redes manuseadas e os problemas destas. Assim, o gerenciamento de redes destina-se a auxiliar os gerentes a trabalhar com a complexidade dos dados envolvidos, de modo a garantir a máxima eficiência e transparência da rede para os seus usuários.

Este capítulo apresenta os principais conceitos de gerência de redes, os sistemas de registro de problemas e o uso de sistemas especialistas no domínio, destacando o uso de raciocínio baseado em casos e as aplicações da área que utilizam esta abordagem.

3.1 Introdução

O gerenciamento de redes pode ser definido como a prática de monitorar e controlar uma rede, de modo que ela corresponda às expectativas de seus usuários; o planejamento de ampliações ou modificações na rede, a fim de aumentar a demanda nas operações da rede e a incorporação de novos elementos na rede sem interferir nas operações existentes [LEW 95]. Possui diversos objetivos [UDU 96]: garantir sua disponibilidade, reduzir seus custos operacionais, aumentar a flexibilidade de operação e integração das redes, aumentar a eficiência das redes, permitir facilidades de uso, garantir características de segurança.

Gerenciar redes de computadores é uma tarefa complexa, envolvendo a monitoração e o controle de diferentes componentes de hardware e software, tais como equipamentos de nível físico e de enlace, componentes de computadores (processadores, impressoras, etc.), componentes de interconexão (*bridges*, roteadores, *switches*, etc.), hardware de telecomunicações (modems, multiplexadores, etc.), sistemas operacionais, aplicações e ferramentas de software, softwares de interconexão (presentes nas *bridges*, roteadores, etc.), aplicações cliente-servidor (servidores de banco de dados, servidores de arquivos, servidores de impressão, etc.), software de comunicação de dados [UDU 96].

Um importante fator no gerenciamento é a habilidade de adquirir informações sobre os equipamentos envolvidos e as mudanças ocorridas nestes [LEI 96]. Até alguns

anos atrás, a coleta de informações nos diferentes dispositivos de rede era feita através de métodos distintos, proprietários [NUN 97]. Com o passar do tempo, porém, foram desenvolvidos protocolos de gerenciamento padronizados específicos para o gerenciamento de redes, que fornecem métodos para monitorar e configurar os diversos equipamentos na rede. Atualmente, os dois protocolos mais comuns são o protocolo SNMP (*Simple Network Management Protocol*), desenvolvido pela comunidade Internet, e o CMIS/CMIP (*Common Management Information Services/Common Management Information Protocol*), padrão da ISO [LEI 96].

3.2 Áreas Funcionais da Gerência de Redes

Para melhor definir o seu escopo, o gerenciamento de redes foi dividido em cinco áreas funcionais pela ISO, cada uma com uma responsabilidade específica: **gerenciamento de configuração**, **gerenciamento de segurança**, **gerenciamento de performance**, **gerenciamento de contabilização** e **gerenciamento de falhas** [LEI 96][UDU 96] [CAR 93][LEW 95].

O **gerenciamento de configuração** fornece facilidades para a obtenção dos dados sobre a rede e utilização destes dados para gerenciar a configuração de todos os equipamentos da rede, de modo a aumentar o controle dos gerentes através do acesso rápido aos dados de configuração. Ele deve incluir facilidades como capacidade de descobrir uma rede, manter o inventário dos tipos de equipamentos de uma rede e detalhes sobre eles, detectar mudanças ocorridas e executar mudanças. O gerenciamento de configuração consiste de três etapas[LEI 96]: **obtenção das informações** sobre o ambiente de rede corrente, de modo que problemas gerados por simples erros de configuração sejam rapidamente descobertos, utilização dos dados obtidos para **modificar a configuração** dos equipamentos de rede necessários, o que precisa ser feito em tempo real em virtude da contínua modificação dos dados no ambiente da rede, e **armazenamento dos dados**, mantendo uma relação atualizada de todos os componentes de rede e produzindo diversos relatórios. Estas etapas podem ser efetuadas através de métodos manuais ou automáticos.

O **gerenciamento de segurança** envolve o controle de pontos de acesso às informações sensíveis em uma rede — as informações armazenadas nos equipamentos de rede que a organização deseja manter em segurança e não devem estar disponíveis para todos os usuários. O gerenciamento de segurança é responsável pela proteção destas informações e detecção e relato das tentativas de intrusão na rede ocorridas com sucesso ou não, incluindo facilidades para procedimentos de controle, tais como estabelecer políticas para o uso da rede, estabelecer e manter chaves criptografadas e códigos de autorização, manter um registro (*log*) do acesso à rede, prevenir e relatar acessos não autorizados, iniciar procedimentos de investigação em resposta a acessos não autorizados, detectar e prevenir vírus de computadores [LEW 95]. O gerenciamento de segurança envolve quatro etapas: **identificar as informações sensíveis**, **encontrar os pontos de acesso** (que podem incluir serviços de software, componentes de hardware, o meio de comunicação da rede), **tornar** os pontos de acesso **seguros** e **manter a segurança** destes pontos [LEI 96].

O **gerenciamento de performance** envolve a medida da performance do hardware, software e meios de comunicação da rede de modo que a rede permaneça acessível e sem congestionamentos, oferecendo um nível de serviço consistente para os

usuários. Utilizando o gerenciamento de performance, é possível monitorar a utilização dos enlaces e equipamentos da rede e, em posse dessas informações, determinar tendências de utilização, isolar problemas de performance e resolvê-los mesmo antes que eles causem impacto na performance da rede. As etapas envolvidas nesse processo são **coletar** os dados de utilização dos equipamentos e enlaces de rede, **analisar os dados relevantes**, de modo a identificar tendências de utilização elevadas, **estabelecer limiares** de utilização e usar procedimentos de simulação para identificar como a rede pode ser alterada para **maximizar sua performance** [LEI 96].

O **gerenciamento de contabilização** diz respeito à análise de como os recursos disponíveis são utilizados e os custos envolvidos no uso desses recursos. Envolve o rastreamento e geração de relatórios da utilização dos recursos da rede por cada usuário ou grupo de usuários, a fim de estabelecer métricas, verificar cotas, determinar custos e cobrar usuários. Permite um aumento do entendimento da utilização da rede que pode auxiliar também no desenvolvimento de uma rede mais produtiva. O gerenciamento de contabilização envolve os processos de [LEI 96]: **obter dados sobre a utilização da rede**, **estabelecer quotas** de utilização utilizando métricas e **cobrar os usuários** pelo uso da rede.

Por fim, o **gerenciamento de falhas** envolve a localização e correção dos problemas ocorridos nas redes. Provê meios para o reconhecimento dos problemas da rede, relatório e registro desses problemas, correlação e avaliação destes, identificação das causas dos problemas, inicialização dos procedimentos de correção das falhas, verificação da correção [LEW 95]. O gerenciamento de falhas contribui para o aumento da confiabilidade da rede, em virtude de suas ferramentas permitirem uma rápida detecção de problemas e inicialização de procedimentos de recuperação: com elas, são fornecidas as informações necessárias sobre o estado corrente da rede, e os gerentes podem trabalhar na resolução de sua falha mesmo antes desta ser detectada pelos usuários. Conforme aponta Allan Leinwand [LEI 96], o gerenciamento de falhas compreensivo é uma das tarefas mais importantes envolvidas no gerenciamento de redes.

O gerenciamento de falhas consiste primeiramente da **detecção e aviso do problema**, que pode ser efetuado através de alarmes e eventos. O filtro e a correlação de alarmes são funções importantes, além do registro (*log*) das falhas e erros ocorridos. A etapa seguinte consiste em **isolar a causa do problema**, utilizando procedimentos de diagnóstico e testes. Quando um problema ocorre, é importante identificar a sua causa correta. A análise do problema requer, além dos registros de *log*, detalhes das prováveis causas e as ações de diagnóstico recomendadas, que devem estar disponíveis imediatamente para os gerentes envolvidos na resolução do problema. Como os problemas podem ser resultantes de diversas causas, ou um problema pode ser responsável por diferentes alarmes sendo gerados, estes devem ser correlacionados em uma causa simples se há chances de múltiplas mensagens estarem sendo geradas. Por fim, o problema é **corrigido**, através de métodos manuais ou automáticos [UDU 96]. Um **rastro** de todo o ciclo de vida do problema deve ser mantido, sendo usualmente utilizados para isso *trouble tickets* (registros de problemas).

A seguir apresentamos algumas considerações sobre os sistemas de registro de problemas e seu uso para o armazenamento do ciclo de vida de um problema.

3.3 Sistemas de Registro de Problemas

Os *trouble ticket systems* (TTS - sistemas de registro de problemas) são utilizados para monitorar os problemas de uma rede, mantendo um *rastro* do ciclo de vida de um problema, e têm sido amplamente discutidos na bibliografia [JOH 92][LEI 96][UDU 96][MAD 94]. Eles podem, também, ser utilizados para manter o registro de outros aspectos da rede que não apenas digam respeito ao gerenciamento de falhas, tais como modificações de configuração, modificações de segurança, requisições e melhoramentos de performance. [LEI 96].

São atribuídos para esses sistemas muitas funções ou propósitos [JOH 92]:

- servir como uma memória dos problemas específicos ocorridos no domínio de gerência, mantendo a história completa do problema. Com isso, é possível a comunicação entre os gerentes envolvidos na resolução do problema, de modo que os problemas que se estendem por diversos turnos de trabalho possam ser analisados e trabalhados imediatamente por um novo gerente quando necessário;
- servir como um visualizador da lista de problemas correntes em um domínio, que deve preferencialmente ser fornecida em ordem de prioridade. Eles auxiliam, assim, o escalonamento do fluxo de trabalho dos diversos gerentes de um domínio;
- enviar atribuições de tarefas ou consultas através de sistemas de correio eletrônico que estejam integrados ao sistema;
- atribuir temporizadores para cada registro de problema, que quando decorridos gerem um alerta para o registro associado, lembrando sobre o problema;
- 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;
- permitir a análise estatística dos equipamentos da rede e da produtividade do domínio de gerência, através do processamento dos campos fixos dos registros. Permitem que sejam gerados relatórios de Tempo Médio entre Falhas e Tempo Médio de Reparo, além de relatórios estatísticos de controle de qualidade, que propiciam a detecção de equipamentos defeituosos antes de uma falha efetiva;
- atuar como filtro dos alertas que sejam relacionados a um registro de problema em aberto;
- permitir aos usuários 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.

Além destes propósitos, os sistemas de registro de problemas servem também para a interação entre diversos domínios envolvidos em um problema. Cabe lembrar que a gerência de redes no 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 94], como no caso da rede desta universidade, de redes regionais ou nacionais. Porém, se de um lado a divisão da responsabilidade facilita o diagnóstico, uma vez que os administradores locais têm grande conhecimento daquela parte da rede, por outro lado, a possibilidade dos 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 administradores das diversas sub-redes. Assim, sistemas de registro de problemas podem ser usados para compartilhar as informações sobre a resolução do problema, desde que possam ser acessados de modo distribuído com os controles e restrições adequadas, permitindo a colaboração dos especialistas dos diversos domínios envolvidos no diagnóstico do problema.

Um sistema de registro de problemas cria para cada problema informado um novo registro, atribuindo para este um número único, e registra os dados sobre o problema e ações efetuadas 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 que o problema é registrado, o sistema interage com sua base de dados ou com bancos de dados da topologia, de modo a preencher automaticamente as informações solicitadas pelo registro que ele tem condições de responder.

Cada problema registrado deve ter uma categoria assinalada automaticamente pelo sistema ou manualmente pelo gerente encarregado do problema, o que pode auxiliar no futuro a identificar os problemas que ocorrem mais freqüentemente. Algumas classificações comuns, apontadas em [LEI 96], seriam: 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 um rede, variando o formato dos registros principalmente nos campos fixos.

As informações podem ser solicitadas através de campos fixos ou de texto de forma livre [JOH 92]. Os campos fixos têm a vantagem de serem utilizados mais facilmente para busca e ter sua consistência verificada com mais exatidão. Além disso, estes campos são apropriados para dados que são fornecidos automaticamente pelo sistema. Eles são, porém, mais apropriados para ambientes de resolução de problemas bem compreendidos e estáveis, sendo utilizados para problemas específicos. Isso acontece porque embora tendam a tornar os dados mais consistentes e confiáveis, os campos fixos têm a desvantagem de forçar os usuários a escolherem entre os valores preparados e permitidos que nem sempre representam a situação com precisão.

A estrutura de um registro de problema sugerida por [JOH 92] consiste de três partes: **cabeçalho**, **atualizações** e **dados da resolução**. O **cabeçalho** é responsável pelas informações de abertura do problema, que incluem [JOH 92]:

- 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 centro de operações 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.

Os quatro primeiros itens apresentados são sugeridos para todos os sistemas. Os demais são apresentados como opções para coleta de informações com propósitos específicos ou informações associadas a diferentes tipos de problemas.

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 não é conhecida e a descrição fornecida pode ser imprecisa e demasiado complexa. É sugerido que ao menos um campo de texto livre seja fornecido nesse estágio do problema, para esse tipo de informação. Os campos de atualização seguintes podem ser bastante simples, tais como o exemplificado em [JOH 92] “*Site chamado; sem resposta*”, e podem ser armazenados em campos fixos ou campos de texto livre, variando conforme a implementação. 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 campo de texto livre.

Por fim, os **dados da resolução** representam as informações que resumem o problema para futuras análises estatísticas, e também para uso em problemas similares. Os campos apontados em [JOH 92] que seriam úteis para esta etapa são:

- hora e data da resolução do problema;
- duração;
- 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 para investigações estatísticas.

Outras informações que poderiam ser solicitadas são ainda sugeridas na bibliografia consultada. Entre estas, estão o estado corrente do problema, os usuários afetados e as prováveis causas para o problema [UDU 96][LEW 95].

3.4 Sistemas Especialistas para Gerência de Redes

O processo de tomada de decisão em um ambiente, incluindo o gerenciamento de redes, consiste em obter informações, analisá-las e tomar decisões baseadas nestas

informações. Esse é um processo frequentemente decisivo para determinar o sucesso ou a falha de um ambiente, de modo que grandes esforços são investidos nos estudos desse processo. Exemplos de processos de tomada de decisão dentro da gerência de redes incluem gerenciamento da detecção e correlação de falhas na rede, do roteamento das redes, do planejamento de configuração e controle de configuração *on-line*, da análise e otimização da performance dos sistemas de comunicação, da análise da segurança da rede, da detecção de intrusão na rede [ERI 89].

O processo pode ser feito de modo automático. Com a automação dessas atividades, a produtividade da equipe dos especialistas envolvidos é aumentada e mais serviços podem ser oferecidos pela equipe, que pode concentrar-se também em outras atividades em que é requisitada. A automatização de tais sistemas envolve a construção de sistemas que imitem as atividades de tomada de decisão por especialistas humanos. Técnicas da IA podem ser usadas para o desenvolvimento destes sistemas — os sistemas especialistas.

As redes de computadores e telecomunicações estão se tornando maiores e mais heterogêneas, possuindo cada vez mais diferentes características que precisam ser consideradas no processo de gerenciamento de redes. São, assim, um domínio em que o uso de IA traz inúmeros benefícios. Aliado a isso, a gerência de redes é uma área que se beneficia com a presença de especialistas que, via de regra, possuem um considerável domínio técnico em computação. Com isso, expressam mais facilmente seu conhecimento na forma adequada para a representação do conhecimento [STA 93].

Os sistemas especialistas podem ser aplicados na gerência de redes para diversas áreas. No **gerenciamento de configuração**, eles podem ser utilizados para auxiliar no planejamento de redes. Esses sistemas devem possuir informações sobre a topologia física da rede, mapas de roteamento e a topologia lógica-virtual.

O **gerenciamento de falhas** foi a área que teve os primeiros sistemas especialistas. Nesta área, eles podem ser utilizados para o diagnóstico e manutenção das redes. Os sistemas de diagnóstico efetuam a análise das falhas da rede e seus efeitos, a fim de determinar as prováveis causas e definir os reparos e manutenção necessários para resolver o problema. Os benefícios dos sistemas especialistas de diagnóstico incluem [ERI 89]: diminuir o tempo para detectar as causas do problema; sugerir aos gerentes da rede ações para resolver o problema; e automatizar a resolução de problemas pela intervenção direta, resultando em comandos corretivos para uma “rede inteligente”.

Uma outra aplicação de sistemas especialistas para gerenciamento de falhas envolve o controle da rede, sendo utilizados para estender as capacidades dos operadores da rede, e não substituí-los. Os benefícios de tais sistemas são aumentar a precisão e eficiência da intervenção do operador, facilitar seu processo de tomada de decisão e reduzir a quantidade de tempo necessária para restaurar ou alterar a rede [ERI 89]. Além de 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 prioridade associados.

Sistemas especialistas podem ser também aplicados para o **gerenciamento de performance**, **gerenciamento de contabilização** e **gerenciamento de segurança**. Uma aplicação desenvolvida para o último, por exemplo, combina 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 89].

Existem diversos exemplos de sistemas especialistas desenvolvidos para a área de gerência de redes de computadores e de telecomunicações [ERI 89][KRI 91][CRO 88][TAR 90][NUN 97][HAR 97]. Os sistemas especialistas de diagnóstico são as aplicações mais populares. Na área de telecomunicações, a manutenção de *switches*, tipicamente para a tecnologia dos antigos *switches*, é uma aplicação comum. Para a administração de redes em telecomunicações, a aplicação mais comum é para roteamento de tráfego ou gerenciamento de tráfego [GOY 91].

Nas diversas aplicações existentes, o paradigma de representação do conhecimento e **raciocínio baseado em regras (RBR - rule based reasoning)** [JAC 86][HAR 89][CRO 88] tem sido a tecnologia básica utilizada, e tem sido aplicado com sucesso para uma ampla variedade de problemas do mundo real [GOY 91]. Um sistema baseado em regras consiste basicamente em uma memória de trabalho, uma base de regras e procedimentos de controle. Em aplicações no domínio de redes, a memória de trabalho consiste tipicamente na representação das características da rede, incluindo informações topológicas e do estado da rede, enquanto que a base de regras representa o conhecimento sobre as operações que devem ser efetuadas quando a rede entra num estado indesejável. As regras podem efetuar diversas ações, tais como testes na rede, consultas em um banco de dados, invocação de um outro sistema especialista, envio de avisos e criação de registros de problemas. Quando a rede entra num estado indesejável, os procedimentos de controle selecionam as regras aplicáveis para a situação corrente, e uma estratégia de controle pré-determinada seleciona dentre as regras aplicáveis a que será de fato executada [LEW 93].

Os sistemas baseados em regras possuem, porém, algumas limitações, especialmente considerando o uso de sistemas especialistas na gerência de redes. Uma destas limitações é o gargalo apresentado na aquisição de conhecimento. Isto é especialmente aplicado para a gerência de redes, em que os esquemas de representação do conhecimento são freqüentemente inadequados para seu conhecimento — um exemplo é o tipo de conhecimento utilizado em gerentes de tráfego, que é amplamente baseado na experiência, ao invés de em heurísticas pré-compiladas ou modelos de domínios bem conhecidos. O domínio de gerência requer a combinação da avaliação da situação, resolução de problemas e controle em tempo próximo ao real, quase de modo contínuo, num domínio em que o modelo formal não existe e o conhecimento especialista é freqüentemente irregular e sem coordenação. Além disso, a velocidade na mudança tecnológica é muito alta, com mudanças na arquitetura de vários elementos da rede e novos serviços sendo integrados constantemente [GOY 91].

Uma outra característica das operações na rede — a restrição de tempo para tomada de decisões — tem como efeito a necessidade de resposta em tempo real em muitas aplicações do gerenciamento e a valorização de sistemas que apresentam funções com ciclo fechado, já que o tempo de reação humana é muito longo para algumas operações da rede e a tarefa dos especialistas deve ser idealmente apenas de supervisão [GOY 91]. Contudo, já que os sistemas automáticos apresentam um ciclo fechado de execução, a performance da rede se torna dependente da confiabilidade de tais sistemas, que requerem técnicas de verificação e validação precisas.

Novas tecnologias de conhecimento têm sido assim estudadas pela comunidade da IA, freqüentemente apelidando os sistemas em sistemas especialistas de segunda geração. Entre estas, citamos a Inteligência Artificial Distribuída (IAD) [LEW 95a][GOY 91][KRI 91], voltada para compartilhar o conhecimento e melhor coordenar a resolução de problemas, e o Aprendizado Automático [GOY 91], que provê a habilidade de obtenção automática de conhecimento em um ambiente dinâmico. Informações adicionais e exemplos de sistemas baseados em regras e nas demais tecnologias da IA para gerência de redes podem ser obtidas em [ERI 89][KRI 91][CRO 88][TAR 90][NUN 97][HAR 97]. Além destes, estudos sobre a representação do conhecimento tem sido também desenvolvidos, a fim de identificar modos melhores e mais robustos de utilização do conhecimento do especialista [GOY 91].

A representação do conhecimento é o modo de codificar o conhecimento do especialista sobre o domínio, de modo que seja possível armazenar, processar e utilizar o conhecimento codificado. Existem muitos esquemas de representação desenvolvidos, sendo que a maior parte dos sistemas utiliza para a representação regras de produção ou frames [GOY 91]. Os paradigmas baseados em regras têm sido, como foi dito anteriormente, amplamente utilizados no domínio de gerência de redes, mas apresentam algumas limitações quando aplicado à gerência de redes.

Um paradigma que é, segundo [GOY 91], adequado para o diagnóstico e resolução de problemas é o **raciocínio baseado em modelos (MBR - *model-based reasoning*)**. O MBR trabalha através da interação da observação do elemento (o comportamento de um equipamento, por exemplo) e da previsão baseada no modelo. Esta abordagem pode ser utilizada quando o equipamento ou sistema pode ser formalmente modelado e o domínio do problema não contém interações entre muitos equipamentos. É, entretanto, difícil modelar em uma aplicação uma rede completa e suas interações.

Um outro paradigma que pode ser utilizado para o raciocínio em sistemas especialistas em gerência de redes é o **raciocínio baseado em casos (CBR- *case-based reasoning*)**, abordado no capítulo anterior. Sua aplicação no gerenciamento de redes traz diversos benefícios, tais como: possibilidade de incorporar novos conhecimentos; diminuição da fragilidade dos sistemas, já que não precisam casar um conjunto preciso de condições, e sim identificar situações similares; adequabilidade à representação do conhecimento adquirido a partir de experiências passadas.

Cabe comentar que as diferentes representações de conhecimento existentes são apropriadas para porções específicas do conhecimento, que são, por sua vez, necessárias em uma operação mais complexa. Assim, o uso de representações híbridas contribui para o aumento da robustez do sistema especialista, diminuindo a sua fragilidade quando este não está apto para uma situação [GOY 91].

Comentaremos, a seguir, algumas aplicações de gerenciamento de redes que fazem uso do paradigma de raciocínio baseado em casos.

3.5 Raciocínio Baseado em Casos em Gerência de Redes

Existem, ainda hoje, poucas aplicações para gerência de redes que fazem uso de raciocínio baseado em casos. As aplicações encontradas são utilizadas para suporte à

decisão, algumas apresentando execução automática, outras sugerindo ações. Essas aplicações são aplicadas ao domínio das telecomunicações e de redes de computadores.

No domínio das telecomunicações, citamos os sistemas NETTRAC [KOP 88][BRA 91], o sistema NEMS [MAT 95] e o sistema apresentado em [CAU 95], aplicados ao gerenciamento de tráfego, além do sistema ACS [PEN 99], aplicado para diagnóstico e correção de falhas. No domínio de redes de computadores, destacamos o sistema ExSim [STA 93], o sistema CRITTER [LEW 93][LEW 95] e o sistema MASTER [DRE 95].

A seguir, apresentamos os sistemas NETTRAC, ExSim e CRITTER. Esses sistemas representam um exemplo do uso de raciocínio baseado em casos nos diferentes domínios de gerenciamento em que foram encontradas aplicações: gerenciamento em telecomunicações, roteamento e gerenciamento de falhas em redes de computadores, respectivamente. Os aspectos e particularidades em cada abordagem que contribuíram para o desenvolvimento do sistema DUMBO, proposto neste trabalho, são também apresentados.

3.5.1 Sistema NETTRAC

O sistema NETTRAC [KOP 88][BRA 91] é aplicado para o gerenciamento do fluxo de tráfego em uma rede telefônica pública comutada. O controle de tráfego nestas redes é realizado pela alocação de um conjunto variável de recursos da rede, de modo a satisfazer as demandas de uma amostra flutuante de chamadas telefônicas. Esse controle é exercido por um grupo de gerentes de tráfego experientes, localizado em um ponto central, que são responsáveis pelas modificações no processamento das chamadas telefônicas para melhor atender às necessidades. Os problemas apresentados nesse domínio são problemas de *controle contínuo* — um quadro de dados é analisado para fornecer o diagnóstico da situação, um conjunto de ações de controle é efetuado para tratar a situação e os efeitos destas ações são continuamente monitorados e ajustados. [KOP 88]

Os nodos distribuídos de uma rede telefônica são sistemas de comutação telefônica automática (os *switches*) — fontes dos dados da rede e máquinas que executam os controles de tráfego —, cada um capaz de manusear milhares de conexões simultaneamente. Estes *switches*, assim como os grupos de troncos que os interconectam, são os recursos da rede com capacidade finita sendo gerenciados. Quando um *switch* falha, parcialmente ou completamente, ou o tronco é rompido, a capacidade de processamento total da rede é reduzida e os gerentes de tráfego buscam o melhor modo de completar as chamadas. Da mesma forma, mesmo quando a capacidade da rede é total, a demanda pode excedê-la — por exemplo, em horários críticos, em dias especiais como dia de Natal, em chamadas de rádio — e, também nessas situações, os gerentes procuram encontrar modos de satisfazer a demanda local utilizando os recursos disponíveis do resto da rede. Esses sistemas apresentam aspectos relativos a diagnóstico e planejamento, que são, como aponta Kopeikina *et al*, domínios clássicos para uso de IA, e funções de planejamento têm sido amplamente desenvolvidas utilizando CBR, que têm diversas vantagens de serem aplicados para sistemas de tal tipo [KOP 88].

Uma dessas vantagens diz respeito ao tipo de conhecimento utilizado pelos gerentes de tráfego. A partir das entrevistas com os gerentes de tráfego, Kopeikina *et al* descobriram que grande parte do seu conhecimento é episódico, isto é, o especialista

resolve um problema corrente pela lembrança de uma experiência anterior similar, que pode ser tanto uma situação real específica como uma generalização de uma classe de situações similares ocorridas. Quando um gerente de tráfego está passando seus conhecimentos para um novo gerente, ele utiliza essas experiências/casos para transmitir seu conhecimento especialista, o que indica a adequação de utilizar casos para representar o conhecimento envolvido nesse domínio. Assim, a aquisição de conhecimento naturalmente resulta em casos de gerenciamento de tráfego [KOP 88].

Além disso, Kopeikina *et al* apontam que a partir do entendimento do domínio adquirido pelo grupo, a utilização de experiências passadas parece ser o melhor guia disponível para tomada de decisão pois, embora exista algum conhecimento de relações gerais causais entre as ações de controle e respostas da rede a estas, nenhum modelo completo do domínio existe. Assim, os gerentes de rede frequentemente agem de acordo com situações anteriores bem sucedidas, mantendo um entendimento incompleto das razões do sucesso ocorrido. Como comentam Kopeikina *et al*, isso “*é compreensível, dada a intratável escala da dinamicidade da rede e das restrições de tempo para resposta*” [KOP 88]p.251. Assim, o raciocínio baseado em casos oferece um meio adequado para levar vantagem deste conhecimento episódico.

Uma outra questão diz respeito à contínua e gradual mudança sofrida no domínio do gerenciamento de tráfego — mudanças ocorrem quando a companhia telefônica aumenta o número de *switches* sobre gerenciamento de tráfego, quando a companhia adiciona novos tipos de *switch* ou novos componentes de hardware ou software que gradualmente modificam o comportamento da rede. Essas mudanças passam, porém, frequentemente despercebidas pelos especialistas humanos envolvidos neste domínio, que vão adaptando o seu conhecimento gradualmente e aplicando versões modificadas de situações anteriores para resolver novas situações. Assim, sistemas utilizando CBR podem criar novos casos para resolver as situações em que o conhecimento dos casos armazenados é insuficiente, ao passo que a utilização de sistemas especialistas convencionais que não utilizem CBR iriam demonstrar uma contínua degradação de performance com estas mudanças [KOP 88].

Portanto, o uso de raciocínio baseado em casos é muito adequado para domínios como o de gerenciamento de tráfego: ele permite a utilização de experiências anteriores — casos — para resolução de novas situações e permite que novos casos sejam criados para incorporar as situações em que os casos existentes não são aplicáveis adequadamente [KOP 88]. Essas vantagens são compartilhadas pelos sistemas aplicados ao domínio de gerenciamento de falhas em redes de computadores, como o sistema proposto neste trabalho. Nesse domínio, pode ser também identificado nos gerentes de rede uma grande ocorrência de conhecimento episódico, que vai sendo adaptado para as diversas modificações e evolução sofrida pelas redes atuais.

O processo envolvido no sistema NETTRAC inicia com o Pré-processador, que interpreta as informações da rede e forma a representação do problema, denominada *Problem Statement* (PS - Descrição do Problema). O Pré-processador escolhe o esqueleto de PS a ser utilizado, entre os diferentes esqueletos existentes, de acordo com o tipo de problema corrente, e avalia se o problema é antigo e já está sendo tratado pelo sistema ou se é um problema novo. No primeiro caso, os dados sobre o problema são enviados ao Monitor, responsável por relacionar os dados correntes sobre o estado do problema com o estado esperado (as expectativas para as ações já efetuadas), relatado

no caso armazenado. Quando os efeitos esperados não foram atingidos, o Monitor deve propor ajustes ou modificações no tratamento.

Já no segundo caso, isto é, quando o Pré-processador identifica um novo problema, a descrição (PS) é transferida para o Indexador/Casador que recupera entre os casos armazenados na base os casos mais relevantes para o problema corrente. O Seletor escolhe, então, entre os casos selecionados, o que possui maior potencial de ser útil para a situação, e a experiência armazenada nesse caso é aplicada para a situação corrente pelo Modificador, sendo alterada quando necessário.

O caso modificado é proposto ao Criticador, que utiliza procedimentos específicos ao domínio para determinar se o tratamento proposto parece levar a algum efeito danoso. Se o caso proposto for vetado, o Modificador pode efetuar as modificações necessárias no caso armazenado, o Seletor pode escolher um novo caso entre os casos selecionados ou ainda o Indexador/Casador pode realizar uma nova busca para recuperar outros casos candidatos. Se o caso for aprovado pelo Criticador, ele é proposto para o usuário. Uma vez que também o usuário tenha aceito o plano de tratamento do caso, ele é, então, implementado pela Interface de Rede na forma de controles nos processamentos das chamadas dos *switches*. Esse plano é ainda registrado na História do Tratamento e comunicado ao Pré-processador, a fim de que uma nova entrada de problema associada com um tratamento sendo efetuado seja reconhecida. A figura abaixo [KOP 88] apresenta a arquitetura do sistema.

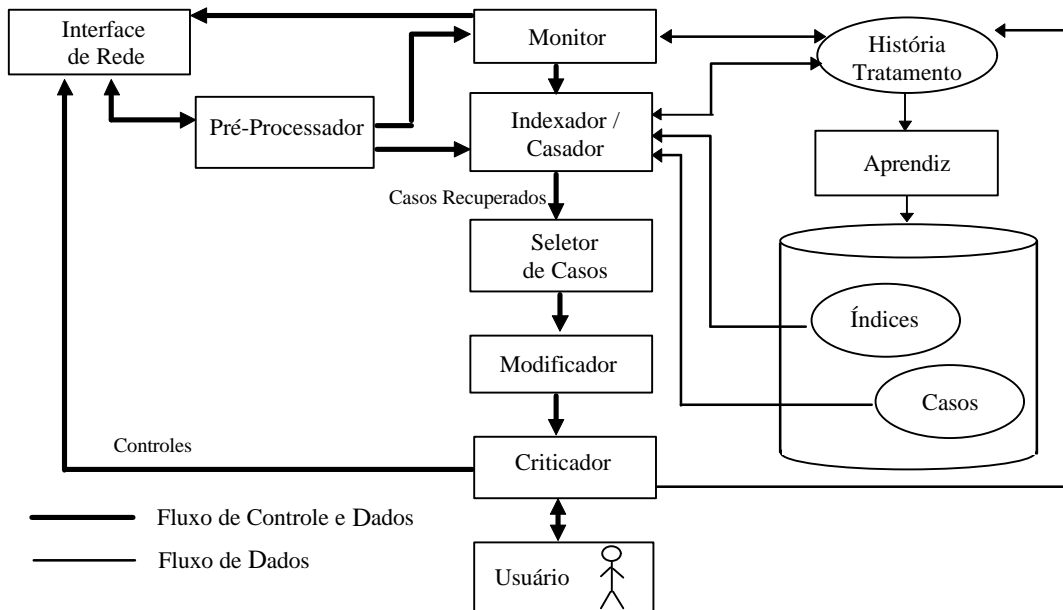


FIGURA 3.1 - Arquitetura do sistema NETTRAC

A estrutura de indexação dos casos, utilizada pelo módulo Indexador/Casador para recuperar os casos similares ao problema corrente, foi escolhida considerando-se que o domínio não suporta um conjunto simples de atributos que seja *sempre* discriminante entre as diversas possíveis situações problema — ao contrário, a relevância dos atributos é definida localmente. Essa abordagem foi também utilizada no sistema DUMBO, em que os problemas do domínio apresentam grande diversidade. Dessa forma, tornou-se necessário utilizar um conjunto de atributos discriminantes variável conforme a situação sendo tratada, como será visto no capítulo 5.

3.5.2 Sistema ExSim

O ExSim Prototype [STA 93] foi desenvolvido para o domínio de gerenciamento de redes automático, enfocando o processo de roteamento. Esse sistema utiliza um programa de simulação que simula a rede WAN sendo gerenciada, composta de *gateways* que trocam mensagens utilizando técnicas de roteamento estático. Em virtude dessa estratégia de roteamento simples, uma sobrecarga local pode ocorrer se as informações de roteamento não forem trocadas de modo adequado pelo gerenciamento, diminuindo a performance da rede. Assim, a tarefa do módulo de raciocínio baseado em casos é detectar os gargalos e mau funcionamento, através da classificação dos estados da rede e comparação desses estados com os problemas ocorridos anteriormente, armazenados na base como casos.

Por ser um sistema sem intervenção do usuário, essa abordagem apresenta aspectos como a decisão automática quanto à aplicação ou não da solução de um caso e a avaliação automática de que procedimentos realizar se o sistema não foi capaz de recuperar uma situação similar. O sistema ExSim ilustra, assim, como o uso de procedimentos sem intervenção humana é bastante complexo num domínio como o gerenciamento de redes. Quando aplicado em um sistema cujo objetivo é tratar uma vasta gama de problemas, como o sistema proposto neste trabalho, esses procedimentos se tornam ainda mais complexos pela dificuldade de avaliar a solução recuperada e simular uma situação similar. O sistema DUMBO engloba, assim, a intervenção e avaliação do usuário na sua arquitetura, como será visto no capítulo 5. Porém, o uso de procedimentos automáticos com a rede é suportado pelo DUMBO em alguns processos, tal como a obtenção de informações diretamente dos equipamentos da rede.

A figura a seguir [STA 93] apresenta a estrutura do sistema ExSim, onde podem ser identificadas as suas três partes componentes (simulador, rede gerenciada e raciocinador baseado em casos).

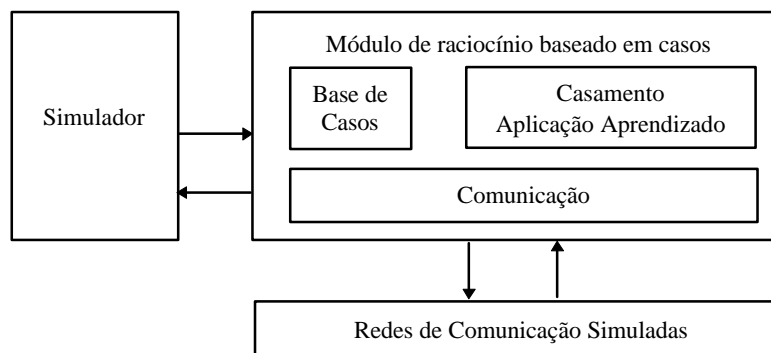


FIGURA 3.2 - Estrutura do ExSim

Os casos contidos no sistema representam situações de falha, sendo formados pela descrição do problema, descrição da solução, um nome único e por dois valores de limiares α e δ . A descrição do problema e da solução é realizada através de um conjunto de pares atributo/valor, em que cada atributo (característica) descreve um aspecto de um possível estado dos componentes da rede.

A descrição do problema consiste em um conjunto de tabelas de roteamento dos *gateways* combinadas em um atributo, a informação da carga em todos os enlaces da rede, a tabela da topologia da rede e o estado dos *gateways*. O atributo correspondente à tabela de roteamento é representado por um conjunto de matrizes inteiras, enquanto o

correspondente a carga da rede é formado por um conjunto de números positivos em ponto flutuante. Os atributos correspondentes ao estado do nodo são identificados com o estado ‘*up*’ ou ‘*down*’. A descrição da solução, por sua vez, consiste em um conjunto de tabelas de roteamento para os *gateways* da rede gerenciada, que também é representado por um único atributo, como na descrição do problema.

O valor de limiar α é utilizado para decidir se um caso é candidato para a solução do problema, enquanto que o valor de limiar δ é utilizado para decidir se a solução do caso deve ser aplicada para o problema corrente. O sistema assegura sempre que $0 < \alpha < \delta < 1$. Quando a similaridade do caso para o problema corrente excede o seu limiar α , o caso é acrescentado para os casos candidatos ao problema corrente; quando a similaridade excede o limiar δ , a solução deve ser aplicada para o problema corrente. Com isso, é possível, através do ajuste de α e δ , influenciar a probabilidade dos casos serem escolhidos. Estes valores são obtidos através do método de tentativas, e na versão do protótipo relatado na referência consultada [STA 93] os melhores resultados foram obtidos utilizando os valores 0,6 para α e 0,8 para δ .

O casamento dos casos recuperados com as descrições do estado da rede é realizado utilizando uma medida de similaridade calculada pela razão entre os indícios indicando similaridade e todos os indícios registrados, conforme apresenta a função *sim* na figura 3.3 [STA 93]. Na figura, *common* representa o número de atributos que estão presentes tanto na descrição do caso recuperado como na descrição do estado da rede e cujos valores são considerados similares, e *different* representa o número de atributos que estão também em ambas as descrições, mas cujos valores não são classificados como similares. A similaridade entre dois valores é considerada verdadeira quando a similaridade entre eles exceder um limiar global *t*.

$$\text{sim}(\text{state}, \text{case}) = \frac{a \cdot \text{common}}{a \cdot \text{common} + b \cdot \text{different}} [0, 1].$$

FIGURA 3.3 - Função de similaridade utilizada no sistema ExSim

Cada tipo de atributo dos casos possui uma função de similaridade própria. Os valores correspondentes ao estado do nodo possuem similaridade 1 se ambos possuem o valor igual (seja ele *up* ou *down*), e similaridade 0 caso sejam diferentes. Os valores descrevendo a topologia têm similaridade 1 quando são idênticos, e 0 quando não o são. A similaridade entre as tabelas de roteamento é calculada pela razão do número de entradas que coincidem em relação ao número total de entradas na tabela. Por fim, a similaridade entre os valores da carga do enlace é considerada verdadeira se ambos os enlaces excedem um limiar *C*, que representa enlaces com carga crítica. Este limiar *C* é ajustado de acordo com a carga mínima e máxima ocorrida no estado da rede, para que se garanta que a medida de similaridade seja específica e não seja tratada de modo independente pelo módulo responsável pelo casamento.

Como o sistema opera em modo de ciclo fechado, isto é, as tarefas de monitoração, raciocínio e controle devem ser feitas sem intervenção humana durante operações normais, os autores consideraram importante que ele operasse em modo *pessimista* — as ações de controle devem ser assim verificadas em caso de incerteza antes de que elas sejam aplicadas na rede gerenciada, seja utilizando simulações para verificar os resultados das operações corretivas seja comunicando as ações pretendidas para os operadores humanos a fim de obter uma verificação. Assim, a fim de

implementar uma estratégia pessimista, o sistema definiu o coeficiente a como 1, e o coeficiente b como 2.

O processo de recuperação inicia quando estados críticos na rede são reconhecidos pelo raciocinador. Isso é identificado através de um alarme recebido da rede, indicando uma sobrecarga de um dos *gateways* e informando o estado da rede, ou através de um *pooling* explícito no estado da rede. As informações sobre o estado da rede consistem em um conjunto de tabelas de roteamento dos *gateways*, na carga nos enlaces da rede, na topologia e no estado do *gateway*. As informações recebidas são comparadas às partes problemáticas dos casos armazenados utilizando-se medidas de similaridades. Quando um caso é selecionado como o melhor caso, a sua solução é enviada para os componentes ativos da rede a fim de que a situação crítica seja aliviada. Essa solução consiste de um novo conjunto de tabelas de roteamento para os *gateways* atingidos pela sobrecarga ou que são origem dela.

Quando nenhum caso similar é selecionado, diferentes ações são tomadas, dependendo do problema ter sido identificado através de um alarme da rede ou não. Se o problema não foi identificado através de um alarme, então significa que em termos do conhecimento do sistema a rede está operando corretamente e nenhuma ação precisa ser tomada. Se, entretanto, o problema foi identificado através de um alarme da rede, é identificado que para um problema da rede existente não há solução no sistema e assim novos conhecimentos devem ser adquiridos. As informações sobre o estado da rede são então passadas para o programa simulador, que simula uma rede similar à rede gerenciada, com a diferença que na rede simulada é utilizada uma estratégia de roteamento dinâmica dependente da carga. Assim, quando a simulação termina, o seu resultado representa um conjunto de tabelas de roteamento ótimo aplicável para a rede sendo gerenciada, dando origem a um novo caso, que é armazenado na memória de casos, sendo sua solução repassada para a rede gerenciada [STA 93].

O protótipo deste sistema foi implementado em linguagem C++.

3.5.3 Sistema CRITTER

O sistema CRITTER [LEW 93][LEW 95] é baseado na adição de um componente de resolução de problemas que utiliza o paradigma de raciocínio baseado em casos sobre um sistema de registro de problemas simples. O contexto para o qual este sistema foi desenvolvido é similar ao do sistema DUMBO, enfocando falhas em redes de computadores e o uso de raciocínio baseado em casos associado a sistemas de registros de problemas. Além disso, a abordagem utilizada nesse sistema apresenta diversos aspectos que influenciaram o sistema DUMBO, tais como o uso de uma linguagem estruturada para o registro de problema e o uso de atributos relevantes distintos, variando conforme o tipo de problema sendo tratado, que são comentados a seguir.

Um exemplo de um registro do CRITTER [LEW 93] é apresentado na figura 3.4. Alguns campos do registro são usados apenas para propósitos de gerenciamento, tais como a prioridade ou o estado do registro, enquanto outros descrevem a natureza do problema (tais como o nome e o tipo do equipamento). Os registros podem ser formados de modo mais ou menos estruturados, com linguagem estruturada ou livre. No exemplo da figura, por exemplo, o campo *Additional Data* contém uma lista de dados que o gerente pode estar interessado em conhecer antes de começar a tentar solucionar o problema. Esse campo é descrito usando uma linguagem estruturada. Isso ocorre

porque, como aponta Lundy Lewis [LEW 93], é aconselhável desenvolver uma linguagem descritiva estruturada para a informação para os campos como *'Trouble'*, *'Additional Data'* e *'Resolution'*. É importante ressaltar, porém, que, embora essa abordagem evite algumas difíceis questões referentes ao processamento de linguagem natural, ela tem a desvantagem de impor limites para a quantidade de aprendizado que o sistema pode atingir [LEW 93].

| | | | |
|--------------------|--|-------------------|--|
| Entry-ID | <input type="text" value="0000000116"/> | Submitter | <input type="text" value="SPECTRUM"/> |
| Alarm ID | <input type="text" value="57"/> | Create-date | <input type="text" value="07/24/92 08:51:08"/> |
| Alarm Date/Time | <input type="text" value="07/24/92 08:51:07"/> | Notify-method | <input type="radio"/> None <input type="radio"/> Notifier <input type="radio"/> E-mail |
| Condition | <input type="radio"/> Red <input type="radio"/> Orange <input type="radio"/> Yellow | Assigned-Priority | <input type="radio"/> Low <input type="radio"/> Medium <input type="radio"/> High |
| Device Name | <input type="text" value="Ethernet-Randy"/> | Assigned-to | <input type="text"/> |
| Device Type | <input type="text" value="Ethernet"/> | Last-modified-by | <input type="text" value="SPECTRUM"/> |
| IP Address | <input type="text"/> | Modified-date | <input type="text" value="07/24/92 08:51:08"/> |
| Trouble | <input type="text" value="file_transfer_throughput=slow"/> | | |
| Additional Data | <input type="text" value="network_load=20, collision_rate=15,deferment_rate=20,users=31"/> | | |
| History of Trouble | <input type="text"/> | | |
| Probable Cause | <input type="text"/> | | |
| Resolution | <input type="text"/> | | |
| Resolution Status | <input type="radio"/> Good <input type="radio"/> No Good <input type="radio"/> In Progress | | |
| Ticket Status | <input type="radio"/> New <input type="radio"/> Assigned <input type="radio"/> Reject <input type="radio"/> Closed | | |

FIGURA 3.4 - Exemplo de um registro de problema

O cálculo da similaridade entre dois registros do sistema foi concebido de modo a dar importância para o número de *casamentos entre atributos que são relevantes para um particular tipo de problema*. Para isso, Lewis aponta que a base de registros deve ser acrescentada com um conjunto de *determinadores* que registrem informações de relevância entre as classes de problemas da rede e os conjuntos de atributos dos registros [LEW 93]. A figura a seguir, extraída de [LEW 93], apresenta um exemplo de um determinador, onde são indicados os atributos relevantes para os problemas do tipo *'vazão_da_transferência_de_arquivo'*. Os determinadores, assim, permitem que o sistema focalize os registros de problemas mais prováveis de conterem estratégias que levem à resolução do problema corrente.

Exemplo de um determinador:

A solução para o problema 'vazão da transferência de arquivos está lenta' é determinado observando a largura de banda, carga da rede, taxa de colisões e taxa de retardo dos pacotes.

FIGURA 3.5 - Exemplo de um determinador

Como aponta Lewis, existem diversas abordagens para a criação dos determinadores. A primeira delas se baseia na observação de que os gerentes de rede aplicam determinadores conceituais quando vão inspecionar uma base de dados de registros de problemas em busca de registros úteis. Tipicamente, é recuperado um conjunto de registros que case com o campo problema da situação corrente, e este conjunto é, então, *'podado'*, através do casamento de outros atributos. Assim, com o

uso de sistemas de registro de problemas, é possível armazenar as ações efetuadas pelos gerentes durante o processo de seleção e definir estes processos como macros, que se tornam os determinadores, podendo ser usados novamente em problemas similares [LEW 93].

Uma segunda abordagem estabelece que os especialistas do domínio definam um pequeno, mesmo que imperfeito, conjunto de determinadores. Essas regras podem ser então refinadas automaticamente com o aprendizado do sistema e ser modificadas com as mudanças da rede. Com isso, a obstrução da aquisição de conhecimento, que ocorre em domínios de rápidas mudanças, torna-se menos problemática. Por fim, uma terceira alternativa seria aplicar um algoritmo de informação teórico para a base de dados dos registros de problemas, que indicaria uma lista de determinadores que casam com os atributos do registro para os procedimentos de resolução de falhas [LEW 93]. O sistema CRITTER foi desenvolvido utilizando a segunda abordagem citada.

Após a recuperação ser efetuada e o registro com maior similaridade ser escolhido, um estratégia de resolução deve ser proposta ao usuário (ou executada automaticamente), o que é realizado pelo mapeamento da solução do registro recuperado para o registro da situação corrente.

Quando a resolução do registro selecionado não pode ser diretamente aplicada para o problema, o sistema CRITTER aplica sobre ela uma estratégia de adaptação, possuindo três estratégias utilizáveis [LEW 93]. Uma primeira estratégia é a adaptação parametrizada, descrita no capítulo 2. Com esse método, a variável da solução do registro corrente é ajustada de modo relativo às variáveis do problema, usando para isso a relação entre a variável solução e as variáveis do problema do caso recuperado. Observando o exemplo da figura 3.6, com as demais variáveis sendo iguais, para um problema corrente $vazão_transferência_arquivo=F'$, seria proposto a resolução $ajustar_carga_rede=A'$ (b), onde F' e A' teriam a mesma relação que F e A do problema recuperado (a). Nesse exemplo, há vários modos de representar a função f . O modo mais simples seria uma tabela em que os valores de F que não estão na tabela seriam calculados por interpolação. Um outro modo seria representar a função utilizando inferências de lógica *fuzzi* [LEW 92]. Outros tipos de problema poderiam utilizar uma seqüência de passos ou uma árvore de decisão [LEW 93].

Caso Recuperado:

problema: vazão_transferência_arquivo=F

dados adicionais: nenhum

solução: $A=f(F)$, ajustar_carga_rede=A

estado da solução: bom

(a)

Situação corrente:

problema: vazão_transferência_arquivo=F'

dados adicionais: nenhum

solução: $A=f(F)$, ajustar_carga_rede=A'

estado da solução: bom

(b)

FIGURA 3.6 - Exemplo de adaptação parametrizada

Uma segunda estratégia que é utilizada para a adaptação é o método da abstração ou reinstanciação. Com essa técnica, se há uma restrição proibitiva em uma solução proposta, o sistema abstrai o registro que contém a solução e se volta para um registro que contém uma solução alternativa. Um exemplo dessa estratégia pode ser visto considerando-se novamente o problema da figura anterior (3.6 b), e supondo-se que tenham sido recuperados dois casos igualmente similares para o problema — o caso exemplificado na figura anterior (3.6 a) e o caso exemplificado na figura a seguir (3.7 a). Se uma restrição para uma possível resolução de um registro corrente é

'*não_ajustar_carga_rede*', ou se a execução desta função não resolve o problema, então o sistema propõe '*aumentar_largura_de_banda*' e acrescenta um novo registro a base conforme mostrado na figura a seguir (3.7 b) [LEW 93].

Caso Recuperado:

problema: vazão_transferência_arquivo=F

dados adicionais: nenhum

solução: $B=g(F)$, aumentar_largura_de_banda=B

estado da solução: bom

(a)

Caso aprendido:

problema: vazão_transferência_arquivo=F

dados adicionais: ajustar_carga_rede=não

solução: $A=f(F)$, ajustar_carga_rede=A

estado da solução: não bom

(b)

FIGURA 3.7 - Exemplo de adaptação por abstração

Por fim, o sistema pode utilizar também a adaptação baseada em crítica, que corresponde à adaptação realizada quando um especialista repara a solução recuperada, de modo que esta seja aplicada ao caso corrente. Exemplos dessa forma de adaptação são adicionar, remover, reordenar e substituir passos na solução recuperada. Os registros cuja solução sofreram essa forma de adaptação devem ser, então, armazenados na base de casos, de forma que seu conhecimento seja incorporado.

O sistema CRITTER está envolvido em uma arquitetura conforme apresentado na figura 3.8 [LEW 93]. O sistema SPECTRUM provê as funções de gerenciamento de configuração e detecção de falhas, enquanto que o AR System provê as funções de gerenciamento de falhas [CAB 97]. O sistema CRITTER é resultado da adição do componente de resolução de falhas CBR para o sistema de gerenciamento de falhas (AR System).

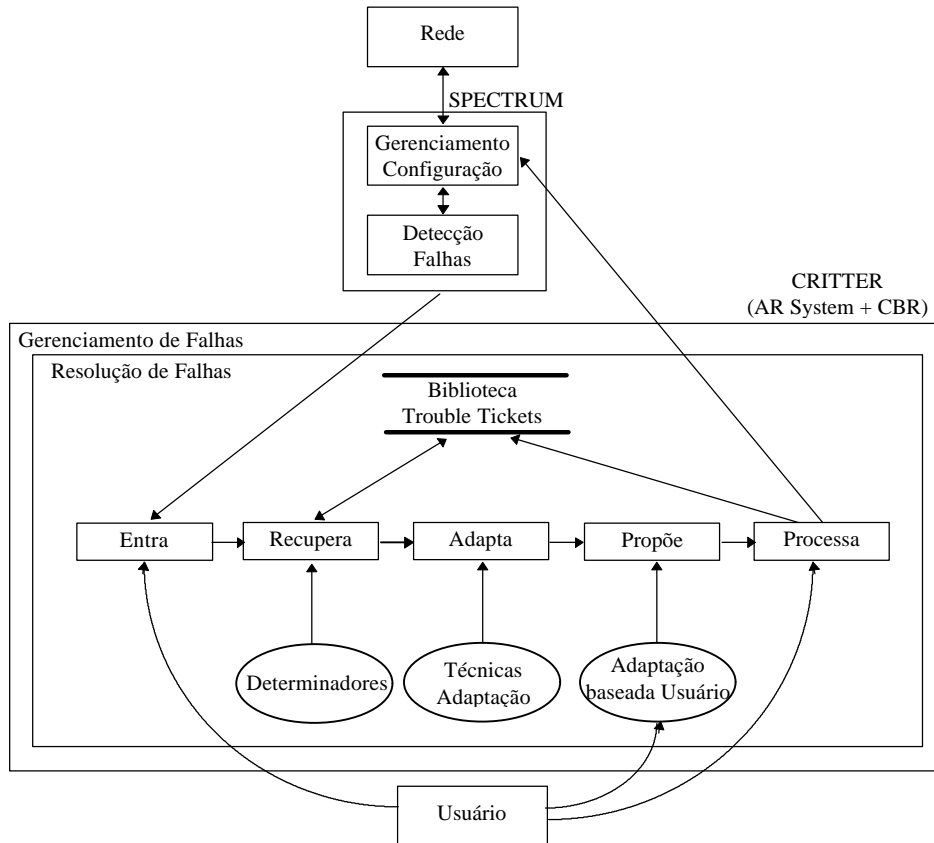


FIGURA 3.8 - Arquitetura do sistema CRITTER

O processo inicial envolvido no sistema CRITTER obtém informações sobre o problema corrente na forma de um registro de problema, através do módulo Entra. Essas informações podem ser obtidas de modo manual por um usuário ou automaticamente por uma aplicação que une o sistema CRITTER ao SPECTRUM. A recuperação é realizada pelo módulo Recupera de modo transparente ao usuário, fazendo uso dos determinadores para recuperar o conjunto de registros armazenados que são similares ao problema corrente. O conjunto inicial das regras de determinação do sistema podem ser fornecidos por especialistas do domínio ou serem extraídos de um documento de diagnóstico. Estas regras não são, porém, perfeitas, sendo melhoradas com o uso do sistema.

Uma vez que os registros similares foram recuperados, o registro com maior similaridade é selecionado pelo módulo de Adaptação. Se este registro possui um casamento total com todos os campos relevantes, então a resolução é mantida igual. Caso contrário, o módulo aplica a técnica de adaptação parametrizada ou a adaptação por abstração, comentadas anteriormente. As soluções potenciais encontradas pelo sistema são apresentadas ao usuário pelo módulo Propõe, e o usuário pode inspecionar e manualmente adaptar as soluções. Por fim, o registro é armazenado na base de casos pelo módulo Processa. Além disso, esse módulo pode enviar instruções ao sistema SPECTRUM, tornando o gerenciamento um ciclo fechado e fazendo o gerenciamento de falhas de modo completamente automatizado.

Como pode ser visto na figura, o usuário pode interagir em três pontos da arquitetura: pode filtrar os registros de problemas que são submetidos ao sistema CRITTER, pode adaptar as soluções propostas através do módulo Propõe e pode regular as instruções que serão enviadas para o sistema SPECTRUM.

3.6 Considerações Finais

Este capítulo apresentou os principais tópicos de gerência de redes e comentou alguns exemplos do uso de raciocínio baseado em casos — já apresentado no capítulo 2 — neste domínio.

Apresentaremos, a seguir, o modelo proposto neste trabalho — um modelo que utiliza raciocínio baseado em casos sobre um sistema de registro de problemas para diagnóstico de falhas ocorridas em uma rede. O modelo será introduzido no capítulo 4 e descrito no capítulo 5, sendo o seu protótipo comentado no capítulo 6.

4 Raciocínio Baseado em Casos Aplicado a um Sistema de Registro de Problemas: Sistema DUMBO

4.1 Motivação

Comentamos, no capítulo anterior, a ampla utilização dos sistemas de registro de problemas no gerenciamento de falhas em redes de computadores [JOH 92][MAD 94]. Esses sistemas mantêm o rastro do ciclo de vida dos problemas enfrentados, armazenando, assim, uma memória histórica dos problemas do domínio de gerência.

Como grande parte dos problemas que ocorrem em uma rede constituem repetição de uma situação similar já ocorrida no passado, tais sistemas são muito utilizados para acumular o conhecimento derivado do processo de diagnóstico e resolução dos problemas anteriores. Todavia, a consulta manual por uma solução de uma situação similar em sistemas de registro com número grande de ocorrências acaba por se tornar muito morosa e imprecisa.

Assim, uma solução apropriada para consolidar a memória histórica de um domínio de gerência é a criação de um sistema especialista que faça uso do conhecimento armazenado nestes sistemas para propor soluções na ocorrência de um novo incidente. Como foi comentado no capítulo anterior, existem diversos paradigmas que têm sido utilizados no desenvolvimento de sistemas especialistas na área de gerência de redes em geral.

O raciocínio baseado em regras tem sido a tecnologia mais utilizada [GOY 91]. Os sistemas com este paradigma possuem, porém, algumas limitações, especialmente quando utilizados em domínios com alta taxa de mudanças e evolução [LEW 93]. Uma destas limitações diz respeito ao gargalo apresentado na aquisição do conhecimento [GOY 91][LEW 93]. Esse problema pode ocorrer quando um engenheiro de conhecimento tenta identificar regras e procedimentos de controle que irão abranger situações imprevisíveis e o sistema acaba por se tornar imprevisível, difícil de manejar e de manter [LEW 93]. Razões para tal problema ocorrer incluem dificuldade dos especialistas em articular o raciocínio que utilizaram, dificuldade dos especialistas em lembrar de detalhes de como solucionar um problema em particular até que o tenham feito, dificuldade em prever os problemas que podem ocorrer [LEW 95].

Outra limitação de tais sistemas se relaciona à sua fragilidade quando uma situação nova é apresentada. Essa limitação está relacionada à falta de habilidade desses sistemas em adaptar o conhecimento existente para novas situações e de aprender com a experiência [LEW 93]. Como aponta Lewis, os seres humanos aumentam claramente suas habilidades de diagnóstico com a experiência [LEW 95]. Sistemas especialistas tradicionais são, entretanto, difíceis de serem projetados para aprender novas soluções. Aqueles sistemas especialistas tradicionais que são desenvolvidos com o cuidado de tratar de uma situação bem compreendida e relativamente constante, em um domínio que apresenta poucas surpresas e que o conhecimento necessário para resolver os problemas é na maior parte rígido e fixo, são bem sucedidos e provêem um serviço útil. Mas, como comenta Lewis, “*infelizmente, muito poucas tarefas no gerenciamento das redes de hoje são assim*” [LEW 95]p.34. Nas redes de hoje, novas aplicações e usuários são introduzidos naturalmente, os componentes são atualizados rotineiramente para inserção

de novas funcionalidades, novas tecnologias são adicionadas às redes existentes. E essas evoluções e mudanças nas redes impõem uma grande sobrecarga aos gerentes de redes e aumenta a dificuldade de construir um sistema especialista para auxílio das tarefas de diagnóstico [LEW 95]. Assim, em domínios com rápidas mudanças, como ocorre em muitas tarefas do gerenciamento de redes, esses sistemas podem acabar por se tornarem obsoletos rapidamente [LEW 93].

Outro paradigma que pode ser utilizado para o diagnóstico de problemas é o raciocínio baseado em modelos, também comentado anteriormente. Sua aplicação para o domínio de gerência de redes sofre, entretanto, pela dificuldade de modelar uma rede completa, com suas interações, em uma aplicação [GOY 91].

Assim, uma abordagem alternativa para sistemas especialistas em gerência de redes é o raciocínio baseado em casos, apresentado no capítulo 2. Esse paradigma traz benefícios como a capacidade de aprender com a experiência naturalmente com a utilização do sistema e evitar a manutenção excessiva [LEW 93].

Quando, porém, é utilizado junto a sistemas de registro de problemas, em que a estrutura de um registro já é semelhante à de um caso, soma-se a essas vantagens o fato de tais sistemas já representarem tecnologia consolidada na área. Isso proporciona uma utilização e aprendizado mais natural da aplicação CBR, já que esta se integra às facilidades colaborativas inerentes aos sistemas de registro aceitos e utilizados normalmente nos centros de gerência. Assim, o uso de raciocínio baseado em casos junto a um sistema de registro de problemas é um meio bastante adequado para utilizar o conhecimento armazenado nestes sistemas.

A partir do estudo dos sistemas de registro apresentado na seção 3.3 e do estudo de raciocínio baseado em casos comentado no capítulo 2, é fácil conceber um registro de problema como um caso, e a base de registros de problemas como a base de casos do sistema. Portanto, a fim de desenvolver uma aplicação de raciocínio baseado em casos sobre um sistema de registro de problemas tradicionais, é necessário acrescentar à arquitetura destes sistemas os processos de raciocínio e acrescentar sobre os registros de problemas as informações adicionais necessárias para que esses processos possam ser executados.

Este trabalho apresenta um sistema de raciocínio baseado em casos que foi desenvolvido a partir da arquitetura e tecnologia de sistemas de registro de problemas, denominado sistema DUMBO. Neste capítulo, abordaremos o processo de aquisição de conhecimento efetuado para o desenvolvimento deste sistema. No capítulo 5, abordaremos a modelagem do sistema, apresentando como a representação do conhecimento e os processos de raciocínio foram abordados. Por fim, no capítulo 6, apresentaremos o protótipo do sistema implementado.

4.2 Aquisição do Conhecimento

O processo de aquisição do conhecimento para a modelagem do sistema foi realizado utilizando-se entrevistas com especialistas, análise dos registros de problemas armazenado no sistema de registro de problemas CINEMA [TAR 96][TAR 96a] — desenvolvido e utilizado pelo grupo do centro de gerência do POP-RS — e por análise da bibliografia. Essa análise bibliográfica compreendeu estudo de referências teóricas e de situações problemas relatadas por livros de *troubleshooting* [MIL 96]

[MIL 95][NAS 94], por manuais de *troubleshooting* de fabricantes de equipamentos de rede [CIS 97][CIS 97a][CIS 97b][CIS 97c], de casos problemas fornecidos por fabricantes [CAB 97a][CAB 97b], de referências teóricas de redes de computadores e gerência de redes [HUN 98][NEM 95][SIN 96][STA 97] e de projetos de sistemas especialistas desenvolvidos na área [NUN 97][HAR 97].

As entrevistas foram realizadas com especialistas do centro de gerência do POP-RS e do centro de gerência da rede do Instituto de Informática/UFRGS. O ambiente e tecnologias das redes gerenciadas por esses centros caracterizam o ambiente para o qual o modelo foi concebido e onde será validado — redes TCP/IP em ambiente Unix, com presença de redes Ethernet e de linhas seriais.

Este estudo foi realizado enfocando de modo especial, portanto, ambientes com as tecnologias presentes nessas redes. Entretanto, ele foi concebido objetivando facilitar sua adaptação e portabilidade para redes com outras tecnologias, variando a facilidade desta tarefa de acordo com as características do ambiente adotado. Assim, existem algumas novas tecnologias, como tecnologias de hardware e acesso ao meio e novos tipos de equipamentos de interconexão, que podem ser acrescentadas automaticamente com o uso do sistema.

Do estudo realizado, foram identificados os tipos de problemas típicos que podem ser encontrados em redes TCP/IP. Esses problemas abrangem falhas na comunicação entre nodos, em aplicações e nos demais serviços fornecidos pelas redes, podendo se manifestar com a total interrupção do serviço ou na qualidade oferecida por este. O estudo enfocou a identificação dos sintomas gerados pelos diversos problemas, ambiente e contexto em que podem acontecer e que outros problemas poderiam fornecer informações que auxiliassem no diagnóstico daqueles problemas.

Na seção a seguir, comentaremos brevemente sobre a atividade de diagnóstico de problemas e apresentaremos algumas situações de problemas típicos que podem ser encontrados nas redes TCP/IP. Para uma análise mais aprofundada dos problemas citados, remetemos o leitor à bibliografia referenciada anteriormente.

4.3 Problemas Típicos das Redes Pesquisadas

As tarefas de gerenciamento de falhas são fundamentais para manter um serviço de rede confiável e estável. Entretanto, os problemas em redes são freqüentemente difíceis de serem solucionados. O diagnóstico efetivo de problemas requer uma abordagem metódica do problema, além de um entendimento de como a rede funciona — incluindo como os dados são roteados através da rede entre estações individuais e entre os níveis da pilha de protocolos [HUN 98].

Como aponta Craig Hunt, os problemas não são todos iguais e não podem, portanto, ser abordados da mesma maneira [HUN 98]. Mas o entendimento de qual problema está ocorrendo é a chave para sua solução. Um problema superficial é algumas vezes enganoso e o problema real pode estar obscurecido por muitos níveis de software. Assim, uma vez que a natureza real do problema for identificada, a solução para ele se torna freqüentemente óbvia.

O primeiro passo para uma atividade de diagnóstico de um problema é a obtenção de informações detalhadas sobre o que está exatamente ocorrendo, a fim de identificar e eliminar causas potenciais do problema. Exemplos de tais informações

incluem a identificação da aplicação que está com falha, o nome e endereço IP da estação remota, o nome e endereço IP da estação do usuário (local), as mensagens informadas [HUN 98]. O correto e amplo entendimento do problema também auxilia a determinar quais testes e dados adicionais serão necessários para o diagnóstico [SIN 96].

Alguns sintomas que devem ser observados no problema corrente são [HUN 98] [SIN 96]:

- O problema ocorre em apenas uma aplicação na estação do usuário? Se for apenas uma aplicação, a aplicação pode estar mal configurada ou desabilitada na estação remota.
- O problema ocorre somente com uma estação remota, todas as estações remotas, ou somente certos grupos de estações remotas? Se somente uma estação remota está envolvida, o problema pode facilmente estar nesta estação. Se todas as estações estão envolvidas, o problema provavelmente está no sistema do usuário (particularmente se nenhuma outra estação na sua rede local está experimentando o mesmo problema). Se somente estações em certas sub-redes ou redes externas estão envolvidas, o problema pode estar relacionado a roteamento.
- O problema ocorre em outras estações da mesma sub-rede? Se ele ocorre somente na estação do usuário, os testes devem ser concentrados nesta estação, no cabeamento e nos concentrados aos quais a estação está ligada. Se ele afeta todas as estações na sub-rede, a atenção deve ser concentrada no roteador da sub-rede, no servidor, nos equipamentos de interconexão ao longo caminho para o servidor ou estação destino.
- O problema impede totalmente os serviços da rede ou é um problema de performance que está causando atrasos e interrupções?
- Há algum tráfego nos canais de comunicação? Pouco ou nenhum tráfego podem indicar um componente maior com falha. O tipo de tráfego inexistente pode também auxiliar na identificação da causa do problema.
- Há mensagens de erro? As mensagens de erros devem ser observadas com atenção, pois podem conter informações importantes para a resolução da situação.
- Houve modificações na rede? Muitas falhas em redes ocorrem após modificações ou mudanças na topologia, na configuração, aplicações e usuários. Novos componentes inseridos, novas versões de aplicações e sistemas de gerenciamento, reconfiguração de componentes de conexão como roteadores, *bridges* (pontes) e *gateways* são exemplos de mudanças que podem ser relacionadas ao problema.

Existem, ainda, alguns procedimentos que podem auxiliar na correta abordagem da situação. Entre eles, destacamos [HUN 98]:

- Os problemas devem ser trabalhados cuidadosamente, sendo divididos em partes gerenciáveis. Cada parte deve ser testada antes de ser abordada a próxima. A divisão da rede em segmentos LAN, MAN ou WAN, por

exemplo, é essencial para tornar um problema gerenciável [SIN 96]. Nessa abordagem, cada segmento pode então ser examinado e pode ser excluído ou identificado como causador do problema — é raro dois segmentos falharem simultaneamente [SIN 96][MIL 96].

- Devem ser mantidos bons registros de testes que foram executados e seus resultados. Registros históricos de problemas devem também ser mantidos para ocorrências que se repetem.

Muitas falhas em redes são diretamente relacionadas aos seus componentes. Exemplos de componentes que podem estar envolvidos são [SIN 96]:

- cabeamento, conectores, concentradores, adaptadores de rede, placas de rede;
- pontes, *switches*, roteadores e *gateways*;
- servidores, estações de trabalho.

As diversas funções exercidas por estes componentes, assim como os diversos protocolos associados, são enquadrados nas diversas camadas da Arquitetura TCP/IP [COM 91] que compõem a arquitetura do ambiente. Quando há, por exemplo, problemas na comunicação de uma estação para outra numa rede, a causa do problema pode estar em qualquer ponto no caminho da comunicação entre as estações, podendo envolver equipamentos que tratam os dados apenas fisicamente, como repetidores e concentradores; equipamentos que tratam os dados segundo o hardware de rede, como pontes e placas de rede; equipamentos encarregados do roteamento dos pacotes, como roteadores; equipamentos que tratam os dados também no nível de aplicação, como estações de trabalho e servidores. Esses problemas podem envolver falhas físicas nos diversos componentes, como falhas no hardware de placas de rede e cabeamento, e falhas nos diversos protocolos e softwares envolvidos, como incompatibilidade de protocolos, roteamento, controle de fluxo, etc., que se enquadram nos diversos níveis do modelo.

Para permitir a melhor apresentação e análise dos problemas, iremos abordá-los examinando cada uma das camadas do modelo. Assim, comentaremos brevemente a seguir cada um dos problemas típicos e suas características nas camadas do modelo TCP/IP [COM 91]: camada interface de rede, camada internet, camada de transporte e camada de aplicação.

4.3.1 Camada Interface de Rede

A camada interface de rede conecta a estação local ao hardware da rede local, fazendo a conexão física ao cabeamento do sistema, acessando o cabeamento no momento apropriado e transmitindo os dados em um frame no formato do hardware de rede [MIL 96]. A configuração do hardware sobre o qual os protocolos TCP/IP podem operar incluem opções para LAN, MANs e WANs. Como aponta Mark Miller [MIL 96], a resolução de problemas de hardware — isto é, no nível de Interface de Rede — consome tanto tempo, se não mais, que os problemas nos níveis superiores.

As falhas na camada interface de rede são, juntamente com as falhas na camada internet, responsáveis pelos problemas com a conectividade da rede. O comando *ping* é exemplo de uma função que pode ser usada para testar a conexão da rede, permitindo

que seja determinado se o diagnóstico da falta de conectividade em uma aplicação deve ser direcionado para a própria conexão da rede, englobando os níveis inferiores, ou se deve focar a aplicação em que o problema foi detectado, nos níveis superiores.

Na camada interface de rede, o diagnóstico pode ser auxiliado, por exemplo, com a verificação da configuração do estado da interface (comando *ifconfig*, por exemplo), que permite identificar interfaces desabilitadas que não puderam ser ativadas por problemas físicos na interface. Outras informações que podem auxiliar nesta etapa são o número de pacotes enviados e recebidos, erros de entrada e de saída e taxa de colisão, entre outras, que podem ser verificadas, em sistemas Unix, através do comando *netstat*. Alguns exemplos de problemas que podem ser detectados através de comandos como este são mostrados na tabela abaixo.

TABELA 4.1 - Alguns problemas da camada interface de rede

| Causas | Possíveis Sintomas | Referências |
|--|--|----------------------|
| <ul style="list-style-type: none"> • cabeamento partido • interface defeituosa | <ul style="list-style-type: none"> • estado interface <i>up</i> e <i>running</i> mas o sistema não envia pacotes para a rede (ocorrência de pacotes na fila da interface). | [HUN 98] |
| <ul style="list-style-type: none"> • má conexão física, problemas físicos na rede | <ul style="list-style-type: none"> • alta taxa de erros de saída e • alta taxa de erros de entrada | [HUN 98] |
| <ul style="list-style-type: none"> • meio saturado | <ul style="list-style-type: none"> • alta taxa de erros de saída na interface • alta taxa de erros de entrada na interface • alta taxa de colisões (Ethernet) | [HUN 98] [MIL 96] |
| <ul style="list-style-type: none"> • estação local saturada | <ul style="list-style-type: none"> • alta taxa de erros de entrada | [HUN 98] |

Nos problemas dessa camada, porém, é importante focar a tecnologia de rede envolvida, já que a topologia de cada tipo de rede tem seu próprio conjunto de erros [NAS 94]. O próprio tipo de cabeamento utilizado pode trazer diferentes tipos de problemas. Apresentaremos, a seguir, alguns exemplos de erros frequentes presentes em algumas tecnologias de rede, enfocando as redes selecionadas para a validação do sistema — Ethernet, para rede local, e linhas seriais — que são as tecnologias de rede presentes na rede da UFRGS onde o sistema será validado.

Como na tabela anterior, a primeira coluna de cada uma das tabelas indica algumas causas de problemas em redes, enquanto a segunda coluna apresenta alguns possíveis sintomas para os problemas. Nem todos os sintomas abordados, entretanto, ocorrem sempre para os problemas apresentados. Exemplo é uma falha em uma placa de interface de rede, que pode ser responsável por um grupo de diferentes tipos de erros [NAS 94]. Maiores informações sobre cada problema podem ser obtidas nas referências correspondentes, indicadas na última coluna. Características adicionais aos sintomas apresentados que podem auxiliar no diagnóstico do problema são a sua classificação em constante ou intermitente, se componentes ou softwares foram modificados ou inseridos e a abrangência do problema na sub-rede e no segmento [NAS 94].

4.3.1.1 Ethernet

As causas de problemas em redes Ethernet incluem problemas de projeto do cabeamento, configuração do nodo, componentes defeituosos e configuração dos equipamentos de interconexão envolvidos, que podem atingir apenas um nodo ou todos os nodos da rede. A tabela a seguir exemplifica alguns destes problemas.

TABELA 4.2 - Alguns problemas típicos de redes Ethernet

| Causas | Possíveis Sintomas | Referências |
|---|---|----------------------|
| <ul style="list-style-type: none"> • cabos Ethernet sem terminação | <ul style="list-style-type: none"> • alta taxa de colisões | [CIS 97a] |
| <ul style="list-style-type: none"> • cabos Ethernet com comprimento excessivo • número excessivo de repetidores na rede | <ul style="list-style-type: none"> • colisões tardias | [CIS 97a] |
| <ul style="list-style-type: none"> • uso de cabeamento incorreto: 10BaseT4 com somente 2 pares de fios disponíveis, erro em 10BaseT, 100BaseT4 e 100BaseTx (por exemplo, placa de rede diferente da porta em um hub) | <ul style="list-style-type: none"> • ausência de integridade no enlace em 10BaseT, 100BaseT4 ou 100BaseTX | [CIS 97a] |
| <ul style="list-style-type: none"> • incompatibilidade entre IEEE 802.3 e Ethernet | <ul style="list-style-type: none"> • ausência de acesso para nodos configurados diferentes, estações recém configuradas | [MIL 96] [SIN 96] |
| <ul style="list-style-type: none"> • software defeituoso na placa de rede | <ul style="list-style-type: none"> • taxa excessiva de frames pequenos, sem alta taxa de colisões simultânea | [CIS 97a] |
| <ul style="list-style-type: none"> • cabeamento defeituoso associado a um nodo Ethernet (falhas na placa de rede, transceiver, conectores) | <ul style="list-style-type: none"> • ocorrência de falha em um único nodo Ethernet • alta taxa de colisões locais (transceiver, cabeamento) • colisões remotas (placa de rede ou transceiver de segmento remoto) • pacotes longos e curtos (placa de rede, transceiver) • jabber (placa de rede, transceiver) • alta taxa de CRC (conectores, transceiver, placa de rede) | [NAS 94] [MIL 96] |
| <ul style="list-style-type: none"> • alta taxa de colisões | <ul style="list-style-type: none"> • taxa excessiva de frames pequenos | [CIS 97a] |

| Causas | Possíveis Sintomas | Referências |
|--|--|-------------|
| <ul style="list-style-type: none"> • ruído excessivo no meio, causado por cabeamento defeituoso, pancadas espaçadas causando reflexões | <ul style="list-style-type: none"> • muitos erros de CRC com baixa taxa de colisões • o tipo de cabeamento utilizado pode ser relevante para a solução: por exemplo, em 100BaseTX, verificar a utilização de UTP Cat. 5 | [CIS 97a] |
| <ul style="list-style-type: none"> • porta de repetidor ou hub com falha | <ul style="list-style-type: none"> • ocorrência de falha associada a uma porta específica do equipamento — no segmento específico, nodos não operam ou congelam e penduram durante operação | [NAS 94] |
| <ul style="list-style-type: none"> • repetidor ou hub com falha | <ul style="list-style-type: none"> • ocorrência de falha em múltiplas portas ou com toda a rede Ethernet — nodos não operam ou congelam durante operação • tráfego pode não ser transmitido de um lado para o outro do equipamento • alta taxa de colisões e CRC de múltiplos nodos da rede | [NAS 94] |
| <ul style="list-style-type: none"> • configuração de hardware ou software de uma ponte incorreto | <ul style="list-style-type: none"> • nenhum tráfego atravessa a ponte ou dados são corrompidos intermitentemente quando atravessam o equipamento | [NAS 94] |
| <ul style="list-style-type: none"> • configuração de parâmetros que controlam a quantidade de tráfego a ser enviada na ponte incorretos | <ul style="list-style-type: none"> • tráfego excessivo presente em um certo segmento • broadcast storms excessivas vindo da ponte | [NAS 94] |
| <ul style="list-style-type: none"> • equipamentos conectados a um nodo Ethernet (como modems e impressoras sem sua própria placa de rede) ou nodo Ethernet ao qual são ligados com hardware ou software mal configurados • cabeamento defeituoso entre o nodo Ethernet e o equipamento sem placa de rede | <ul style="list-style-type: none"> • equipamento conectado ao nodo Ethernet não pode ser acessado | [NAS 94] |

4.3.1.2 Linhas Seriais

As linhas seriais podem apresentar diversas falhas na conexão. A tabela abaixo apresenta alguns dos problemas que podem ocorrer nestes enlaces. Além dos sintomas apresentados, uma característica que pode auxiliar no diagnóstico desses incidentes é o estado da interface. Testes de loop local e remoto são também ferramentas poderosas para isolar a origem do problema.

TABELA 4.3 - Alguns problemas típicos de linhas seriais

| Causas | Possíveis Sintomas | Referências |
|---|--|------------------------|
| <ul style="list-style-type: none"> • equipamentos da companhia telefônica defeituosos • linha serial com ruído • cabos incorretos ou longos demais | <ul style="list-style-type: none"> • conectividade intermitente • erros de entrada de CRC, <i>framing</i> e <i>aborts</i> na transmissão | [CIS 97c] |
| <ul style="list-style-type: none"> • tráfego de entrada na interface serial excede a largura de banda disponível | <ul style="list-style-type: none"> • descartes de saída em taxa crescente | [CIS 97c] |
| <ul style="list-style-type: none"> • tráfego na entrada excede a capacidade do roteador • filas de entrada excedem o tamanho das filas de saída | <ul style="list-style-type: none"> • descartes na entrada em número crescente • ocorre, geralmente, quando o tráfego está sendo roteado entre interfaces rápidas (Ethernet, FDDI, Token Ring) e interfaces seriais, e há presença de taxa de tráfego elevada | [CIS 97c] |
| <ul style="list-style-type: none"> • congestionamento no enlace | <ul style="list-style-type: none"> • conectividade intermitente, falhando nos períodos de pico • taxa de erros de entrada baixa • ocorrência de resets na interface em número crescente | [CIS 97c] [CIS 97a] |
| <ul style="list-style-type: none"> • problemas de hardware no CSU/DSU, ou no enlace | <ul style="list-style-type: none"> • conectividade intermitente • ocorrência de resets na interface em número crescente • alta taxa de erros de entrada | [CIS 97a] [CIS 97c] |
| <ul style="list-style-type: none"> • interrupções na linha causadas por origem externa (separação física do cabeamento, relâmpagos em algum ponto da rede) | <ul style="list-style-type: none"> • contador das transições na portadora crescente | [CIS 97a] |
| <ul style="list-style-type: none"> • pacotes <i>keepalive</i> não sendo recebidos | <ul style="list-style-type: none"> • ausência de conectividade | [CIS 97a] |

4.3.2 Camada Internet

A camada internet do modelo TCP/IP é análoga à camada de rede do modelo OSI, sendo responsável pela entrega dos dados de sua origem ao seu destino através de uma rede internet. Esta rede pode incluir a Internet, uma rede corporativa ou os meios de transmissão de LANs e WANs.

Uma das principais funções da camada internet é o roteamento. O roteamento possibilita a conectividade entre duas estações através de uma internet, utilizando o endereço lógico da estação que a identifica unicamente dentro da internet — o endereço IP. Para isso, são utilizados roteadores que guiam os pacotes ao destino, identificando a rota apropriada para os pacotes, através da comparação das informações de endereçamento dentro dos pacotes com as informações que eles possuem sobre a topologia da rede. As informações de endereçamento da topologia de rede são trocadas entre os roteadores, através de protocolos de roteamento, como o RIP, OSPF, etc. Associado ao roteamento, estão questões de endereçamento, definição de sub-redes e máscaras [MIL 96], entre outros. Protocolos como ARP/RARP, BOOTP estão também associados à camada.

Um protocolo que auxilia o diagnóstico de problemas de conectividade é o ICMP. Uma das mensagens desse protocolo é a utilizada pelo comando *ping*, a mensagem ICMP Echo, que pode ser usada de modo seqüencial para isolar um problema. Outra ferramenta que pode ser também utilizada é o *traceroute*, que utiliza mensagens ICMP para verificar cada sub-rede ao longo de uma rota para uma estação distante.

Assim, se problemas de conectividade em uma internet não são causados por problemas no hardware e sua configuração (camada interface de rede), as falhas devem ser buscadas na camada internet. O processo de identificação de um endereço, atribuição de endereços, comunicação entre roteadores e notificações de erros dos roteadores podem oferecer indícios sobre a causa do problema na entrega dos datagramas [MIL 96].

TABELA 4.4 - Alguns problemas típicos da camada internet

| Causas | Possíveis Sintomas | Referências |
|---|---|------------------------|
| <ul style="list-style-type: none"> rota <i>default</i> não especificada ou incorreta | <ul style="list-style-type: none"> ausência de conectividade para estações remotas | [CIS 97a] |
| <ul style="list-style-type: none"> <i>gateway default</i> não especificado ou incorreto em estação local ou remota | <ul style="list-style-type: none"> ausência de conectividade para estações de sub-redes remotas pode haver conectividade para algumas sub-redes remotas, enquanto para outras não há acesso | [CIS 97a] [CIS 97c] |
| <ul style="list-style-type: none"> má configuração no roteador, com comandos ausentes ou incorretos (tais como identificador e endereço IP para uma interface do roteador) | <ul style="list-style-type: none"> rotas são perdidas de uma tabela de roteamento estações em uma rede não podem acessar estações em outra | [CIS 97a] |
| <ul style="list-style-type: none"> tabelas de roteamento em roteadores corrompidas | <ul style="list-style-type: none"> métricas incorretas ausência de conectividade | [MIL 96] |

| | | |
|--|--|--------------------------------|
| <ul style="list-style-type: none"> • listas de acesso ou outros filtros mal configurados | <ul style="list-style-type: none"> • conectividade em algumas aplicações ou protocolos, enquanto outros falham [CIS 97a] • conectividade para algumas sub-redes remotas, enquanto outras falham (com listas de acesso atingindo informações de roteamento para algumas rotas, mas não para outras) [CIS 97c] • conectividade para algumas estações remotas, enquanto para outras há falha | <p>[CIS 97a] [CIS 97c]</p> |
| <ul style="list-style-type: none"> • em rede com múltiplas rotas de uma sub-rede a outra há listas de acesso mal configuradas, que bloqueiam o acesso a uma das rotas, ou problemas com balanceamento de carga | <ul style="list-style-type: none"> • performance ruim, um dos canais não parece conter tráfego. Caracteriza, na verdade, um problema de conectividade | <p>[CIS 97c]</p> |
| <ul style="list-style-type: none"> • endereços IP duplicados numa mesma sub-rede | <ul style="list-style-type: none"> • perda de conexão de uma das estações, enquanto uma outra estação possui acesso | <p>[MIL 96]</p> |
| <ul style="list-style-type: none"> • roteador vê atualizações de roteamento duplicadas em duas interfaces, causada por ponte em paralelo com roteador, que faz com que atualizações e tráfego sejam vistos em ambos os lados de uma interface | <ul style="list-style-type: none"> • perda de conexão súbita e performance extremamente ruim | <p>[CIS 97c]</p> |
| <ul style="list-style-type: none"> • em um roteador que está redistribuindo rotas em diferentes domínios de roteamento (tipicamente, RIP e IGRP) há ausência da métrica <i>default</i>, problemas com a distância administrativa <i>default</i>, ausência de outros comandos necessários (como <i>redistribute</i>, <i>distribute-list</i>) | <ul style="list-style-type: none"> • tráfego não é enviado pelo roteador que está redistribuindo rotas em diferentes domínios de roteamento | <p>[CIS 97c]</p> |

| | |
|--|--|
| <ul style="list-style-type: none"> • máscara da sub-rede incorreta | <ul style="list-style-type: none"> • pacotes não são roteados corretamente [CIS 97a] • dependendo da topologia da rede, um roteador com máscara incorreta poderia enviar datagramas para um estação destino incorreta, para uma interface incorreta ou descartá-los. [CIS 97c] • uma estação com máscara incorreta poderia ter, dependendo da topologia, retardo no estabelecimento de conexão para estações de sua própria sub-rede, ou poderia não ter conectividade [MIL 96] • poderia haver ausência de conectividade para estações em sub-redes remotas (enquanto, em certas situações, outras estações poderiam ser acessadas) |
| <ul style="list-style-type: none"> • em uma nova interface em um roteador, há ausência de comandos necessários à configuração do roteador | <ul style="list-style-type: none"> • ausência de conectividade na nova interface [CIS 97c] |

4.3.3 Camada de Transporte

A camada de transporte tem como principal objetivo prover a transferência confiável de dados entre processos de diferentes estações pertencentes ou não à mesma rede [CAR 94]. Essa camada é a responsável por garantir que os dados sejam entregues livres de erro e em seqüência, sem perdas ou duplicação, liberando as camadas superiores da arquitetura TCP/IP da tarefa de gerenciar a infra-estrutura de comunicação utilizadas pelas aplicações.

A camada de transporte da Arquitetura TCP/IP possui dois tipos de protocolos: o protocolo TCP (*Transmission Control Protocol*), que garante a transferência confiável dos dados, e o UDP (*User Datagram Protocol*), que é considerado uma simples extensão do protocolo IP da camada Internet e que não oferece garantia de entrega de dados. A utilização do protocolo TCP ou UDP depende das necessidades da aplicação, tais como qualidade de serviço, *throughput*, tipo de dados [CAR 94].

Para o diagnóstico de problemas ocorridos na conectividade fim-a-fim (ou *host-to-host*), o primeiro passo consiste em determinar qual o protocolo de transporte utilizado. Se o protocolo for o UDP, deve ser verificado se o serviço sem conexão é adequado para a aplicação, e, se for, então problemas como múltiplas retransmissões devem ser resultantes dos protocolos da camada superior condizentes com o mecanismo de transporte. Se o protocolo utilizado for o TCP, o diagnóstico é mais complexo. Além de verificar o número da porta de serviço, eventos significativos na conexão TCP devem ser analisados, tais como o mecanismo *three-way handshake* e o encerramento da conexão [MIL 96].

Enfim, de modo geral, deve ser determinado se os dados transmitidos estão atingindo o seu destino. Se não estiverem, problemas na camada internet devem ser analisados. Se estiverem, o problema pode ser atribuído para a camada de transporte. Um analisador de protocolos pode também auxiliar nesse processo com a análise dos cabeçalhos TCP e UDP. A seguir, apresentamos alguns exemplos de problemas que podem ocorrer na camada de transporte, ou que podem ser aparentemente atribuídos a camada de transporte.

TABELA 4.5 - Alguns exemplos de problemas na camada de transporte

| Causas | Possíveis Sintomas | Referências |
|--|--|--------------------|
| <ul style="list-style-type: none"> condição <i>half-open TCP connection</i> e <i>reset</i> da conexão devido a problemas intermitentes da comunicação (causa real nas camadas inferiores, por exemplo, placa de rede com falhas intermitentes) | <ul style="list-style-type: none"> ausência de resposta na aplicação interrupção da conexão TCP | [MIL 96] |
| <ul style="list-style-type: none"> módulo TCP de equipamento incorreto, que interpreta <i>delay</i> solicitado por equipamento remoto como fim da conexão | <ul style="list-style-type: none"> falha súbita em aplicação com grande transferência de dados interrupção súbita da conexão TCP | [MIL 96] |
| <ul style="list-style-type: none"> buffers de entrada e saída com tamanho inadequado e com parâmetro limiar de descartes (número máximo de frames que podem ser armazenados na fila) inferior ao necessário em equipamentos de interconexão entre estações (por exemplo, em pontes) | <ul style="list-style-type: none"> estações comunicam com outras no mesmo segmento sem problema, enquanto a comunicação com estações em segmentos remotos ocorre com <i>delays</i> excessivos e interrupções em sessões FTP fim abrupto da conexão em conexões FTP | [MIL 96] |

4.3.4 Camada de Aplicação

A camada de aplicação situa-se no topo da Arquitetura TCP/IP. Diferente das demais camadas, que são transparentes para os usuários, a camada de aplicação é acessada pelos usuários diretamente via sistema operacional da estação [MIL 96]. Aplicações padronizadas da Arquitetura TCP/IP incluem correio eletrônico (implementado através do protocolo SMTP - *Simple Mail Transfer Protocol*), FTP (*File Transfer Protocol*), TELNET (Terminal Virtual), NFS (*Network File System*), DNS (*Domain Name System*) e SNMP (*Simple Network Management Protocol*) [CAR 94].

O diagnóstico de problemas na camada de aplicação começa com a verificação das funções de conectividade fim-a-fim, que devem estar transmitindo os dados para o destino solicitado; se não estiverem, falhas nas camadas inferiores à camada de aplicação

devem ser consideradas. Se, entretanto, os dados estão atingindo o destino, então problemas na camada superior podem existir [MIL 96].

Uma questão que deve ser lembrada é que os dados recebidos não são, necessariamente, interpretados corretamente (por exemplo, se um terminal está transmitindo caracteres ASCII e o outro está esperando caracteres EBCDIC). Assim, uma questão a ser analisada é se a aplicação está sendo utilizada adequadamente. Outra possibilidade é que os dois processos de aplicações não estejam aptos a comunicarem devido a diferenças de implementação internas [MIL 96].

A seguir, apresentamos alguns exemplos de problemas na camada de aplicação.

TABELA 4.6 - Alguns exemplos de problemas na camada de aplicação

| Causas | Possíveis Sintomas | Referências |
|---|--|-------------|
| <ul style="list-style-type: none"> utilizado um modo de transferência de dados incorreto em FTP ou TFTP | <ul style="list-style-type: none"> transferência de dados com erro, com arquivo transferido corrompido | [MIL 96] |
| <ul style="list-style-type: none"> erro de configuração na estação de trabalho, tais como erro no nome do servidor | <ul style="list-style-type: none"> uma estação pode enviar e-mails corretamente para outra (com configuração incorreta), que não consegue, entretanto, enviar resposta para a estação original | [MIL 96] |
| <ul style="list-style-type: none"> incompatibilidade entre formatos do tipo de terminal em TELNET | <ul style="list-style-type: none"> interrupção da conexão sendo estabelecida | [MIL 96] |
| <ul style="list-style-type: none"> alguns registros sendo recuperados do servidor de nomes primário pelos secundários estão corrompidos, devido a problemas na configuração do servidor primário | <ul style="list-style-type: none"> servidores de nomes secundários não podem resolver um <i>hostname</i> do domínio em que seu servidor é um secundário. O nome é resolvido pelo servidor primário. | [HUN 98] |

4.4 Considerações sobre os Problemas do Domínio

Como pode ser visto pela relação de problemas citados, o domínio de gerência de redes — mesmo considerando a limitação para o ambiente de redes TCP/IP — possui uma ampla gama de problemas possíveis de serem encontrados. Algumas dessas falhas são facilmente diagnosticadas e solucionadas, outras podem envolver uma série de ações até a correta compreensão do problema, cujos sintomas podem muitas vezes estar mascarando a problema real e apontando para um outro tipo de problema.

Os problemas podem atingir a própria conexão da rede, com interrupção ou má qualidade do acesso a nodos ou sub-redes remotas, ou podem atingir as aplicações ou serviços da rede que utilizam a conexão à rede proporcionada pelas camadas inferiores. Além disso, podem ser identificados problemas no tráfego presentes nos canais de

comunicação da rede, mesmo que ainda não tenham ocasionado problemas na conexão perceptíveis aos usuários em geral.

Assim, através do estudo dos problemas realizados, os problemas do domínio foram caracterizados como problemas de **conectividade**, **performance** da comunicação e **alto tráfego** nos canais, para os problemas na conexão das redes, além dos problemas de **aplicações** e serviços, sendo identificados de modo especial três serviços: **resolução de nomes**, **autenticação de usuários** e **servidor de arquivos**. Em virtude das diferentes situações de problemas de conectividade, foram ainda reconhecidos dois subtipos distintos: problemas de **conectividade físicos e de configuração do hardware** (englobando os problemas na camada interface de rede) e problemas de **endereçoamento e roteamento** (englobando as funções na camada internet).

Esses problemas não são mutuamente exclusivos: alguns problemas que ocorrem nas redes podem ser detectados como fazendo parte de mais de um dos tipos apresentados. Exemplo disso seria uma situação de conectividade intermitente, que poderia ser detectada como uma situação de alto tempo de resposta ou interrupção da comunicação (performance), e que poderia apresentar ainda alto tráfego nos canais. Modos de divisão dos problemas em tipos que não sejam relacionados e que possam ser identificados na visão dos problemas disponível aos usuários não foram identificados.

Além disso, as falhas reais podem estar mascaradas como outros tipos de problemas. Exemplos são uma ocorrência de performance ruim ou alto tráfego num canal causada pela interrupção de um canal de comunicação paralelo, problemas com um serviço de autenticação causado por falhas na resolução de nomes, problemas em uma aplicação ou serviço causada por falhas na conectividade.

Assim, como um problema pode ser enquadrado e informado em diferentes tipos, meios de relacionar estes problemas em sistemas são necessários, a fim de que uma situação vista de diferentes formas possa ser tratada corretamente. Os procedimentos concebidos para esse tratamento no sistema DUMBO serão apresentados no capítulo a seguir.

O estudo dos problemas permitiu, também, a identificação de informações relevantes para o diagnóstico de uma situação: algumas estão presentes em todos os incidentes enfrentados, tais como a abrangência do problema e sua localização; outras estão presentes em apenas um conjunto de incidentes, tais como a aplicação sendo utilizada, a tecnologia de rede, os protocolos envolvidos. Existem outras, ainda, que estão presentes apenas em casos particulares, como o uso de um determinado protocolo ou tecnologia na rede gerenciada, o resultado de um teste em particular, etc. Portanto, para a utilização destas informações na recuperação de situações similares anteriores, a identificação de uma listagem dessas informações é necessária, assim como meios de organizá-la e compará-la.

Apresentaremos, a seguir, o modelo proposto para o uso de raciocínio baseado em casos integrado a um sistema de registro de problemas. Enfocaremos, como já foi comentado, o ambiente de redes TCP/IP com sistemas Unix.

5 Sistema DUMBO: Modelagem

Como foi abordado no capítulo anterior, o conhecimento armazenado nos sistemas de registro de problemas pode ser reutilizado aplicando-se sobre ele técnicas de raciocínio baseado em casos. Desta forma, às funções dos sistemas de registro tradicionais — tais como manter a listagem dos problemas correntes do centro de gerência, enviar mensagens aos especialistas envolvidos, permitir a análise estatística da produtividade do domínio — são agregadas facilidades que auxiliam a resolução dos problemas correntes na inserção de um novo registro, através da recuperação de situações similares anteriores.

A fim de se tornarem amigáveis, esses sistemas devem manter a estrutura utilizada nos sistemas de registro de problemas tradicionais — já consolidada em centros de gerência, sendo nela adicionados os demais procedimentos de raciocínio baseado em casos. Assim, na inserção de um novo registro, além das informações tradicionais solicitadas aos usuários — tais como a prioridade do problema, quem o relatou e o responsável —, o sistema deve obter as informações adicionais necessárias para o processo de recuperação — tais como detalhes dos sintomas identificados e do ambiente em que o problema ocorreu. Com essas informações, mecanismos de raciocínio inseridos no sistema selecionam os registros das situações similares ocorridas no passado que podem contribuir para a solução do problema enfrentado e apresentam-nos ao usuário. Procedimentos adicionais responsáveis pelo aprendizado do caso ocorrido e já solucionado devem também ser acrescidos ao processo de encerramento de um problema dos sistemas tradicionais.

Este capítulo discorre sobre o modelo concebido para o sistema DUMBO. A representação do conhecimento, os procedimentos de raciocínio utilizados e a base de conhecimento do sistema são apresentados.

5.1 Sistema DUMBO - Uma Visão Geral das Técnicas Adotadas

Como foi visto, o uso de CBR em um sistema de registro de problemas exige mecanismos para decidir quais informações sobre o problema devem ser obtidas, quais procedimentos para recuperar registros similares armazenados devem ser utilizados e quais instrumentos para aprender um novo caso podem ser usados.

Inicialmente, é necessário preparar o registro para o diagnóstico, sendo obtidas informações acerca da situação — incluindo sintomas identificados (problema encontrado, taxa de erros da rede, abrangência do problema, equipamentos e redes afetadas, etc.) e ambiente da rede afetada (topologia da rede, protocolos utilizados, etc.), além das demais informações já utilizadas nos sistemas de registro de problemas tradicionais (apresentadas na seção 3.3). As informações acerca de um problema podem ser obtidas através de métodos automáticos diretamente da rede, de informações cadastradas, de uma plataforma de gerência associada ou diretamente do usuário.

Uma importante decisão nessa etapa é a definição da linguagem a ser utilizada nos registros. Como apontam Lewis e Dreo [LEW 93a], a linguagem utilizada deve levar em consideração que existem diferentes modos de expressar um problema. Isso visa evitar que duas descrições expressando de diferentes modos o mesmo problema não sejam relacionadas, como, por exemplo, com múltiplos registros dizendo em diferentes

palavras “*FTP não possibilitado*”. Assim, para propósitos práticos, uma descrição em linguagem natural deve ser evitada, devendo ser utilizada uma linguagem descritiva formal, bem estruturada [LEW 93a][LEW 95][LEW 93]. Podem ser utilizados para isto menus em cascata ou campos independentes para informações, tais como “problema”, “equipamento” [LEW 93], etc., sendo utilizado um campo adicional, em texto livre, para outras informações relevantes [LEW 95].

É possível, porém, utilizar de modo complementar a breve descrição do problema, presente nos sistemas de registro tradicionais, para o cálculo da similaridade do caso. Essa utilização já é feita em algumas aplicações CBR, seja auxiliando numa busca inicial para recuperar alguns casos e as características relevantes destes [ACO 92], seja auxiliando na recuperação principal [RAM 95][CHA 96]. Para isso, a descrição do caso corrente passa por um tratamento em que os principais termos são selecionados e é feito um casamento desses termos com as descrições dos casos armazenados [ACO 92][CHA 96]. Neste trabalho, é considerada também, de modo complementar às demais características, a similaridade das descrições no cálculo da similaridade entre casos, utilizando uma abordagem análoga à utilizada em [CHA 96].

O gerenciamento em redes de computadores envolve, entretanto, a análise e resolução de uma ampla variedade de problemas. Os problemas podem abranger tanto falhas nas camadas inferiores do modelo TCP/IP, englobando desde problemas físicos nos meios e equipamentos de transmissão até problemas de roteamento de pacotes, como podem envolver problemas em aplicações e nos serviços oferecidos pela rede [MIL 96] [NAS 94][CIS 97][HUN 98][TAR 96]. Não basta, portanto, para diagnosticar uma ocorrência, obter informações que sejam referentes a *todos* os problemas que podem ser encontrados em gerenciamento. Embora algumas informações digam respeito a todos os tipos de ocorrências, outras referem-se e são relevantes apenas para um determinado grupo ou tipo de problema, e devem ser elaboradas para que ocorrências similares possam ser recuperadas.

Lewis aponta que, para o desenvolvimento de um sistema CBR de registro de problemas de redes, deve ser elaborada uma lista dos principais tipos de problemas, a qual deve ser implementada no registro como um campo com uma lista de termos fixa [LEW 95]. Tipos de problemas não previstos deveriam ser também permitidos, devendo nesse caso o usuário informar o novo tipo de problema da situação corrente. Essa informação (tipo de problema), que representa o principal sintoma, já é solicitada em alguns sistemas de registro de problemas tradicionais, a exemplo do ambiente CINEMA [TAR 96], embora nesses sistemas a identificação dos problemas não necessite ser tão completa.

Uma vez que os tipos de incidentes prováveis para o domínio são identificados, para cada tipo de problema devem ser analisadas quais informações adicionais contribuiriam para determinar a solução da situação, e as informações relevantes devem ser obtidas [LEW 95]. Estas questões, assim como os meios de obtenção destas informações serão discutidos nas seções 5.4.1 e 5.5.1.

Esta abordagem apresenta, entretanto, algumas dificuldades, especialmente quando aplicada a sistemas de registro de problemas que não estejam ligados a uma estrutura de gerenciamento de falhas, e, por conseguinte, a criação de registros não seja feita de maneira automática, mas sim diretamente pelo usuário. Uma estrutura de gerenciamento de falhas (tal como a apresentada anteriormente na figura 3.8, por

exemplo) pode integrar diversos componentes, incluindo uma plataforma de gerenciamento de redes, um gerador automático de registros de problemas e um sistema de registro de problemas [LEW 95]. Nesse contexto, os registros presentes no sistema de registro podem ser criados pelo gerador automático de registros, a partir dos alarmes detectados na rede pela plataforma de gerência. Estes registros são criados pelo sistema após a aplicação de filtros de alarmes, que evitam múltiplas ocorrências relacionadas a um mesmo problema (como, por exemplo, quando se evita a criação de múltiplos registros para um mesmo problema intermitente) [LEW 95].

Essa criação automática acarreta algumas características nestes registros, que apresentarão uma certa padronização, já que são reflexos da configuração da plataforma geradora dos alarmes e do gerador automático de registros. Assim, incidentes similares serão provavelmente detectados de modo similar e darão origem a registros também similares, com tipo de problema similar. Além disso, a plataforma muito provavelmente identificará um problema já no seu ponto de origem — por exemplo, comunicará de imediato que um determinado enlace apresenta falha na comunicação, e não que a comunicação entre duas estações que possuem este enlace entre elas está interrompida. Assim, a comparação da similaridade entre dois registros gerados automaticamente é mais simplificada, apresentando menos variações que os registros informados por especialistas humanos.

No presente trabalho se pretende, porém, atender problemas informados por especialistas humanos, e não gerados automaticamente. Como comentado anteriormente, foi utilizado neste trabalho uma estrutura baseada no sistema de registro de problemas CINEMA, desenvolvido e utilizado pelo grupo no centro de gerência do POP-RS, acrescida das modificações necessárias para a implantação do raciocínio baseado em casos. Também o CINEMA [TAR 96] trata principalmente de registros criados manualmente por usuários. Assim, mecanismos adicionais para assegurar a identificação da similaridade entre casos é desejável. Neste trabalho, foram, então, definidos alguns mecanismos para tratamento destas questões.

Em primeiro lugar, como já foi comentado, é utilizada uma linguagem estruturada para com o usuário, e um mecanismo desenvolvido para cálculo da similaridade da descrição em texto livre é implementado para que esta descrição contribua também para a similaridade. Mas, além disso, foi também identificada a necessidade de realizar algum tratamento sobre o tipo de problema a ser consultado.

O tipo de problema é um dos principais índices para a recuperação dos casos — ele funciona como um filtro para os casos recuperados, impondo uma hierarquia virtual na base de casos onde só os casos dos tipos desejados são consultados. O estudo efetuado sobre os problemas em redes e a análise dos registros cadastrados no sistema estudado (CINEMA) demonstraram, porém, que os problemas em redes são algumas vezes extremamente complexos, e uma ocorrência que é detectada inicialmente como de um tipo pode ser causada, na realidade, por um outro tipo de problema correlato. Isso causa, assim, algumas dificuldades. Dependendo do estágio de desenvolvimento da situação quando o registro foi criado, este pode ser informado como de diferentes tipos, o que impediria a sua recuperação na base de um caso semelhante cadastrado em etapas diferentes. Além disso, foi percebido que um mesmo problema pode levar a falhas em diferentes níveis e aplicações, e, mais uma vez, problemas com a mesma falha poderiam não ser recuperados como similares. Esta situação é agravada ainda pelo sistema

proposto ser baseado na criação manual de registros, que traz uma maior variação entre os registros criados.

Para minimizar essa situação, percebeu-se a necessidade de relacionar não apenas o tipo de problema usado na criação, mas de considerar, também, o tipo de problema identificado após o encerramento do problema, além de desenvolver um mecanismo de tratamento para essa característica. O mecanismo proposto efetua então, para cada caso, um tratamento para identificar outros tipos de problemas relacionados e prováveis causadores. Este tratamento faz uso de uma relação histórica entre os tipos de problemas cadastrados inicialmente e o tipo de problema real causador indicado na solução, além de utilizar informações específicas da descrição do problema, que podem redirecionar de um tipo de problema para outro. Este mecanismo é apresentado na seção 5.5.1.2.

Na etapa seguinte, em posse das características da situação e dos prováveis tipos de problema, é realizada uma busca na biblioteca de casos, e os casos recuperados são classificados de acordo com o seu grau de similaridade com o caso corrente. Essa classificação foi elaborada levando-se em conta que se percebeu que as informações dos diversos problemas possuem um grau de relevância diferente para a identificação da similaridade entre eles.

Presente em diversas áreas em que os sistemas CBR são aplicados, essa particularidade pode ser tratada atribuindo-se diferentes pesos para cada característica no cálculo da similaridade [KOL 93][WAT 97] ou separando características que são essenciais para um problema de outras que podem contribuir para este, mas não são obrigatórias, tal como em [KOP 88]. Na abordagem proposta pelo presente trabalho, serão consideradas características essenciais algumas informações — tais como a representada pelos prováveis tipo de problema —, que são usadas como índices para a recuperação inicial na base, enquanto as demais receberão diferentes graus de importância, sendo representadas por um peso diferente no algoritmo para cálculo da similaridade entre dois casos. Este tópico será visto no item 5.5.2.2.

Uma vez que os casos recuperados são classificados, os melhores casos são apresentados ao usuário, ou o usuário pode solicitar um processo de refino. Esse processo foi incorporado ao sistema porque o estudo efetuado sobre os incidentes do domínio demonstrou que algumas características extremamente importantes acerca dos problemas não fazem parte do conjunto de características de um tipo de problema que serve a *todos* os problemas desse tipo, porém estão presentes em *diversos* casos desse tipo, formando um subgrupo dentro do tipo.

Assim, optou-se por uma abordagem que se baseia no conteúdo dos casos recuperados para obter características mais discriminantes para a nova situação. Cada caso pode conter uma ou mais informações relevantes para o problema em particular, denominadas *características específicas*. Quando é feita a recuperação, as características específicas dos casos selecionados são solicitadas ao usuário e utilizadas posteriormente para o refino da recuperação, podendo modificar a ordem de casamento, selecionar novos casos ou eliminar casos já selecionados.

Essas características permitem que ações de diagnóstico — atividades essenciais para a identificação da solução em problemas de gerenciamento de redes — sejam propostas pelo sistema e que seus resultados sejam utilizados para a recuperação. Outra contribuição dessa abordagem é o aumento do conhecimento do sistema, já que permite que, no momento de encerramento de um registro que será aprendido, características

específicas sejam informadas, seja selecionando-as entre as já cadastradas, seja cadastrando uma nova característica, e permitindo, assim, que um problema com facetas diferentes dos já armazenados seja aprendido adequadamente pelo sistema. Este processo será comentado na seção 5.5.3.

Os tópicos aqui comentados permitiram uma visão geral da modelagem de raciocínio utilizada no sistema proposto neste trabalho — sistema DUMBO. Apresentaremos, a seguir, a representação dos problemas, utilizando como formalismo redes semânticas. Nas seções posteriores abordaremos os processos de raciocínio utilizados e a base de conhecimento, que foram comentados brevemente nesta seção.

5.2 Representação dos Tipos de Problemas

Existem diversas formas de representar o conhecimento quando se utiliza um sistema especialista, tais como redes semânticas, sistemas de produção, frames [STE 95][TUR 92]. A representação por redes semânticas é uma forma gráfica de representar o conhecimento, composta por nodos e arcos, sendo basicamente uma descrição gráfica do conhecimento que mostra relações hierárquicas entre objetos. Uma das características mais interessantes e úteis das redes semânticas é sua capacidade de mostrar herança. Assim, uma idéia comum em tais representações é que as informações gerais devem ser armazenadas o mais alto na hierarquia em que for aplicável e herdadas pelos nodos abaixo durante o processo [STE 95].

A representação por redes semânticas foi utilizada neste trabalho para permitir a visualização dos atributos e da hierarquia dos tipos de problemas identificados para o ambiente de redes, através da aquisição do conhecimento, e que são utilizados no sistema, permitindo identificar quais as informações o sistema considera relevantes no tratamento de uma situação. Estas informações, como pode ser visto pelas figuras, variam conforme os principais sintomas e o contexto da rede em que o problema se desenvolve, e representam sintomas que podem ser encontrados nas situações, topologia da rede, etc. Em virtude da complexidade e ampla gama de problemas encontrados, optou-se por dar uma ênfase maior para as falhas na comunicação, realizando um tratamento mais simplificado nas falhas dos níveis superiores.

Inicialmente, na figura 5.1, é apresentada a visão global dos tipos de problemas, que podem ser *problemas na comunicação* e *problemas nas camadas superiores*. Nesta figura e na figura 5.2, os tipos de problemas, seus atributos e valores que podem assumir são indicados. Como pode ser visto, um dos atributos dos problemas de comunicação é a *localização do problema*, que pode ser *entre uma mesma sub-rede IP*, *entre outras sub-redes IP*, *em uma linha serial*, etc. (figura 5.3). Essas especializações conduzem a diversos outros atributos específicos para a localização do problema em questão, que são apresentados no anexo 1.

Nessa rede, nem todas as especializações de um atributo herdado são obrigatoriamente assumidas em uma situação. Um problema de roteamento/endereçamento, por exemplo, não assumirá a especialização *canal de comunicação de uma LAN* para a *localização do problema*. Da mesma forma, o atributo *interface de rede* também não será relevante para esta situação. Além disso, existem atributos que podem ser, em algumas situações, relacionados a outros da hierarquia superior de modo que podem ser identificados automaticamente, recebendo o mesmo valor do outro atributo. Um exemplo é quando um problema de conectividade para uma

outra sub-rede IP aponta que o primeiro roteador na rota para esta sub-rede já não é atingido — neste caso, a abrangência do problema da sub-rede afetada terá o mesmo valor que a abrangência do problema na sub-rede origem, pois a sub-rede afetada é a própria sub-rede origem.

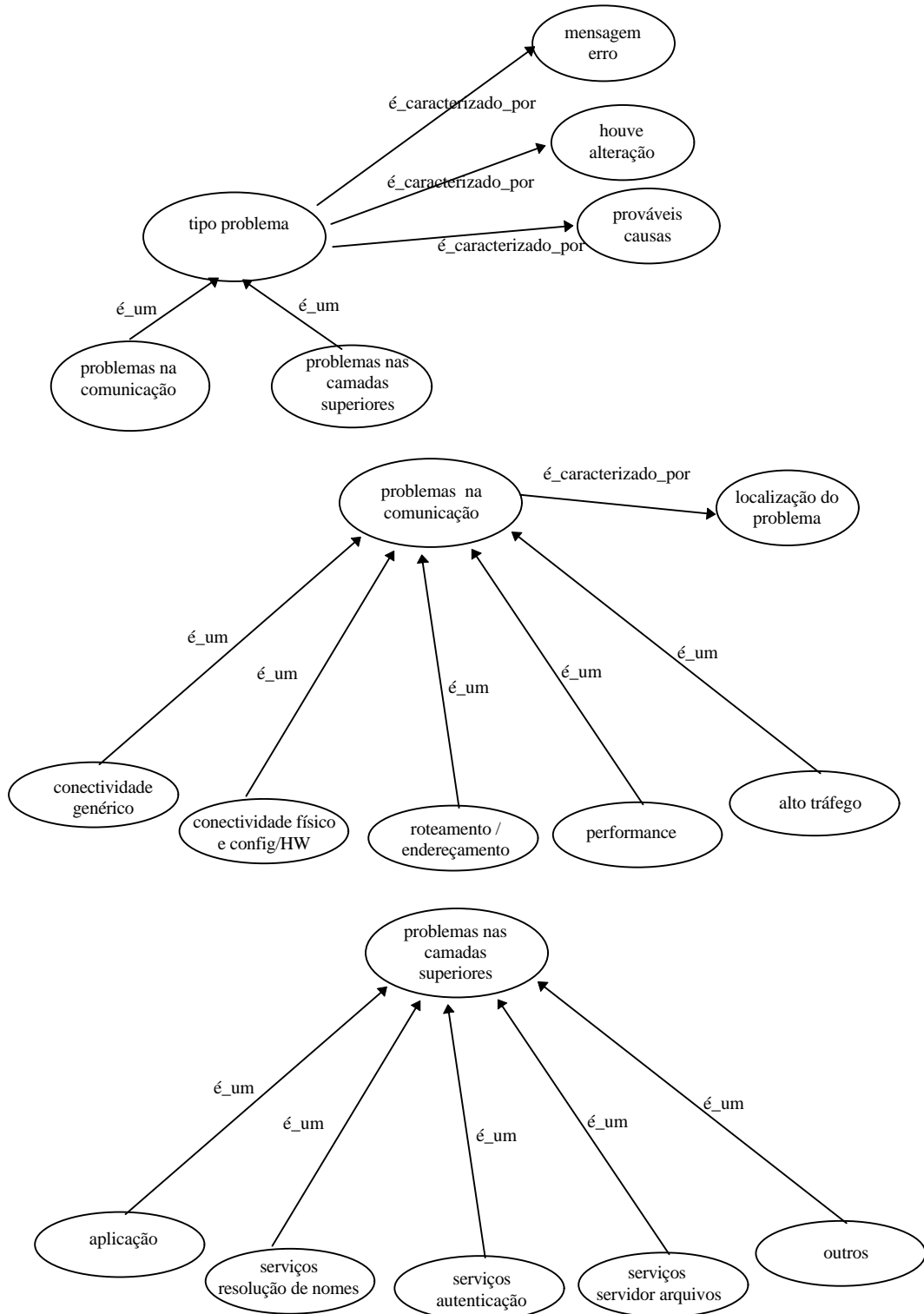


FIGURA 5.1 - Rede semântica dos tipos de problemas

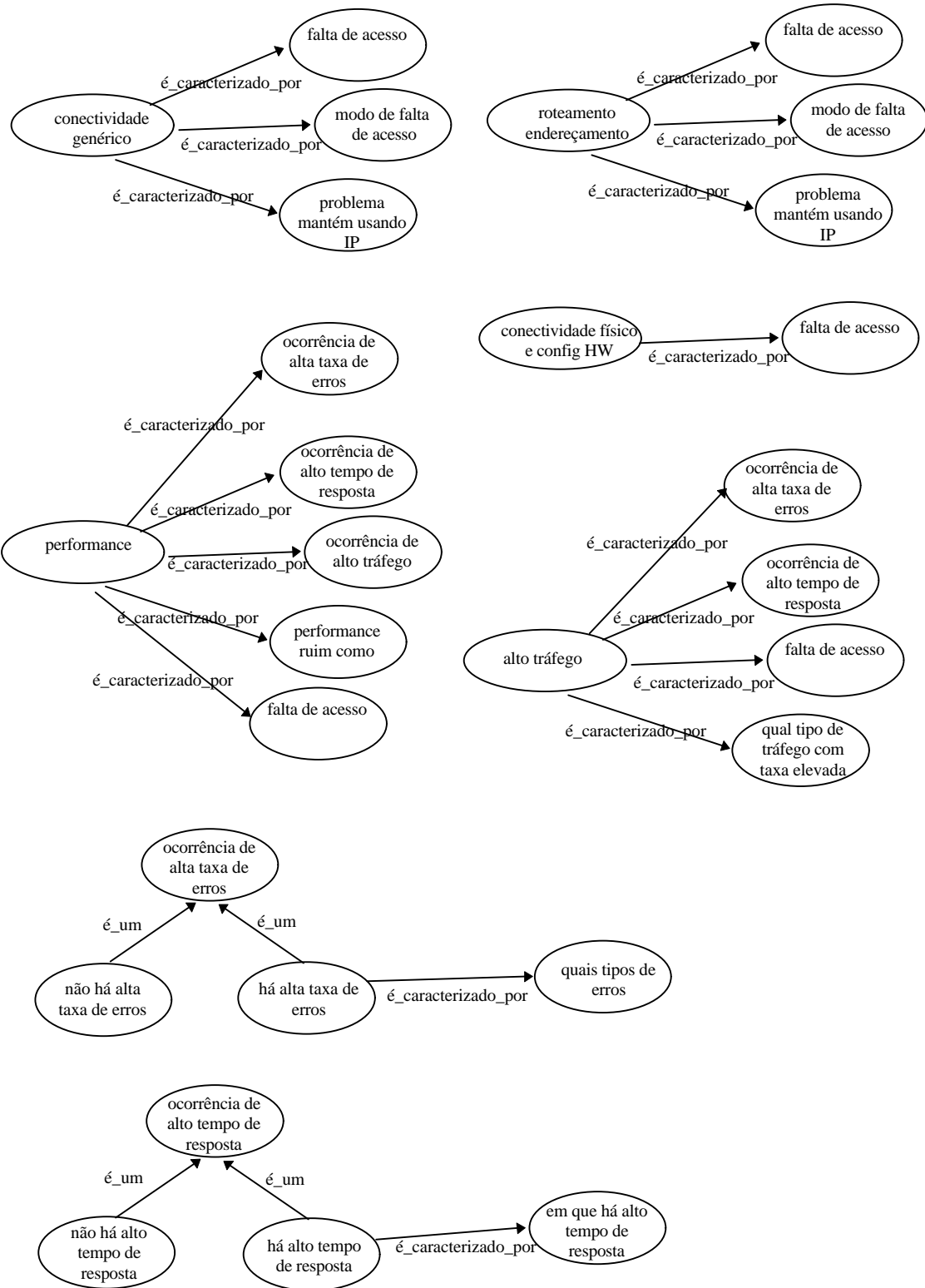


FIGURA 5.2 - Rede semântica dos tipos de problemas (parte 2)

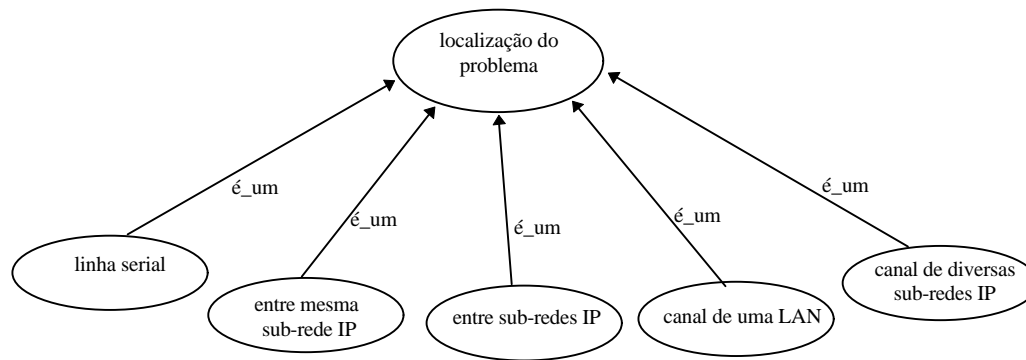


FIGURA 5.3 - Rede semântica da localização dos problemas

5.3 Estrutura do Sistema

O sistema DUMBO foi estruturado de modo a manter as funções dos sistemas de registro de problemas tradicionais, inserindo os procedimentos de raciocínio na etapa de criação de um registro, de encerramento de um registro e na criação de notas. A estrutura do processo de raciocínio pode ser visto na figura a seguir.

No momento de criação de um registro, o módulo **avaliador da situação - definição do contexto** é ativado e é o responsável pela obtenção das informações referentes à descrição do problema. O tipo de problema informado é utilizado então para identificar as informações adicionais que devem ser obtidas. Em alguns problemas, situação análoga ocorre para a *localização do problema*. As informações que puderem ser obtidas de modo automático são processadas, enquanto que as demais são solicitadas ao usuário.

O módulo efetua o tratamento sobre o tipo de problema, identificando os demais tipos de problemas relacionados segundo a experiência do sistema e fazendo uso das regras *redirecionadoras*, que são aplicadas sobre algumas características, para verificar se um outro tipo de problema deve também ser consultado e se as probabilidades de cada tipo de problema devem ser alteradas. Essas regras constituem regras simples, aplicadas sobre algumas características pré-definidas que contribuem para isolar o tipo de problema.

O registro de problema/caso é então conduzido ao módulo **recuperador inicial**, que recupera na biblioteca os casos cujo tipo corresponda a algum dos tipos de problemas selecionados para busca — os tipos de problema avaliados como prováveis pelo módulo anterior. Em alguns tipos de problema, filtros podem ser também aplicados. Os casos recuperados são classificados pelo módulo **seletor** segundo as suas características, utilizando para o cálculo da similaridade entre os casos o grau de relevância de cada característica, que é influenciado pelo tipo de problema final do caso recuperado, e os melhores casos são selecionados e apresentados ao usuário.

O usuário pode aceitar diretamente esses casos ou solicitar um processo de refino da recuperação. Para isso, informa todas ou algumas das características específicas solicitadas, que foram recuperadas pelo módulo **reutilizador-revisor** dos casos recuperados, e o processo de recuperação é reiniciado utilizando agora também as características específicas.

Situações em que o sistema não foi capaz de propor casos similares são aprendidas pelo sistema no momento de encerramento do registro, a fim de que o conhecimento adquirido pelo grupo na experiência possa ser utilizado novamente. Nas demais situações, o caso é armazenado apenas com fins de gerenciamento. Esses procedimentos são controlados pelo módulo **aprendiz**.

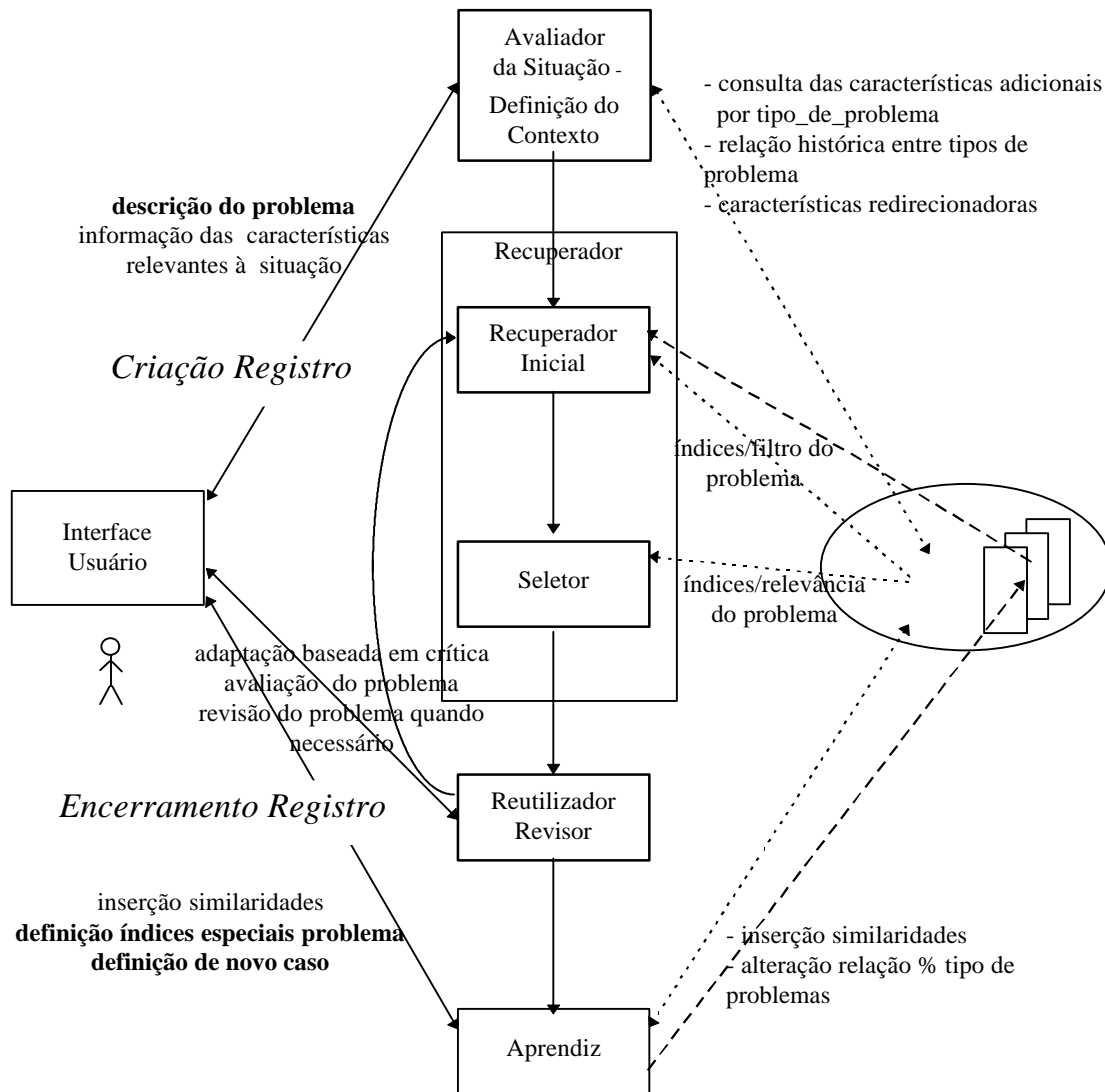


FIGURA 5.4 - Estrutura do sistema proposto

Essa seqüência de processos não precisa ser efetuada ao mesmo tempo pelo usuário — a estrutura apresentada diz respeito à seqüência de raciocínio do sistema, e não necessariamente o desenvolvimento do problema e seu registro seguirão essa mesma seqüência de etapas diretamente (criação do registro, visualização dos similares, reutilização de uma solução proposta com sucesso e encerramento do registro). Após a recuperação dos casos, quando soluções são oferecidas, através dos casos, e atividades de diagnóstico são sugeridas, através destes e das características específicas, o usuário pode interromper o processo, a fim de realizar alguma atividade de diagnóstico ou testar alguma solução proposta. Após esse período, pode fornecer novas informações e solicitar a recuperação de novos casos, pode inserir uma nota ao registro ou pode encerrar o registro, caso a solução proposta tenha surtido o efeito desejado.

Ao inserir uma nota, os casos recuperados a partir das informações já disponíveis sobre o problema são fornecidos ao usuário, assim como as características específicas são solicitadas, e o processo de recuperação pode ser reiniciado.

A seguir, apresentamos cada um dos componentes da estrutura: a base de conhecimento e os processos envolvidos no raciocínio.

5.4 Base de Conhecimento

Conforme comentado no capítulo 2, a base de conhecimento de um sistema CBR é responsável por armazenar os seus casos. Diz respeito a ela, portanto, decidir o que armazenar em um caso, qual a estrutura apropriada para descrever o conteúdo dos casos e como organizar e indexar a base de casos para que os processos de recuperação e reutilização se tornem eficientes [AAM 94]. Uma questão adicional envolve a decisão de como integrar a estrutura da base de casos em um modelo de conhecimento geral do domínio, de modo que este conhecimento seja incorporado.

Comentaremos, a seguir, a representação dos casos, a organização da base e o conhecimento geral do domínio do modelo proposto.

5.4.1 Representação dos Casos

A partir das discussões acerca de sistemas de registro de problemas, apresentadas na seção 3.3, e de representação dos casos, comentadas na seção 2.4.1, torna-se simples relacionar um registro de problema como um caso. Resta, assim, determinar quais informações adicionais são necessárias para transformar um registro de problema em um caso que possa ser utilizado de modo eficiente nos processos de acesso à base de casos.

Antes disso, porém, é necessário avaliarmos como será a **linguagem de um caso**, a fim de determinar as transformações da linguagem do registro — muitas vezes uma linguagem em texto livre, sem restrições — para a linguagem de um caso. Conforme aponta Lewis [LEW 95], quanto mais liberal se é na definição da linguagem para descrever as experiências nos problemas das redes, mais provável se é de ser mal interpretada a natureza do problema. Assim, a melhor abordagem é impor restrições para a linguagem do caso de modo que o usuário deva ser preciso sobre a experiência representada no caso.

Deve-se cuidar, entretanto, para que a linguagem não se torne excessivamente precisa e restritiva, de modo que não existam formas de expressar pedaços importantes de informação na linguagem. Assim, é necessário estabelecer inicialmente um meio termo entre uma linguagem sem restrições e outra excessivamente restrita, e fornecer modos para que a linguagem evolua a partir de novas experiências.

Neste trabalho, a fim de minimizar os problemas que envolvem as linguagens estruturadas — necessárias para o melhor entendimento da natureza do problema — será usado um campo para descrição em texto livre, que será utilizado de modo complementar para o cálculo da similaridade entre os casos, e se possibilitará que novas características sejam inseridas, através das *características específicas*, permitindo uma flexibilização da linguagem. Estes tópicos serão abordados novamente em seções subseqüentes. Resta agora definir as modificações de um registro tradicional para um

caso, a fim de que os diversos processos envolvidos no raciocínio possam ser incorporados de modo eficiente no sistema.

Como apresentado no capítulo 2, em um caso de um sistema CBR, a **descrição de um caso** representa as informações acerca do problema e do ambiente onde ocorreu. Os componentes que fazem parte dela são os **objetivos** a que o caso se propõe, as **restrições** associadas a obtenção do objetivo e as **características da situação** representada pelo caso. Esses componentes variam conforme a aplicação sendo desenvolvida, sendo enfatizados em algumas delas apenas um ou dois componentes.

De modo similar, as informações em um registro de problema trazem os dados referentes aos sintomas enfrentados e o ambiente de rede onde o problema foi detectado. Os registros de problemas se propõem a diagnosticar um problema — um **objetivo** simples, único, que é implícito a todos os registros de problemas tratados. Assim, este componente, que não está presente nos registros tradicionais, também não é necessário aos casos.

As **restrições** e **características da situação** em um caso, por sua vez, correspondem às informações presentes nos registros que descrevem os sintomas identificados e as condições do ambiente onde o problema ocorreu. São as informações mais importantes no caso para aplicações de diagnóstico [KOL 93]. Em registros de problemas, entretanto, nem todas as informações auxiliarão a descobrir a solução de um problema: algumas delas possuem propósitos apenas de gerenciamento. Essas informações devem ser, porém, mantidas nos casos, a fim de que o sistema preserve as funções legadas dos sistemas de registro de problemas tradicionais.

Os registros de problemas não contêm, contudo, todas as informações relevantes sobre o problema que poderiam ser utilizadas para a identificação de situações similares. Existem informações adicionais, tais como perguntas que devem ser feitas aos usuários e testes que devem ser efetuados, que são identificadas pelos técnicos especialistas a partir de alguns dados dos campos dos registros de problema tradicionais. Estes campos são utilizados como impulso para a identificação de novas informações a serem buscadas — e a extensão do quanto um técnico responsável pelo diagnóstico de um problema avança a partir dos dados de um registro é o que faz dele um especialista [LEW 95].

Essas informações, entretanto, não estão usualmente presentes nos registros, ou então são registradas através de campos de texto livre — indesejáveis, como já foi comentado, nestes sistemas de raciocínio. Assim, a transformação do registro em um caso demanda a identificação das demais informações sobre o problema que poderiam contribuir para a recuperação de um caso similar — aquelas informações que trazem aspectos relevantes sobre o problema, que são identificadas através da experiência pelos especialistas como características que podem contribuir para a solução de um problema, seja por indicar um sintoma específico, seja por indicar uma condição do ambiente da rede que pode ser relacionada ao problema.

O estudo dos problemas em redes, apresentado na seção 4.3, demonstrou que uma das informações que subdivide os problemas e que pode ser usada para identificar dados relevantes a eles é o tipo de problema. Diferentes tipos de problemas apresentam diferentes sintomas e são relacionados a diferentes condições do ambiente. Problemas com a conectividade em uma rede local, por exemplo, possuem diferentes características relevantes de problemas de acessibilidade, ocorrendo com o uso de uma única aplicação. Assim, uma abordagem que pode ser utilizada, sugerida por Lewis [LEW 95], é a

identificação dos prováveis tipos de problemas do domínio e, para cada um dos tipos identificados, analisar as informações adicionais que poderiam auxiliar para resolver o problema e adicionar campos ao caso-registro para cada uma destas informações. Essa abordagem foi aplicada no modelo identificando, para cada tipo de problema, um conjunto de informações que contribui para descrever as informações relevantes *daquele* tipo de problema, sendo chamadas *características adicionais*.

Além do tipo de problema, existem também outras características que influenciam a determinação das características relevantes para um problema quando combinadas com alguns valores particulares de outras. Um exemplo é a interface de rede local. Em uma rede onde a característica interface de rede corresponde a uma Ethernet, por exemplo, as características relevantes (tais como taxa de colisões e carga da rede) são diferentes das características relevantes de uma rede FDDI. Essas inter-relações entre as características, que serão comentadas posteriormente na seção 5.5.2, são usadas para o processo de classificação. Já os tipos de problemas, que envolvem não apenas a relação entre algumas poucas características mas influenciam todo o conjunto de características adicionais relevantes, são usados como índices principais no processo de busca, e serão comentados na seção referente à organização da base de casos.

As características aqui apresentadas — informações com propósitos de gerenciamento, informações gerais sobre o problema, informações adicionais referentes ao tipo de problema — são obtidas pelo sistema no processo de definição do contexto, que é inserido no sistema de registro de problemas no momento da criação do registro. O estudo efetuado sobre os problemas enfrentados nas redes demonstrou, porém, que casos formados apenas por estas características deixariam lacunas no sistema.

A primeira delas diz respeito à grande variação de problemas enfrentados, aliada à complexidade destes. Assim, percebeu-se que mesmo identificando características relevantes para cada tipo de problema, entre um mesmo tipo de problema existe uma grande variação de casos, que possuem características relevantes diferentes de *todos* os demais casos do grupo, embora suas características possam ser semelhantes a *alguns* dos problemas do grupo. Assim, incorporar essas características ao questionário das informações adicionais do tipo de problema torna-lo-ia exaustivo e poderia obter informações não importantes para o problema e nem mesmo para a maior parte dos problemas do grupo.

A segunda lacuna diz respeito às restrições impostas pela utilização de uma linguagem estruturada. Conforme citado anteriormente, um sistema deve estabelecer inicialmente um meio termo entre uma linguagem excessivamente estruturada (que peca pela restrição às informações sobre o problema e sobre os problemas que podem ser manuseados eficientemente) e uma excessivamente flexível (que peca por permitir uma má interpretação do sistema), permitindo que a linguagem evolua a partir das novas experiências do sistema [LEW 95]. Assim, a estrutura dos casos formada apenas pelas características adicionais demanda ainda por componentes para expressar informações adicionais às identificadas inicialmente como relevantes para cada um dos tipos e meios de permitir a evolução da linguagem.

Por fim, uma terceira lacuna diz respeito aos modos de incorporar no sistema informações provenientes das ações de diagnóstico efetuadas para o isolamento de um problema. Essas ações, representando muitas vezes informações com um alto custo para obtenção, são escolhidas a partir dos dados iniciais sobre a situação (representados na

criação do registro) e de resultados de ações anteriores, e são algumas vezes armazenadas nos registros tradicionais, através das notas. A seqüência de sua execução, entretanto, não é necessariamente completa: certas vezes, alguns especialistas, a partir da sua experiência e de determinados sintomas ou condições da rede, tendem a “pular” algumas das ações que especialistas novatos efetuariam. Assim, demonstra-se a necessidade de representar as informações provenientes dos resultados dessas ações em um modo que possam ser utilizadas para os processos de raciocínio, permitindo ainda que a ordem de sua execução não seja obrigatória.

Uma alternativa para solução de algumas dessas lacunas seria a criação de questionários alternativos para subconjuntos de problemas, solicitados pela combinação de valores de algumas das características já informadas. Essa alternativa se refletiria, porém, sobre a flexibilidade do sistema, tornando a linguagem ainda mais restritiva, ao invés de flexibilizá-la e permitir que ela evolua com a experiência do sistema. Ela exigiria, também, a identificação exaustiva de todos os possíveis problemas das redes em detalhes e não apenas os tipos de problemas possíveis, uma tarefa complicada em virtude da complexidade, variação e grau de transformação das tecnologias envolvidas nos problemas em redes. Além disso, esta alternativa propiciaria que as informações fossem fornecidas no momento de criação do registro, o que se mostra inadequado, porque muitas dessas características importantes são obtidas ao longo da execução do problema, através de ações de diagnóstico.

Assim, foi elaborada para o modelo proposto uma abordagem diferente, que é aplicada após uma primeira recuperação, formando o refino dessa recuperação. Essa abordagem faz uso do conteúdo de características discriminantes presentes nos casos já recuperados para identificar novas informações *relevantes para o caso corrente*, já que são provenientes dos casos similares a este. Tal abordagem é semelhante à já utilizada em outros sistemas de raciocínio baseado em casos, tais como CASCADE [SIM 92] e SMART [ACO 92], além de ferramentas para desenvolvimento de sistemas como a família CBR3 (Inference Corp.) [WAT 97][WIL 94], que fazem uso de informações provenientes dos casos recuperados, a partir de uma busca inicial com base em uma breve descrição do problema [ACO 92] ou em características iniciais deste [SIM 92] para a identificação de outras possíveis informações relevantes ao problema. O sistema MASTER [DRE 95] pode ser também aqui citado, pelo seu conceito de *Master Tickets*, em que é feita a recuperação do *master ticket* mais similar e as atividades de diagnóstico deste caso são solicitadas.

Cada caso, pode, assim, conter uma ou mais destas características discriminantes que serão utilizadas posteriormente, quando o caso for recuperado, para o refino da recuperação. Essas características, denominadas *características específicas*, são as responsáveis por armazenar informações importantes do caso que não estão presentes entre as características do tipo de problema, sendo utilizadas, na recuperação, para identificar um subconjunto de casos similares dentre os selecionados. Tais características permitem que o resultado de ações de diagnóstico sejam utilizadas para a recuperação, não exigindo que todas sejam respondidas numa ordem pré-definida. Além disso, contribuem para o aumento do conhecimento do sistema e flexibilizam a linguagem dos casos, já que permitem que novas características sejam acrescentadas ao sistema no momento de aprendizado de um caso.

Assim, a parte **descrição** do caso é formada pelas informações apresentadas no momento de criação do registro e pelas características específicas do caso. Além disso, é

formada também pelas notas que são acrescentadas ao registro ao longo da evolução do problema. A figura seguir demonstra os componentes do caso.

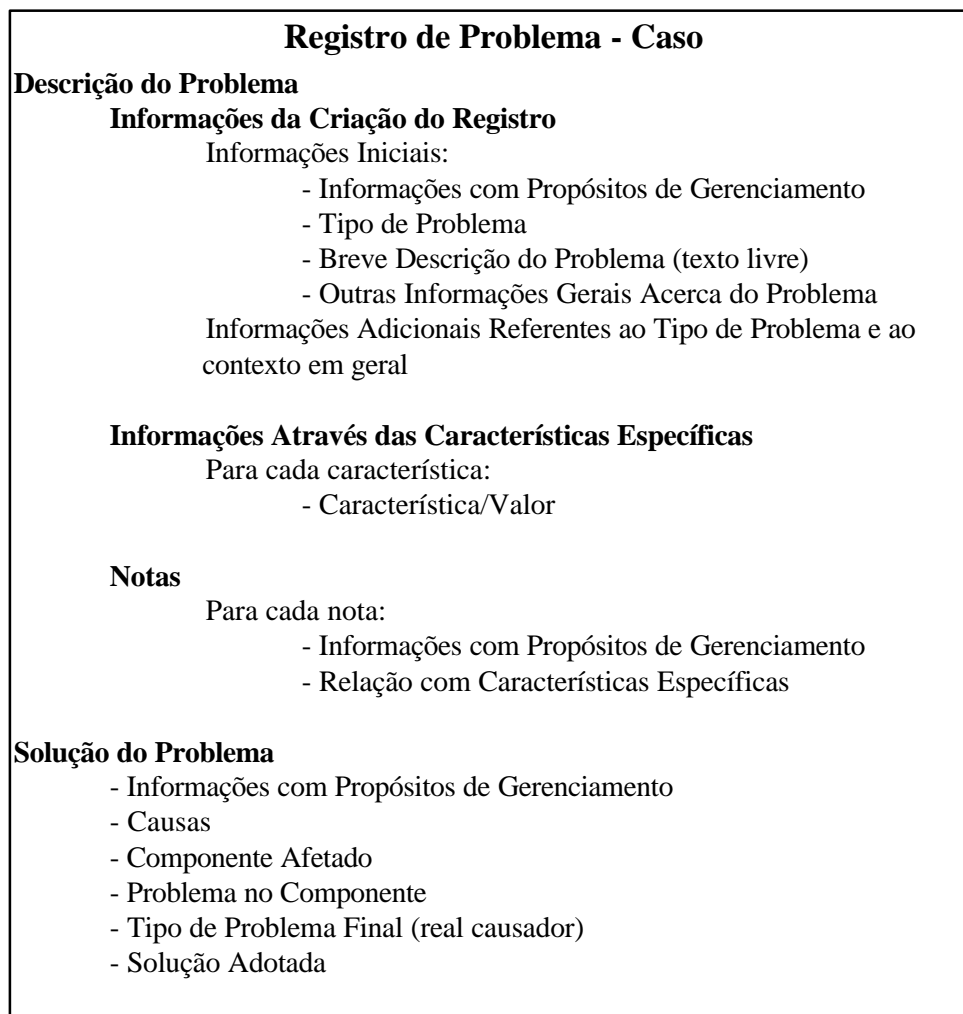


FIGURA 5.5 - Componentes de um caso

Além da descrição, o caso é formado também pelas informações referentes à **solução do problema**. Similar às informações em um registro tradicional, a solução é formada pelas informações com propósitos exclusivamente de gerenciamento (tais como data de encerramento, usuário que encerrou o problema, etc.) e pelas informações referentes às causas do problema, componente afetado (função deste equipamento no problema), descrição do problema no componente, solução empregada e tipo de problema final.

A característica tipo de problema final foi acrescentada por ter sido constatado que muitos problemas identificados inicialmente como de um tipo são, ao final, causados por outros. Assim, os registros devem ser cadastrados pelo tipo de problema final, que foi o problema que realmente causou os sintomas encontrados, a fim de que esse registro possa, no futuro, contribuir melhor para a solução de um problema com causa similar à ocorrida. Para que esse problema contribua também para problemas com características similares às enfrentadas, identificados também como de um tipo diferente do real, são fornecidos mecanismos de relacionamento entre os problemas, que serão comentados na seção referente a definição do contexto.

5.4.2 Base de Casos: Organização e Indexação

Além da representação dos casos, o desenvolvimento de uma base de casos envolve a escolha dos índices para eles. Os índices, formados por combinações de características importantes que distinguem os casos entre si, são os responsáveis por tornar os casos acessíveis no momento apropriado. São utilizados em diversos processos de recuperação para restringir o espaço de busca na base de casos, a fim de que algoritmos de recuperação parcial — algoritmos dispendiosos — sejam aplicados apenas nos casos com algum potencial de relevância para o caso corrente. São, portanto, papel importante nos algoritmos de recuperação para organizar e direcionar a busca.

Buscou-se, assim, uma forma de organizar os casos na base de modo que a busca por casos similares a uma situação corrente fosse restrita àqueles casos que podem contribuir com sua solução para a resolução da situação corrente. Aliando essas questões às discutidas anteriormente sobre a representação dos casos e às informações relevantes para um problema, optou-se, assim, por uma estrutura hierárquica para a base de casos que se baseasse na informação sobre o tipo de problema enfrentado (tipo de problema final). Essa abordagem para organizar a base é também proposta em [LEW 95].

Além disso, identificou-se que existem informações úteis para restringir a busca na base, variando com o tipo de problema — tais como o padrão de interface de rede encontrado na rede gerenciada, que é relevante para os problemas físicos e de configuração do hardware. Estas características serão utilizadas para criação de filtros no momento da recuperação da base.

A figura a seguir apresenta a hierarquia utilizada na base de casos.

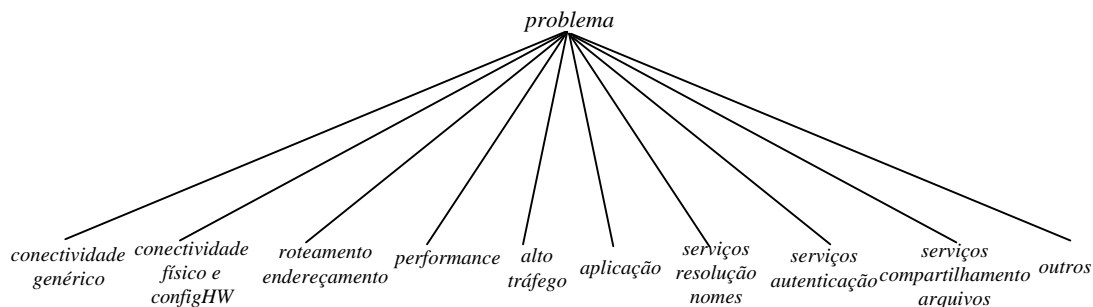


FIGURA 5.6 - Hierarquia da base de casos

5.4.3 Base de Conhecimento Geral do Domínio

Os sistemas CBR possuem, usualmente, também uma base de conhecimento que armazena o conhecimento do domínio, que permite identificar fatores tais como a similaridade entre as características, as relações entre elas, etc. Também no presente trabalho, a partir das necessidades envolvidas nos processos de raciocínio e na representação dos problemas, foram identificados elementos do conhecimento do domínio que deveriam ser representados no sistema.

Assim, o conhecimento do domínio presente no sistema inclui várias informações:

- conhecimento das relações entre os tipos de problema, adquirido a partir da experiência do sistema e do relacionamento entre o tipo de problema aparente e o tipo de problema identificado ao final como causador. Essas relações são armazenadas em tabelas, representando a probabilidade de um problema comunicado inicialmente como de um tipo ser, ao final, identificado como cada um dos outros tipos do sistema.

- conhecimento da relação entre algumas características com os tipos de problema prováveis. Essas relações são expressas através de regras em que determinado valor para determinada característica leva a uma probabilidade maior que um tipo de problema seja considerado provável. Um exemplo pode ser encontrado na informação de que um problema comunicado como de performance da comunicação apresenta também falta de acesso. Assim, um tipo de problema provável, que poderia ter sido usado em situações similares anteriores ou mesmo que pode ser causador da degradação de performance, é o problema de tipo conectividade - genérico.

- conhecimento sobre a similaridade entre os valores das características dos casos. Permite identificar, por exemplo, que dois equipamentos de uma mesma família são similares ou que determinados valores para o estado de uma interface são mais similares que outros.

- conhecimento sobre as inter-relações entre algumas características e sua relevância. Permite identificar que uma variável pode transmitir similaridade (ou seja, é relevante) e deve ser comparada a partir do valor de uma outra, como, por exemplo, o conhecimento de que a taxa de colisão é relevante e sua similaridade deve ser verificada quando a interface de rede sendo utilizada na rede com problema é uma Ethernet.

- conhecimento sobre a similaridade de expressões para cálculo de similaridade entre características em texto livre.

5.5 Modelagem dos Processos CBR

Os processos de raciocínio envolvidos no sistema são separados em quatro etapas: **definição do contexto**, recuperação com **busca e classificação**, **reutilização e revisão** e, após a solução ser obtida, **aprendizado**.

Os processos de raciocínio baseado em casos iniciam com a **avaliação da situação** — processo responsável por analisar a situação e determinar quais seriam os índices para identificar um caso similar que estivesse armazenado na base se a situação fosse bem entendida, o que é feito através de uma interpretação do problema que busca identificar que tipo de situação está ocorrendo, o que é importante nela, e o que poderia ser concluído a partir dela. A avaliação é finalizada através de um processo de *elaboração* — as características deriváveis do caso são calculadas e utilizadas junto com a descrição original para a busca e classificação.

A avaliação da situação pode ser desenvolvida em três fases. Pode ser executada antes da busca ser realizada, elaborando um conjunto de características que sejam adequadas para a recuperação — processo conhecido como **definição do contexto**. Pode ser também utilizada durante a busca, utilizando a base de casos para designar que novas dimensões podem ser geradas para discriminar entre os vários itens da memória — processo conhecido como **refinamento do contexto**. Por fim, ela pode também ser

implementada depois da recuperação — processo conhecido como **redefinição do contexto** [KOL 93].

Nesse modelo, no processo de **definição do contexto**, a partir da descrição inicial do problema ocorrido é feita a análise da situação e o sistema identifica quais as características adicionais sobre ela podem contribuir para a identificação de problemas similares. A avaliação da situação é efetuada também depois da etapa de recuperação, quando novas características importantes para a recuperação são identificadas e obtidas. Por fim, a avaliação é realizada novamente no momento de aprendizado.

Trataremos aqui inicialmente da definição do contexto. A seguir, apresentaremos o processo de recuperação referente à busca e classificação e o processo de reutilização e revisão, enfocando a avaliação da situação. Por fim, abordaremos o processo de aprendizado.

5.5.1 Definição do Contexto

No momento inicial da criação de um registro, são obtidos inicialmente apenas dados iniciais acerca do problema (tais como responsável pelo registro, usuário reclamante, breve descrição do problema, tipo de problema). O processo de definição do contexto é o responsável então por especificar ao sistema quais informações adicionais são necessárias, definindo a lista de características que devem ser elaboradas, os mecanismos que podem ser utilizados para essa elaboração e elaborando-as.

Conforme comentado anteriormente, na seção referente à representação dos problemas, a lista de características para elaboração no sistema é sensível ao contexto, sendo associada aos diferentes tipos de problema que o sistema trata, a fim de que a elaboração seja especializada para cada problema. Esse tipo de abordagem para definição das características é adotado também em outros sistemas, como o CYRUS, desenvolvido por Janet Kolodner *Apud* [KOL 93], que se baseia nos diferentes tipos de situação para agrupar os casos em redes de discriminação redundante, e o sistema CRITTER [LEW 93], que se baseia também nos tipos de problemas de rede.

O processo de definição das características se desenvolve da seguinte forma. O sistema analisa a situação e identifica um conjunto de características que deve ser elaborado a partir do tipo de problema do caso corrente. A partir dos valores obtidos com a elaboração destas características, algumas delas (tais como localização do problema) são também utilizadas para a definição de outras características que devem ser elaboradas. Essas características, elaboradas a partir do contexto, são denominadas *características adicionais*.

No desenvolvimento do sistema Dumbo, a identificação destas características foi feita manualmente, através da identificação de que informações poderiam contribuir para a identificação de casos similares e do julgamento da relação custo-benefício da disponibilidade de cada uma. Outras informações sobre este processo podem ser buscadas na seção 5.4.1.

Após serem identificadas quais são as características adicionais que devem ser utilizadas para a situação, o sistema implementa então mecanismos para executar esta elaboração. Esses mecanismos podem ser implementados com execução automática ou, caso não seja possível computar a característica automaticamente pelo sistema, ela é solicitada ao usuário.

A elaboração automática das características adicionais é feita através de um dos procedimentos abaixo:

- obtenção a partir de uma ou mais combinações de características já obtidas, em conjunto com o conhecimento do domínio presente. Um exemplo é a característica *tipo de equipamento* (tal como roteador), que pode ser elaborada a partir da característica *produto* (tal como Cisco 7000), desde que o sistema possua cadastrado o produto correspondente, podendo assim identificar qual tipo de equipamento aquele produto é. Outro exemplo é a característica dos prováveis componentes com falha física, que será comentado na seção a seguir.
- obtenção a partir do conhecimento topológico da rede, obtido com a utilização e aprendizado do sistema. Exemplos dessas características são as informações referentes a um equipamento como *número IP*, *sistema operacional* e *produto* que podem ser elaboradas a partir das características de identificação do equipamento (seu *nome*, por exemplo), desde que este equipamento já tenha sido referenciado em algum caso anterior no sistema e estes dados tenham sido informados em tal situação. Essas características poderiam ser também obtidas diretamente da rede utilizando uma plataforma de gerência que fosse integrada ao sistema e que possuísse informações sobre o referido equipamento ou através de serviços de rede como o *nslookup* (que pode ser usado para retornar o número IP de um nome do servidor DNS). Nesta versão do trabalho, porém, esses procedimentos não estão disponíveis.
- obtenção de condições do ambiente diretamente da rede. Assim, poderiam ser elaboradas características relacionadas às informações de gerenciamento SNMP associadas a MIB dos equipamentos envolvidos. São exemplos a taxa de erros de descartes de entrada e saída de uma interface de um roteador para uma linha serial. Outras características poderiam ser também obtidas utilizando-se ferramentas de monitoração integradas ao sistema (como *taxa de colisões* em uma rede Ethernet), objeto de estudo de uma próxima versão. Na versão do protótipo implementada, entretanto, esses procedimentos não estão ainda integrados.

Assim, para cada característica a ser elaborada, o sistema avalia se é possível obtê-la automaticamente e, em caso contrário, solicita-a ao usuário. Essas características podem ser solicitadas diretamente (solicitando-se exatamente a informação que se deseja) ou através de uma questão que envolva duas ou mais características que devem ser elaboradas, a fim de que o usuário forneça o mínimo de informações necessário.

O processo de definição do contexto não é, entretanto, responsável por elaborar apenas as características adicionais. Conforme comentado anteriormente, ele efetua também a elaboração dos tipos de problemas que são prováveis para a situação além do tipo de problema já informado. Essa informação é importante em virtude da complexidade dos problemas, uma vez que um problema aparentemente de um tipo pode ser causado na verdade por um outro diferente. Assim, é necessário analisar a situação a fim de identificar quais tipos de problemas adicionais devem ser também consultados.

5.5.1.1 Elaboração dos Prováveis Componentes com Falha Física

Uma das informações presentes freqüentemente da parte **solução** de sistemas de registro de problema é o componente afetado. Deste modo, essa informação pode ser usada para casamento do caso corrente se dispusermos, para essa situação, dos componentes que podem estar envolvidos no problema. Por ser uma comparação com

uma informação de um caso já resolvido — parte da própria solução — essa característica pode trazer dados importantes para a similaridade entre os casos.

Como foi comentado anteriormente, falhas em um componente podem ser causadas por problemas físicos no hardware ou na configuração do hardware, problemas na camada Internet (por falhas em roteamento, endereçamento IP, etc.) e problemas nas camadas superiores.

Buscou-se, assim, elaborar quais seriam os problemas em componentes que seriam causados por falhas físicas (incluindo configuração do hardware envolvido no acesso ao meio, como, por exemplo, configuração do frame em Ethernet ou IEEE 802.3). Os componentes identificados foram as entidades origem e destino (ponto de onde partiu e para onde se destina a comunicação), equipamentos de interconexão de rede (incluindo, assim, aqueles equipamentos que tem função de roteador, de servidor de comunicação, etc. na rota entre a estação local e remota), equipamentos de interconexão de enlace (tais como pontes presentes nas LANs local, remota ou intermediária entre a origem e destino) e cabeamento (incluindo, neste item, o cabeamento propriamente dito e os equipamentos que manipulam os dados apenas na camada física, tal como repetidores), como apresentado na figura 5.7.

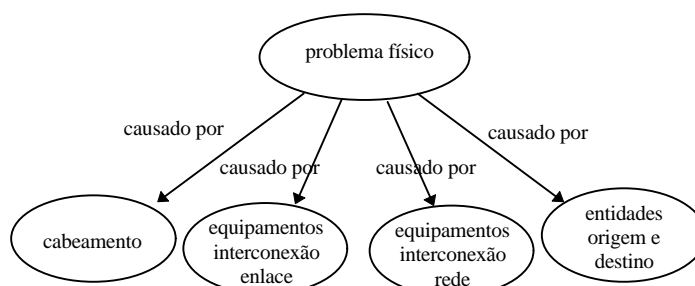


FIGURA 5.7 - Possíveis componentes envolvidos em falhas físicas e de configuração do HW

A posse de algumas informações presentes nas características adicionais, entretanto, permite que os prováveis componentes envolvidos em falha física sejam melhor interpretados. Assim, optou-se por um modo de raciocínio em que é feita a exclusão dos componentes (segundo a função desempenhada por eles no problema) que, na situação corrente, não devem estar envolvidos em falhas físicas. Isso pode ser feito pela não existência do componente na rede gerenciada ou porque, com a interpretação da situação, é possível eliminar alguns deles.

Duas regras pertencentes ao conjunto das regras do sistema são apresentadas a seguir como exemplo. A primeira regra se destina a eliminar os componentes das entidades origem e destino no caso corrente se o problema é encontrado na comunicação para todas as estações da sub-rede ou do segmento (eliminando, portanto, a estação remota de ser a causadora do problema), partindo de todas as estações (eliminando, portanto, a estação local de ser o componente com falha), naquelas situações em que a sub-rede não foi recém configurada nem alterada. A segunda regra, por sua vez, visa eliminar os componentes equipamentos de interconexão de enlace se o problema for em uma sub-rede IP de uma rede local que só possui um segmento/domínio de colisão.

SE localização do problema = 'entre mesma sub-rede IP' E
 (abrangência problema estações origem da sub-rede = 'todos' E
 abrangência problema estações origem dos segmentos envolvidos = 'todos') E
 (abrangência problema estações destino da sub-rede = 'todos' OU
 abrangência problema estações destino dos segmentos envolvidos = 'todos') E
 alteração = 'funcionavam anteriormente sem alteração recente'
 ENTÃO provável componente físico não possui 'entidades origem e destino'

SE localização do problema = 'entre mesma sub-rede IP' E
 interface de rede é uma 'LAN' E
 segmentos envolvidos = 'sub-rede só possui um segmento/domínio de colisão'
 ENTÃO provável componente físico não possui 'equipamento interconexão enlace'

A elaboração dos componentes com falhas de roteamento e endereçamento IP está sendo também estudada. Futuramente, a elaboração de características para falhas nas camadas superiores pode também ser acrescentada e comparada de modo análogo ao realizado com as falhas físicas.

5.5.1.2 Elaboração dos Tipos de Problema Relacionados

A elaboração da característica referente aos tipos de problema relacionados à situação se baseia em dois pontos distintos: em regras associadas às características já informadas que buscam direcionar o tipo de problema para outros tipos prováveis e na relação histórica entre os tipos de problema.

O primeiro ponto — uso de regras associadas à algumas características — visa atingir aqueles registros onde o tipo de problema foi informado sem levar em consideração todos os dados já disponíveis no momento, dados estes que indicam um tipo de problema diferente. Assim, as regras visam direcionar o tipo de problema para o tipo ou os tipos de problemas que poderiam já ter sido informados. Um exemplo de um caso como este seria a criação de um registro como um problema de *aplicação* para uma situação em que o usuário não conseguiu estabelecer uma conexão de terminal virtual com uma estação remota, estação remota que não estava, contudo, acessível via rede. Assim, uma vez que a estação não está acessível, o tipo de problema informado preferencialmente deveria ter sido *conectividade*, já que a falha com a aplicação foi apenas um sintoma superficial do sintoma principal relativo à falta de acessibilidade.

Para estas situações, o sistema implementa um mecanismo em que associa para algumas informações solicitadas ao usuário regras que, dependendo do valor da resposta, privilegiam um tipo de problema e, no cálculo final, podem adicioná-lo à característica relativa aos tipos de problemas prováveis sendo elaborada. Algumas regras definidas são apresentadas a seguir, denominadas *regras redirecionadoras*.

SE problema mantém utilizando IP = 'não'
 ENTÃO tipo de problema 'serviços-resolução de nomes' PESO 3

SE falta de acesso = 'constante' ou falta de acesso = 'intermitente' E
 tipo de problema não é 'conectividade-genérico' E
 tipo de problema não é 'conectividade-físico e config/HW' E
 tipo de problema não é 'roteamento/endereçamento'
 ENTÃO tipo de problema 'conectividade-genérico' PESO 3

SE falta de acesso = 'intermitente' e tipo de problema != 'performance'
 ENTÃO tipo de problema 'performance' PESO 3

As características relacionadas a essas regras, como pode ser visto pelo exemplo, podem não ser relativas a todos os tipos de problemas, sendo utilizadas para apenas um

ou alguns tipos de problemas. A segunda regra apresentada, por exemplo, aplica-se somente para problemas que não sejam de tipo conectividade, a fim de identificar se estes problemas poderiam ter sido causados por uma falta de acesso ou se, por envolverem também uma acessibilidade, poderiam ter sido em outras situações cadastrados desta forma. De forma semelhante, a terceira regra é relacionada a problemas que não sejam de performance, a fim de identificar se, por ser uma falha intermitente, o problema poderia ter sido percebido desta forma. Neste caso, a função da regra é obter tipos de problemas que poderiam também ter sido escolhidos em outras situações, embora não sejam o sintoma principal.

Desta forma, essas regras visam analisar a situação em função das informações já disponíveis, a fim de elaborar quais tipos de problemas devem também ser consultados na base. O cálculo da probabilidade de cada tipo de problema segundo as regras redirecionadoras é feito da seguinte forma: cada tipo de problema tem peso inicial zero. O caso passa por todas as regras, e, para cada regra verdadeira, o peso da regra é somado ao peso do tipo de problema correspondente. Após todas as regras serem executadas, é feita a soma dos pesos resultantes de todos os tipos de problemas e o peso de cada tipo de problema é dividido pelo peso total, formando um valor para cada tipo de problema denominado **probabilidade segundo regras redirecionadoras**. Esse valor será relacionado à relação histórica para cálculo da probabilidade final do tipo de problema, conforme veremos a seguir.

O segundo ponto considerado para elaboração dos prováveis tipos de problema — que se baseia na *relação histórica entre os tipos de problema* — visa atingir os casos em que o tipo de problema real da situação só será identificado na fase final de diagnóstico da situação, sendo que os dados iniciais apontam para um tipo de problema superficial causado, na verdade, por outro. Um exemplo desse tipo de situação pode ser encontrado no caso 2, apresentado no anexo 3. Nesse caso, um problema de compartilhamento de arquivos mostrou, ao final, ser causado por um problema de autenticação. Assim, outros casos de autenticação seriam também similares ao problema corrente, que serviriam para indicar ao usuário as atividades de diagnóstico e soluções que poderiam ser empregadas. Portanto, casos de tipo *servicos-autenticação* deveriam também ser consultados.

Essas situações são tratadas fazendo-se uso da relação histórica da rede entre os tipos de problema, identificada através de uma tabela em que é armazenado para cada tipo de problema o percentual de vezes que este tipo de problema foi no passado causado pelos outros tipos. Busca-se, com isso, fazer o relacionamento de tipos de problemas que podem ser causados por outros, por utilizarem seus serviços (como no exemplo citado), e usar esta relação para a busca de uma nova situação. Essas relações são atualizadas a cada caso encerrado, e o sistema pode assim, ao longo da sua utilização, adaptar essas relações para o seu domínio em particular. O valor resultante para cada tipo de problema segundo esta relação histórica é chamado **probabilidade segundo a relação histórica**.

A elaboração da **probabilidade final** de cada tipo de problema é realizada utilizando a probabilidade segundo as regras redirecionadoras (se alguma das regras tiver sido aplicada) e a probabilidade segundo a relação histórica. Seu cálculo é feito dando maior importância à relação histórica, pois esta evolui com o uso do sistema e, com isso, torna-se mais adequada para o domínio em que está sendo usada. Atualmente, a

importância dada à probabilidade da relação histórica tem valor 3. A função para obtenção da probabilidade final é apresentada na figura abaixo.

$$prob(T, C) = \frac{pred(T, C) + I * phist(T, C)}{I + 1}$$

onde T : tipo de problema
 C : caso corrente
 $pred(T, C)$: probabilidade segundo regras redirecionadoras
 $phist(T, C)$: probabilidade segundo relação histórica
 I : importância da relação histórica

FIGURA 5.8 - Função para cálculo da probabilidade final de cada tipo de problema

O processo de elaboração se encerra com a seleção automática para busca dos tipos de problemas prováveis que possuem uma probabilidade maior que um valor pré-determinado (atualmente, o valor 0,2). Os tipos de problema não selecionados podem, porém, ser incluídos no grupo dos selecionados por determinação do usuário na etapa de reutilização e revisão, quando a avaliação da situação é realizada novamente. Assim, essa característica é formada por uma listagem dos diversos tipos de problemas, sendo, para cada um, associado um valor referente à probabilidade resultante para aquele tipo e a indicação de sua seleção ou não. Esses valores serão utilizados posteriormente no processo de recuperação.

Em posse da descrição do problema adicionada das demais características elaboradas nesta etapa de definição do contexto, o caso segue para o processo de busca da base de casos.

5.5.2 Recuperação: Busca e Classificação

O processo de recuperação envolve, após a definição do contexto, a busca na base pelos casos com potencial de serem similares à situação corrente e a seleção, dentre estes, dos casos com maior similaridade com esta situação. Esta similaridade não é, entretanto, uma similaridade realizada através de casamento exato: não se pode esperar que um caso na base de casos seja o casamento exatamente da nova situação, de modo que a recuperação deve resultar no *melhor casamento parcial*.

Todavia, a utilização de casamento parcial no processo de busca dos casos na base resulta num algoritmo de busca excessivamente dispendioso. Assim, deve ser buscado um modo eficiente para particionar o espaço de busca de modo que o casamento parcial completo seja feito em apenas um pequeno número de casos [KOL 93].

5.5.2.1 Algoritmo de Busca

O algoritmo de busca na base de casos utiliza-se de índices para buscar os casos na base segundo a estrutura organizacional desta, procurando cumprir seu objetivo: *selecionar um pequeno número de casos potencialmente relevantes, certificando-se ao mesmo tempo que no mínimo alguns casos que têm o potencial de serem os mais úteis estão no conjunto* [KOL 93].

Como foi comentado na seção 5.4.2, a base de casos neste trabalho foi organizada hierarquicamente segundo o tipo de problema dos casos (tipo de problema final). Esta organização visa selecionar do conjunto de todos os problemas armazenados aqueles problemas que dizem respeito à situação corrente por envolverem o mesmo tipo de problema.

A busca na base de casos deve almejar, contudo, que sejam selecionados ao menos um grupo dos casos com maior potencial de similaridade para a situação, os quais, como já foi discutido anteriormente, não é garantido que sejam do mesmo tipo de problema identificado no início da situação corrente. É, portanto, a fim de garantir este conjunto que é feita a elaboração dos tipos de problemas prováveis na definição do contexto, os com maior probabilidade são selecionados e estes tipos de problemas são então usados como índices na busca na base.

Assim, o algoritmo de busca percorre a base selecionando os casos que pertencem a cada um dos tipos de problemas identificados nos tipos de problema prováveis selecionados para busca. A figura a seguir esquematiza o processo.

```

nodo_atual = raiz
para cada nodo folha tipo_de_problema repetir
    se tipo_de_problema ∈ tipos_de_problemas_prováveis_selecionados
        nodo_atual = nodo folha
        se nodos folhas nodo_atual são casos
            selecionar casos
        senão
            nodo_atual = nodo folha cujo índice arco seja melhor casamento
            selecionar casos representados pelos folhas do nodo atual

```

FIGURA 5.9 - Algoritmo para recuperação na base de casos

Após o processo de busca, tem-se o grupo de casos que parcialmente casaram com a situação corrente. Esse grupo deve, então, ser submetido a uma avaliação mais compreensiva do grau de casamento de cada caso com o caso corrente, considerando agora todas as características do caso, a fim de que os casos mais similares sejam identificados e apresentados para o usuário. Este processo é conhecido como classificação.

5.5.2.2 Classificação: Similaridade entre Características

O cálculo confiável do grau de casamento entre casos envolve diversos processos: (i) a determinação de que características correspondem entre si e devem portanto ser comparadas, (ii) o cálculo do grau de casamento entre estas características e (iii) a identificação da importância de cada característica.

No sistema DUMBO, a maior parte das características possui uma associação direta (por exemplo, mensagem com mensagem, interface de rede com interface de rede), mesmo que não seja comparação das mesmas características (provável componente com falha física com componente afetado do caso recuperado, por exemplo). Assim, as características que serão comparadas já possuem correspondência direta, não sendo necessário determinar a correspondência entre características (etapa i). Resta, portanto, abordar o modo como será calculada a similaridade entre as características (ii) e o modo como foi determinada e como é registrada no sistema a relevância destas (iii).

A avaliação da similaridade de uma característica e, conseqüentemente, o cálculo do seu grau de casamento dizem respeito ao tipo da característica e conjunto de valores que ela pode assumir. Algumas suportam apenas casamento exato, seja por possuírem apenas dois valores válidos, seja por seus valores não possuírem similaridade alguma quando diferentes. Outras suportam casamento parcial. Nesse sistema, a partir da avaliação das características definidas para os casos e dos tipos de similaridade utilizadas nos sistemas apresentados na bibliografia [KOL 93], foram identificados seis tipos de características e, por conseguinte, seis tipos de similaridades que o sistema deve suportar. São elas:

- **características numéricas, similaridade numérica por distância de regiões qualitativas:** utilizada para características que assumem apenas valores numéricos. Esse método foi escolhido porque pequenas diferenças nos valores destas características são irrelevantes para o grau de casamento, sendo também utilizado em outros sistemas como o CASEY, de Koton *Apud* [KOL 93]. Nesse método, são estabelecidas regiões ou intervalos qualitativos que abrangem os possíveis valores numéricos para cada característica segundo o seu significado no domínio. O valor da característica é atribuído para o intervalo correspondente e o grau de similaridade é então calculado pela distância (número de regiões) entre os dois valores. Assim, dois valores que se encontram numa mesma região possuem grau de similaridade máximo, enquanto que valores que se encontram em regiões vizinhas possuem uma similaridade menor. A fim de acrescentar maior acurácia aos valores localizados próximos as bordas ou limites dos intervalos, estes possuem limites que se sobrepõem.
- **características binárias, similaridade exata:** utilizada em características cujas respostas incluem apenas o valor *SIM* e *NÃO*, esta similaridade admite apenas casamento exato. Geralmente, salvo aquelas características cuja resposta é exigida para o correto andamento do processo, o valor *NÃO_RESPONDIDO* é também aceito; neste caso, a similaridade não pode ser calculada. Entre as características que utilizam este tipo de similaridade podemos citar *há muito tráfego e ambiente tem múltiplos protocolos*.
- **características com termos fixos, similaridade qualitativa:** este tipo de similaridade ocorre com características que admitem vários termos (expressões) como respostas e cujos valores para resposta são previamente identificados e cadastrados no sistema. A similaridade entre eles é definida de acordo com o conhecimento do domínio, pela comparação qualitativa dos possíveis valores. Um exemplo de característica que utiliza essa similaridade é a *abrangência do problema na sub-rede origem*. Nessa característica, o valor *um único equipamento da sub-rede* é considerado mais similar quando comparado à *alguns equipamentos da sub-rede* do que quando comparado ao valor *todos equipamentos da sub-rede*.
- **características com termos variáveis, similaridade qualitativa:** semelhante à similaridade qualitativa de termos fixos, com a diferença de que novos termos podem ser acrescentados ao longo da utilização e aprendizado sofrido pelo sistema. Novos valores são inseridos, a fim de cadastrar termos tais como novas versões de sistemas operacionais, novos produtos, etc. A similaridade nessas características permite identificar uma maior similaridade entre, por exemplo, as distribuições Red Hat e Slackware (sistema operacional Linux) do que quando essas são comparadas ao sistema Solaris 2.x, ao mesmo tempo que permite demonstrar uma maior similaridade

entre estes produtos do que quando comparados a um sistema operacional Windows 95. Outro exemplo pode ser encontrado numa maior similaridade entre produtos de uma mesma família, tal como entre um roteador Cisco 2524 e um roteador Cisco 2525.

- **características com termos variáveis, similaridade exata:** utilizada em características que descrevem um determinado elemento da rede, como o número IP de um equipamento ou sub-rede. Estas características admitem apenas casamento exato.
- **característica com texto livre, similaridade textual por palavras chave:** utilizada para comparar a característica *breve descrição do problema*. Nesta característica, uma maior similaridade é identificada por um número maior de ocorrências comuns às duas descrições de expressões pré-cadastradas que possuem um valor semântico maior em uma descrição, tais como *cabeamento*, *TX*, *rotas*. Expressões iguais com variação em número, gênero e grau são cadastradas com grau de casamento 1, enquanto que expressões similares (como *par trançado* e *UTP*) representam uma similaridade em menor grau. Cada expressão está cadastrada em um ou mais conjuntos semânticos. O cálculo da similaridade entre duas característica é feito pela comparação de cada expressão cadastrada da descrição no problema corrente com a descrição do caso recuperado (sendo atualmente atribuído o valor 1 para a ocorrência de expressão igual e valor 0,7 para expressão similar), e pela soma do grau de casamento de todas as expressões comparadas. Uma expressão pode ser aprendida pelo sistema na fase de aprendizagem, quando o usuário informa as expressões similares entre as já cadastradas. Termos como *o*, *um* e *aquela* são cadastrados como termos que podem ser automaticamente anuláveis, e não são disponibilizados para aprendizagem.

Os diversos tipos de características do sistema acarretam em diferentes modos de cálculo da similaridade da característica, porém o grau de um casamento foi projetado visando que, independente como tenha sido calculado (por que tipo de similaridade), seja consistente com as demais características, como propõe Janet Kolodner [KOL 93]. Assim, um valor de casamento 0,75, por exemplo, representa o mesmo grau de similaridade independente da dimensão (característica) analisada e do método de cálculo de similaridade utilizado.

Para isso, os graus de similaridade possíveis entre as características, sejam estas com termos fixos ou variáveis, foram definidos como *total*(1,0), *alto*(0,75), *médio*(0,5), *baixo*(0,3), *nenhum*(0). As características do tipo *binárias* e do tipo *com termos variáveis e similaridade exata* admitem, por sua vez, o subconjunto de valores *total* e *nenhum*. As características numéricas possuem o grau adaptado para o número de regiões que a característica suporta, de modo que a diferença de *uma região* em uma característica com *cinco regiões*, por exemplo, seja menor que se a característica possuir, por exemplo, apenas *duas regiões*. A característica com similaridade textual, por sua vez, utiliza sua fórmula própria para atribuir o grau de casamento. A tabela a seguir resume os tipos de características e similaridades apresentados.

TABELA 5.1 - Tipos de similaridades das características do sistema

| Similaridade | Valores Possíveis da Característica | Tipo de Casamento | Meios de Cálculo de Casamento Parcial | Exemplos |
|-----------------------------------|---|-------------------|---|---|
| numérica | números reais positivos | exato e parcial | atribuição para regiões, cálculo pelo número de regiões entre valores | taxa de colisões, percentual de utilização da rede |
| binária | <i>SIM, NÃO</i> | apenas exato | - | ambiente tem múltiplos protocolos, há muito tráfego |
| qualitativa para termos fixos | termos (expressões) pré-definidos para a característica | exato e parcial | similaridade de termos pré-definida de acordo com conhecimento do domínio | abrangência do problema na sub-rede origem |
| qualitativa para termos variáveis | termos (expressões) cadastrados com possibilidade de novos termos | exato e parcial | similaridade de termos cadastrada com possibilidade de inserir novas relações para novos termos | sistema operacional, produto |
| exata para termos variáveis | termos (expressões) cadastrados com possibilidade de novos termos | apenas exato | - | identificação do equipamento, identificação da sub-rede |
| textual | texto livre | parcial | comparação de expressões similares cadastradas | breve descrição do problema |

5.5.2.3 Classificação: Relevância das Características

Uma vez que o modo como é feita a análise da similaridade das características já foi abordado (*ii*), resta ainda comentar a atribuição de valores de importância para as características do sistema (*iii*) antes de abordarmos o processo de classificação como um todo. Como comenta Kolodner, a função que calcula o grau de casamento de uma situação tem sua eficiência de acordo com o conhecimento que se possui da importância dos campos ou dimensões, que indica quais informações devem receber mais atenção — aquelas informações que contribuem mais para a similaridade entre casos.

Essa similaridade, como foi dito anteriormente, não representa necessariamente os casos *mais parecidos*, mas sim os casos que têm o maior potencial de *serem úteis* para contribuir para a solução daquela situação, seja por possuírem causas semelhantes, seja por necessitarem de ações de diagnóstico semelhantes. Assim, não basta para o cálculo da similaridade entre dois casos avaliar o número de características casadas de modo simples, pois isso desprezaria a maior relevância que algumas características possuem em indicar pontos importantes da situação [KOL 93][LEW 95].

A atribuição da importância das características do sistema foi desenvolvida através do estudo geral do domínio efetuado e dos casos coletados, quando foram identificadas informações ou combinações de informações (tais como alguns sintomas em especial ou condições do ambiente) mais aptas a indicar a similaridade entre casos.

Foram, assim, estabelecidos cinco graus de importância para as características de um caso:

- **características essenciais** (filtros), que eliminam o caso da lista dos selecionados se ele não possuir um grau de similaridade mínimo pré-determinado, que pode ser grau 1 ou um grau definido de similaridade parcial. As características essenciais representam as informações necessárias para o problema, sem as quais o caso não pode ser selecionado. Fazem parte desse grupo algumas das características específicas, a característica tipo de problema, que já é utilizada para a recuperação inicial, e as características que para alguns tipos de problemas constituem filtros, como o padrão de interface de rede para problemas de conectividade físicos e de configuração do hardware.
- **características com importância excessiva**, que representam informações que devem idealmente ser similares ao problema corrente por dividir geralmente o grupo de problemas em um subgrupo com características muito mais específicas. Fazem parte desse grupo características como tipo de interface de rede em problemas de comunicação em redes locais, que indicam situações com características particulares; tipo de servidor em problemas com serviços, etc.
- **características muito importantes**, tais como abrangência do problema e mensagem.
- **características importantes**, tais como equipamento envolvido, enlace ou sub-rede atingida.
- **características sem importância**, englobando as características com propósitos apenas de gerenciamento e que, portanto, não contribuem na identificação de situações similares.

Apenas algumas destas características são utilizadas no processo de classificação (características filtro, excessivamente importantes, muito importantes, importantes), sendo atribuído a cada uma destas um valor numérico que o sistema utiliza para o cálculo de similaridade do caso (valores 5, 5, 3 e 1, respectivamente). As demais não são utilizadas no processo de ordenamento dos casos. Essa abordagem, em que são associados graus de importância de acordo com valores fixos, não fornece uma determinação exata da importância das informações — muitas vezes até mesmo desconhecida já que o caso representa uma situação ainda incompreendida — mas captura a importância relativa dos problemas.

Entretanto, a atribuição de um valor simples estático para cada característica dos casos não é adequada para este sistema, já que, dependendo da situação enfrentada, a importância de algumas informações varia muito, sendo até mesmo relevante para alguns tipos de problemas e para outros não. Assim, um valor simples assinalado globalmente para cada característica não permitiria o julgamento de similaridade necessário no domínio, sendo necessário assinalar valores de importância que sejam relativos ao contexto da situação corrente para algumas características — as características adicionais do caso. Essa particularidade foi implementada assinalando-se diferentes graus de relevância para cada uma das características adicionais, cada grau correspondente a um contexto, representado no sistema pelo tipo de problema. A tabela abaixo apresenta a relevância de algumas características do sistema.

TABELA 5.2 - Relevância de algumas características

| Tipo de Problema | Relevância das Características |
|----------------------------------|--|
| Conectividade Genérico | modo falta acesso PESO 3 falta acesso PESO 3 padrão interface rede PESO 5 |
| Conectividade Físico e Config/HW | falta acesso PESO 3 padrão interface rede PESO 5 |
| Roteamento/Endereçamento | modo falta acesso PESO 3 falta acesso PESO 3 |
| Performance | modo falta acesso PESO 3 falta acesso PESO 3 performance ruim quando PESO 3 há muitos erros PESO 5 há muito tráfego PESO 3 há alto tempo resposta PESO 5 tipo alto tempo resposta PESO 3 padrão interface rede peso 5 |
| Alto Tráfego | modo falta acesso PESO 3 falta acesso PESO 3 alta taxa de tráfego quando PESO 3 alto tráfego em PESO 3 há muitos erros PESO 3 há muito tráfego PESO 3 há alto tempo resposta PESO 3 tipo tráfego alta taxa PESO 3 padrão interface rede PESO 5 |
| Aplicação | aplicação em uso PESO 3 abrangência do problema nas estações PESO 3 abrangência do problema para usuários PESO 3 |
| Serviços Compartilhamento Arq. | tipo de servidor arquivos PESO 5 |
| Serviços Autenticação | tipo de servidor autenticação PESO 5 |
| Serviços Resolução Nomes | tipo de servidor nomes PESO 5 |

Além do tipo de problema, a combinação de alguns valores para certas características é também responsável por indicar um diferente grau de relevância para algumas características. Essas inter-relações entre características são implementadas através de regras, que são aplicadas nos casos que possuem determinados valores para as características. Exemplo de uma característica que determina uma maior relevância é o *tipo de interface de rede*, que para o valor *Ethernet* indica uma maior relevância da característica *tipo específico de interface*. Três destas regras podem ser vistas a seguir. Exemplos adicionais destas regras são apresentados no anexo 2.

SE tipo de problema = 'conectividade-genérico' OU
 tipo de problema = 'conectividade-físico e config/HW' OU tipo de problema = 'performance' OU
 tipo de problema = 'alto tráfego' E provável componente falha física possui 'cabearmento' OU
 provável componente falha física possui 'equipamento interconexão enlace'
 ENTÃO tipo interface rede peso 5

SE tipo de problema = 'conectividade-genérico' OU
 tipo de problema = 'conectividade-físico e config/HW' OU tipo de problema = 'performance' OU
 tipo de problema = 'alto tráfego' E provável componente falha física possui 'cabearmento' OU
 provável componente falha física possui 'equipamento interconexão enlace' E
 tipo interface rede contém Ethernet
 ENTÃO tipo específico interface PESO 3

SE tipo de problema = 'conectividade-genérico' OU tipo de problema = 'roteamento/endereçamento' E
 localização do problema = 'entre mesma sub-rede IP'
 ENTÃO acessibilidade entre hosts e roteador PESO 1

As demais características do caso, tais como breve descrição do problema e mensagem, possuem valores de importância assinalados globalmente, independente do contexto da situação.

5.5.2.4 Classificação e Ordenamento

O processo de classificação dos casos consiste na avaliação do grau de similaridade entre cada um dos casos recuperados e o caso corrente, ordenando os casos em ordem decrescente de similaridade e selecionando os casos melhor classificados.

O processo inicia com a análise das *características específicas* do caso corrente. Assim, os casos recuperados são passados por filtros onde, para cada característica específica filtro do caso corrente, são eliminados os casos que possuem aquela característica e esta possuir similaridade inferior à mínima definida para ela. Foi estudada, também, a inclusão de filtros adicionais no problema, representados pelas características extremamente relevantes. Optou-se, porém, por não fazer a eliminação dos casos por essas características e, sim, utilizá-las atribuindo um alto valor de relevância na função de avaliação, direcionando, assim, os casos com essas características para os casos melhores classificados.

A etapa seguinte consiste então na avaliação numérica dos casos recuperados que foram mantidos após passarem pelo filtro das características específicas. Essa avaliação é efetuada utilizando um cálculo em que é considerada a similaridade entre todas as características do caso: tipo de problema, breve descrição do problema, demais características gerais acerca do problema, características adicionais, características específicas.

A similaridade do tipo de problema é obtida pela probabilidade do tipo de problema do caso recuperado resultante da elaboração da característica tipos de problemas relacionados (vide seção 5.5.1.2). As características gerais acerca do problema, assim como a breve descrição, possuem grau de relevância igual para todos os problemas. As características adicionais, por sua vez, possuem grau de relevância relativo ao contexto, conforme comentado na seção anterior, sendo utilizado o contexto do *caso recuperado*. Por fim, as características específicas são utilizadas no casamento segundo sua relevância.

O cálculo da similaridade entre o caso corrente e cada caso recuperado é realizado pelo somatório da similaridade entre *cada característica com valor elaborado* multiplicado pela *relevância desta para o contexto do caso recuperado*, conforme apresenta a função de avaliação numérica na figura abaixo.

$$\text{Similaridade}(Cr, R) = \frac{\sum_{i=1}^n Wi * \text{sim}(f_i^C, f_i^R) * Ci}{\sum_{i=1}^n Wi * Ci}$$

onde Wi : importância da característica i
 sim : função de similaridade para a característica
 f_i^C e f_i^R : valores para a característica f_i no caso corrente e caso recuperado
 Ci : confiança da similaridade da característica i , sendo 0 para valor elaborado, 1 para valor não elaborado

FIGURA 5.10 - Função de avaliação numérica utilizada para a classificação dos casos recuperados

Entretanto, quando uma característica não foi respondida, ela não é calculada no fator similaridade do caso — a similaridade contabiliza apenas aquelas características que podem ser comparadas, aquelas que estão elaboradas (representadas pelo fator 1 para *confiança* na característica), indicando a similaridade entre os dados disponíveis. Percebeu-se, com isso, a necessidade da criação de um outro fator que levasse em conta o quanto a similaridade entre dois casos calculada é confiável, o quanto ela representa de fato a similaridade confirmada no caso. Esse fator foi chamado de *confiabilidade*.

A confiabilidade é calcula pelo somatório das relevâncias das características que foram contabilizadas na similaridade (as que estavam elaboradas) dividido pelo somatório das relevâncias de todas as características que deveriam ter sido contabilizadas. A função para cálculo da confiabilidade é expressa na figura abaixo.

$$\text{Confiabilidade}(Cr, R) = \frac{\sum_{i=1}^n Wi * Ci}{\sum_{i=1}^n Wi}$$

onde Wi : importância da característica i
 Ci : confiança da similaridade da característica i , sendo 0 para valor elaborado, 1 para valor não elaborado

FIGURA 5.11 - Função para cálculo da confiabilidade do casamento entre dois casos

Por fim, é feito o ordenamento dos casos de acordo com o resultado da similaridade e confiabilidade de cada um. A função para cálculo do fator de ordenamento, utilizando a confiabilidade e similaridade é apresentada na figura abaixo, sendo atualmente utilizado o valor 2 para a importância da similaridade.

$$FatorOrdenamento(R) = \frac{I * Similaridade(Cr, R) + Confiabilidade(Cr, R)}{I + 1}$$

onde *Similaridade(Cr,R)*: similaridade entre os casos
Confiabilidade(Cr,R): confiabilidade do casamento entre os casos
I: importância da similaridade frente a confiabilidade

FIGURA 5.12 - Função para cálculo do fator de ordenamento

5.5.3 Reutilização e Revisão

Após os melhores casos serem selecionados entre os casos recuperados da base, a experiência destes deve ser reutilizada na situação corrente — seja a experiência referente à solução encontrada, seja a experiência referente à identificação de que informações ou ações de diagnóstico devem ser aplicadas à situação para levar à solução do problema.

Os casos recuperados num primeiro ciclo de execução, contudo, têm a descrição formada apenas por características gerais dos problemas e gerais do seu tipo de problema, não tendo ainda sido identificadas informações específicas para o problema corrente. A fim de tratar isso, o sistema oferece nessa etapa um processo de refino da recuperação, fazendo com que o usuário possa optar por aplicar uma solução no ambiente real apenas depois que todas as informações sobre a situação recuperada sejam similares à situação corrente.

Esse processo é realizado através de um refinamento do contexto, que permite que o módulo de recuperação garanta casos mais similares, já que a nova descrição do problema provê um maior entendimento da situação corrente. Esse processo pode ser usado, também, depois que uma solução proposta já tenha sido avaliada no mundo real e não tenha obtido sucesso, fornecendo meios para o aprimoramento da recuperação de um caso melhor para a situação corrente.

Optou-se, nesse sistema, por esta abordagem, em que o usuário não necessita utilizar obrigatoriamente o processo de refino, para permitir uma maior flexibilidade de uso do sistema para especialistas. Os usuários podem, portanto, por conhecimento especialista próprio, optar por aplicar imediatamente no ambiente real uma solução proposta que não tenha todas as informações validadas, seja por avaliar que esta solução se adequa perfeitamente à situação corrente, seja por conhecer mecanismos de adaptação que permitam reutilizá-la mesmo sem dispor das informações específicas sobre os casos recuperados que são propostas pelo processo de refino.

Nesse caso, o refino não precisa ser utilizado e a experiência proposta é reutilizada diretamente, sendo aplicada ao ambiente real e avaliados os seus resultados. Nessa versão do sistema não foi desenvolvido um processo de adaptação automática pelo sistema da solução proposta — mecanismo que, como aponta Watson, é freqüentemente efetuado em muitas aplicações apenas pelos usuários [WAT 97]. Desta forma, os casos recuperados são oferecidos aos usuários a fim de que sua solução possa ser reutilizada no problema corrente, podendo, quando necessário, ser adaptada pelo usuário (adaptação baseada em crítica [LEW 95]).

Se os resultados da avaliação no ambiente real mostrarem que o problema foi solucionado com a experiência reutilizada, o caso corrente deve ser encerrado, sendo ou não aprendido pelo sistema (dependendo da sua solução ter sofrido ou não alguma alteração pelo usuário). Se, entretanto, os resultados da experiência proposta indicarem que a experiência não pôde ser aplicada, uma nova solução deve ser proposta. Para isso, é aplicado o processo de refino. A figura a seguir esquematiza o processo.

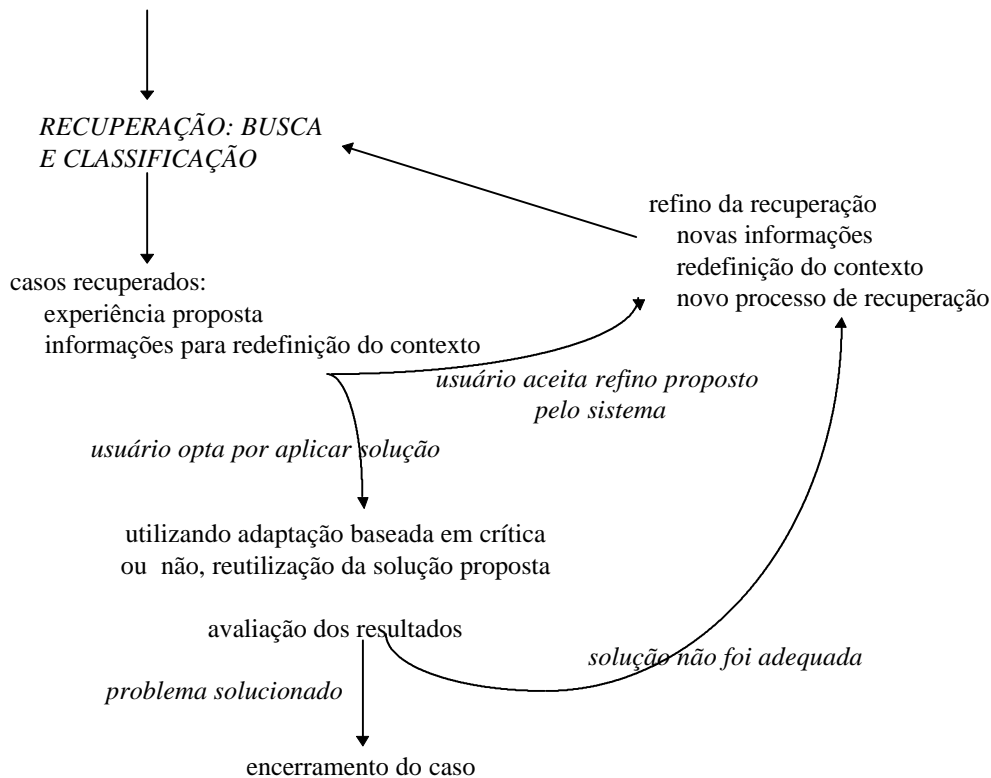


FIGURA 5.13 - Esquema do processo de reutilização e revisão do sistema

5.5.3.1 Redefinindo o Contexto

O processo de refino consiste numa nova avaliação da situação, através de um processo de redefinição do contexto, já utilizado em outras aplicações [SIM 92] [KOL 93]. Na abordagem adotada nesse sistema, o processo é realizado dinamicamente, definindo novas elaborações úteis baseado nas similaridades entre os casos recuperados. Assim, uma elaboração é escolhida porque identifica uma informação específica dos casos recuperados que se avalia ser também útil para melhor descrever o problema corrente, e que com isso contribuirá por selecionar um subconjunto melhor definido de casos similares entre os casos recuperados.

Para isso, são extraídas dos casos recuperados características que contribuíram, na situação anterior, para a evolução do diagnóstico do problema, a fim de que a descrição do caso atual seja melhor definida por possuir também estas informações (que são implementadas através das características específicas, comentadas anteriormente na seção 5.4.1). Para que as informações referentes a essas características sejam fornecidas pelos usuários, é muitas vezes necessário que estes efetuem ações de diagnóstico para possuí-las. Assim, essa abordagem aplicada no domínio de gerenciamento de redes contribui de modo conjunto para o fornecimento de uma melhor definição do problema e

para a identificação de ações que podem ser efetuadas no problema corrente, permitindo a evolução do diagnóstico desta situação, já que fornecem novos dados que produzem um andamento do processo de diagnóstico.

O processo inicia com a identificação de todas as características específicas dos casos fornecidos pelo módulo seletor. A seleção de que características devem então ser elaboradas e a ordem como estas características serão solicitadas é realizada considerando quatro fatores: *(i)* o custo de obtenção da informação, *(ii)* a probabilidade desta característica contribuir no problema, *(iii)* condições desta característica influenciar um maior número de casos e *(iv)* a ordem em que esta informação foi fornecida no caso armazenado.

O custo da obtenção da informação *(i)* diz respeito à dificuldade do usuário em obter a resposta para a informação solicitada. Nesse princípio, características que podem ser informadas mais facilmente são preferidas frente àquelas que envolvem ações mais demoradas ou complexas de serem efetuadas. A probabilidade da característica contribuir para o problema *(ii)* é identificada pelo grau de similaridade do caso recuperado que a indicou, ou, em caso de estar presente em vários casos, pelo grau de similaridade do melhor caso. Dessa forma, características resultantes dos casos com maior similaridade são preferidas frente às características dos demais, já que esses casos possuem uma maior probabilidade de serem úteis para o problema corrente.

O número de vezes que uma característica aparece nos casos selecionados é também considerado. Esse fator permite identificar as características com condições de influenciar um maior número de casos *(iii)*, que são preferidas em relação às características que aparecem em menos casos. Por fim, para selecionar a ordem em que as características são solicitadas é também considerada a ordem em que uma característica foi informada no caso recuperado *(iv)*. Esse princípio visa privilegiar a ordem original dessas informações nos casos armazenados, evitando que uma informação que é relevante por resultado de outra seja solicitada antes da que indicou sua relevância. Nessas situações, ambas características podem ser solicitadas, mas a ordem em que elas foram informadas no caso é mantida. Juntos, estes quatro fatores contribuem na escolha das características que serão elaboradas.

Uma vez que as características são selecionadas, elas são solicitadas ao usuário. Não foram concebidos, nesta versão, meios de obtenção das características específicas automaticamente, a menos aquelas características específicas que são operações sobre uma característica adicional. O usuário pode informar aquelas que achar mais conveniente, respondendo apenas às primeiras ou pulando algumas, conforme seu conhecimento e condições de obtê-las no momento. Uma vez informadas, a descrição do caso corrente é refinada com as novas características e uma nova recuperação é efetuada. Com essas características, as soluções propostas anteriormente, através dos casos recuperados, podem ser validadas ou consideradas impróprias pelo sistema, podendo, portanto, ser selecionados novos casos, ser eliminados casos selecionados anteriormente ou alterando o grau de similaridade dos casos já recuperados.

Após os novos casos serem recuperados, um novo processo de refino pode ser solicitado ou uma solução proposta pode ser reutilizada se for julgada adequada, sendo aplicada no ambiente real diretamente ou sofrendo algumas adaptações pelo usuário. Se o problema for solucionado, o caso é encerrado e o sistema solicita do usuário se a solução foi utilizada diretamente ou se foi modificada — devendo ser, então, aprendida

pelo sistema. Se, entretanto, o problema não foi solucionado, um novo refino pode ser aplicado, reiniciando o ciclo de recuperação.

5.5.4 Aprendizado

Após o caso ser solucionado, ele é encerrado podendo ou não ser aprendido pelo sistema. Um caso é aprendido quando ele representa uma experiência nova, uma experiência para a qual o sistema não foi capaz de propor uma solução adequada — seja porque o sistema propôs uma solução que precisou ser adaptada, seja porque não propôs a melhor solução para a situação. Nesses casos, a experiência obtida com o processo deve ser retida no sistema através de um novo caso.

Se, entretanto, a solução proposta pelo sistema foi adequada, o caso não precisa ser aprendido como um caso para CBR. Ele deve, porém, ser armazenado no sistema como um registro simples a fim de permitir que sejam mantidas no sistema as funções colaborativas de sistemas de registro de problemas tradicionais (tais como análise estatística dos equipamentos da rede, análise da produtividade do centro de gerência, criação de relatórios estatísticos de controle de qualidade, etc.). Para isso, o registro de problema (caso corrente) é encerrado, sendo atualizado com as informações referente ao encerramento de um registro (tais como data, hora, causas, tipo de problema real, solução, autor) e registrado como um registro simples, que não será utilizado nos processos CBR subsequentes. A única modificação no sistema diz respeito à atualização da relação entre os tipos de problemas.

O encerramento de um caso que será aprendido, contudo, exige algumas etapas adicionais. Inicialmente, o registro-caso é atualizado com as informações referentes ao encerramento, de modo similar ao encerramento de um registro simples. Após esta etapa, as informações do caso devem ser confirmadas e novas informações devem ser fornecidas, tanto para o caso como para o aumento do conhecimento geral do sistema.

Em primeiro lugar, as características adicionais sobre o tipo de problema final que ainda não tenham sido informadas pelo usuário são solicitadas e as demais são apresentadas para que sejam confirmadas. Isso significa, portanto, que se o tipo de problema real (informado no encerramento do registro) for diferente do tipo de problema inicial, as características adicionais que não forem comuns aos dois serão solicitadas. Nesta etapa, são também solicitadas informações que não tenham sido fornecidas inicialmente por representarem uma nova informação — tais como um novo sistema operacional, um novo produto, uma nova aplicação. Para isso, o novo termo deve ser informado e os demais termos daquela dimensão (referentes a mesma característica) já cadastrados são apresentados, a fim de que a similaridade do novo termo com os demais seja informada, utilizando-se para isso dos conceitos referentes ao grau de similaridade *total*(1,0), *alto*(0,7), *médio*(0,5), *baixo*(0,3), *nenhum*(0), sendo este último o grau atribuído quando o grau entre dois termos não for fornecido.

Embora esta abordagem exija um certo conhecimento especialista do usuário, pois exige que o usuário relacione o novo termo com os demais, ela é necessária porque permitirá a evolução do sistema com a utilização de novas tecnologias (representadas pelos novos termos dentro de cada categoria). Entretanto, essas informações não apresentam de modo geral uma complexidade elevada (já que muitas vezes novos elementos representam novas versões, ou são desenvolvidos como pequenas modificações de outros elementos, ou mesmo porque a categoria a que eles fazem parte

é conhecida usualmente pelos usuários — que sabem, por exemplo, que um sistema operacional Solaris é um sistema operacional Unix, informação necessária ao se acrescentar este sistema operacional ao sistema). Portanto, acredita-se que essa abordagem possa permitir uma certa evolução do conhecimento do domínio, ao passo que se não fosse utilizada, o sistema acabaria por apresentar uma gradativa perda de desempenho, mesmo com a inserção de novos casos, já que as informações solicitadas são aquelas que se acreditam ser importantes para diferenciar problemas, e esta comparação baseia-se sobretudo na similaridade de seus termos.

Por fim, é necessária a manipulação sobre as características específicas do caso. Para isso, devem ser indicadas quais dentre as características específicas já informadas não seriam necessárias para a evolução do problema — aquelas, portanto, que foram solicitadas pelo sistema, foram respondidas, mas que ao final pôde ser identificado que não contribuíram de modo relevante para a evolução do problema, e não seriam necessárias para que o problema fosse identificado e a solução encontrada. Essas características serão identificadas como não relevantes no caso e não serão utilizadas para propor novas informações no processo de reutilização nem serão utilizadas como filtros.

Além de identificar quais informações não foram relevantes, o encerramento do problema permite ainda identificar informações relevantes que não tenham sido informadas na descrição do problema. Informações, portanto, que são importantes para descrever o problema e devem ser inseridas para que um caso similar seja identificado. Para isso, é apresentado ao usuário o conjunto de características específicas cadastradas, assim como os possíveis valores para cada uma, e as características desejadas podem ser inseridas no caso. Podem existir, entretanto, informações que não estejam representadas pelas características cadastradas: nessa circunstância, é permitido ao usuário inserir uma nova característica específica no caso. Nesta versão do sistema, entretanto, não foram identificados meios para controlar uma inserção de características específicas que seja feita livremente. Assim, é aceito apenas que sejam cadastradas características específicas do tipo característica binária (vide seção 5.5.2.2), deixando-se para uma próxima versão o estabelecimento de processos que controlem a criação de características com termos variáveis (e, portanto, controlem seus termos e a similaridade entre eles).

5.6 Considerações Finais

Este capítulo apresentou o modelo proposto para o sistema DUMBO. A representação dos problemas do domínio no qual o sistema é validado foi abordada. A representação dos casos e a organização da base de casos utilizada para o sistema foram comentadas, bem como as abordagens concebidas para os processos de raciocínio.

No capítulo a seguir, apresentaremos o protótipo desenvolvido para validação deste modelo, comentando aspectos de sua implementação e a avaliação de seus resultados.

6 Protótipo do Sistema

Este capítulo visa, com os subsídios fornecidos pelos capítulos anteriores, apresentar a arquitetura do protótipo desenvolvido para o sistema DUMBO, os resultados obtidos por ele e comentar sobre sua comparação com os outros sistemas similares encontrados na bibliografia.

6.1 Implementação

A complexidade e variedade das falhas em redes possibilita que ocorram situações que atinjam mais de um domínio de gerência. Essas situações são tratadas pela interação dos especialistas dos diversos domínios envolvidos no problema. Se uma situação for facilmente solucionada pelo sistema, essa interação se torna reduzida. Em ocorrências que não possuam antecedentes na rede, entretanto, o diagnóstico pode levar horas ou mesmo dias, exigindo, dessa forma, uma grande interação entre os especialistas. A fim de suportar esta interação, a interface do ambiente é apoiada pelo ambiente WWW, através de CGI. No ambiente disponibilizado, os diversos especialistas podem interagir com o sistema criando novos registros ou anexando novas informações a um registro corrente (aberto) e solicitando novas consultas ao sistema CBR, que pode aprimorar seu processo de raciocínio com os novos dados fornecidos.

Para disponibilizar o acesso aos vários usuários de diferentes domínios, o sistema possui um cadastro destes e apenas usuários autorizados têm acesso à criação de registros e interação com registros de situações correntes. O aprendizado de um novo caso é também controlado, sendo autorizado apenas para usuários com nível de acesso superior. Isso é necessário devido à relevância das informações fornecidas no aprendizado de um novo caso, as quais devem ser fornecidas corretamente para não prejudicarem o conhecimento do sistema.

O sistema foi implementado em plataforma Unix, em sistemas SunOS4.x e Linux. Foi utilizada a linguagem C. O banco de dados utilizado foi o POSTGRES [RHE 90][STO 90], escolhido por ser utilizado pelos projetos do grupo e pelo sistema de registro de problema do ambiente CINEMA e por ser de domínio público, ampliando assim a portabilidade do sistema.

Comentaremos a seguir o protótipo implementado. Utilizaremos para a especificação do protótipo a Linguagem de Descrição e Especificação SDL (*Specification and Description Language*). Detalhes sobre esta linguagem podem ser encontrados em [TRI 92].

6.1.1 Arquitetura do sistema

O sistema foi projetado numa arquitetura cliente-servidor. Os módulos cliente são representados pelos diversos módulos CGI, que são responsáveis pelas funções de interfaces e pequenos procedimentos relativos a ela. O raciocínio principal do sistema é implementado pelo módulo servidor, que é consultado e atualizado pelos diversos módulos CGI clientes.

Aos módulos CGI cabe a interface com o usuário. Através deles, é feita a solicitação de criação de um novo registro-caso, a coleta de informações sobre um caso

corrente do usuário seguindo as indicações providas pelo servidor, a apresentação dos casos recuperados e a solicitação de novas informações para refino, a inserção de notas e de novas informações, o encerramento de um caso. O módulo servidor, por sua vez, é o responsável por processar os dados, criando novos casos, realizando os procedimentos de recuperação de casos similares, selecionando as características específicas mais relevantes, etc. A figura abaixo apresenta a definição do sistema.

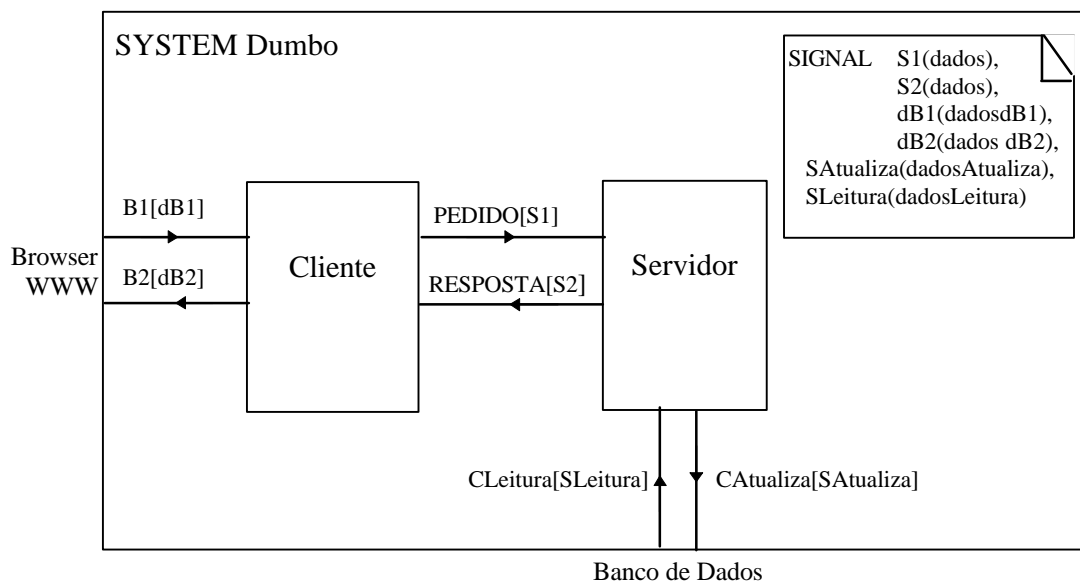


FIGURA 6.1 - Diagrama do sistema Dumbo

Os clientes, para executarem suas tarefas, fazem uso de funções para a comunicação com o servidor e para a interface com o usuário, além do processamento das informações fornecidas. O diagrama do bloco cliente pode ser visto na figura a seguir.

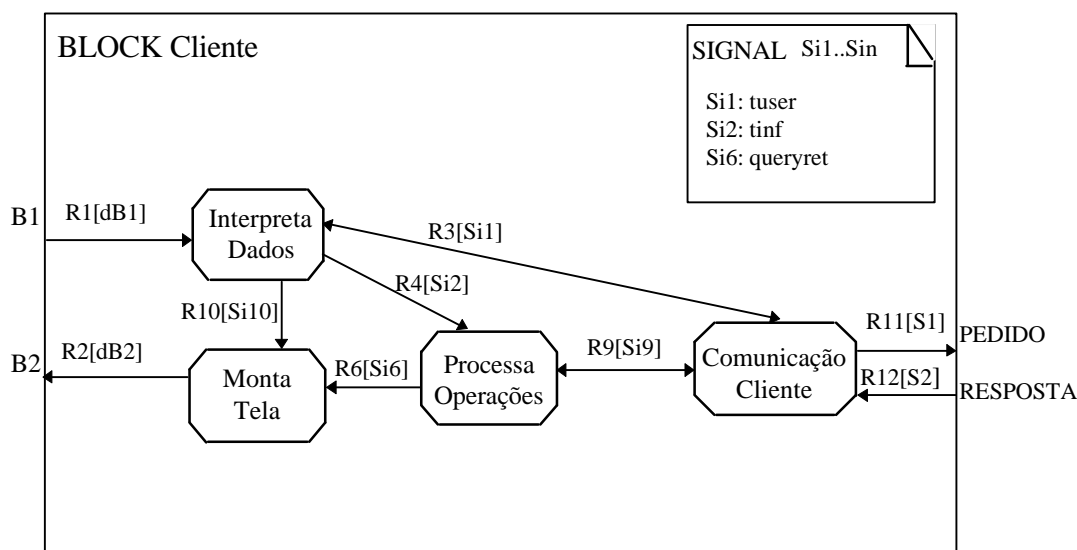


FIGURA 6.2 - Diagrama do bloco cliente

Cada cliente é formado por quatro módulos. Um primeiro módulo é responsável pela interpretação dos dados no formato CGI, denominado *InterpretaDados*. Após a

interpretação dos dados fornecidos pelo usuário, o cliente faz a validação do usuário, consultando o servidor, através do módulo *ComunicaçãoCliente*. Pela estrutura dos programas CGI e por questões de segurança, cada cliente faz, de modo transparente ao usuário, a validação do usuário. Quando um usuário não é validado, uma mensagem é apresentada ao usuário e o cliente termina suas tarefas, não fazendo nenhuma alteração sobre o sistema. Em situações normais, entretanto, um usuário não será validado apenas no programa CGI de entrada no sistema, sendo o mecanismo completamente transparente ao usuário nos programas posteriores.

Após o usuário ser validado, a comunicação estabelecida com um processo filho do bloco servidor é mantida e utilizada pelo cliente até que nenhuma consulta ao servidor seja necessária, quando uma mensagem de encerramento é enviada e o processo filho do bloco servidor é finalizado. Após a validação, o cliente passa ao módulo *ProcessaOperações*, que é o responsável por realizar as operações referentes à função do programa CGI em execução. Essas funções incluem entrada no sistema, opção por criação de um novo registro de problema, fornecimento das informações iniciais sobre o problema, fornecimento das informações adicionais sobre o problema, solicitação da recuperação de caso similar, encerramento de um registro de problema, etc. Cada programa, de acordo com sua função, pode executar uma ou várias consultas ao servidor, fazendo leituras e enviando atualizações de informações do banco, solicitando consultas da característica prováveis tipos de problemas, solicitando os casos similares, entre outras.

Após todas as operações necessárias serem executadas pelo módulo, ele solicita ao módulo *ComunicaçãoCliente* o envio de uma mensagem de encerramento e envia ao módulo *MontaTela* os dados resultantes das suas operações, que devem ser apresentados ou solicitados ao usuário. Os diagramas dos módulos do bloco cliente podem ser vistos no anexo 4.

O bloco servidor, por sua vez, é o responsável pela interface com o banco de dados e pela maior parte das funções de raciocínio do sistema. Esse bloco está estruturado em quatro módulos: módulo responsável pela comunicação, denominado *ComunicaçãoServidor*; módulo responsável pelo encaminhamento das solicitações, denominado *Servidor*; módulo *CBR*, responsável pela execução dos processos de raciocínio do sistema e módulo *InterfaceBD*, responsável pelo acesso ao banco de dados. A figura a seguir apresenta o diagrama do bloco servidor.

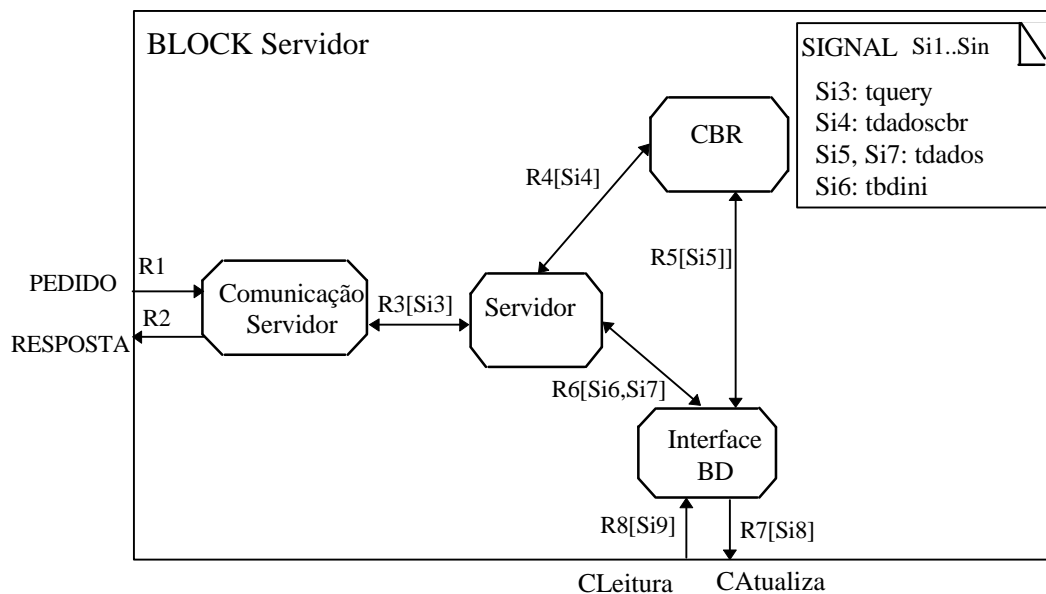


FIGURA 6.3 - Diagrama do bloco servidor

O módulo responsável por coordenar os pedidos recebidos e distribuí-los para os módulos correspondentes é o módulo *Servidor*, apresentado na figura a seguir. Inicialmente, no momento da ativação do programa, o módulo *Servidor* envia para o banco de dados os procedimentos de inicialização do banco, de acordo com o número IP e a porta de comunicação com o banco, além de selecionar a base que será consultada (base dumbo). Após esses procedimentos, ele entra num estado em que fica aguardando mensagens do módulo *ComunicaçãoServidor* com consultas solicitadas pelo bloco cliente.

Quando uma mensagem é recebida, para que o servidor possa atender outros pedidos de outros clientes, ele cria um processo filho para tratar a mensagem recebida e efetuar as consultas solicitadas pelo cliente. A primeira mensagem recebida traz os dados do usuário e sua senha de acesso ao sistema. Com essas informações, o servidor consulta o módulo *InterpretaBD* sobre o acesso do usuário, e envia para o *ComunicaçãoServidor* se o usuário foi ou não validado. No caso do usuário não ter sido aceito, após o envio do sinal ao *ComunicaçãoServidor*, o processo filho é finalizado e nenhuma outra consulta do cliente será aceita enquanto nova validação não for realizada. No caso do usuário ter sido aceito, o processo filho entra num estado em que aguarda consultas vindas do cliente, enviando-as para o módulo *InterpretaBD* ou *CBR* de acordo com o seu tipo. As consultas do tipo BD representam operações de leitura, criação ou atualização do banco de dados diretamente, sem necessitar que os dados sejam manuseados pelo sistema. As consultas do tipo CBR, entretanto, representam operações que requerem tratamento do sistema, representando funções que fazem parte dos processos do ciclo CBR.

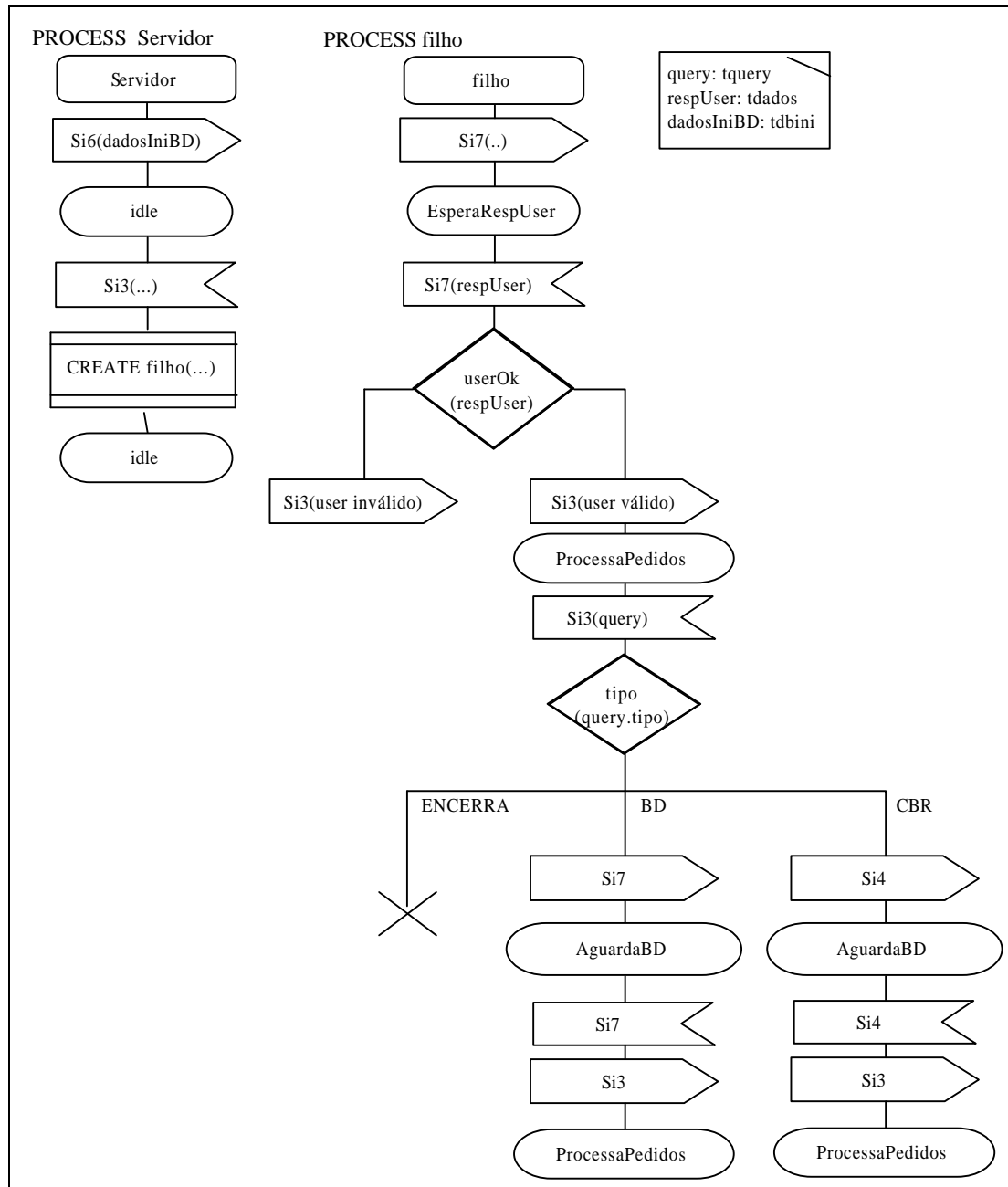


FIGURA 6.4 - Definição do Servidor

O módulo *InterpretaBD* é apresentado na figura A4.5, no anexo 4. Este módulo é o responsável pela interface com o banco de dados, que é realizada através da linguagem POSTQUEL [STO 90][RHE 90]. O módulo é composto por procedimentos que acessam as diversas tabelas do sistema DUMBO, realizando procedimentos de leitura de dados, criação de novos registros no banco e atualização de dados existentes.

A base de dados do sistema DUMBO é composta por diversas tabelas que armazenam os dados gerais do sistema, tais como dados de controle e usuários, e a base de conhecimento, formada pelos registros-casos e pelo conhecimento utilizado pelo sistema, representado pelas informações de similaridade entre os dados, pela relação histórica entre os tipos de problemas, etc. A tabela abaixo apresenta as principais tabelas presentes no sistema e os dados que elas armazenam.

TABELA 6.1 - Tabelas do banco de dados do sistema DUMBO

| Tabela | Descrição |
|---------------|---|
| USER | Informações sobre os usuários cadastrados no sistema. |
| CONTROLE | Informações de controle do sistema. |
| TTMANAG | Dados dos casos armazenados com as informações com caráter de gerenciamento. |
| TTCBR | Dados dos casos armazenados com as informações utilizadas pelo raciocínio do sistema. |
| TIPOSREL | Relação histórica entre os tipos de problemas. |
| TTCARAC_ESP | Características específicas dos casos armazenados. |
| CAD_CARACESP | Cadastro das características específicas do sistema. |
| TIPOSEQUIP | Cadastro dos tipos de equipamentos já informados. |
| SISOPER | Cadastro dos sistemas operacionais já informados. |
| PRODUTOS | Cadastro dos produtos já informados. |
| FUNCOES | Cadastro das funções dos equipamentos já informadas. |
| FUNCTIPO | Cadastro das relações entre as funções e tipos dos equipamentos. |
| PRODSO | Cadastro das relações entre produtos e sistemas operacionais. |
| ESTADOSITF | Cadastro dos estados de interface já informados. |
| TIPOSITF | Cadastro dos tipos de interfaces já informados. |
| NUMCADASTRO | Cadastro da similaridade das características numéricas. |
| TERMOS | Cadastro dos termos registrados no sistema para similaridade da característica breve descrição. |
| TERMOSNULOS | Cadastro dos termos sem significado especial, registrados como nulos. |
| TERMOSSIM | Informações sobre a similaridade entre termos. |

A tabela USER e CONTROLE armazenam dados gerais do sistema. Os casos são formados pelas tabelas TTMANAG, TTCBR e TTCARAC_ESP. As tabelas TIPOSEQUIP, SISOPER, PRODUTOS, FUNCOES, ESTADOSITF, TIPOSITF, FUNCTIPO e PRODSO contêm o cadastro das informações dos diversos tipos de características presentes no sistema. Estas tabelas são utilizadas no momento da inserção da característica do tipo correspondente em um novo caso, a fim de apresentar os elementos já cadastrados, e no momento do casamento entre casos, quando suas informações sobre a similaridade entre os diversos elementos são utilizadas. Um exemplo pode ser visto na tabela PRODUTOS, que, além do nome do produto, permite armazenar sua família. Produtos diferentes mas de uma mesma família possuem similaridade maior que produtos diferentes que não fazem parte da mesma família.

Outro exemplo pode ser observado na tabela SISOPER, que permite armazenar, além do sistema operacional, sua família e seu tipo geral. Assim, por exemplo, dois sistemas operacionais Solaris podem ser cadastrados como pertencentes a uma mesma família, e um sistema operacional Solaris e um Linux podem ser cadastrados como do mesmo tipo geral Unix. Com isso, se dois produtos forem diferentes, e sua família também, a similaridade pelo tipo (com valor inferior ao da similaridade da família) ainda é permitida. Além destas, a tabela NUMCADASTRO contém informações sobre os intervalos que são utilizados para a similaridade entre características numéricas.

A tabela TIPOSREL contém as informações sobre a relação histórica entre os tipos de problema. Por fim, as tabelas TERMOS, TERMOSNULOS e TERMOSSIM contêm informações sobre termos de uma descrição que são utilizados para fazer o casamento do campo breve descrição.

Por fim, o bloco servidor é composto pelo módulo *CBR*, o módulo responsável por executar aquelas funções de raciocínio do sistema que são executadas pelo bloco servidor (e não pelos clientes CGI). Para a execução destas funções, podem ser necessárias uma ou mais consultas ao banco de dados (através do *InterpretaBD*), como demonstra o diagrama na figura abaixo, variando de acordo com a função solicitada.

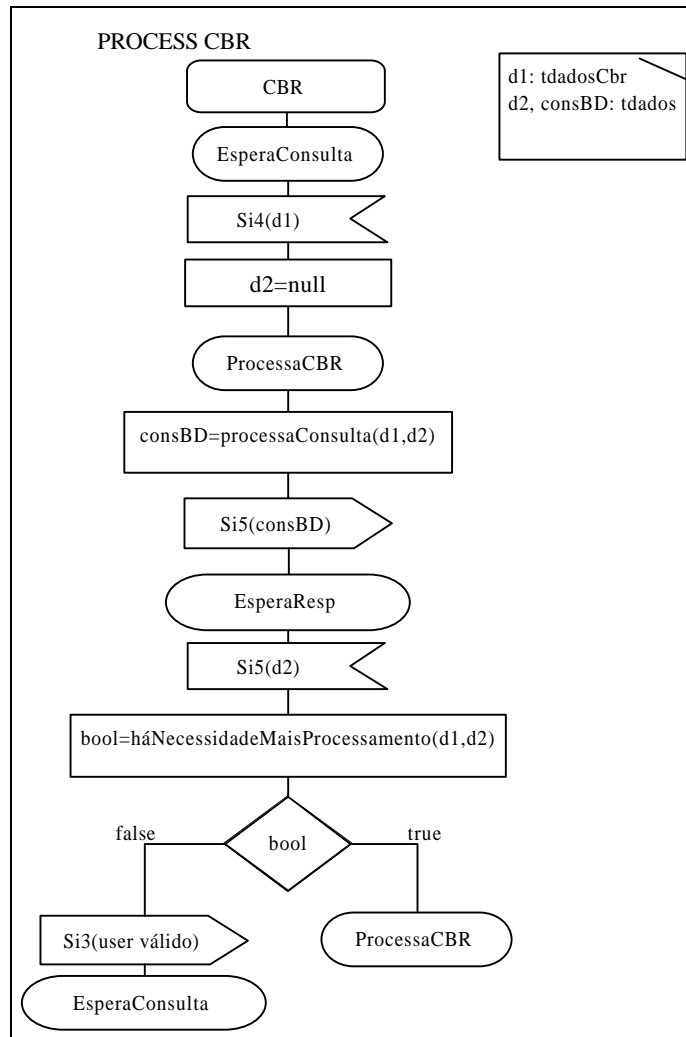


FIGURA 6.5 - Definição do CBR

As funções executadas pelo módulo *CBR* incluem a identificação dos prováveis tipos de problemas e a recuperação de casos similares e de características específicas relevantes. A identificação dos prováveis tipos de problemas faz parte do processo de **definição do contexto**, apresentado na seção 5.5.1. Esse processo também compreende funções executadas pelo cliente, que faz a elaboração das informações iniciais e características adicionais sobre o problema, e solicita então a **relação dos prováveis tipos de problemas** ao servidor para apresentá-la ao usuário no momento da recuperação.

A solicitação da **recuperação de casos similares** pelo cliente, por sua vez, inclui um conjunto de operações em que são feitas consultas às informações do banco e processamento sobre estas. Inicialmente, o módulo recebe o identificador do caso que deseja recuperar e os tipos de problemas que deve consultar (que podem ter sido alterados pelo usuário, embora mantenham o valor para casamento idêntico ao encontrado na definição do contexto). O caso completo é, então, recuperado do banco, e são efetuadas algumas operações.

Inicialmente, a característica referente aos prováveis componentes com possível falha física é elaborada pelo sistema. Essa elaboração é feita com a implementação das regras comentadas na seção 5.5.1.1, que foram implementadas através de comandos *if-then*. Após esta etapa, é feita a definição da relevância das características do caso para cada tipo de problema que será consultado. Esta definição é realizada utilizando valores fixos para as características de acordo com o tipo de problema e um conjunto de regras que são relacionadas a certas características e valores do caso corrente (vide seção 5.5.2.3), que também são implementadas através de comandos *if-then*.

Após, é feita a recuperação no banco dos casos cujos tipos de problemas devem ser consultados, podendo ser ainda aqui aplicados filtros para determinados tipos de problemas. Exemplo de filtro é o aplicado para problemas de conectividade físico e de configuração do HW, em que só são consultados aqueles casos que possuem ao menos um dos padrões de tecnologia de rede do caso corrente (problemas que dizem respeito somente a uma LAN, por exemplo, não serão recuperados para um problema corrente em linha serial).

Uma vez que a busca na base foi executada, cada um dos casos selecionados passa pelo processo de casamento. Inicialmente, as características específicas do caso recuperado que têm grau de relevância filtro são comparadas às do caso corrente para verificar se o caso deve ser eliminado (filtrado). Se o caso permanecer, é calculada a similaridade, a confiabilidade e o fator de ordenamento do caso, conforme descrito na seção 5.5.2.4. A similaridade de cada característica é implementada segundo seu tipo (vide tabela 5.1): algumas, utilizando similaridade exata ou proporcional ao número de termos iguais em características com termos múltiplos (tais como enlaces envolvidos, sub-redes envolvidas); outras, que possuem termos fixos, utilizando valores pré-definidos para o casamento destes termos (tais como abrangência do problema, localização do problema); um outro grupo, utilizando a similaridade própria da característica envolvida, fazendo, para isso, uso de informações armazenadas na tabela do tipo da característica (tal como produtos, sistemas operacionais) ou na tabela com os intervalos numéricos (tal como intervalos de percentual de utilização da rede); e, por fim, a característica breve descrição tem sua similaridade segundo a ocorrência de termos cadastrados iguais ou similares.

Uma vez que o casamento foi realizado para todos os casos, é feito o ordenamento destes segundo o fator de ordenamento de cada um, e as características específicas de um número pré-definido de melhores casos são selecionadas, ordenadas e enviadas ao cliente, junto aos casos com melhor casamento.

Os diagramas SDL dos módulos *ComunicaçãoServidor*, *InterfaceBD* e dos módulos do bloco cliente podem ser encontrados no anexo 4.

6.1.2 Interação com o sistema

Como comentado anteriormente, a interação do usuário com o sistema se dá através do ambiente WWW. Após a validação do usuário no sistema, o usuário dispõe de um menu de opções, em que ele pode escolher a criação de um novo registro de problema e posteriormente consultar similares a este.

A figura a seguir apresenta a tela para obtenção das informações iniciais do problema na criação de um novo registro. No exemplo da figura, um problema está sendo cadastrado como de tipo Conectividade-Genérico, descrevendo um problema no enlace entre Porto Alegre e SP. Como pode ser observado, são solicitadas informações com propósito apenas de gerenciamento, tais como dados do reclamante e responsável, e informações sobre o problema corrente.

The screenshot shows a Netscape browser window with the following form fields:

- File Edit View Go Bookmarks Options Directory Window Help** (Menu bar)
- Address bar:** cristina@inf.utrgs.br
- Tipo de Problema:** Conectividade - Genérico (dropdown menu)
- Breve Descrição do Problema:** À FAPESP avisa que circuito com SP esta fora. (text area)
- Mensagem de erro do sistema/aplicacao:** - (dropdown menu)
- Rede/equipamentos/serviços envolvidos no problema:** funcionavam anteriormente, sem alteracao recente (dropdown menu)
- Identifique as prováveis causas para o problema, inserindo outras se necessario:**
 - NFS mal configurado (checkbox)
 - Servidor com problemas (checkbox)
 - Rede down (checkbox)
 - Outros: (checkbox) [text area]
- Informações Adicionais:** (caso originado do CINEMA) (text area)
- Status bar:** Document: Done

FIGURA 6.6 - Tela inicial no registro do caso corrente

Após as informações iniciais serem fornecidas pelo usuário, o sistema irá solicitar informações adicionais, a fim de elaborar as *características adicionais* sobre o problema. As informações que irão ser solicitadas variam, como já foi comentado e pôde ser visualizado na rede semântica dos tipos de problemas, apresentada no capítulo 5, de acordo com o contexto da situação — o tipo de problema e a localização do problema são as duas principais características que guiam a escolha dessas informações.

Assim, no problema exemplo da figura anterior, a criação de um problema do tipo Conectividade-Genérico leva à solicitação da localização do problema e outras informações, como o tipo de falta de acesso ocorrendo (constante, intermitente) e se esta é parcial ou total. Após a localização do problema ter sido obtida pelo sistema, questões específicas para o contexto encontrado são então solicitadas. Voltando ao exemplo anterior, em que é informado que a localização do problema é em linha serial, o sistema apresenta a tela indicada na figura abaixo, em que é feita a finalização da obtenção das características adicionais.

The screenshot shows a Netscape browser window with a form for configuring serial lines. The form is titled "uma única linha serial" and includes several sections:

- Enlaces:** A dropdown menu with options: POA-UCS, POA-Unisinos, and Outros.
- Enlace:** A dropdown menu with the option: Novo.
- Enlace entre:** A text input field containing: POA-SP.
- Equipamento Local:** A dropdown menu with the option: Novo.
- Numero IP ou Nome:** A text input field containing: 200.132.0.17.
- Sistema Operacional:** A dropdown menu with the option: IOS 10.3(7).
- Modelo:** A dropdown menu with the option: Cisco 7000.
- Equipamento Remoto:** A section with a text input field for "Numero IP ou Nome".
- Estado Interface Local:** A dropdown menu with the option: SERIAL UP, LINE PROTOCOL DOWN.
- Estado Interface Remota:** A dropdown menu with the option: -.

The browser's status bar at the bottom shows "Document: Done".

FIGURA 6.7 - Tela de coleta de características adicionais para linhas seriais

Após essa etapa, o sistema identifica (através de uma solicitação do cliente ao servidor) quais são os prováveis tipos de problemas, que serão utilizados na busca na base, e o processo de recuperação é solicitado pelo cliente ao servidor. A recuperação segue conforme apresentado na seção anterior, e os casos recuperados são enviados ao cliente, que os apresenta ao usuário juntamente com informações sobre os tipos de problemas consultados, de modo que no refino o usuário possa alterar esta lista se julgar conveniente. A figura abaixo apresenta os casos recuperados para o exemplo anterior.

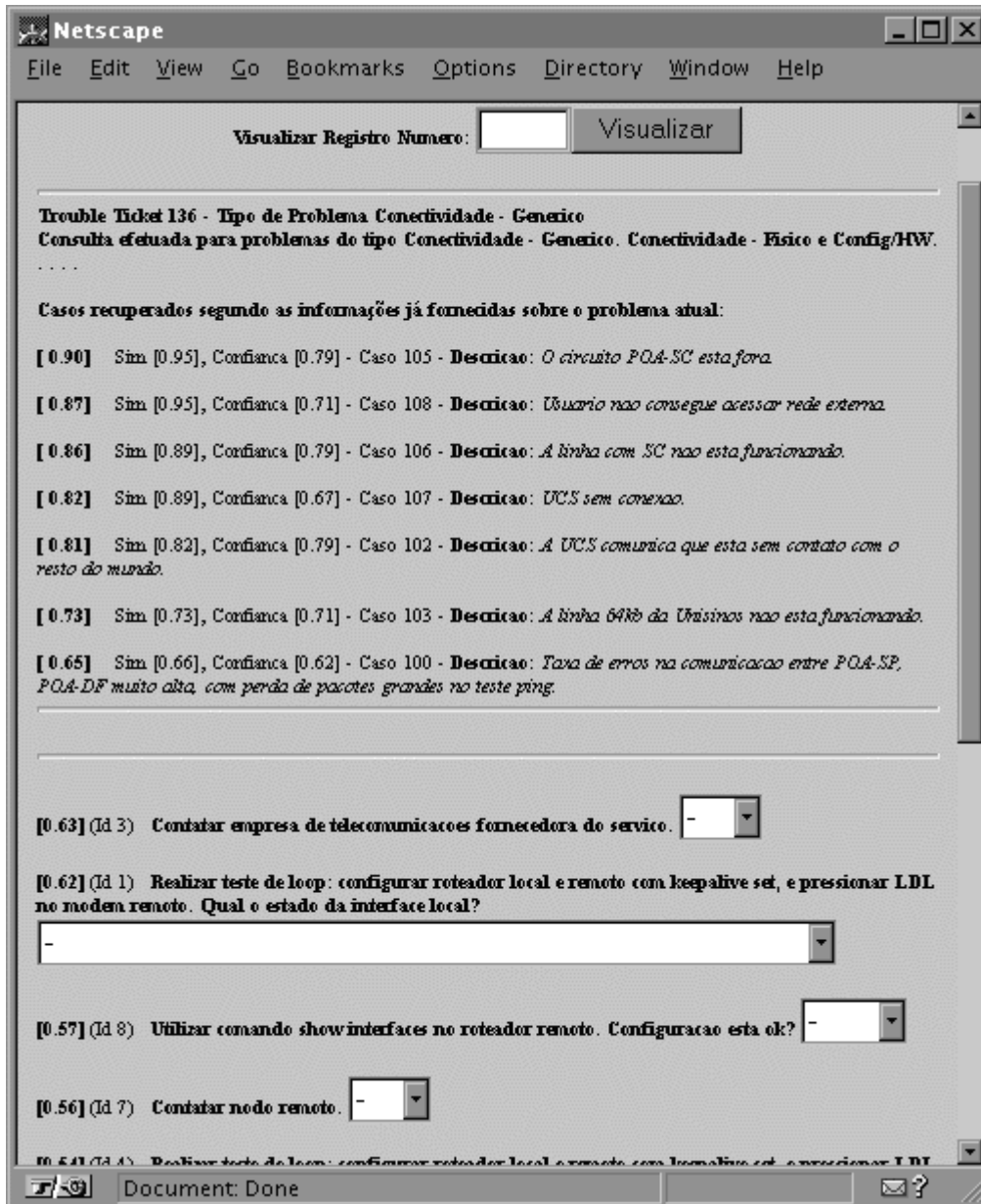


FIGURA 6.8 - Tela com casos recuperados

Por fim, quando um problema é resolvido, seu registro é encerrado e ele pode ser aprendido pelo sistema se o usuário julgar que o sistema não foi capaz de propor um caso similar, e portanto, essa situação representa uma situação nova que o sistema deve armazenar como *exemplar* e ser utilizado para futuras consultas CBR. Se, contudo, o sistema recuperou uma situação similar, o registro é armazenado apenas para fins de gerenciamento, como utilizado nos sistemas de registro de problemas tradicionais, com classificação *normal*. Para a inserção de um caso *exemplar*, algumas questões adicionais sobre a solução devem ser informadas pelo usuário, como comentado no capítulo 5, o que está implementado, nesta versão do protótipo, parte pela interface via CGIs e parte via interação direta com o banco.

A implementação do protótipo foi desenvolvida priorizando os problemas das camadas inferiores, correspondentes aos problemas na comunicação da rede — os tipos de problemas das camadas superiores não foram ainda implementados.

6.2 Avaliação do Protótipo

A avaliação do protótipo foi desenvolvida em duas etapas. A primeira consistiu num ajuste do grau da relevância e similaridade das características do sistema atribuídos inicialmente na modelagem. A segunda consistiu na avaliação propriamente dita, quando foram feitas recuperações para diversas situações e foram avaliados os pontos negativos e positivos do protótipo. Essas etapas serão comentadas a seguir.

Inicialmente, foram inseridos no sistema um conjunto de casos que haviam sido definidos na modelagem. Como foi visto anteriormente, em virtude do tempo disponível e do volume dos dados tratados no sistema, o protótipo foi desenvolvido apenas para os problemas na comunicação. Problemas de tipo aplicação e serviços, embora tenham sido tratados na modelagem, não foram implementados nesta versão.

Os casos inseridos representam problemas que (i) ocorreram no ambiente real estudado e estavam armazenadas no ambiente CINEMA (10 casos); (ii) ocorreram no ambiente real estudado e foram adquiridos a partir do relato dos especialistas entrevistados (6 casos) e (iii) foram retirados da bibliografia (8 casos). O maior número de casos encontrados foram de problemas que se diagnosticou como dos tipos Conectividade - Físico e Config/HW e Roteamento/Endereçamento.

A tabela abaixo apresenta como os casos do sistema estão distribuídos por tipo de problema. No anexo 3, uma lista com maiores informações sobre os casos, tais como o tipo de problema inicial e a localização, é apresentada.

TABELA 6.2 - Distribuição dos casos do protótipo por tipo de problema

| Tipo de Problema | Número de Casos |
|------------------------------------|------------------------|
| Conectividade - Genérico | 0 |
| Conectividade - Físico e Config/HW | 11 |
| Roteamento/Endereçamento | 12 |
| Performance | 0 |
| Alto Tráfego | 1 |

Na primeira etapa, foram feitas diversas recuperações no sistema, buscando analisar e ajustar a relevância e similaridade das informações. A análise da relevância atribuída a cada uma das características, para o contexto das situações de testes, foi realizada, e pequenos ajustes nos valores iniciais foram efetuados. Um exemplo foram as informações sobre o tipo de equipamento e produto (modelo) para equipamentos com função de roteamento que estivessem sendo analisados para falhas físicas. Inicialmente atribuídos como de grau 3 (características muito importante), estas características tiveram sua relevância modificada para grau 1 (características importantes) para problemas como os de *Conectividade - Físico e Config/HW*, já que pôde ser percebido que para problemas deste tipo estas características não têm tanta relevância. Assim, o grau 3 foi mantido apenas para problemas cuja *localização* é informada como *equipamentos de interconexão de rede*.

A similaridade, por sua vez, foi analisada buscando-se uma equivalência nos valores de similaridade atribuída para as diversas características, independente do tipo de

informação sendo tratada. Essa primeira etapa consistiu, assim, num ajuste das características de modo que as recuperações da fase posterior pudessem ser realizadas.

A segunda etapa, por sua vez, teve como objetivo avaliar se os casos recuperados traziam informações com potencial de contribuir para a solução corrente, incluindo os próprios casos recuperados e as características específicas selecionadas pelo sistema. Para efetuar esse teste, situações/casos novas teriam de ser inseridas no sistema, deveria ser avaliado se o sistema propôs uma situação adequada e se, dentre o conhecimento presente nele, esta solução fazia parte das melhores soluções disponíveis.

Como se desejava fazer testes com situações que fossem o mais próximo possível do real, utilizar unicamente a criação manual de casos poderia resultar num conjunto irreal de situações corrente, que não refletisse de modo adequado o ambiente real. Assim, optou-se por buscar a solução para uma situação corrente utilizando, para algumas recuperações, casos presentes no próprio sistema. Para isso, cada caso selecionado para corrente foi excluído do conjunto de casos disponíveis para recuperação e foi reutilizado como se representasse uma situação nova. Em outras recuperações, foram utilizados casos adicionais adaptados de situações relatadas em entrevistas, e que não haviam sido ainda incluídas no conjunto de casos.

A seguir, apresentamos dois exemplos de recuperações efetuadas, um para problemas em linha seriais, um para problemas em uma LAN, executados sobre o protótipo contendo os casos apresentados no anexo 3. Para cada um, são apresentadas as etapas envolvidas na recuperação. Inicialmente, podem ser visualizadas as informações da criação do registro com suas principais informações adicionais. Logo após, é apresentada uma ou duas recuperações efetuadas.

Cada recuperação é formada pelo conjunto dos casos recuperados e o conjunto das informações adicionais solicitadas pelo sistema (*características específicas*). As informações respondidas na situação são indicadas, e os resultados da nova recuperação são apresentados. Por fim, os dados referentes à solução são mostrados, as características específicas que foram cadastradas pelo caso (se houveram) são indicadas, e o conjunto das características específicas que foram mantidas após o caso ser encerrado é apresentado.

Caso 132 - Criação de Registro e Recuperação

Tipo de Problema Inicial: Conectividade - Genérico

Breve Descrição: Estações em sub-rede pararam de responder.

Outras Informações: Estações da sub-rede não respondem e não conseguem acesso externo. Há mensagem do NFS informando que servidor NFS não responde. (Caso adaptado de caso modelado número 4 - NFS)

Localização do problema: mesma_subrede_IP-LAN_com_um_segmento

Falta de Acesso: CONSTANTE

Área do Problema: ANTES_PRIMEIRO_ROTADOR

Padrão de Interface: LAN

Caso 132 - Resultados da primeira recuperação

Trouble Ticket 132 - Tipo de Problema Conectividade - Genérico

Consulta efetuada para problemas do tipo Conectividade - Genérico. Conectividade - Físico e Config/HW. Performance
Casos recuperados segundo as informações já fornecidas sobre o problema atual:

[0.80] Similaridade [0.80], Confiança [0.80] - Caso 122 -
Descrição: Estação não consegue acessar demais nodos.

[0.71] Similaridade [0.66], Confiança [0.80] - Caso 123 -
Descrição: Estação não consegue acessar nodos.

Informações Adicionais Solicitadas:

[0.69] (Id 19) Há alta taxa de colisões?

Solução Caso 132:

Causas: Um dos cabos de rede estava defeituoso.

Solução Adotada: Troca do cabo resolveu o problema.

Tipo de Problema Final: Conectividade - Físico e Config/HW

Características Específicas Cadastradas no Sistema (pelo caso 132):

Id: 21 Descrição: Os cabos e transceivers da rede estão ok, trocando-os o problema mantém? Grau: relev.5 Tipo: binária

Características Específicas Finais do Caso 132:

Id: 21 Valor: "NÃO", Ordem: 1

Caso 136 - Criação de Registro e Recuperação

(correspondente a caso 104 retirado)

Tipo de Problema Inicial: Conectividade - Genérico

Breve Descrição: A FAPESP avisa que o circuito com SP esta fora.

Outras Informações: (Caso correspondente a recriação do caso 104 excluído - caso 104 originado do CINEMA).

Localização do Problema: linha_serial

Falta de Acesso: CONSTANTE

Padrão de Interface: SERIAL

Enlaces com Problema: UMA_UNICA_LINHA

Estado Interface Local: SERIAL UP, LINE PROTOCOL DOWN

Caso 136 - Resultados da primeira recuperação

Trouble Ticket 136 - Tipo de Problema Conectividade - Genérico
Consulta efetuada para problemas do tipo Conectividade - Genérico. Conectividade - Físico e Config/HW

Casos recuperados segundo as informações já fornecidas sobre o problema atual:

[0.90] Similaridade [0.95], Confiança [0.79] - Caso 105 -
Descrição: O circuito POA-SC esta fora.

[0.87] Similaridade [0.95], Confiança [0.71] - Caso 108 -
Descrição: Usuário não consegue acessar rede externa.

- [0.86]** Similaridade [0.89], Confiança [0.79] - Caso 106 -
Descrição: A linha com SC não esta funcionando.
- [0.82]** Similaridade [0.89], Confiança [0.67] - Caso 107 -
Descrição: UCS sem conexão.
- [0.81]** Similaridade [0.82], Confiança [0.79] - Caso 102 -
Descrição: A UCS comunica que esta sem contato com o resto do mundo.
- [0.73]** Similaridade [0.73], Confiança [0.71] - Caso 103 -
Descrição: A linha 64kb da Unisinos não esta funcionando.
- [0.65]** Similaridade [0.66], Confiança [0.62] - Caso 100 -
Descrição: Taxa de erros na comunicação entre POA-SP, POA-DF muito alta, com perda de pacotes grandes no teste ping.

Informações Adicionais Solicitadas:

- [0.63]** (Id 3) Contatar empresa de telecomunicações fornecedora do serviço.
- [0.62]** (Id 1) Realizar teste de loop: configurar roteador local e remoto com keepalive set, e pressionar LDL no modem remoto. Qual o estado da interface local?
- [0.57]** (Id 8) Utilizar comando show interfaces no roteador remoto. Configuração esta ok?
- [0.56]** (Id 7) Contatar nodo remoto.
- [0.54]** (Id 4) Realizar teste de loop: configurar roteador local e remoto com keepalive set, e pressionar LDL no modem remoto. Qual o estado da interface remota?

Informações Respostadas:

- (Id 3) Contatar empresa de telecomunicações fornecedora do serviço.
- (Id 1) Realizar teste de loop: configurar roteador local e remoto com keepalive set, e pressionar LDL no modem remoto. Qual o estado da interface local? Resposta: SERIAL UP, LINE PROTOCOL DOWN
- (Id 4) Realizar teste de loop: configurar roteador local e remoto com keepalive set, e pressionar LDL no modem remoto. Qual o estado da interface remota? Resposta: SERIAL UP, LINE PROTOCOL UP (LOOPED)

Caso 136 - Resultados da segunda recuperação

Trouble Ticket 136 - Tipo de Problema Conectividade - Genérico
Consulta efetuada para problemas do tipo Conectividade - Genérico. Conectividade - Físico e Config/HW.
Casos recuperados segundo as informações já fornecidas sobre o problema atual:

- [0.90]** Similaridade [0.95], Confiança [0.79] - Caso 105 -
Descrição: O circuito POA-SC esta fora.
- [0.86]** Similaridade [0.89], Confiança [0.79] - Caso 106 -
Descrição: A linha com SC não esta funcionando.
- [0.81]** Similaridade [0.82], Confiança [0.79] - Caso 102 -
Descrição: A UCS comunica que esta sem contato com o resto do mundo.

[0.73] Similaridade [0.73], Confiança [0.71] - Caso 103 -
 Descrição: A linha 64kb da Unisinos não esta funcionando.

[0.65] Similaridade [0.66], Confiança [0.62] - Caso 100 -
 Descrição: Taxa de erros na comunicação entre POA-SP, POA-DF
 muito alta, com perda de pacotes grandes no teste ping.

Informações Adicionais Solicitadas:

[0.63] (Id 6) Ha perda de pacotes grandes com teste ping?

[0.56] (Id 7) Contatar nodo remoto.

[0.46] (Id 5) Contatar nodo remoto. Outros enlaces deste
 ponto apresentam problemas?

[0.54] (Id 22) Ha alta taxa de erros CRC?

Solução Caso 136:

Causas: Problema foi isolado entre EMBRATEL e FAPESP.

Solução Adotada: Problema foi solucionado pela EMBRATEL.

Tipo de Problema Final: Conectividade - Físico e Config/HW

Características Específicas Cadastradas no Sistema (pelo caso
 100/134):

Id: 2 Descrição: Fazendo teste de loop ate ponto
 intermediário do enlace, qual o estado da interface local? Grau:
 relev.5 Tipo: Sobre carac. adicional itfst, operação com mesmos
 valores e similaridade.

Características Específicas Finais do Caso 134:

Id: 1 Valor: "SERIAL UP, LINE PROTOCOL DOWN", Ordem: 1

Id: 4 Valor: "SERIAL UP, LINE PROTOCOL UP (LOOPED)",
 Ordem: 1

Id: 3

Id: 2 Valor: "SERIAL UP, LINE PROTOCOL UP (LOOPED)",
 Ordem: 1

O exemplo de recuperação do caso 132 corresponde a um problema em uma rede Ethernet, causado por cabo defeituoso. Os casos retornados foram os casos 122 e 123. O caso 122, retornado em primeiro lugar, corresponde ao caso cadastrado no sistema mais próximo à situação corrente. Este caso representa um problema em que uma estação não conseguia acessar os demais nodos porque havia uma placa de rede defeituosa — e casos com problemas causados por falhas em cabeamento mais semelhantes ao corrente não estão cadastrados no protótipo.

O segundo exemplo, por sua vez, corresponde a um problema causado por uma falha no enlace fornecido pela EMBRATEL. Também nesse caso foram retornados problemas similares. Na primeira recuperação, o primeiro caso selecionado (caso 105) corresponde a um problema também causado por falha no enlace serial. Além disso, as características específicas propostas são relevantes para problemas deste tipo.

A característica 3, apenas informativa, sugere que a empresa de telecomunicações seja contatada, sugestão relevante e que, nos problemas similares coletados, representa uma das primeiras ações que foram efetuadas. A característica 1 apresenta também uma

ação de diagnóstico utilizada frequentemente nesses tipos de problemas, e pode contribuir de modo significativo para o isolamento da falha.

Para o caso real sendo tratado, as características 1 e 3 foram relevantes, e haviam sido utilizadas de fato no caso coletado no ambiente CINEMA — de modo que o sistema foi capaz de propor as ações adequadas e que já haviam, na situação real que deu origem ao registro no ambiente CINEMA, contribuído no diagnóstico.

Além dessas, a característica 4 é também relevante. Idealmente, ela deveria ter sido proposta em seqüência à característica 1, pois ambas representam resultados da mesma ação de diagnóstico — assim, meios de relacionar características específicas poderiam ser estudados para uma próxima versão, contribuindo para usuários que prefiram refinar muitas vezes uma recuperação respondendo poucas específicas por vez.

Na recuperação sendo comentada (caso 136), foram inseridas as características específicas 1, 3 e 4. A recuperação subsequente, mais uma vez, propôs em primeiro lugar o caso 105, que foi causado também por falha na linha serial. Mas além do caso proposto ser similar, a sugestão do sistema de ações de diagnóstico relevantes já contribui para a evolução da situação corrente — daí a importância das características específicas.

O conjunto dos testes desenvolvidos permitiu que algumas conclusões fossem retiradas sobre o protótipo em geral e aprimoramento que pode ser efetuado sobre ele.

Como foi comentado, foram implementados nesta versão apenas dois tipos de *características específicas*: do tipo binária e do tipo que usa os *mesmos valores para similaridade* de uma *característica adicional*. Assim, características específicas como “Há alta taxa de colisões?” e “Há os protocolos RIP e IGRP no ambiente?” tiveram que ser cadastradas para permitir que estas informações fossem utilizadas.

Entretanto, versões posteriores podem aprimorar o sistema, implementando os tipos de características específicas que elaboram sua resposta automaticamente, utilizando valores já fornecidos para *características adicionais*. Assim, as duas características específicas citadas acima, por exemplo, poderiam ser elaboradas automaticamente, a partir de respostas das características adicionais *taxa de colisões* e *protocolos de roteamento do ambiente*. Com isso, a recuperação será aperfeiçoada pelo sistema, que pode fazer um refino da recuperação mesmo antes de consultar o usuário (elaborando automaticamente estas características e utilizando-as imediatamente), e, assim, fornecer casos com similaridade maior. Essa abordagem, contudo, ainda mantém a necessidade do usuário identificar, no momento do aprendizado de um caso — após, portanto, a situação ter sido solucionada —, quais características específicas foram de fato relevantes para o diagnóstico, a fim de evitar que informações específicas que tenham sido elaboradas mas que não contribuam para o entendimento do caso sejam utilizadas em recuperações futuras.

Uma dificuldade encontrada na atual implementação das características específicas foi a ausência de meios para restringir estas informações apenas para problemas num determinado contexto. Um exemplo disso foram as informações específicas 6 e 22, solicitadas na segunda recuperação do caso 136. Embora elas tenham sido requeridas apenas numa segunda recuperação, após as informações mais relevantes cadastradas no sistema já terem sido solicitadas, elas não serão úteis para problemas desse contexto, pois, no caso 136, há falta de acesso constante. Assim, uma questão que pode contribuir para o aprimoramento do sistema é o estudo de meios de definir para

algumas específicas contextos onde elas não devem ser selecionadas. Uma proposta inicial seria permitir que a característica só fosse aplicada se uma característica adicional com um valor pré-determinado casasse com o valor daquela característica do caso corrente com similaridade mínima de um valor X. Entretanto, essa abordagem poderia restringir a flexibilidade e simplicidade do sistema, e estudos adicionais são necessários.

Sobre as características específicas, pode ser vislumbrado ainda que a associação de notas a essas informações, como proposto na modelagem, poderia auxiliar o usuário a compreender a ação de diagnóstico proposta, que se torna, muitas vezes, pouco clara se feita unicamente através da descrição. Assim, para cada específica retornada, a nota associada a esta característica poderia ser fornecida ao usuário, que teria mais informações sobre os procedimentos envolvidos na ação se esta não fosse familiar a ele.

Algumas considerações podem ser levantadas também sobre a probabilidade dos tipos de problemas calculada na **definição do contexto** e utilizada para identificar o conjunto de casos que será consultado. Nos testes avaliados, pode ser constatado que, muitas vezes, o valor 0,2 não é adequado para ser o limiar da similaridade mínima para a probabilidade de um tipo de problema a fim de que este seja consultado. Dessa forma, percebeu-se que realizar um refino deste limiar à medida que o problema vai se desenvolvendo pode contribuir para aperfeiçoar a recuperação, selecionando, numa segunda busca, casos com limiar menor se os casos recuperados anteriormente não foram os mais adequados. Com isso, mesmo quando o usuário não selecionar um tipo de problema com probabilidade reduzida por considerar que os casos recuperados não foram suficientes, o sistema pode, automaticamente, ajustar esse valor e garantir que um caso similar presente na base não seja excluído da busca.

Uma das dificuldades da avaliação da acurácia do protótipo para os diversos contextos encontrada foi causada pelo número de casos cadastrados que representam um grupo inicial coletado e que não correspondem ao aprendizado pelo uso normal do sistema. Assim, o conjunto dos casos recuperados pode, muitas vezes, indicar os melhores dentre os casos do sistema, embora não represente a melhor situação possível, o que dificulta sua avaliação.

Os testes efetuados permitiram, contudo, verificar que o sistema tem a capacidade de recuperar situações similares adequadas para uma situação, e seu uso pode ser aplicado num ambiente real. Nesse ambiente, juntamente com uma etapa inicial de acompanhamento para ajustes finais dos graus de relevância e similaridades decorrentes das diferenças em um ambiente real, o aprendizado do sistema — principalmente através dos novos casos e do aprimoramento a relação histórica — irá aumentar seu conhecimento e permitir que situações mais similares à corrente possam ser recuperadas.

Em relação às demais abordagens encontradas na bibliografia, o sistema DUMBO pode ser comparado ao CRITTER [LEW 93][LEW 95] e MASTER [DRE 95], ambos sistemas de gerenciamento de falhas para redes de computadores que usam CBR. Entre seus pontos negativos, pode ser citada a ausência de adaptação automática e a sugestão da solução de modo indireto, diferente do que ocorre no CRITTER. Isso acontece, de certo modo, porque o DUMBO foi desenvolvido com o propósito de permitir grande interação dos usuários no processo — os casos serão criados, utilizados e encerrados por usuários e não de modo automático, o que é uma importante contribuição do DUMBO. Para que isso fosse possível, entretanto, uma representação dos casos mais flexível e

menos estruturada teve que ser utilizada para os campos da solução, o que trouxe maiores dificuldades para adaptação e para propor soluções de modo direto.

Outra contribuição do DUMBO diz respeito à sugestão de ações de diagnóstico além da solução, que permitem a evolução do problema, o que não ocorre no CRITTER, onde a solução é proposta imediatamente. No MASTER, há também a sugestão de ações ao usuário, que permitem descrever a situação melhor. O sistema DUMBO, contudo, parece possuir uma estrutura muito mais simplificada e de fácil manutenção, enquanto que no MASTER, pelo seu conceito de *master tickets* que envolve a generalização das informações sobre a falha, há uma complexidade maior no sistema para aprendizado e manutenção.

Além disso, uma importante contribuição do DUMBO diz respeito a sua estrutura, que foi concebida com o propósito de ser flexível e incluir a maior parte das situações que podem ser encontradas no domínio.

Em relação aos resultados, a sua comparação entre os sistemas não foi possível pela ausência de informações nas referências que permitam a comparação com os resultados obtidos nos testes com o DUMBO.

7 Considerações finais

O presente trabalho se propôs a desenvolver um sistema destinado ao auxílio de gerentes de redes no diagnóstico de falhas. Para isso, foi feita a definição de um modelo apropriado, desenvolvido utilizando o paradigma de raciocínio baseado em casos aplicado sobre um sistema de registro de problemas. Uma vasta revisão bibliográfica sobre o paradigma foi feita (capítulo 2), visando identificar as particularidades do paradigma para possibilitar sua correta utilização na modelagem do sistema. Além disso, a revisão bibliográfica da área de gerenciamento de redes foi também desenvolvida (capítulo 3), enfocando, principalmente, os sistemas de registro de problemas, o uso de paradigmas de raciocínio no domínio e o uso do paradigma de raciocínio baseado em casos em aplicações referenciadas na bibliografia.

Uma vez efetuado o estudo do paradigma e da área de gerenciamento de redes, o desenvolvimento do modelo exigiu uma etapa de aquisição do conhecimento para o sistema, quando foi feito o estudo dos problemas típicos do domínio e das particularidades de cada tipo de informação relevante para reconhecer dois problemas similares (capítulo 4). Em posse desses dados e fazendo uso do conhecimento obtido na revisão do paradigma e da área de gerenciamento, os dados a serem representados no sistema foram organizados na representação em redes semânticas e a representação dos casos e os processos de raciocínio necessários foram concebidos para o sistema.

O sistema DUMBO, nome escolhido para o sistema desenvolvido, permite que a solução ou auxílio ao diagnóstico em uma situação de problema corrente seja feito através de casos similares ocorridos anteriormente, os quais estão armazenados na base do sistema. Para isso, faz uso da comparação de informações relevantes entre os dois casos, utilizando o conhecimento do sistema para efetuar, nesta comparação, não apenas um casamento simples, mas permitir também o casamento parcial das informações.

Cabe, agora, apresentarmos uma análise crítica sobre o trabalho desenvolvido, salientando as falhas encontradas durante o percurso e os aspectos positivos que o mesmo apresenta.

O primeiro dos pontos a ser comentado diz respeito aos problemas tratados pelo sistema. Em virtude da ampla gama de problemas do ambiente de rede focado e das restrições de tempo disponíveis para o desenvolvimento do trabalho, o estudo mais detalhado dos problemas encontrados foi dificultado. Entretanto, a ênfase em um único tipo de problema ou em um ambiente com tecnologias reduzidas prejudicaria o objetivo do trabalho, que visava auxiliar o diagnóstico das falhas nas redes de modo geral.

Assim, como não foram encontrados trabalhos disponíveis que pudessem ser encaixados ao sistema e aplicados para alguns problemas, deixando ao DUMBO o enfoque apenas dos demais, foi necessário o desenvolvimento de uma estrutura para as informações que permitisse que a maior parte dos problemas possíveis de ocorrer pudesse ser tratada pelo sistema, mesmo que um enfoque maior fosse dado para um grupo dentre estes. Desta forma, um aspecto positivo a ser ressaltado é o desenvolvimento dessa estrutura para representação dos problemas. Esta estrutura poderá ser utilizada posteriormente em outros trabalhos, que podem até mesmo focar apenas um tipo específico de problemas, e se associar ao sistema DUMBO para executar o tratamento dos demais.

No modelo proposto para o sistema, em virtude das restrições de tempo e volume de dados encontrados pelo trabalho, uma ênfase maior foi dada aos problemas das camadas inferiores do modelo TCP/IP, englobando os problemas na comunicação. Destes, como pôde ser avaliado pelo protótipo desenvolvido, resultados melhores foram obtidos com falhas físicas e de configuração do hardware do que os obtidos em falhas de roteamento, em virtude da complexidade envolvidas nos problemas deste tipo.

Entre as deficiências constatadas no tratamento de falhas de roteamento, pode ser citado o uso de informações pouco precisas, uma vez que a situação foi tratada com a visão de um problema de acessibilidade, havendo um número menor de informações nas características adicionais relevantes que façam uso das informações mais detalhadas em uma situação que já foi identificada como de roteamento. Esse aspecto foi contornado no sistema pelo uso das características específicas para estes dados.

Um aspecto que traz dificuldades no uso deste sistema é a necessidade de informar os diversos dados solicitados. Como não foi possível implementar, em virtude das restrições comentadas, esquemas de elaboração automática das informações no protótipo desenvolvido, o usuário tem de responder a quase todas as informações, o que torna seu uso menos amigável. Entretanto, como foi comentado amplamente no capítulo 5, o uso destas informações, com uma linguagem estruturada, é necessário para o desenvolvimento de sistemas deste tipo.

Um dos importantes aspectos positivos do modelo foi a possibilidade de identificar informações de diagnóstico, além dos casos recuperados propriamente ditos. No domínio do gerenciamento de falhas em redes, o estudo efetuado sobre os problemas demonstrou que estas ações são utilizadas de modo corrente na atividade de diagnóstico, e sua integração no sistema contribui de modo significativo para auxiliar estas atividades.

Por fim, a avaliação do protótipo desenvolvida mostrou que o modelo contribui para a identificação de casos similares e atividades de diagnóstico relevantes para os problemas de rede, e está apto para ser testado e aplicado a um ambiente maior e real.

7.1 Trabalhos Futuros

A partir dos pontos negativos identificados no sistema, e visando também o aprimoramento dos processos envolvidos e do uso do sistema, algumas metas foram estabelecidas para futuro desenvolvimento do trabalho. Entre estas, podemos citar:

- Estudo e implementação das informações adicionais identificadas no sistema para os problemas nas camadas inferiores e ainda não implementadas, tais como as citadas no anexo 2. Estudos complementares e implementação das informações adicionais dos problemas das camadas superiores.
- Desenvolvimento das características específicas com elaboração automática.
- Desenvolvimento de uma interface mais adequada para o aprendizado do sistema, assim como os meios de controle deste aprendizado apenas para usuários capacitados.
- Integração do sistema em plataformas ou outros sistemas de gerenciamento, de modo a fornecer novas informações e dados mais específicos disponibilizados por estes sistemas nas características adicionais ou específicas. Um exemplo seria a integração com

o MAD [NUN 97], que poderia fornecer dados específicos sobre os erros encontrados em uma situação. O uso de plataformas de gerenciamento integradas também proporcionariam que informações disponibilizadas pelo protocolo SNMP ou em agentes distribuídos na rede fossem identificadas automaticamente pelo sistema. Com isso, a interface com o usuário também seria simplificada, liberando-o da necessidade de informar um número grande de informações.

- Identificação de formas de implantar técnicas de adaptação automática ao ciclo de raciocínio.

- Estudo de meios de integrar ao sistema aplicações que possuam facilidades de interpretação de textos livres, que poderiam ser encaixadas ao sistema preenchendo, a partir da descrição em texto livre fornecida pelos gerentes, algumas das características adicionais e específicas já comentadas, de modo análogo ao comentado em [DRE 95].

Anexo 1 Rede Semântica dos Tipos de Problemas

No capítulo 5, a rede semântica dos problemas foi apresentada, com as categorias de problema que uma situação pode assumir. Os problemas de comunicação, como foi visto, possuem um atributo localização do problema, que pode assumir os valores linha serial, entre mesma sub-rede IP, entre sub-redes IP, canal de uma LAN e canal de diversas sub-redes IP. A seguir, apresentamos a parte da rede semântica referente à estas especializações da localização do problema.

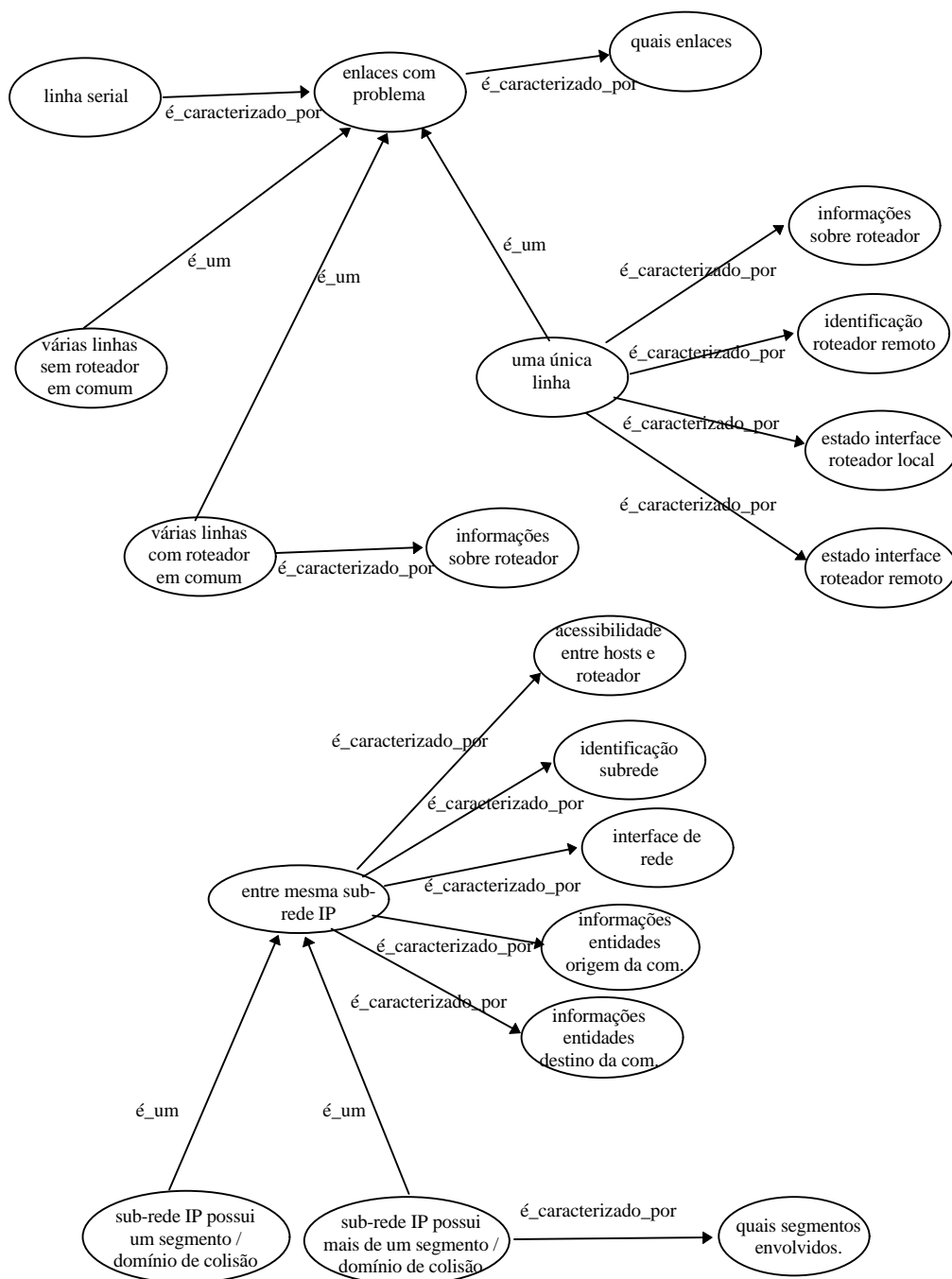


FIGURA A1.1 - Rede semântica da localização dos problemas (parte 1)

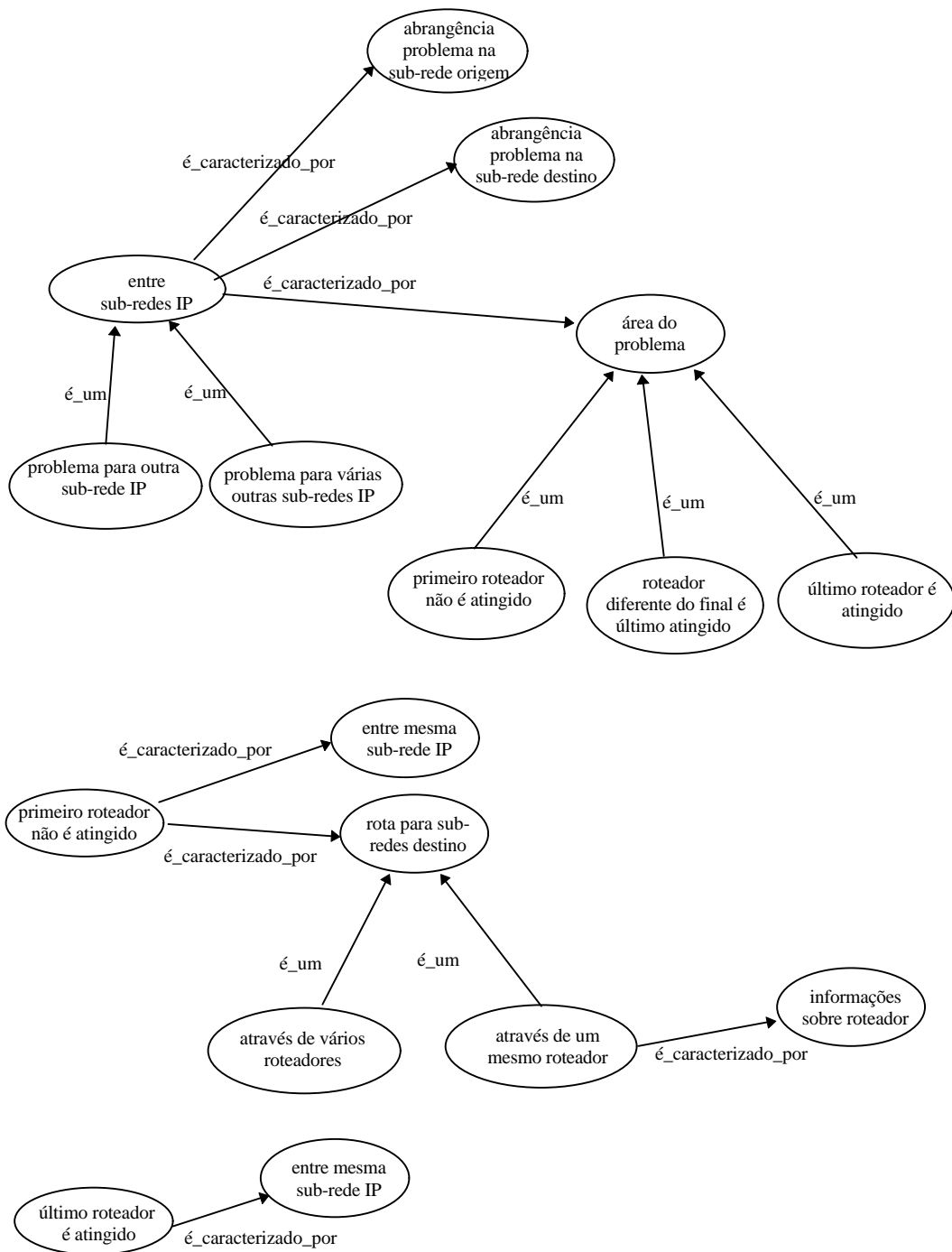


FIGURA A1.2 - Rede semântica da localização dos problemas (parte 2)

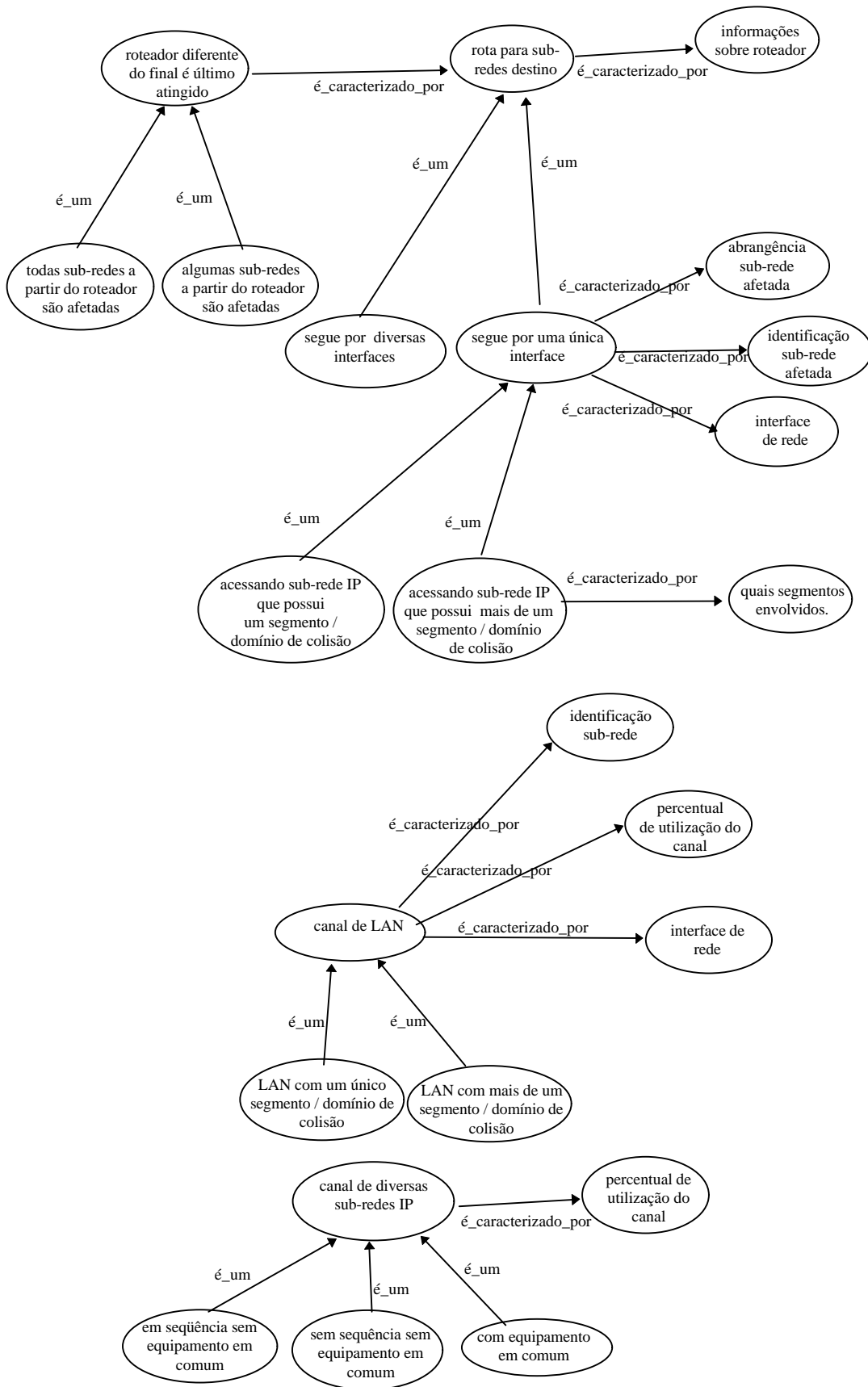


FIGURA A1.3 - Rede semântica da localização dos problemas (parte 3)

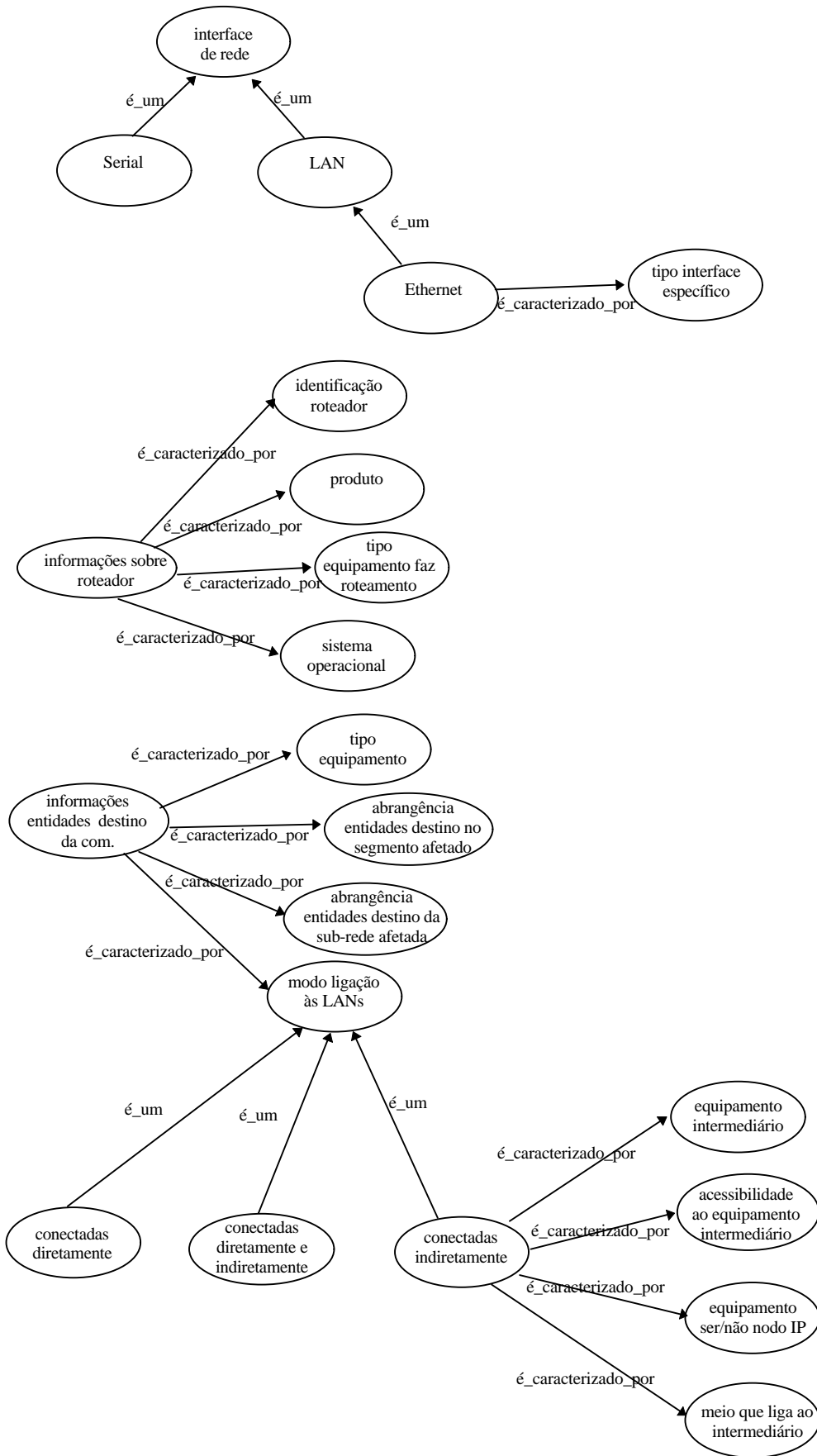


FIGURA A1.4 - Rede semântica da localização dos problemas (parte 4)

Anexo 2 Características de um Caso

Características Adicionais

Relevância para características que tem relevância variável

Como foi apresentado na seção 5.5.2.2, algumas características têm a relevância variável de acordo com o contexto da situação. A seguir, apresentamos algumas das principais regras utilizadas para identificar esta relevância presentes no protótipo implementado.

SE tipo de problema = 'conectividade-genérico' OU tipo de problema = 'conectividade-físico e config/HW' OU tipo de problema = 'performance' OU tipo de problema = 'alto tráfego' E provável componente falha física possui 'cabramento' OU provável componente falha física possui 'equipamento interconexão enlace'
ENTÃO tipo interface rede peso 5

SE tipo de problema = 'conectividade-genérico' OU tipo de problema = 'conectividade-físico e config/HW' OU tipo de problema = 'performance' OU tipo de problema = 'alto tráfego' E provável componente falha física possui 'cabramento' OU provável componente falha física possui 'equipamento interconexão enlace' E tipo interface rede contém 'Ethernet'
ENTÃO tipo específico interface PESO 3

SE tipo de problema = 'conectividade-genérico' OU tipo de problema = 'roteamento/endereçamento' E localização do problema = 'entre mesma sub-rede IP'
ENTÃO acessibilidade entre hosts e roteador PESO 1

SE tipo de problema = 'conectividade-genérico' OU tipo de problema = 'roteamento/endereçamento' OU tipo de problema = 'performance' OU tipo de problema = 'alto tráfego' E localização do problema = 'entre sub-redes IP' OU localização do problema = 'canal de diversas sub-redes IP'
ENTÃO área do problema PESO 3

SE tipo de problema = 'conectividade-genérico' OU tipo de problema = 'conectividade-físico e config/HW' OU tipo de problema = 'performance' OU tipo de problema = 'alto tráfego' E provável componente físico contém 'equipamento interconexão rede'
ENTÃO equip. interconexão rede falha física número IP PESO 1,
equip. interconexão rede falha físico tipo PESO 1,
equip. interconexão rede falha física produto PESO 1

SE tipo de problema = 'conectividade-genérico' OU tipo de problema = 'conectividade-físico e config/HW' OU tipo de problema = 'performance' OU tipo de problema = 'alto tráfego' E localização do problema = 'linha serial' E enlaces com problema = 'uma única linha'
ENTÃO equip. interconexão rede falha física estado interface PESO 5

SE tipo de problema = 'conectividade-genérico' OU tipo de problema = 'roteamento/endereçamento' OU tipo de problema = 'performance' OU tipo de problema = 'alto tráfego' E localização do problema = 'entre mesma sub-rede IP' OU localização do problema = 'entre sub-redes IP' OU localização do problema = 'canal de LAN'
ENTÃO abrangência sub-rede afetada destino PESO 3, abrangência sub-rede origem PESO 3

SE tipo de problema = 'conectividade-genérico' OU tipo de problema = 'roteamento/endereçamento' OU tipo de problema = 'performance' OU tipo de problema = 'alto tráfego' E localização do problema = 'entre mesma sub-rede IP' OU localização do problema = 'entre sub-redes IP' OU localização do problema = 'canal de LAN' E padrão interface rede possui 'LAN'

ENTÃO abrangência das entidades destino do segmento afetado PESO 3,
abrangência das entidades origem do segmento afetado PESO 3,
quais segmentos envolvidos PESO 3

SE tipo de problema = 'roteamento/endereçamento' E localização do problema = 'entre mesma sub-rede IP' OU (localização do problema = 'entre sub-redes IP' E área do problema = 'primeiro roteador não é atingido')

ENTÃO tipo de equipamento origem PESO 1

SE tipo de problema = 'roteamento/endereçamento' E
localização do problema = 'entre mesma sub-rede IP' OU
(localização do problema = 'entre sub-redes IP' E área do problema = 'último roteador é atingido')

ENTÃO tipo de equipamento destino PESO 1

Similaridade das Características Adicionais

A seguir, apresentamos o conjunto das características adicionais implementadas no protótipo desenvolvido, comentando qual o tipo de similaridade utilizada em cada uma delas. Como será visto, o cálculo da similaridade de algumas é feito independente do tipo de informação, tais como a similaridade exata entre os dois valores, ou a similaridade em que é feita a comparação dos diversos elementos em cada valor e a similaridade final é proporcional ao número de elementos iguais pelo número de elementos presentes em cada uma das características (chamada *possui termos, percentual simples*). Já em outras, a similaridade é própria do tipo de informação, sendo manuseada, tais como a similaridade de produtos, de interfaces de rede, de estados de interfaces de rede, podendo ser aplicada para valores sempre fixos ou variáveis, que podem aprendidos pelo sistema. Essa similaridade própria pode ser usada em apenas uma característica ou para várias características que trazem o mesmo tipo de informação (estado da interface do roteador local e remoto, por exemplo).

TABELA A2.1 - Similaridade das características adicionais implementadas

| Característica | Similaridade |
|--|---|
| tipo de problema | probabilidade calculada |
| tipo de problema inicial | exata com tipo corrente |
| breve descrição | textual |
| houve alteração | própria |
| padrão interface rede | possui termos, percentual simples |
| localização do problema | própria |
| problema mantém usando IP | usada apenas para regras redirecionadoras do tipo de problema |
| falta de acesso (intermitente, constante, nunca) | exata |
| modo de falta de acesso (parcial, total) | exata |
| performance ruim como (intermitente, constante) | exata |
| alto tráfego (constante, intermitente) | exata |
| qual tipo de tráfego com taxa elevada | exata |
| ocorrência de alta taxa de erros | exata |
| ocorrência de alto tráfego | exata |
| ocorrência de alto tempo de resposta | exata |
| identificação sub-rede | exata |
| interface de rede | própria |
| tipo interface específico | possui termos, percentual simples |
| percentual de utilização do meio | numérica valores próprios |
| acessibilidade entre hosts e roteador | exata |
| área do problema | própria |
| identificação roteador com possível falha | exata |

| Característica | Similaridade |
|---|---|
| roteamento | |
| tipo equipamento faz roteamento com possível falha roteamento | própria |
| produto (modelo) roteador com possível falha roteamento | própria |
| sistema operacional roteador com possível falha roteamento | própria |
| identificação roteador com possível com possível falha física | exata |
| tipo equipamento faz roteamento com possível falha física | própria |
| produto (modelo) roteador com possível falha física | própria |
| sistema operacional roteador com possível falha física | própria |
| estado da interface do roteador com possível falha física | própria |
| percentual de utilização da interface | numérica com valores próprios segundo tipo de interface |
| rota para sub-rede destino segue por uma/várias interfaces | exata |
| identificação roteador remoto | exata |
| estado interface roteador remoto | própria |
| enlaces com problema | própria |
| quais enlaces | possui termos, percentual simples |
| quais segmentos envolvidos | própria tipo um, alguns, todos |
| abrangência entidades destino no segmento afetado | própria tipo um, alguns, todos |
| abrangência entidades origem no segmento afetado | própria tipo um, alguns, todos |
| abrangência entidades destino da sub-rede afetada | própria tipo um, alguns, todos |
| abrangência problema na sub-rede destino | própria tipo um, alguns, todos |
| abrangência problema na sub-rede | própria tipo um, alguns, todos |
| tipo equipamento destino | possui termos, percentual simples |
| tipo equipamento origem | possui termos, percentual simples |
| modo ligação às LANs entidades destino | própria |
| modo ligação às LANs entidades origem | própria |
| equipamento intermediário entidades destino | exata |
| equipamentos intermediário entidades origem | exata |
| equipamento destino ser/não nodo IP | exata |
| equipamento origem ser/não nodo IP | exata |
| qual tipo de tráfego com taxa elevada | possui percentual simples |
| qual tipo de erros | possui percentual simples |

Existem, ainda, outras características que foram identificadas no desenvolvimento do sistema DUMBO como indicadoras de similaridade entre casos em alguns tipos de problemas e contextos. Essas características, entretanto, não puderam ser implementadas no protótipo, em virtude das restrições de tempo e volume de informações do sistema. A seguir, apresentamos uma lista destas características, bem como a similaridade proposta inicialmente para elas.

TABELA A2.2 - Algumas características adicionais propostas

| Característica | Similaridade |
|---|---|
| qual tipo de tráfego com taxa elevada | similaridade através de características do tráfego envolvido, como protocolo de transporte utilizado, função do protocolo, p. e. |
| em que há alto tempo de resposta | própria, para valores fixos como no estabelecimento da conexão, após estabelecimento da comunicação, em todas as etapas da comunicação, etc. |
| mensagem | preferencialmente, com tabela própria com cadastro das mensagens mais frequentes de cada sistema e a que tipo de mensagem pertence |
| possíveis causas | própria, com cadastro das causas mais frequentes |
| definição do problema | não utilizada no casamento de modo direto, mas que permita elaborar uma ou mais características específicas de acordo com esta definição fornecida no momento da inserção do caso por usuário com mais conhecimento do problema |
| quais protocolos não tem acesso | própria, com similaridades do protocolo envolvido, tais como protocolo de transporte |
| taxa de erros CRC da rede | numérica com valores próprios |
| taxa de colisões da rede | numérica com valores próprios |
| taxa de broadcast da rede | numérica com valores próprios |
| protocolos de roteamento envolvidos no ambiente | própria, com similaridades através de características do protocolo como seu tipo, se tem sub-endereçamento, etc |
| ambiente tem múltiplos protocolos | exata, sendo a característica elaborada automaticamente a partir dos protocolos de roteamento envolvido ambiente |
| função do equipamento de interconexão de enlace/físico | própria, de acordo com a camada em que atua, por exemplo |
| identificação do equipamento de interconexão de enlace/físico | exata |
| tipo de equipamento de interconexão de enlace/físico | própria segundo tipo de equipamento (já implementado) |
| produto (modelo) do equipamento de interconexão de enlace/físico | própria segundo produto (já implementado) |
| protocolos manuseados pelo roteador com possível falha roteamento | própria segundo características dos protocolos |
| taxa de erros de entrada da interface do roteador | numérica com valores próprios |
| taxa de erros de saída da interface para roteador | numérica com valores próprios |
| taxa de descartes de entrada da interface do roteador | numérica com valores próprios |
| taxa de descartes de saída da interface do roteador | numérica com valores próprios |
| taxa de erros CRC da interface do roteador | numérica com valores próprios |
| ocorrência de resets na interface do roteador | numérica com valores próprios |
| taxa de colisões da interface do roteador | numérica com valores próprios |
| identificação do servidor de nomes | exata |
| identificação do tipo de servidor de nomes | própria para tipo de servidores de nomes |
| identificação do produto (modelo) do servidor de nomes | própria para produtos |
| identificação do sistema operacional do servidor de nomes | própria para sistemas operacionais |
| identificação do servidor de autenticação | exata |

| Característica | Similaridade |
|--|---|
| identificação do tipo de servidor de autenticação | própria para tipo de servidores de autenticação |
| identificação do produto (modelo) do servidor de autenticação | própria para produtos |
| identificação do sistema operacional do servidor de autenticação | própria para sistemas operacionais |
| identificação do servidor de arquivos | exata |
| identificação do tipo do servidor de arquivos | própria para tipo de servidores de arquivos |
| identificação do produto (modelo) do servidor de arquivos | própria para produtos |
| identificação do sistema operacional do servidor de arquivos | própria para sistemas operacionais |
| prováveis componentes com falha física | exata, por comparação com componente envolvido informado na solução (sua elaboração já está implementada, sendo usada, entretanto, apenas para definição de relevância) |
| prováveis componentes com falha de roteamento | exata, por comparação com componente envolvido informado na solução |

Alguns grupos do sistema implementados em tabelas

A seguir, apresentamos como exemplo dois grupos do sistema que são utilizados para cadastrar valores para aquelas características que possuem um tipo de similaridade próprio e com termos variáveis, que podem ser aprendidos pelo sistema. Inicialmente, é apresentado o grupo dos estados das interfaces, e em segundo lugar, o grupo produtos.

TABELA A2.3 - Grupo estados das interfaces

| Estado Indicado | Estado Geral |
|--|---------------------|
| DOWN | 0 |
| SERIAL DOWN, LINE PROTOCOL DOWN | 0 |
| SERIAL UP, LINE PROTOCOL DOWN | 0 |
| SERIAL UP, LINE PROTOCOL DOWN (DISABLED) | 0 |
| SERIAL ADMINISTRATIVELY DOWN, LINE PROTOCOL DOWN | 0 |
| UP | 1 |
| SERIAL UP, LINE PROTOCOL UP (LOOPED) | 1 |
| SERIAL UP, LINE PROTOCOL UP | 1 |

TABELA A2.4 - Grupo produtos

| Produto | Familia/Sim.Versões | Tipo Equipamento |
|----------------|----------------------------|-------------------------|
| RISC 6000 | | estação de trabalho |
| SPARC 20 | | estação de trabalho |
| PC Pentium 133 | Pentium | microcomputador |
| Cisco 2524 | Cisco 2500 Series | roteador |
| Cisco 2525 | Cisco 2500 Series | roteador |
| Catalyst 5002 | Catalyst 5000 Series | switch |
| Catalyst 5005 | Catalyst 5000 Series | switch |
| Catalyst 5500 | Catalyst 5000 Series | switch |

Características Específicas

A seguir, apresentamos uma lista de algumas das características específicas cadastradas no protótipo.

TABELA A2.5 - Algumas características específicas do protótipo

| Descrição | Grau | Observações | Custo |
|---|-------------|--|-------|
| Fazendo teste de loop ate ponto intermediário do enlace, qual o estado da interface local? | relev. 5 | tipo carac. ad. estados de itf., operação usar mesmos valores similaridade | 2 |
| Contatar nodo remoto. Outros enlaces deste ponto apresentam problemas? | relev. 5 | binária | 2 |
| Há perda de pacotes grandes com teste ping? | relev. 5 | binária | 1 |
| Contatar nodo remoto. | informativa | | 2 |
| Utilizar comando show interfaces no roteador remoto. Configuração está ok? | relev. 5 | binária | 2 |
| O roteador afetado está recebendo as rotas do roteador remoto que deveria anunciar a rota para a sub-rede sem acesso (comando sh ip route, por exemplo, onde deve ser verificado se aparecem entradas enviadas por este roteador remoto)? | filtro | binária | 2 |
| Há os protocolos RIP e IGRP no ambiente? | filtro | binária | 1 |
| As sub-redes que não conseguem acesso externo usam sub-endereçamento? | filtro | binária | 1 |
| O protocolo de roteamento utilizado possui campo para sub-endereçamento? | relev. 5 | binária | 2 |
| Contatar empresa de telecomunicações fornecedora do serviço. | informativa | | 2 |
| As tabelas de roteamento dos roteadores envolvidos possuem gateway default correto? | filtro | binária | 2 |
| Há alto percentual de tráfego IP em IP? | filtro | binária | 2 |
| Problema se agrava em certos horários definidos (como horários de pico, horário de ligação das estações, etc)? | relev. 5 | binária | 2 |
| Há alta taxa de colisões? | relev. 5 | binária | 1 |
| Rotas para as sub-redes envolvidas difundidas pelos roteadores estão corretas? | relev. 5 | binária | 2 |
| Há alta taxa de erros CRC? | relev. 5 | binária | 1 |
| Realizar teste de loop: configurar roteador local e remoto com keepalive set, e pressionar LDL no modem remoto. Qual o estado da interface local? | filtro | tipo carac. ad. estados de itf., operação usar mesmos valores similaridade | 2 |
| Realizar teste de loop: configurar roteador local e remoto com keepalive set, e pressionar LDL no modem remoto. Qual o estado da interface remota? | relev. 5 | tipo carac. ad. estados de itf., operação usar mesmos valores similaridade | 2 |

Anexo 3 Alguns Casos do Sistema

Na tabela a seguir, é apresentada uma lista dos casos presentes no protótipo. Exemplos de dois casos completos assim como a recuperação resultante destes casos é comentada na seção 6.2.

TABELA A3.1 - Lista dos casos presentes no protótipo

| Id | Tipo de Problema | Tipo de Problema Inicial | Localização do Problema |
|-----------|------------------------------------|---------------------------------|--|
| 100 | Conectividade - Físico e Config/HW | Performance | linha_serial |
| 102 | Conectividade - Físico e Config/HW | Conectividade - Genérico | linha_serial |
| 103 | Conectividade - Físico e Config/HW | Conectividade - Genérico | linha_serial |
| 104 | Conectividade - Físico e Config/HW | Conectividade - Genérico | linha_serial |
| 105 | Conectividade - Físico e Config/HW | Conectividade - Genérico | linha_serial |
| 106 | Conectividade - Físico e Config/HW | Conectividade - Genérico | linha_serial |
| 107 | Conectividade - Físico e Config/HW | Conectividade - Genérico | linha_serial |
| 108 | Conectividade - Físico e Config/HW | Conectividade - Genérico | linha_serial |
| 111 | Roteamento/Endereçamento | Roteamento/Endereçamento | outra_sub-rede_IP |
| 115 | Roteamento/Endereçamento | Roteamento/Endereçamento | outra_sub-rede_IP |
| 116 | Roteamento/Endereçamento | Roteamento/Endereçamento | outra_sub-rede_IP |
| 117 | Roteamento/Endereçamento | Roteamento/Endereçamento | outra_sub-rede_IP |
| 118 | Roteamento/Endereçamento | Roteamento/Endereçamento | outra_sub-rede_IP |
| 119 | Roteamento/Endereçamento | Alto Trafego | canal_diversas_sub-redes_IP |
| 121 | Alto Tráfego | Performance | mesma_subrede_IP- LAN_com_um_segmento |
| 122 | Conectividade - Físico e Config/HW | Conectividade - Genérico | mesma_subrede_IP- LAN_com_um_segmento |
| 123 | Conectividade - Físico e Config/HW | Conectividade - Genérico | mesma_subrede_IP- LAN_com_um_segmento |
| 125 | Roteamento/Endereçamento | Conectividade - Genérico | mesma_subrede_IP- LAN_com_um_segmento |
| 127 | Roteamento/Endereçamento | Conectividade - Genérico | outra_sub-rede_IP |
| 128 | Roteamento/Endereçamento | Performance | mesma_subrede_IP- LAN_com_um_segmento |
| 129 | Roteamento/Endereçamento | Conectividade - Genérico | outra_sub-rede_IP |
| 130 | Roteamento/Endereçamento | Roteamento/Endereçamento | outra_sub-rede_IP |
| 132 | Conectividade - Físico e Config/HW | Conectividade - Genérico | mesma_subrede_IP- LAN_com_um_segmento |
| 133 | Roteamento/Endereçamento | Roteamento/Endereçamento | outra_sub-rede_IP |

Além dos casos citados, foram coletados também casos para os tipos de problemas envolvendo as aplicações e serviços, que não foram, entretanto, implementados no protótipo. Apresentamos, a seguir, dois exemplos de casos coletados para tais problemas.

Caso 1Dados iniciais:

Tipo de Problema Inicial: Aplicação

Breve Descrição: não se consegue fazer FTP para máquina servidora.

Outras informações: Percebe-se que o problema não é com a rede, mas não se consegue completar a conexão.

Aplicação utilizada: FTP

Ações efetuadas:

1. Acompanhando o processo (com telnet para a porta correspondente ao FTP) é possível identificar onde está ocorrendo algo incorreto? Resposta: "SIM"

Solução:

Causas: Alguns servidores, como o do exemplo, exigem uso de IP reverso na máquina cliente. Uma vez que a máquina cliente não estava com o IP reverso configurado, ela não era autenticada e o processo de conexão era interrompido.

Solução Adotada: Configurar o IP reverso na máquina envolvida no problema.

Caso 2Dados iniciais:

Tipo de Problema Inicial: Serviços - Servidor de Arquivos

Breve Descrição: uma única estação da sub-rede não consegue ler mails.

Outras Informações: verificou-se que o problema ocorre devido a incapacidade da estação em montar o /var/mail (estação gordini).

Houve Alteração: recém configurados

Mensagem: estação sem permissão no servidor NFS

Possíveis Causas: NFS mal configurado, não havendo permissão para exportar na estação origem dos arquivos.

Tipo de servidor de arquivos: NFS

Ações efetuadas:

1. A estação consta no /etc/exports do servidor de arquivos, com atributos e permissões corretos? Resposta: "SIM"

2. No /etc/netgroups do servidor de NIS a máquina consta como membro da sub-rede a qual faz parte? Resposta: "SIM"

3. Pela data de criação dos .db no NIS e a data de alteração do /etc/netgroups, as alterações foram atualizadas (foi dado o push)? Resposta: "SIM"

4. O nome está sendo resolvido corretamente? (isso pode ser feito através de um ping na estação envolvida via IP, para verificar NIS, e um ping na estação envolvida via nome, para verificar DNS. Resposta "NÃO"

Solução:

Causas: Mapa do NIS estava inconsistente em relação ao mapa do DNS, e nome resolvido pelo NIS não obtinha a mesma resposta do nome resolvido pelo DNS. O mapeamento estava errado no NIS (mapa com precedência), então, na validação do NFS, o IP não conferia com o IP do pacotes de pedido de montagem.

Solução Adotada: Tabela de resolução no NIS foi corrigida.

Tipo de Problema Final: Serviços - Autenticação

Anexo 4 Especificação SDL do Sistema

O diagrama do sistema, dos blocos cliente e servidor e dos principais processos foi apresentado no capítulo 6. Apresentamos, aqui, o diagrama dos demais processos do sistema, que foram abordados nos diagramas de blocos mas ainda não foram definidos.

Inicialmente, apresentamos as definições dos módulos do bloco cliente nas figuras A4.1(*InterpretaDados*), A4.2(*MostraTela*), A4.3(*ProcessaOperações*) e A4.4 (*ComunicaçãoCliente*). Por fim, apresentamos a definição do *InterpretaBD* e do *ComunicaçãoServidor*, módulos do bloco servidor, nas figuras A4.5 e A4.6.

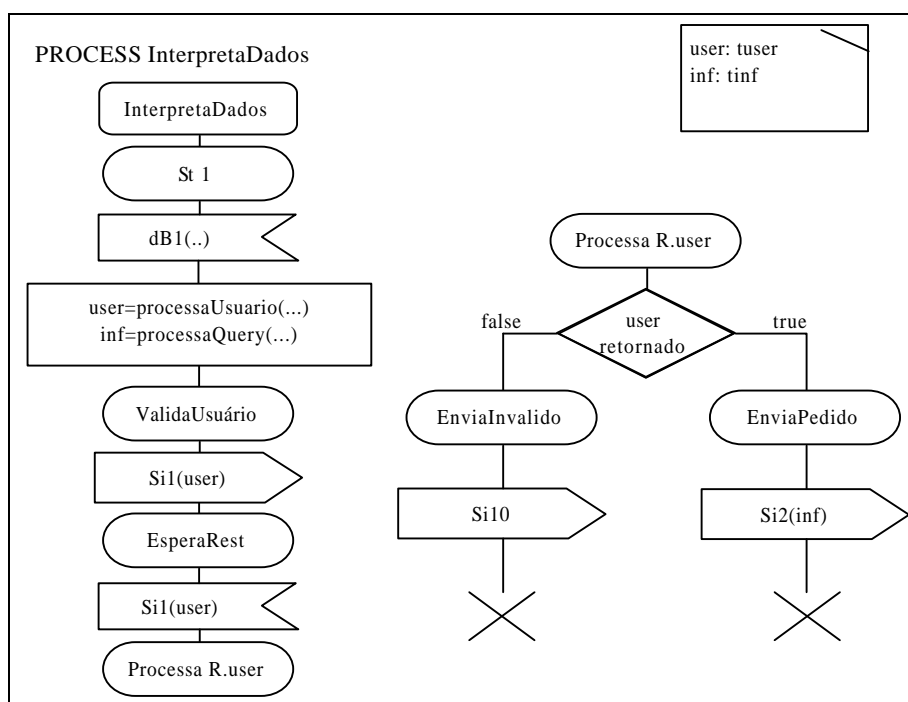


FIGURA A4.1 - Definição do InterpretaDados

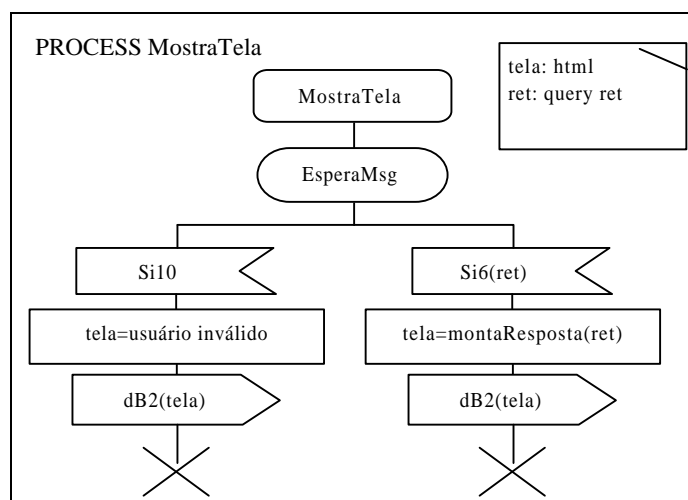


FIGURA A4.2 - Definição do MostraTela

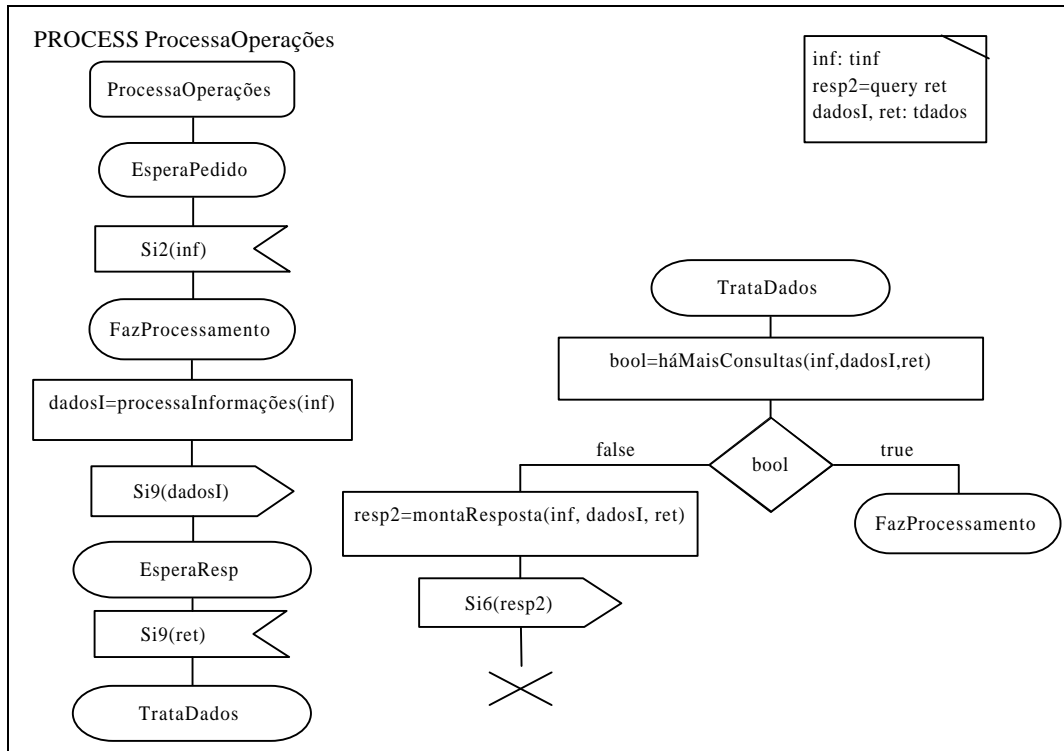


FIGURA A4.3 - Definição do ProcessaOperações

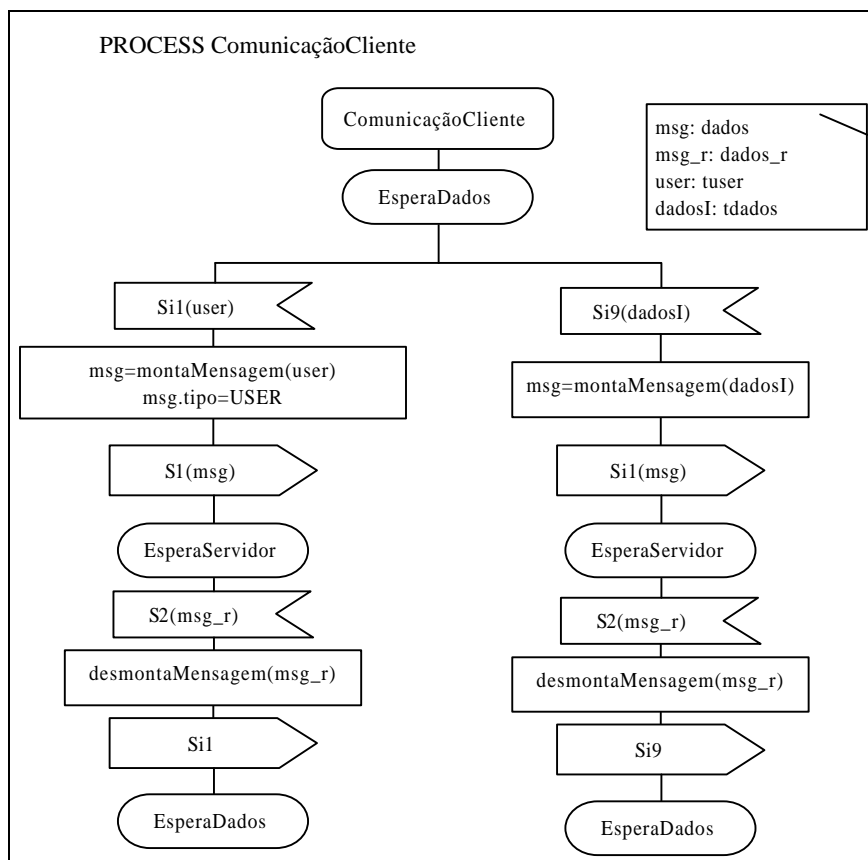


FIGURA A4.4 - Definição do ComunicaçãoCliente

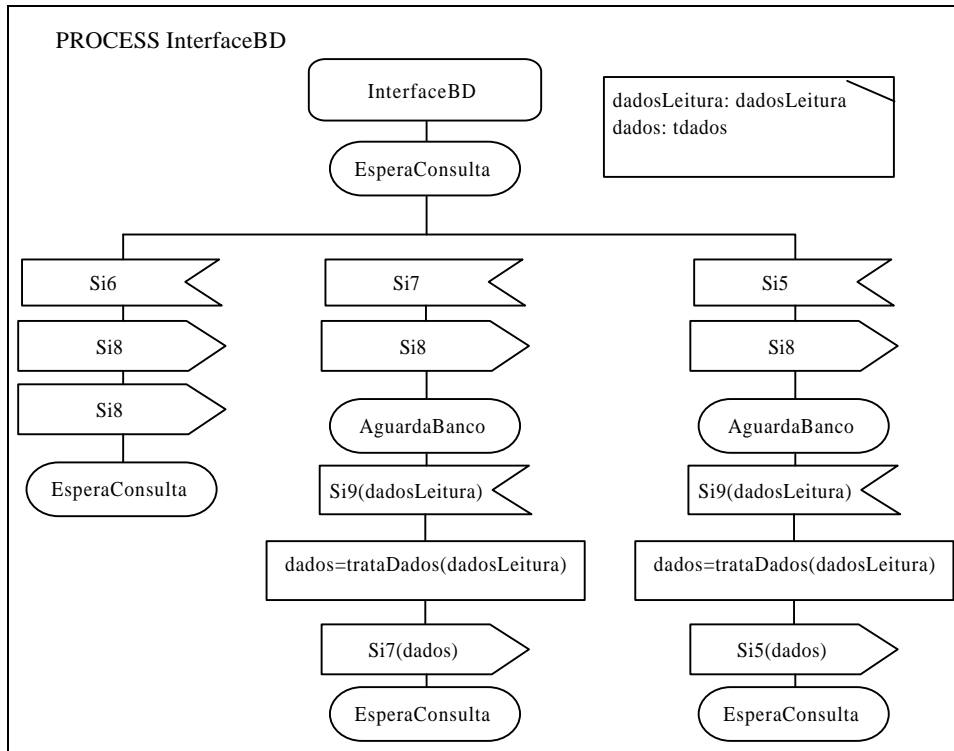


FIGURA A4.5 - Definição do InterfaceBD

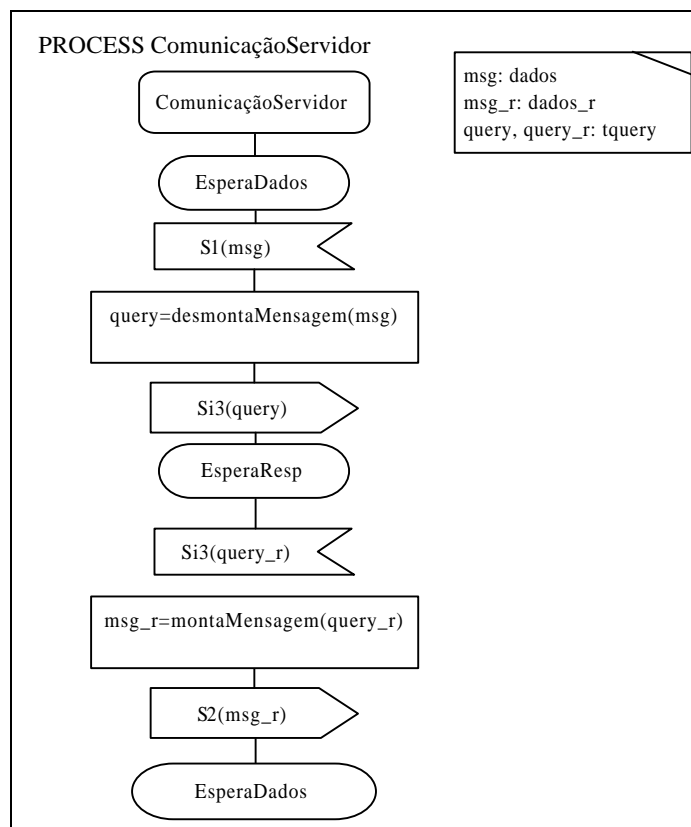


FIGURA A4.6 - Definição do ComunicaçãoServidor

Bibliografia

- [AAM 94] AAMODT, A; PLAZA, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. **AI Communications**, [S.l.], v.7, n.1, p.39-59, Mar. 1994.
- [ABE 95] ABEL, Mara; REATEGUI, Eliseo B.; CASTILHO, José M.V. Aquisição, Modelagem e Processamento de Conhecimento utilizando Raciocínio Baseado em Casos. In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, 12., 1995, Canela. **Anais...** Porto Alegre: Instituto de Informática da UFRGS, 1995. p.363-374.
- [ABE 95a] ABEL, Mara et al. Evaluating Case-Based Reasoning in a geological Domain. In: INTERN. CONFERENCE AND WORKSHOP ON DATABASE AND EXPERT SYSTEMS APLICATIONS, DEXA, 6., 1995, London. **Proceedings...** Berlin: Springer-Verlag, 1995. p.364-373.
- [ABE 96] ABEL, Mara. **Um Estudo sobre Raciocínio Baseado em Casos: Trabalho Individual.** Porto Alegre: CPGCC da UFRGS, 1996.
- [ACO 92] ACORN, T.; WALDEN, S. H. **SMART: Support Management Automated Reasoning Technology for Compaq Customer Service.** [S.l.:s.n.], 1992
- [BRA 91] BRANDAU, Richard.; LEMMON, Alan.; LAFOND, Carol. Experience with Extended Episodes: Cases with Complex Temporal Structure In: DARPA WORKSHOP ON CASE-BASED REASONING, 1991, Washington. **Proceedings...** San Francisco: Morgan Kaufmann, 1991.
- [CAB 97] CABLETRON SYSTEMS. **SpectroRX: Add-on module for network and systems management.** Rochester: Cabletron Systems, 1997.
- [CAB 97a] CABLETRON SYSTEMS. **Technical Tips.** Cabletron Systems. Documento disponível em <http://www.cabletron.com/support/techtips/> (1997).
- [CAB 97b] CABLETRON SYSTEMS. **Troubleshooting Tips.** Cabletron Systems. Documento disponível em <http://www.cabletron.com/support/trbltips/> (1997).
- [CAR 93] CARVALHO, T.C.M.B. (Org.) **Gerenciamento de Redes: Uma Abordagem de Sistemas Abertos.** São Paulo: Makron Books, 1993.
- [CAR 94] CARVALHO, T.C.M.B. (Org.) **Arquiteturas de Redes de Computadores OSI e TCP/IP.** São Paulo: Makron Books, 1994.
- [CAU 95] CAULIER, P.; HOURIEZ, B. A Case-Based Approach in Network Traffic Control. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS. INTELLIGENT SYSTEMS FOR THE 21ST CENTURY. 1995, Vancouver, Canada. **Proceedings...** Piscataway: IEEE, 1995. p.1430-1435.
- [CHA 96] CHANG, Kai H. et al. A self-improving helpdesk service system using case-based reasoning techniques. **Computers in Industry**, Eindhoven, v.30, n.2, p.113-125, Sept. 1996.

- [CIS 97] CISCO SYSTEMS, Inc. **Internetwork Design Guide**. 1997. Documento disponível em <http://www.cisco.com/univercd/cc/td/doc/cisintwk/idg4/index.htm>.
- [CIS 97a] CISCO SYSTEMS, Inc. **Internetwork Troubleshooting Guide**. 1997. Documento disponível em http://www.cisco.com/univercd/cc/td/doc/cisintwk/itg_v1/index.htm.
- [CIS 97b] CISCO SYSTEMS, Inc. **Internetworking Case Studies**. 1997. Documento disponível em <http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/index.htm>.
- [CIS 97c] CISCO SYSTEMS, Inc., **Troubleshooting Internetworking Systems**. 1997. Documento disponível em http://www.cisco.com/univercd/cc/td/doc/cisintwk/tis_doc/index.htm.
- [COM 91] COMER, Douglas E. **Internetworking With TCP/IP Vol I: Principles, Protocols, and Architecture**. 2.ed. Englewood Cliffs: Prentice-Hall, 1991. 547p.
- [CRO 88] CRONK, R. N.; CALLAHAN, P. H.; BERNSTEIN, L. Rule-Based Expert Systems for Network Management and Operations: An Introduction. **IEEE Network Magazine**, New York, v.5, n.2 , p.7-21, Sept. 1988. Artigo reimpresso em ERICSON, E.C.; ERICSON, L.T.; MINOLI, D. Expert systems applications in integrated network management. Norwood: Artech House, 1989. 451p. p.94-104.
- [DRE 95] DREO, Gabi., VALTA, Robert. Using master tickets as a storage for problem-solving expertise. In: INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, 4., 1995. **Proceedings...** London: Chapman & Hall, 1995. 716p. p.328-340.
- [ENC 95] ENCK, John; BECKMAN, Mel. **LAN to WAN interconnection**. New York: McGraw-Hill, 1995. 265p.
- [ERI 89] ERICSON, E.C.; ERICSON, L.T; MINOLI, D. (Eds.) **Expert systems applications in integrated network management**. Norwood: Artech House, 1989. 451p.
- [GOY 90] GOYAL, S.; WEIHMAYER, R.; BRANDAU, R. Intelligent systems in the future network In: IEEE NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM OPERATIONS FOR THE INFORMATION AGE, 1990, San Diego. **Proceedings...** New York: IEEE, 1990.
- [GOY 91] GOYAL, S. Knowledge technologies for evolving networks. In: INTEGRATED NETWORK MANAGEMENT, 2., 1991. **Proceedings...** Amsterdam: Elsevier Science, 1991.
- [HAR 89] HARMON, P.; SAWYER, B. **Creating Expert Systems For Business and Industry**. New York: John Wiley, 1989. 329p.
- [HAR 97] HARTMANN, L. **Gerência de Roteamento em Redes Interconectadas**. Porto Alegre: CPGCC da UFRGS, 1997. Dissertação de Mestrado.
- [HUN 98] HUNT, Craig. **TCP/IP Network Administration**. 2.ed. Sebastopol: O'Reilly, 1998.
- [JAC 86] JACKSON, P. **Introduction to expert systems**. Wokingham: Addison-Wesley, 1986.

- [JOH 92] JOHNSON, D. **NOC Internal Integrated Trouble Ticket System**: Funcional Specification Wishlist. RFC 1297. [S.l.]: IAB, 1992.
- [KOL 93] KOLODNER, Janet. **Case-Based Reasoning**. San Mateo: Morgan Kaufmann, 1993. 668p.
- [KOP 88] KOPEIKINA, Ludmila et al. Case Based Reasoning for Continuous Control In: WORKSHOP ON CASE-BASED REASONING, DARPA, 1988. **Proceedings...** San Mateo: Morgan Kaufmann, 1988.
- [KRI 91] KRISHNAN, I.; ZIMMER, W. (Eds.). INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, 2., 1991, Crystal City, US. **Proceedings...** Amsterdam: North-Holland, 1991.
- [LEA 96] LEAKE, David (Ed). **Case-Based Reasoning: Experiences, Lessons & Future Directions**. Menlo Park: AAAI Press/MIT Press, 1996.
- [LEI 96] LEINWAND, A.; CONROY, K. F. **Network Management**. A Practical Perspective. 2.ed. Menlo Park: Addison-Wesley, 1996.
- [LEW 92] LEWIS, Lundy. Case-based Reasoning with Fuzzi Inference: A Backend to Network Analyzers. In: INTERNATIONAL CONFERENCE ON FUZZI THEORY AND TECHNOLOGY, 1., 1992. **Proceedings...** [S.l.:s.n.], 1992.
- [LEW 93] LEWIS, Lundy. A Case-Based Reasoning Approach to the Management of faults in Communications Networks. **IEEE INFOCOM**, San Francisco, v.3, p.1422-1429, Apr. 1993. Trabalho apresentado no Annual Joint Conference of The IEEE Computer and Communications Societies, 20., 1993.
- [LEW 93a] LEWIS, Lundy.; DREO, Gabi. Extending Trouble Tickets Systems to Fault Diagnostics. **IEEE Network**, New York, v.7, n.6, p.44-51, Nov. 1993.
- [LEW 95] LEWIS, Lundy. **Managing Computer Networks: A Case-Based Reasoning Approach**. Norwood: Artech House, 1995. 205p.
- [LEW 95a] LEWIS, Lundy. AI and Intelligent Networks in the 1990s and into the 21st Century. In: LIEBOWITZ, J; PRERAU, D. (Eds.) **Worldwide Intelligent Systems: Approaches to Telecommunications and Network Management**. Amsterdam: IOS Press, 1995.
- [MAD 94] MADRUGA, E. L. **Ferramentas de Apoio à Gerência de Falhas e Desempenho em Contexto Distribuído**. Porto Alegre: CPGCC da UFRGS, 1994. Dissertação de Mestrado.
- [MAR 94] MARIR, Farhi; WATSON, Ian. Case-Based Reasoning: A Categorised Bibliography. **The Knowledge Engineering Review**, London, v.9, n.4, p.355-381, 1994. Documento disponível também em <http://www.salford.ac.uk/docs/depts/survey/staff/IWatson/cbrefs.htm>
- [MAT 95] MATSUMOTO, K.; HASHIMOTO, K.; OBANA, S. Design and implementation of real-time expert system for troubleshooting in international telephone networks. In: INTERNATIONAL CONFERENCE. INDUSTRIAL AND ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE AND EXPERT

- SYSTEMS, 8., 1995, Melbourne, Australia. **Proceedings...** Newark: Gordon & Breach, 1995. p.345-352.
- [MEI 97] MEIRA, D.M; NOGUEIRA, J.M.S. Métodos e Algoritmos para Correlação de Alarmes em Redes de Telecomunicações. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 15., 1997, São Carlos. **Anais...** São Carlos: UFSCAR, 1997. p.79-98.
- [MEL 97] MELCHIORS, Cristina. Suporte para Gerenciamento Distribuído Colaborativo. In: SEMANA ACADÊMICA DO CPGCC, 2., 1997, Porto Alegre. **Anais...** Porto Alegre: CPGCC da UFRGS, 1997. 243p. p.211-214.
- [MEL 99] MELCHIORS, Cristina. **Um Estudo sobre Raciocínio Baseado em Casos**: relatório técnico. Porto Alegre, PPGC da UFRGS, 1999. (RP 301).
- [MEL 99a] MELCHIORS, Cristina; TAROUCO, Liane M. R. Fault Management in Computer Networks Using Case-Based Reasoning: DUMBO System. In: INTERNATIONAL CONFERENCE ON CASE-BASED REASONING, 3., 1999, Seon Monastery, Germany. **Proceedings...** Berlin: Springer, 1999. p. 510-524. (Lecture Notes in Artificial Intelligence, v. 1650).
- [MEL 99b] MELCHIORS, Cristina; TAROUCO, Liane M. R. DUMBO: Uma Abordagem para Gerenciamento de Falhas Utilizando Raciocínio Baseado em Casos. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 17., 1999. Salvador. **Anais...** Salvador: SBC, 1999. p. 575-576.
- [MIL 95] MILLER, Mark. A. **Troubleshooting Internetworks**: tools, techniques, and protocols. San Mateo: M&T Books, 1995. 528p.
- [MIL 96] MILLER, Mark. A. **Troubleshooting TCP/IP**: analyzing the protocols of the Internet. New York: M&T Books, 1996. 772p.
- [NAS 94] NASSAR, D. J. **Network Optimization and Troubleshooting**. Indianapolis: New Rider Publishing, 1994. 639p.
- [NEM 95] NEMETH, Evi et al. **UNIX System Administration Handbook**. New Jersey: Prentice Hall, 1995. 779 p.
- [NUN 97] NUNES, C. M. **Um Discriminador Inteligente de Eventos de Rede para o Ambiente CINEMA**. Porto Alegre: CPGCC da UFRGS, 1997. Dissertação de Mestrado.
- [PAR 97] PARNELL, Terè. **LAN Times Guide to Wide Area Networks**. Berkeley: McGraw-Hill, 1997. 528p.
- [PEN 99] PENIDO, G.; NOGUEIRA, J.M.; MACHADO, C. An automatic fault diagnosis and correction system for telecommunications management. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, 6. 1999. **Proceedings...** [S.l.]:IEEE, 1999.
- [PHD 97] PHD. **When Someone on eht Other End of the Line Depends on You Depend On Us PHD**: The Virtual Help Desk. Stamford: PHD, 1997.
- [RAM 95] RAMAN, Pradeep. **Case-Based Reasoning**: An Implemented Methodology. Auburn:

Auburn University, 1995. (Technical Report 04-95, MCSE Desingner Project Report).

- [RHE 90] RHEIN, Jon et al. **The POSTGRES User Manual**. Berkeley: University of California, 1990. 31p.
- [SIM 92] SIMOUDIS, Evangelos. Using Case-Based Retrieval for Customer Technical Support. **IEEE Expert**, Los Alamitos, v.7, n.5, p.7-12, Oct.1992.
- [SIN 96] SINGH, Harry. **Heterogeneous Internetworking: networking technically diverse operanting systems**. New Jersey: Prentice-Hall, 1996. 644 p.
- [SNG 90] SNG, Dennis Cheng-Hong. **Network Monitoring and Fault Detection on the University of Illinois at Urbana-Champaign Campus Computer Network**. Urbana-Champaign: DCS/UIUC, 1990. (Technical Report UIUCDCS-R-90-1595).
- [STA 93] STADLER, M. Case-Based Reasoning for Network Management. In: TOPICS IN CASE-BASED REASONING EUROPEAN WORKSHOP EWCBR-93, 1., 1993. Berlin: Springer-Verlag, 1994. p. 414-423.
- [STA 97] STALLINGS, William. **Local & Metropolitan Area Networks**. 5.ed. New Jersey: Prentice-Hall, 1997. 605 p.
- [STE 95] STEFIK, M. **Introduction to Knowledge Systems**. San Francisco: Morgan Kaufmann Publishers, 1995. 870p.
- [STO 90] STONEBRAKER, Michael et al. **POSTGRES Reference Manual**. Berkeley, CA, EUA: University of California, 1990. 128p.
- [TAN 96] TANENBAUM, Andrew S. **Computer Networks**. 3.ed. New Jersey: Prentice-Hall, 1996. 814p.
- [TAR 90] TAROUCO, L. M. R. **Inteligência Artificial Aplicada ao Gerenciamento de Redes de Computadores**. São Paulo: Escola Politécnica da USP, 1990. Tese de Doutorado.
- [TAR 96] TAROUCO, Liane M.R. et al. Um ambiente para gerenciamento integrado e cooperativo. In: WORKSHOP SOBRE ADMINISTRAÇÃO e INTEGRAÇÃO DE SISTEMAS, WAIS, 2., 1996, Fortaleza. **Anais...** Fortaleza: UFC, 1996. p.235-246.
- [TAR 96a] TAROUCO, Liane M.R. et al. Um ambiente para gerenciamento integrado e cooperativo. In: WORKSHOP DE GERÊNCIA DE REDES DE COMPUTADORES, 1996, Fortaleza. **Anais...** Fortaleza: UFC, 1996.
- [TIS 97] TISCHLER, F.; LAMM, J. **The Intranet/Support Connection Support Management**. [S.l.]: PHD, 1997.
- [TRI 92] TRINDADE, Rodrigo S. **Um Estudo da Linguagem SDL para Especificação e Teste de Protocolos: Trabalho Individual**. Porto Alegre: CPGCC da UFRGS, 1992. 87p.
- [TUR 92] TURBAN, Efraim. **Expert Systems and Applied Artificial Intelligence**. New York:

Macmillan Publishing, 1992.

- [UDU 96] UDUPA, D. K. **Network Management Systems Essentials**. New York: McGraw-Hill, 1996.
- [WAT 94] WATSON, Ian; MARIR, Farhi. Case-based Reasoning: A review. **The Knowledge Engineering Review**, London, v.9, n.4, p. 327-354, 1994.
- [WAT 95] WATSON, Ian. **Case-Based Reasoning Development Tools: A Review**. 1995. Documento obtido em <http://www.salford.ac.uk/docs/depts/survey/staff/IWatson/cbrefs.htm>
- [WAT 97] WATSON, Ian. **Applying Case-Based Reasoning: Techniques for Enterprise Systems**. San Francisco: Morgan Kaufmann, 1997. 289p.
- [WIL 94] WILLIAMS, C. CLAYTON, B. D. **Case Based Retrieval**. [S.l.]: Inference Corporation, 1994.