UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

INSTITUTO DE INFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Raciocínio Baseado em Casos Aplicado a Diversos Domínios de Problema

por

CINARA TEREZINHA MENEGAZZO

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

Prof. Dra. Maria Janilce B. Almeida
Orientadora

Prof. Dra. Liane M. R. Tarouco
Co-Orientadora

Porto Alegre, maio de 2001.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Menegazzo, Cinara Terezinha

Raciocínio Baseado em Casos Aplicado a Diversos Domínios de Problema / por Cinara Terezinha Menegazzo – Porto Alegre: PPGC da UFRGS, 2001.
166f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR – RS, 2001. Orientadora: Almeida, Maria Janilce B.; Co-orientadora: Tarouco, Liane M. R.

1. Raciocínio baseado em casos. 2. Sistema Especialista . 3. Gerenciamento de Segurança. 4. Segurança em Redes TCP/IP. 5. Redes de Computadores. I. Tarouco, M. R. II. Almeida, Maria Janilce B. III. Título.

Agradecimentos

Existem momentos na nossas vidas que dependemos exclusivamente de apoio e ajuda, os passos galgados no desenvolvimento deste trabalho foi um desses momentos que só cheguei ao final devido ao apoio recebido.

Meu primeiro e mais especial obrigado vai ao meu querido “Luzinho”, por tudo o que teve que compartilhar e suportar junto comigo e acima de tudo pela amizade, amor e aquele “Claro que vais terminar, com certeza”. Me fez acreditar que eu poderia.

Um agradecimento muito especial a minha Co-Orientadora Drª. Profrª Liane M. Tarouco, por sem me conhecer acreditar em mim e me fazer trabalhar sempre entendendo o que cada um pode fazer e como. Obrigada pelos ensinamentos e pela oportunidade de convívio.

A minha orientadora Drª. Profrª Maria Janilce B. Almeida, por me fornecer esta oportunidade de trabalho e realização.

Aos meus colegas de serviço que agüentaram a minha fase crítica, minha divisão e acima de tudo por terem acompanhado todos os passos do meu trabalho, dando-me sempre as melhores respostas em tudo, inclusive me ensinando. Aos meus três grandes amigos para toda a vida e para tudo o que acontecer Dorian, Julíbio e Luciano. Obrigado por serem bons, verdadeiros e companheiros, obrigado a vida por eles.

Meu muito obrigado ao Ricardo (como é bom saber matemática), a Cristina, ao Jean (especialista em segurança) e ao Mag, pelo apoio recebido e compartilhamento de trabalho, me fez saber que ainda existem pessoas de boa intenção.

A UDESC (Universidade do Estado de Santa Catarina) em especial na pessoa do magnífico reitor Raimundo Zumblick,do por investir em mim e me fazer crescer profissionalmente, além de todo o apoio recebido.

Aos professores do PPGC da UFRGS, pelo valioso conhecimento que me passaram ao longo desta etapa e aos colegas de mestrado pelo convívio proporcionado.

Sumário

Lista de Abreviaturas **7**

Lista de Figuras **9**

Lista de Tabelas **10**

Lista de Gráficos **11**

Resumo **12**

Abstract **13**

1 Introdução **14**

1.1 Objetivos **16**
1.2 Organização deste Trabalho **17**

2 Sistemas Especialistas **18**

2.1 Case-Based Reasoning **20**
2.1.1 Histórico *22*
2.1.2 Componentes de um Sistema Baseado em Casos *24*
2.1.3 Processos de Desenvolvimento de um Sistema CBR *26*

3 Ferramentas Existentes **35**

3.1 INRECA **36**
3.2 CASUEL **37**
3.3 ReMind **39**
3.3.1 Representação e Conteúdo dos Casos *39*
3.3.2 Aquisição *39*
3.3.3 Mecanismos de Aprendizagem *39*
3.3.4 Indexação e Similaridade *40*
3.3.5 Adaptação *41*
3.3.6 Comunicação com outros Mecanismos de Raciocínio *41*
3.3.7 Apresentação ao Usuário *41*
3.4 CaseAdvisor **41**
3.4.1 Representação e Conteúdo dos Casos *41*
3.4.2 Aquisição de Casos *42*
3.4.3 Mecanismo de Aprendizado *43*
3.4.4 Indexação e Similaridade *43*
3.4.5 Adaptação *44*
3.4.6 Comunicação com outros Mecanismos de Raciocínio *45*
3.4.7 Apresentação para o Usuário *45*
3.5 CBR3 ..... **45**
3.5.1 Representação e Conteúdo dos Casos *45*
3.5.2 Aquisição *46*
3.5.3 Mecanismos de Aprendizagem *46*
3.5.4 Indexação e Similaridade *46*

3.5.5 Adaptação *47*
3.5.6 Comunicação com outros Mecanismos de Raciocínio *47*
3.5.7 Apresentação ao Usuário *48*
3.6 Avaliação das Ferramentas *Remind*, *CaseAdvisor* e *CBR3* **48**

4 Gerenciamento de Redes **51**

4.1 Importância de um Gerenciamento **52**
4.2 Gerenciamento de Segurança **53**
4.3 Política de Segurança **56**
4.4 Gerenciamento e Sistema de Apoio à Decisão **58**
4.5 Sistemas de *Trouble Ticketing* (STT) ou Sistemas *Help Desk* (SHD) **59**
4.5.1 Estrutura de um Trouble Ticket *61*
4.5.2 Estabelecendo um Sistema de Trouble Ticket para Redes *62*
4.6 Tendências para Soluções ao Gerenciamento **63**
4.7 Sistemas Especialistas para o Gerenciamento de Segurança **65**

5 Segurança em Redes TCP/IP **67**

5.1 Identificação **67**
5.2 Formas e Tipos de Ataques **69**
5.3 Localização do Ataque **72**
5.4 Técnicas mais Utilizadas **72**
5.4.1 Scan *74*
5.4.2 AW (Ataque a Servidor Web) *75*
5.4.3 Bug de Software *75*
5.4.4 Backdoors *75*
5.4.5 Vírus *75*
5.4.6 Denial of Service (DoS) *76*
5.4.7 Distributed Denial of Service (DDoS) *76*
5.4.8 Spoofing *76*
5.4.9 Sniffers *76*
5.4.10 Buffer Overflow *76*
5.5 Protocolo, Infra-estrutura, Serviço e Procedimento Utilizado para Ataque **76**
5.5.1 ICMP (Internet Control Message Protocol) *77*
5.5.2 TCP (Transfer Control Protocol) *77*
5.5.3 UDP (User Datagram Protocol) *79*
5.5.4 IP (Internet Protocol) *80*
5.5.5 RIP (Routing Information Protocol) *80*
5.5.6 Serviço de FTP (File Transfer Protocol) *80*
5.5.7 DNS (Domain Name Service) *81*
5.5.8 Proxies e Servers *81*
5.5.9 Internet Firewall *81*
5.5.10 SMTP (Simple Mail Transfer Protocol) *82*
5.6 Resolução de Problemas **82**

6 União de Sistema Especialista, Sistema de Registro de Problemas e Segurança em Redes de Computadores **85**

6.1 Estrutura do Sistema SABER - Sistema Apoiador Baseado em Raciocínio **86**
6.1.1 Definições e Aquisições do Conhecimento *87*
6.1.2 Módulo de Interface Especialista *91*
6.1.3 Interface Consulta de Problema *98*

7 Implementação do Protótipo 101

7.1 Plataforma de Hardware 101
7.2 Recursos de Software 101
7.3 Arquitetura do Sistema 101
7.4 Acesso para Engenheiro de Conhecimento 102
7.5 Acesso a Usuário 107
7.6 Avaliação do Protótipo 112
7.6.1 Capacidade de Representação Formal dos Casos 113
7.6.2 Avaliação da Máquina de Inferência 114
7.6.3 Uso do Protótipo para Vários Domínios de Problema 117

8 Conclusão 118

8.1 Trabalhos Futuros 120
Anexo 1 Criação de Base e Tabela de Dados no MySQL 122
Anexo 2 Redes Semânticas do Domínio de Segurança 123
Anexo 3 Relação de Características de Casos Cadastrados 134
Anexo 4 Problemas Consultados e Casos Recuperados 138
Anexo 5 Especificação do Sistema SABER em SDL 144

Bibliografia 160

Lista de Abreviaturas

AW	Ataque a Servidor WEB
BC	Base de Casos
BD	Banco de Dados
CAIS	Centro de Atendimento a Incidentes de Segurança
CBR	*Case Based Reasoning*
CERT	*Computer Emergency Response Team*
CGI	*Common Gateway Interface*
CMIP	*Common Management Information Protocol*
DdoS	*Distributed Denial of Service*
DNS	*Domain Name System*
DoS	*Denial of Service*
FTP	*File Trensfer Protocol*
HD	*Help Desk*
IA	Inteligência Artificial
ICMP	*Internet Control Message Protocol*
INRECA	*Induction and Reasoning from Cases*
ISS	*Initial Sequence Number*
ITU	*Internacional Telecomunications Union*
KBS	*Knowledge Based Systems*
MIB	*Management Information Base*
MOP's	*Memory Organization Packets*
NFS	*Network File Systems*
NIC	*Network Information Center*
NOC	Centro de Operações de Rede

RBC	Raciocínio Baseado em Casos
RIP	*Routing Information Protocol*
RMON	*Remote Network Monitoring*
SABER	Sistema Apoiador Baseado em Raciocínio
SBC	Sistemas Baseados em Conhecimento
SDL	*Specification and Description Languague*
SE	Sistemas Especialistas
SGR	Sistema de Gerenciamento de Rede
SHD	Sistemas de *Help Desk*
SMTP	*Simple Mail Tranfer Protocol*
SNMP	*Simple Network Management Protocol*
SQL	*Structured Query Language*
STT	Sistemas de *Trouble Ticketing*
TCP/IP	*Transfer Control Protocol*/*Internet Protocol*
UDP	*User Datagram Protocol*

Lista de Figuras

FIGURA 2.1 - Esquema de um Sistema Especialista 18
FIGURA 2.2 - Ciclo de vida de um sistema CBR 20
FIGURA 3.1 - A arquitetura do INRECA. 37
FIGURA 3.2 – Exemplo de objeto para o INRECA. 38
FIGURA 3.3 - Herança e Hierarquia no CASUEL. 38
FIGURA 3.4 – Um Q-model do ReMind. 40
FIGURA 3.5 – O mecanismo do CaseAdvisor 44
FIGURA 4.1 – Exemplo de um trouble ticket. 61
FIGURA 4.2 - Procedimentos para solução de problemas por SE. 66
FIGURA 5.1 - Ataque simples 70
FIGURA 5.2 - Ataque doorknob. 70
FIGURA 5.3 - Ataque em cadeia. 70
FIGURA 5.4 - Ataque looping. 71
FIGURA 5.5 – Interrupção. 71
FIGURA 5.6 – Modificação. 71
FIGURA 5.7 – Interceptação. 72
FIGURA 5.8 – Fabricação. 72
FIGURA 5.9 - Desenvolvimento de um ataque por Syn Flood 78
FIGURA 5.10 - Ataque por predição TCP. 79
FIGURA 5.11 - Procedimentos básicos de combate a ataques e invasões. 84
FIGURA 6.1 - A estrutura do Sistema SABER 86
FIGURA 6.2 - Descrição de um problema em um STT. 88
FIGURA 6.3 – Rede semântica de um problema de Syn Flood 90
FIGURA 7.1 - Inicialização do protótipo. 102
FIGURA 7.2 - Busca por similaridade no protótipo SABER 104
FIGURA 7.3 - Ordem de cadastramento do protótipo SABER. 105
FIGURA 7.4 - Inclusão de uma pergunta no protótipo SABER. 105
FIGURA 7.5 - Inclusão de respostas para uma pergunta. 106
FIGURA 7.6 - Tela de opções de registros de problemas do protótipo. 107
FIGURA 7.7 - Tela de inclusão de registro de problemas no protótipo. 107
FIGURA 7.8 - Processo de consulta de registro de problema. 109
FIGURA 7.9 - Resultado de busca por similaridade na descrição do usuário. 110
FIGURA 7.10 - Relação de perguntas expostas ao usuário. 110
FIGURA 7.11 - Resultado de uma busca refinada. 112

Lista de Tabelas

TABELA 3.1 - Disposição de dados na inclusão de caso no ReMind pelo Case Editor.39
TABELA 3.2 – Comparação de casos feita pelo ReMind, são mostrados os casos recuperados, lado a lado, tendo o caso proposto sempre mais a direita. 40
TABELA 3.3 – Exemplo de como é colocado um caso na interface do CaseAdvisor.42
TABELA 3.4 – Casos detectados como redundantes tratados pelo CaseMantainer... 42
TABELA 3.5 – Comparação das Ferramentas 50
TABELA 4.1 - Tabela de Relação de Segurança com Serviço, Uso e Custo. 57
TABELA 5.1 - Tabela de portas TCP e respectivo serviço. 73
TABELA 6.1 – Perguntas para guiar o problema. 88
TABELA 6.2 – Descrição do problema de Syn Flood, segundo uma OVL..... 92
TABELA 6.3 – Estrutura completa de uma instância de problema para o SABER.... 93
TABELA 6.4 – Atribuição de Índices para casos 95
TABELA 6.5 – Abertura de um registro de problema. 100
TABELA 7.1 - Tabelas do protótipo para um domínio de problema..... 103
TABELA 7.2 - Relação das classes de casos testados. 114
TABELA A 3.1. – Características de DoS 134
TABELA A 3.2. – Características de AW..... 134
TABELA A 3.3. – Características de Vírus 136
TABELA A 3.4. – Características de Axfr..... 136
TABELA A 3.5. – Características de Af..... 136
TABELA A 3.6. – Características de Maçã 137
TABELA A 3.7. – Características de Uva..... 137
TABELA A 3.8. – Características de Laranja 137

Lista de Gráficos

GRÁFICO 4.1 - Relação Custo e Risco de uma Política de Segurança. 57
GRÁFICO 5.1 – Incidentes Reportados para o NIC-BR de Janeiro a Junho de 2000. . 74

Resumo

A procura por uma forma de, fácil e rapidamente, retirar conhecimento de um especialista e transmiti-lo a outros profissionais tem levado ao desenvolvimento de diversas pesquisas que unem metodologias de raciocínio com o atendimento a problemas.

Os ambientes corporativos demonstram sua eficiência baseados na maneira como desenvolvem suas tarefas. Cada vez mais, buscam alternativas que os ajudem a responder por atividades e problemas ocorridos, as quais podem significar um processo de manutenção que pode decidir o nível de eficiência e competência da organização.

Estas alternativas compreendem ferramentas ou profissionais, os quais transformaram-se em especialistas por adquirirem conhecimento e capacidade em prover soluções a problemas. As características de um problema podem ser alteradas sob alguns aspectos mas, mesmo em domínios mais complexos como a gerência de redes de computadores ou a medicina, algo que foi aprendido sempre tem utilidade em novas situações. Este é o tipo de conhecimento e raciocínio próprios de um especialista, ou seja, o uso de suas experiências.

Raciocínio Baseado em Casos (*case-based reasoning*) é uma metodologia de inteligência artificial que apresenta a forma de raciocínio semelhante à de um especialista, onde o raciocínio é obtido por um processo de recordar um exemplo concreto. Porém, as pesquisas que a utilizam, geralmente, desenvolvem trabalhos para um específico tipo de domínio de problema, o que resulta em alterações de programação, caso estes desejem ser adaptados para outros domínios. Este procedimento, muitas vezes, é tido como difícil e trabalhoso.

Baseando-se neste contexto, o presente trabalho apresenta um mecanismo que fornece inclusões de conhecimento de especialistas, independente do tipo de domínio de problema, e raciocínio sobre este conhecimento, de forma a auxiliar usuários com problemas referentes ao domínio cadastrado.

Para tanto, a implementação baseou-se na união de sistemas de registros de problemas (*trouble ticket systems*) com raciocínio baseado em casos, propondo uma forma auxiliar no conhecimento e busca de soluções em domínios de problema. O estudo, além de fazer uso das metodologias citadas acima, usa o domínio de gerenciamento de segurança em redes de computadores para exercitar suas funções, provar sua utilidade e dificuldades. Assim, um estudo mais detalhado sobre os problemas que cercam o domínio de segurança em redes TCP/IP foi desenvolvido.

Palavras-Chave: Raciocínio Baseado em Casos, Inteligência Artificial, Sistemas Especialistas, Gerenciamento de Redes de Computadores, Gerenciamento de Segurança em Redes TCP/IP, Sistemas de Registros de Problemas.

TITLE*:* "CASE-BASED REASONING APPLIED TO VARIOUS PROBLEM DOMAINS"

Abstract

The search for an easy and fast way to join together the knowledge of an expert in a problem domain with the transmission of this very knowledge to other professionals with the same problem has led to the development of a range of research linking reasoning methodologies and problem tending.

Corporate environments demonstrate their efficiency based on the way they develop their tasks. More and more, they search for alternatives to help them answer for activities and occurring problems, which may mean a maintenance process that can decide the level of efficiency and competence of the corporation.

These alternatives always revolve around hired tools or professionals, who become experts due to the amount of acquired knowledge about a certain matter and their ability in providing solutions to problems. The characteristics of a problem can be altered under certain aspects but, even in more complex domains like computer network management or medicine, something that has been learned is always useful in new situations. This is the type of knowledge and reasoning characteristic of an expert, i.e., the use of their own experiences.

An AI methodology that develops the same pattern of thinking of an expert is the Case-based Reasoning. However, the researches usually develop work for a specific type of problem domain, causing programming alterations when adapted to other domains, a procedure usually perceived as difficult and labor intensive.

Based on the context above, this work presents a mechanism that provides inclusions of experts' knowledge, no matter the problem domain type and, reasoning about this knowledge to help users with problems regarding the recorded domain.

Thus, the implementation has been based on linking up trouble ticket systems (TTS) with case-based reasoning (CBR), with the proposal of an auxiliary model in the search for and knowledge about solutions in problem domain. The study, besides utilizing the methodologies already mentioned, also utilizes the security management domain in computer networks to exercise its functions and prove its usefulness. Therefore, a more detailed study about problems regarding security domain in TCP/IP networks has been developed.

Keywords: Case-based Reasoning; Artificial Intelligence; Expert Systems; Network Management; TCP/IP Networks Security Management; Trouble Ticket Systems.

1 Introdução

O homem desenvolveu uma forma de raciocínio que lhe forneceu a capacidade de pensar, comunicar-se, transmitir conhecimento de forma estruturada e ultra-sofisticada, raciocinar e decidir, chamada de inteligência humana.

A habilidade em resolver problemas é, freqüentemente, usada como uma medida de inteligência tanto para humanos quanto para máquinas.

As soluções de problemas são fundamentais para todo tipo de profissão ou área de atuação. O ser humano consegue alcançá-las ao fazer uso de sua experiência em fatos passados. Com o computador, este tipo de tarefa não é tão simples, pois ele não possui capacidade de, naturalmente, deduzir situações ou auto-organizar sua memória.

Desenvolver modelos que permitam simular o comportamento inteligente em computadores tem sido a principal meta das pesquisas em Inteligência Artificial (IA).

Schank [SCH 82] descobriu que a compreensão da linguagem humana está diretamente relacionada com a informação em memória e, através de estudos e definições, chegou à criação da Teoria da Memória Dinâmica, na qual afirma que toda a capacidade do ser humano em aprender e compreender está relacionada com a sua capacidade de recordar. Ao tentar compreender algum fato, o ser humano busca em sua memória algo que já foi compreendido no passado e que, de alguma forma, lhe é útil para compreender a situação atual.

Na década de setenta, surgiram os primeiros programas que simulavam o conhecimento de um especialista – chamados de sistemas especialistas [SCH 86]. Estes sistemas utilizavam raciocínio baseado em regras para fazer suas deduções. Isto significa que todo conhecimento do especialista era compilado em regras.

A aquisição de conhecimento em forma de regras gera uma grande dificuldade, porque não é desta maneira que um especialista resolve seus problemas. Em geral, quando um especialista precisa resolver um problema difícil, ele prefere utilizar a experiência obtida ao resolver um problema semelhante, em vez de repetir todo o processo de raciocínio.

Na década de oitenta, da Teoria da Memória Dinâmica surgiu o Raciocínio Baseado em Casos (*Case Based Reasoning* - CBR) que se caracteriza por cultivar a premissa de que todo ser humano utiliza-se de raciocínio analógico ou experimental para aprender a resolver problemas complexos [KOL 93]. CBR é visto como um modelo para solucionar problemas que, ao invés de começar do nada, utiliza a solução ou o raciocínio usado na solução de um problema (caso) anterior para auxiliar na resolução de um novo problema. O caso anterior é escolhido da memória (é recordado) por sua similaridade com o caso atual. Essa similaridade é medida tendo por base elementos que foram importantes na resolução do velho caso e elementos que são considerados importantes no novo caso. Esses elementos são os índices dos casos e servem para identificá-los de forma única.

A boa performance de um sistema que se utiliza do paradigma de CBR depende diretamente da qualidade e quantidade de conhecimento coletado. Desta forma, se não

existir conhecimento suficiente, a solução de um problema ficará limitada. A aquisição de conhecimento, juntamente com outros tipos de aprendizagem são características essenciais de um sistema inteligente.

Em muitas áreas profissionais, o processo é semelhante, uma solução só é alcançada pela ação do conhecimento de um especialista no problema, o qual coletou as informações necessárias para a tarefa de suas experiências.

Uma prática comum na resolução de um problema é descartar todos os diagnósticos e etapas percorridas na busca pela solução. Acarreta, desta forma, alguns defeitos, podendo-se destacar:

- não transmissão de conhecimento,
- toda vez que um problema igual ocorre, as etapas de sua solução precisam ser refeitas.

Tais fatos, levam a desníveis de conhecimento dentro de uma equipe, profissionais limitados a apenas algumas áreas de atuação e inexistência ou deficiência em processos de treinamento.

Alguns estudos geraram Sistemas de Registros de Problemas [JOH 92] como uma forma de eliminar estas tarefas de aprendizagem, onde os registros fazem parte de um repositório de informações sobre soluções conseguidas por especialistas. O uso constante desta alternativa, porém, leva a um acúmulo de registros, tornando-se inviável uma busca que tenha como resultado um problema armazenado semelhante ao enfrentado.

A utilização deste tipo de sistema seria muito satisfatório, se nele fossem aplicados mecanismos capazes de realizar procura em pouco tempo e com uma certa taxa de precisão, exigindo pouco conhecimento e descrição por parte do usuário, ou seja, fazer com que os Sistemas de Registro de Problemas consigam responder por um nível de raciocínio característico a um especialista ao resolver um determinado problema. Tal ambiente pode ser conseguido por adicionar-se, a estes sistemas, procedimentos de raciocínio, como os fornecidos pelos sistemas de Raciocínio Baseado em Casos.

Uma área de atuação que responde por avanços tecnológicos e onde os problemas são bastante acentuados, exigindo especialistas sempre bem preparados, é o gerenciamento de redes de computadores.

Com o crescimento constante de serviços e componentes de uma rede, particularmente com sua inclusão no mundo da Internet e pela criação do comércio eletrônico, os propósitos de acesso vêm chocando-se com a segurança da mesma. As redes têm se tornado excelentes alvos de ataques por pessoas não autorizadas, que se utilizam de diferentes procedimentos e ferramentas, provocando paralisação de segmentos da rede (ataques de *denial of service*) ou de sistemas servidores (ataques a *hosts*)[BER 97]. As formas de ataque ou tentativas apresentam um crescimento maior e muito mais rápido do que a capacidade de um técnico em se atualizar profissionalmente. O número de ataques e invasões tem duplicado a cada ano, segundo estatísticas de órgãos responsáveis por estes registros, como CERT [CER 2000] e NIC [NIC 2000].

A solução destes problemas é obtida, geralmente, junto à experiência dos gerentes da rede envolvida. O problema é que poucos são os gerentes que têm conhecimento para diagnosticar problemas de intrusão em redes e sistemas. Além disso, esta experiência, muitas vezes, não é replicada devido às pressões vividas nestes locais, tornando-os ambientes carentes por sistemas conciliadores de ocorrências e correções de segurança [GAR 96].

Dependendo dos estragos causados, o conserto poderá consumir, dos responsáveis pela rede, muitas horas ou dias de trabalho. Estes estragos poderiam ser evitados, se estes profissionais soubessem ou dessem atenção aos pontos de vulnerabilidade que sua rede apresenta. Desta forma, fica evidente a necessidade de implementar-se uma metodologia para transmissão e ajuda em resolução de problemas para o domínio de segurança em redes de computadores.

Este trabalho, portanto, proporciona uma estrutura para que especialistas consigam reproduzir suas experiências profissionais auxiliando pessoas que tenham ambientes de problemas semelhantes. Este resultado é obtido ao produzir-se um sistema que envolva problemas e soluções com dois paradigmas, que são: Sistemas Especialistas Baseados em Casos e Sistemas de Registros de Problemas, mantendo uma organização de memória e um *framework* para resolver problemas. Assim, surgiu um sistema chamado SABER, o qual raciocina sobre os problemas de um determinado domínio, gerando consultas de modo rápido e eficiente, e tendo como resultado diretrizes que ajudam na tomada de decisão.

Este protótipo se propõe a auxiliar na tomada de decisão, não a resolver os problemas para o usuário. Nada mais próprio que ele seja aplicado sobre o domínio de problemas de um gerenciamento de segurança em redes de computadores, para desta forma ser testado e analisado quanto ao seu funcionamento, vantagens e problemas.

Facilidade para a aquisição de conhecimento, potencial para trabalhar com domínios imprecisos e complexos e prototipação rápida, entre outros, foram os motivos para a escolha deste paradigma como ferramenta básica de estudo para o desenvolvimento deste trabalho.

Para tanto, aqui está descrito todo o processo de estudo, de projeto, de implementação e de testes, dividido em 8 capítulos, dispostos numa seqüência lógica, para facilitar a compreensão de todo o sistema.

1.1 Objetivos

Este trabalho tem por principal objetivo fornecer um sistema onde seja possível registrar as experiências de especialistas na resolução de problemas, independente do domínio em que estiver sendo utilizado, para que, de maneira rápida e simples, seja feita a descrição dos passos que levaram a solucionar um determinado problema e as soluções que foram resultantes destes passos. A aplicação também deve possibilitar ao seu usuário compartilhar estas lições do especialista, sem que, para isso sejam necessários conhecimentos detalhados sobre o que está sendo procurado, nem tão pouco que realize processos exaustivos na busca de seus objetivos.

A estrutura de raciocínio do protótipo será gerada de forma que a base de conhecimento seja independente do domínio de problema. O objetivo deste procedimento é que um mesmo mecanismo possa ser utilizado em vários domínios, sem que, para isso, sejam necessárias atualizações de programação, apenas construções de conhecimento específico.

1.2 Organização deste Trabalho

No capítulo a seguir, é feito um estudo sobre Sistemas Especialista com um enfoque no paradigma de Sistemas Baseados em Casos, que será utilizado no desenvolvimento deste trabalho. Nele é descrito, passo a passo, seu ciclo de vida, seu conceito e forma de armazenamento de dados.

Uma análise comparativa de ferramentas existentes que utilizam-se da metodologia de CBR é feita no capítulo 3.

O capítulo 4 apresenta alguns conceitos e idéias fundamentais sobre gerenciamento de redes, com enfoque no gerenciamento de segurança, políticas de segurança e desenvolvimento de sistemas de apoio à decisão para o domínio.

Problemas mais comuns em segurança são abordados no capítulo 5. Nele são descritas as características de problemas de segurança em redes TCP/IP com o objetivo de analisar um domínio de problema para validar o estudo feito.

O modelo de raciocínio que é proposto neste trabalho é apresentado, em detalhes, no capítulo 6. Todo o funcionamento da estrutura de controle proposta é detalhado, expondo como o ciclo de vida de um sistema CBR está sendo garantido nos processos desenvolvidos, como os casos são organizados na base e que informações são consideradas para alimentar o modelo.

No capítulo 7, é apresentada a forma como foi implementado o modelo proposto, além da descrição dos aspectos de sua implementação e validação.

O capítulo 8 descreve as considerações finais deste trabalho, onde são apresentadas as contribuições trazidas com a utilização da solução adotada, uma análise crítica sobre o trabalho desenvolvido, bem como alguns pontos que surgiram no decorrer de seu desenvolvimento e que ficaram como sugestões para trabalhos futuros.

Ainda fazem parte deste trabalho cinco anexos distribuídos desta forma: no anexo 1 comandos de mysql necessários a implementação do protótipo; no anexo 2 descrição das redes semânticas gerais e algumas específicas de problemas de segurança; no anexo 3 descrições de características que compõem casos registrados para a validação; no anexo 4 uma relação dos testes feitos para validação e; no anexo 5 a especificação do protótipo em SDL (*Specification and Description Language*), descrevendo o comportamento real do sistema, independentemente da linguagem utilizada para a sua implementação.

2 Sistemas Especialistas

Sistemas especialistas (SE) são, segundo Harmon [HAR85], programas possuidores de conhecimento intensivo que se utilizam de regras ou heurísticas para focalizar os aspectos-chave de problemas particulares e manipular descrições simbólicas para raciocinar sobre este conhecimento.

O processo de construção de um SE é geralmente chamado de engenharia do conhecimento (EC) e envolve uma forma especial de interação entre o construtor do SE, chamado engenheiro de conhecimento, e um ou mais especialistas em alguma área. O engenheiro do conhecimento extrai dos especialistas seus procedimentos, estratégias e regras práticas para solução de problemas e constrói este conhecimento em um SE.

O coração de um SE é um poderoso corpo de conhecimento sobre um domínio específico acumulado durante a construção do sistema. O conhecimento é explícito e organizado de forma a simplificar a tomada de decisões [WAT 97]. Possui também outra característica importante: seu poder de atuar como uma teoria de processamento de informação ou modelo de solução de problemas em um dado domínio, fornecendo as respostas desejadas para um dado problema e mostrando como eles poderiam se ajustar a novas situações.

A arquitetura deste tipo de sistema é composta pela *interface* do usuário, pelo banco de dados, pela base de conhecimento, pela memória de trabalho e pela engenharia de conhecimento, pode ser observado na Figura 2.1.

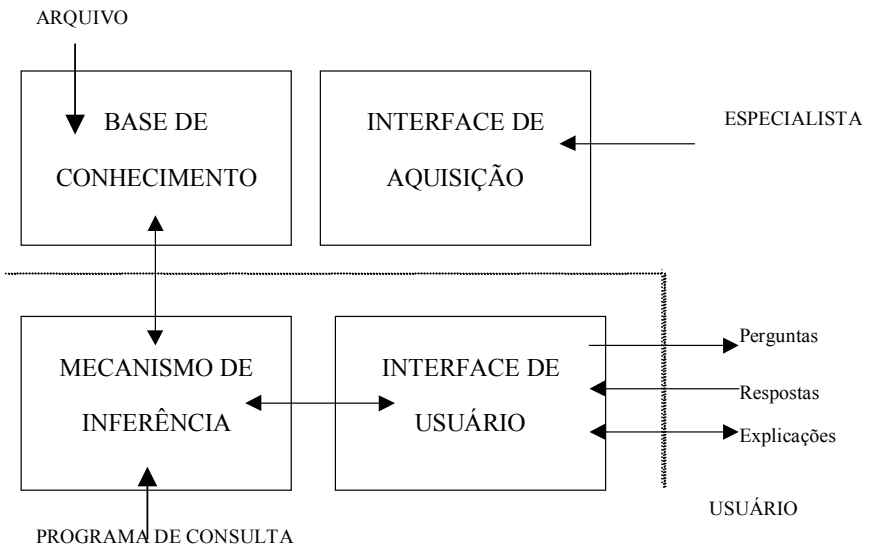


FIGURA 2.1 - Esquema de um Sistema Especialista [BAL 98].

Os SE são sistemas que não trabalham com possibilidades ou certezas provavelmente corretas, mas sim, afirmativas. Os SE ajudam a resolver problemas por conseguir ordenar os fatos relevantes, fazer distinção crítica sobre os tipos de problemas, procurar e eliminar redundância, reduzir ambigüidades, explorar conhecimento para complementar instruções e analisar problemas de diferentes níveis de perspectivas ou níveis de abstração. Eles têm como resultado final o fornecimento de conclusões peritas acerca de assuntos especializados.

Os sistemas especialistas são diferenciados pela estrutura da base de conhecimento, mecanismo de inferência e *interface* do usuário, sendo que sistemas especialistas ou simplesmente sistemas baseados em conhecimento (*Knowledge-Based Systems* - KBS ou SBC), estão entre as maiores histórias de sucesso da IA.

Winston [WIN 93] definiu conhecimento como sendo um conjunto de convenções semânticas e sintáticas que tornam possível a definição de certas coisas. Um sistema baseado em conhecimento deve conter inferências e conhecimento específico do domínio representado em formalismo próprio. SBC consegue montar explicações de seu comportamento na resolução de problemas por transformar afirmações e regras em uma linha de raciocínio, utilizando-se da engenharia de inferência, que tem sua escolha definida pelo tipo de problema que se deseja resolver.

O conhecimento é representado por objetos em bases de conhecimentos. Objetos são dados estruturados usados para representar conhecimento sobre coisas físicas (computadores, equipamentos) ou coisas conceituáveis (plantas, projetos e pedidos). Quando uma base de conhecimento é organizada dentro de objetos, esta representação é feita por uma árvore que mostra como diferentes objetos se relacionam.

Um sistema é capaz de aprender quando consegue adquirir novos conhecimentos e refinar os já existentes a partir de seus usuários, especialistas e de seu próprio ambiente, através da utilização das técnicas de aprendizagem por máquina.

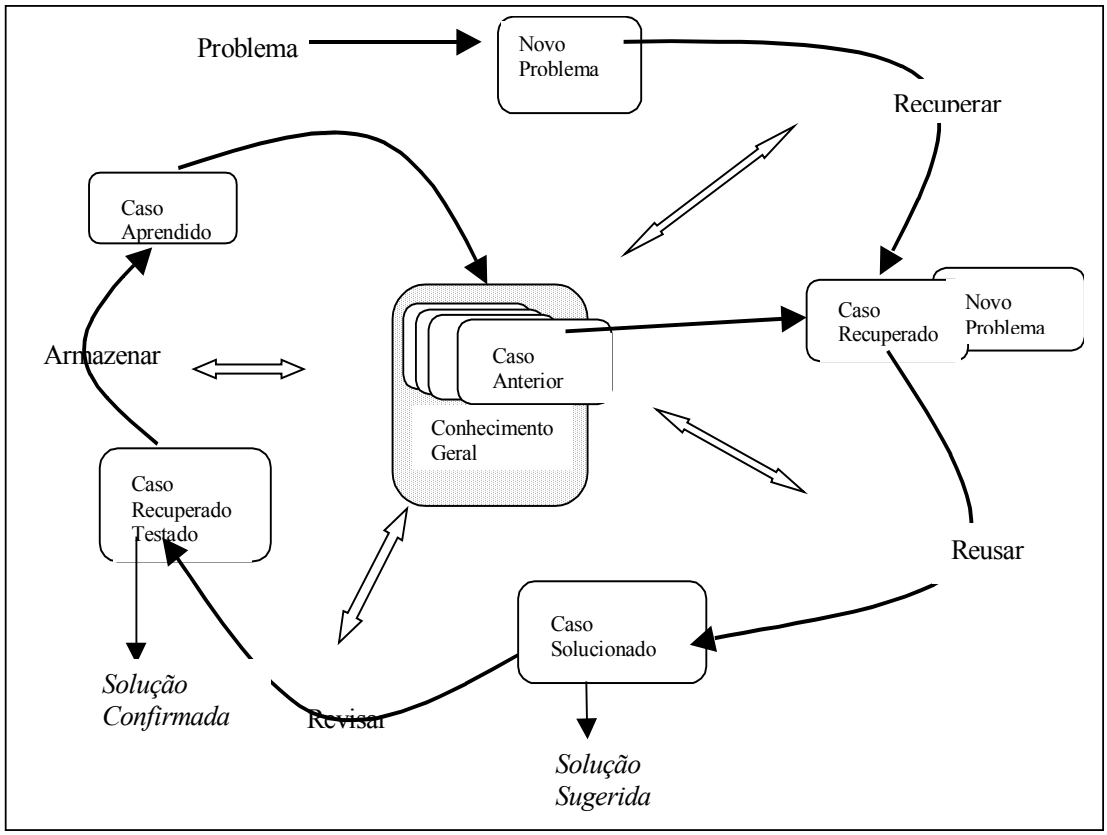
Vários autores reafirmam o sucesso de SE's, mas também apontam dificuldades no seu uso. Segundo Ian Watson [WAT 97], os processos que envolvem a recuperação de conhecimento é muito complexo, o que exige muito trabalho de implementação e muita memória das máquinas; mesmo assim, são incapazes de trabalhar com grandes volumes de dados.

Visando a contornar estas dificuldades, várias pesquisas têm se direcionado para soluções que empreguem a cognição humana, pois o raciocínio é um aspecto fundamental da mesma. Para Agnas Aamodt e Eric Plaza [AAM 94], tal fato se confirma quando se analisa a psicologia cognitiva, onde vários estudos demonstram que o processo humano de aprendizagem e resolução de problemas emprega a utilização de experiências anteriores. Assim, deu-se início aos estudos das técnicas de IA que se utilizam de Raciocínio Baseado em Casos (RBC ou CBR) que cultiva a afirmação (premissa) de que todo o ser humano utiliza-se de raciocínio analógico ou experimental para aprender e resolver problemas complexos. Janet Kolodner [KOL 93] afirma que se pode trabalhar sobre este ângulo, pois problemas similares têm soluções similares e os tipos de problemas se repetem.

2.1 *Case-Based Reasoning*

Segundo Janet Kolodner [KOL 93], CBR (*Case-Based Reasoning* ou Sistema Baseado em Conhecimento) é um modelo cognitivo de raciocínio e um método de construção de sistemas inteligentes. Caracteriza-se por utilizar antigas experiências ao resolver novos problemas. Integra a solução de problemas, entendimento, aprendizado e memória dentro de um *framework*. Procura sempre reproduzir a capacidade de fazer analogias entre situações e a experiência passada.

CBR se aplica a determinadas classes de problemas, entre elas: interpretação de dados, diagnóstico, reparo, monitoramento e instrução [WAT 94]. É freqüentemente utilizado como termo genérico para representar técnicas que se utilizam de raciocínio por analogia. O seu uso na construção de sistemas garante a existência de um raciocinador de casos que poderá, através de relatos já ocorridos, indicar uma similaridade ao problema a ele exposto. Caracteriza-se pela coleta e armazenamento de conhecimento especializado em uma base de casos.



- Ciclo de vida de um sistema CBR [AAM 94].

É um conceito da engenharia de conhecimento, colocado por Ian Watson [WAT 97], dentre outros autores como Janet Kolodner [KOL 93], onde CBR é descrito por um ciclo de nível conceitual que compreende quatro processos chamados quatro RE's que podem, também ser observados na figura 2.2. Os quatro RE's são:

- Recuperar o caso mais similar da base de casos;
- Reutilizar a solução do caso mais similar, se apropriado;
- Revisar ou adaptar o solução adequada, se necessário;
- Reter um novo caso e sua solução para uso no futuro, aprendizagem.

A execução destes processos exige que se tenha bem determinadas as feições e relações estruturais que irão rotular um problema, possibilitando a realização de uma análise do grau de disponibilidade dos casos em relação ao problema e à quantidade de informações que será repassada pelo especialista. Desta forma, é um sistema fortemente dependente das estruturas e contextos das bases de casos, onde a realização de comparações depende de como ela está analisada e armazenada.

Kolodner [KOL 93] apresenta um ciclo com processos mais detalhados, composto pela recuperação, proposta, justificativa ou adaptação, crítica, avaliação e armazenamento de casos. Esta autora segue a afirmação de que RBC pode ser freqüentemente classificado de acordo com suas tarefas e objetivos, o que proporciona a sua classificação em dois tipos:

sistemas interpretativos - utilizam-se de casos passados como pontos de referência para a classificação e caracterização de novos casos, concentram-se na interpretação de novos casos e não na sua resolução. Apresentam um enfoque sobre a etapa de justificação feita pela análise balanceada das similaridades e diferenças com os casos recuperados. Tal forma de aprender é importante, também, para a aplicação no ensino. Interpretar um problema e justificar esta interpretação é uma forma interessante de construir conhecimento sobre um domínio.

sistemas de resolução de problema - concentram-se na utilização de casos passados para sugerir soluções que podem ser aplicadas a novas situações. Seu enfoque está na geração de uma solução adaptada de um caso antigo para o novo problema. A adaptação substitui a etapa de justificação, permitindo uma revisão do caso recuperado para ajustar ao caso de entrada.

Aplicações já desenvolvidas em diversos domínios comprovam a eficiência de sistemas que se utilizam de CBR, como os desenvolvidos em domínios de gerenciamento de redes de computadores dos quais se pode citar CANASTA (*Crash Analysis Trouble-Shooting Assistant*) [LEW 95] ou DUMBO [MEL 98], desenvolvido pela Universidade Federal do Rio Grande do Sul.

Um sistema de CBR deve possuir essencialmente: um solucionador de problemas, um recuperador de casos (*case retriever*) e um analisador de casos (*case reasoner*). Dado um novo problema, o recuperador de casos identifica os casos mais

apropriados na base de casos e os apresenta aos analisadores de casos. A maneira mais simples de identificar os casos mais apropriados é usar a busca do vizinho mais próximo. Porém, o uso de índices evita as comparações exaustivas da operação anterior, sem comprometer a precisão.

O recuperador compara os índices (pré-fixados) com o novo problema e recupera apenas aqueles casos cujos índices solucionam o novo problema. A fim de melhorar o desempenho dessa operação, os índices são freqüentemente organizados hierarquicamente.

O recuperador de casos apresenta duas fases que são o acesso ao caso e a taxação de similaridade. O analisador de casos examina os casos recuperados e tenta resolver os novos problemas, adaptando algumas ações de reparo destes casos ou sintetizando uma solução, a partir destas ações.

A adaptação talvez seja a parte mais crucial de um sistema CBR. Ela compreende a criação de regras específicas, mas é um procedimento mais simples que construir um sistema baseado em regras, desde que os casos armazenados tenham relação com o domínio do problema e estejam uniformemente distribuídos pelo domínio da aplicação. Estes sistemas apresentam, em geral, dois tipos de adaptações [KOL 93]:

- estrutural – aplicam-se regras diretamente sobre a solução armazenada com o caso;

- derivacional – reutilizam-se algoritmos, métodos ou regras que geraram a solução original para proceder uma nova solução ao problema corrente, o que acarreta no armazenamento da seqüência que construiu a solução original juntamente com o caso, na memória de casos.

Em um sistema CBR o conhecimento não está presente somente na sua base de casos através de sua memória mas, também, nas suas etapas de desenvolvimento.

2.1.1 Histórico

A existência de CBR deve-se aos trabalhos do grupo de *Schank* [SCH 77], que iniciou seus estudos por um ponto crítico para seus objetivos, que era descobrir como os humanos pensavam e decidiam sobre as coisas, como eles conseguiam guardar os acontecimentos e desenvolver certos detalhes que contribuíam à tomada de decisão. Com a evolução destes estudos, tais atividades humanas passaram a ser conceituadas como *scripts*, ou seja, a maneira simples como o homem desenvolve a capacidade de entender, aprender e explicar uma sentença de conhecimento sobre o mundo que ele habita. *Scripts* habilitam pessoas a entender sentenças, como quando se ouve:

"A blusa amarela que Márcia comprou não lhe serviu".

Nesta sentença compreende-se que Márcia esteve em uma loja e que provou uma blusa, que ela precisou, ou quis, comprar uma blusa nova e mais: que ela não provou ou provou pouco a blusa antes de comprar. Nada disso está dito na sentença, mas a compreensão humana sobre o assunto já faz com que a experiência conclua o restante, não necessitando que se traduza em palavras tudo o que esteve acontecendo.

A esta maneira de pensar que ajuda a resolver problemas e a ensinar atribuiu-se o nome de MOP's (*Memory Organization Packets*) [SCH 77]. Os autores conceituaram MOP's como *frames* que compõem uma unidade básica da memória dinâmica.

Como *Scripts* ou MOPs contêm vários outros *scripts* menores, após vários estudos [SCH 82] in [SCH 86] renomeou estes como sendo micro-*scripts* ou E-MOPs, (*Memory Organization Packets Episode*), os quais organizam os casos específicos que partilham propriedades similares sob uma estrutura geral. Estes micro-*scripts* foram apresentados como sendo de três tipos:

- cognitivo – refere-se ao conhecimento sobre o uso de alguma coisa. Por exemplo, na frase "Maria fará um rocambole", se o termo rocambole faz sentido em um determinado contexto, então este é um *script* cognitivo, ou micro-*script* cognitivo.

- físico – refere-se ao conhecimento sobre operações, não sobre a explicação da ação. Se "Maria sabe como operar o rocambole", então o rocambole é um micro-*script* físico, sabe-se como desenvolvê-lo, não precisa de explicação.

- perceptivo – refere-se ao conhecimento por observação. Envolve o reconhecimento de coisas, como o reconhecimento de pessoas, ou de perigo, como "O rocambole de Maria está quente".

Outra fase da criação de CBR veio do estudo do raciocínio analógico [GET 83], em que se aplica filosofia e psicologia na formação de conceitos, solução de problemas e aprendizado experimental. Outros autores [AAM 94] observam que conceitos naturais (partes do mundo natural) são polimórficos, ou seja, que suas instâncias devem ser categorizadas de várias maneiras, o que não se tornaria possível com definições clássicas, por não se poder expressar um conjunto de necessidades e características. Assim surgiu a representação definida por um conjunto de instâncias ou casos, em que:

- instâncias representam casos, eventos ou objetos;

- abstrações representam versões generalizadas de instâncias ou de outras abstrações.

Na Universidade de Yale, no início dos anos oitenta, é que se produziu o modelo cognitivo de CBR. O primeiro sistema baseado neste modelo foi o chamado CYRUS, desenvolvido por Kolodner [KOL 93], da mesma universidade e grupo de Schank. Foi baseado nos modelos de memória dinâmica e MOP's, e representou um sistema questão-resposta, o qual relatava, na forma de casos, e implementava como os MOP's, as viagens e encontros de Cyrus Vance, ex-secretário de estado dos Estados Unidos. Mais tarde, este sistema serviu como base para diversos outros que surgiram.

Outros conjuntos de modelos foram desenvolvidos por Bruce Porter e seu grupo da Universidade do Texas, Austin [POR86]. Inicialmente, ele fez uso de classificação heurística e aprendizado de máquina para unificar, em um modelo, o conhecimento genérico do domínio e o conhecimento específico de casos, o que levou à construção do sistema PROTOS, o qual enfatiza uma integração geral do domínio de conhecimento com casos específicos, dentro da mesma estrutura de representação. A combinação de casos com domínios de conhecimento foram enfatizados no sistema GREBE [BRA91].

O potencial deste tipo de sistemas levou à comercialização de ferramentas produzidas a partir dos modelos citados acima, como é o caso da CBR Express, o sistema ECLIPSE, o ReMind ou o europeu ReCall, os quais representam os casos na forma de objetos, indexando-os e fazendo a recuperação em sua forma mais comum, com algoritmos de vizinhança (*nearest neighbour*) [KOL 94]. Todos foram produzidos para os mais diversos tipos de domínios e transformaram-se em *shell* para sistemas CBR. Atualmente, o desenvolvimento deste tipo de sistema está voltado para recursos básicos como os constantes nas primeiras ferramentas de CBR, mas o que é inegável é o desenvolvimento deste modelo em todas as partes do mundo e voltado para os mais diversos domínios, como direito, engenharia ou medicina.

Ficam como destaques de trabalhos recentes, a unificação de conhecimento genérico do domínio com o conhecimento específico de casos [ABE 96] *et al*, [REA 96] *et al, in press*; [HAS 95], o desenvolvimento de algoritmos de recuperação do caso mais útil [KOL 89], aplicação de raciocínio por analogia em CBR [CAM 90], aprendizado automático de novos casos e o gerenciamento de bases de casos [YAN 98].

Existem muitos relatos de sucesso por se utilizar aplicações baseadas em casos, sendo a maioria em aplicações do domínio de sistemas de *help desk*, como por exemplo o aplicado nas Empresas COMPAQ que utilizou um sistema RBC da *Inference Corporations* para um sistema de *help desk*, o qual sugeria reparos em impressoras [NGU 93].

2.1.2 Componentes de um Sistema Baseado em Casos

O principal trabalho ao se produzir um sistema baseado em casos é tentar extrair, de um domínio de problema, feições que possam ser relevantes para caracterizar um caso, por estarem representando uma situação diferente. Assim, este tipo de sistema é composto por:

2.1.2.1 Casos

Casos compreendem o problema e os resultados da solução que podem ser usados em uma nova situação. O conceito de casos é muito bem descrito por Kolodner [KOL 93] e David [DAV 91] *et al* [WAT 94], onde os autores colocam um caso como sendo um pedaço textualizado da representação de conhecimento em uma experiência, contendo as lições passadas (conteúdo do caso) e o contexto no qual a lição pode ser usada. Para Aamodt [AAM 94], um caso é uma situação de um problema.

Tipicamente, compreendem um caso, os seguintes ítens:

- o **problema**, que descreve o estado do mundo onde ocorreu o caso, ou seja, aponta as características da situação onde se insere o problema e as restrições associadas a ele;

- a **solução** derivada para um determinado problema;

- o **resultado da solução**, que descreve o estado do problema após a solução ter sido aplicada; utiliza-se na adaptação de novos casos.

Estes casos são mantidos em bases de casos (BC), que são alimentados pela capacidade de aprendizado do sistema. Casos são chamados para prover soluções e para interpretar situações.

A base de casos junto com as etapas de raciocínio dirão o quanto o sistema conhece sobre o domínio de problema pois, todas as suas fases de construção estarão diretamente ligadas ao conhecimento especialista, como as listadas a seguir:

- o estabelecimento do que deve ser um caso;
- a forma de organização dos casos na sua base;
- a identificação sobre o que faz um caso ser similar a outro;
- a adaptação, que só deve ser resolvida pelo conhecimento especialista.

2.1.2.2 Base de Casos (BC)

É usualmente construída durante um longo tempo exigindo muito trabalho. Nela ficam armazenados os casos que resolvem problemas semelhantes. Uma base pode ser o resultado da união de várias bases menores e diferentes casos, tendo diversos autores em diferentes espaços de tempo. Estas bases devem receber tratamentos e modificações constantes, ou seja, sua manutenção deve ser encarada como um procedimento regular. Tudo o que precisa estar presente ou ficar ausente em uma base de casos é realizado por um serviço de manutenção. Este processo, segundo Trung Nguyen [NGU 93], leva em média de cinqüenta a sessenta por cento do tempo do desenvolvimento de um ciclo de vida de um *software*.

A base de casos contém casos que são descrições de problemas anteriores resolvidos e não resolvidos. Cada caso pode descrever um episódio particular ou uma generalização a partir de um conjunto de episódios relatados, devendo ser responsável por diferentes tarefas, incluindo-se: guardar diversas proporções e tipos de problemas com suas referidas soluções; manter uma relação de problemas com suas possíveis soluções; possuir informações suficientes para que o raciocinador consiga, de seus dados, retirar várias conclusões, como no exemplo:

< Meri, quando come ração de nome *whiskys*, ronrona feliz>

O raciocinador deve, ao vasculhar a base de casos, ter informações suficientes para analisar que Meri é um animal da espécie gato.

Geralmente, a BC é grande e deve considerar os problemas como diferentes uns dos outros, a cada pequena variação, bem como fornecer a solução adequada. Não deve possuir registros de casos inconsistentes e nem redundantes, o que vem a ser uma característica de sistemas baseados em modelo, os quais guardam cada situação representada no domínio de ocorrência. Toda esta consistência na base de casos fica sob a responsabilidade do raciocinador do sistema.

A representação de casos dentro da base ainda não se tornou um padrão, é um ponto muito discutido dentro da comunidade de IA.

2.1.3 Processos de Desenvolvimento de um Sistema CBR

O processo de desenvolvimento de um sistema CBR é uma tarefa que compreende uma seqüência de etapas, que são problemas a serem modelados os quais, quando resolvidos, compreendem um sistema CBR. Tais etapas são:

2.1.3.1 Representação de Casos

A representação deve suceder a uma aquisição de casos. É a tradução deste caso para um sistema, gerando uma imagem que pode ser trabalhada, relacionada com outras imagens, identificada e entendida.

Tendo em vista que um caso representa um conhecimento específico sobre um contexto, o armazenamento de casos iguais não contribui em nada para o raciocínio do sistema. Assim, é preciso fazer uma seleção, para que na BC só estejam registrados casos com feições diferentes, capazes de caracterizar um novo caso, acrescentando conhecimento a ela.

Para que um caso seja coerentemente armazenado, ele deve conter a funcionalidade da informação e a facilidade de prover aquisição de informação. Desta forma, a representação de um caso deve conter características, descrições e dimensões disponíveis para serem trabalhadas.

Uma descrição compreende um par atributo-valor (ex: idade=27), usado na descrição de um caso, podendo representar aspectos da descrição do problema, da situação ou da solução. Pode se referir às características superficiais, abstratas, estruturais ou de relação das características dos casos.

A dimensão representa a parte atributo de uma descrição, são os pontos analisados quando se pretende comparar casos. As características, por sua vez, compreendem todas as outras informações descritivas relevantes sobre uma situação a fim de obter-se seus objetivos.

A representação é um dos pontos polêmicos dentro da comunidade de IA. Como representar os casos? Que padrão utilizar para isso? Até o consenso não chegar, deve-se sempre ter em mente que toda a representação deve ser feita em função dos objetivos a atingir. A representação pode ser feita através de:

- frames - estruturas de dados que representam uma entidade através de suas características e capacidades. As características são representadas por pares atributo-valor e as capacidades são representadas por métodos. Podem ser organizados em estruturas hierárquicas de especialização e todo-parte;

- rede semântica - objetos (nós) interligados através de conexões (arcos) que descrevem as relações entre objetos. Os nós representam objetos, conceitos e eventos. Os arcos são direcionados e representam relações e atributos;

- regras - estruturas compostas de duas partes: premissas, que são conjuntos de expressões que avaliam a presença ou não de determinados fatos e conclusões, que são um conjunto de expressões que modificam fatos existentes ou inserem novos fatos;

- formulários - é um conjunto de campos valorados semelhantes a registros em um banco de dados. Seus campos são chamados de descritores e compostos por pares atributo-valor que caracterizam a informação contida em um caso.

Alguns projetos de âmbito mundial, como o INRECA [MEN 99], tendem a padronizar, não só a representação de casos, como também tudo o que envolve uma construção deste tipo de sistema, resultando na produção de uma espécie de *framework*.

O importante na forma de representação não é o modo como ela será feita, mas sim que ela seja apropriada para o raciocínio que deve ser realizado sobre estes dados. E este ponto depende muitíssimo da aquisição de casos, como obtê-los e que informações são relevantes a ponto de caracterizar um novo caso, visando sempre a ausência de redundância.

2.1.3.2 Aquisição de Casos

O processo de aquisição de casos talvez seja tão minucioso, trabalhoso e difícil quanto as tarefas de construir modelos.

Tanto a aquisição quanto a representação de casos devem iniciar com uma análise para determinar o grau de disponibilidade dos casos e quanta informação adicional deve ser elicitada diretamente do especialista.

A aquisição envolve ações de reunir informação de um ou mais peritos humanos e/ou de fontes documentais, ordenando esta informação e, posteriormente, traduzindo-a para uma linguagem de máquina. Compreende duas fases iniciais: a primeira é a coleta de informações, que permite a construção da base do sistema (o protótipo) e a segunda é um processo de refinamento iterativo, envolvendo ferramentas de suporte, especialista da área e engenheiro de conhecimento (EC), sendo a entrevista o método mais comum de aquisição de conhecimento.

Os métodos de aquisição do conhecimento podem ser divididos em dois grandes grupos: os informatizados e os não informatizados. Os informatizados, além de entrevistas, *brainwriting* (reunião em torno de uma mesa, com um grupo de pessoas que contribuirão no assunto), análise de protocolos (observação do comportamento do especialista em um caso real) e observação direta, têm abordagens específicas. Já existem algumas ferramentas informatizadas para a etapa de aquisição, como descrita por Renato Rabuske [RAB 95].

Independente do método, a parte central da aquisição de conhecimento é o engenheiro de conhecimento (EC). Ele é o gerador de informações para o raciocínio, sendo importante que ele seja diplomático, sensível, agradável e habilidoso ao tentar obter informações dos usuários ou especialistas.

Tendo os casos na base do sistema e seguindo as etapas do ciclo de vida de um sistema que se utiliza do paradigma de CBR, torna-se necessária a construção de um

ambiente que seja capaz de representar, segmentar e indexar os casos apropriadamente, para permitir que sejam recuperados em um determinado espaço de tempo, bem como apresentados de forma acessível ao usuário.

2.1.3.3 Indexação

Uma base de casos, segundo Janet Kolodner [KOL 93], possui duas partes que são: uma biblioteca de casos e um conjunto de procedimentos que os torne acessíveis quando precisarem ser recuperados.

Os casos representados em uma base serão responsáveis por um eficiente raciocínio sobre os problemas quando se encontram devidamente organizados e armazenados. Assim, a tarefa de indexação é a responsável pela etapa de recuperação dos casos de uma base.

A indexação dos casos é um instrumento para orientar a similaridade, ajuda a organizar ou caracterizar os casos em função de uma ou mais características, para isto torna-se necessária a determinação do que é relevante em um caso.

Um índice é uma estrutura de dado computacional que garante uma procura por informações de maneira mais rápida e eficiente [WAT 97]. A existência de um índice garante que o sistema, ao procurar por um caso em sua base, não deverá passar registro à registro. Para sistemas CBR, indexar representa rotular os casos dentro de suas características, faz parte do conjunto de procedimentos definidos por Kolodner, a qual [KOL 93], além de Watson [WAT 97], define que os índices devem ser selecionados observando-se algumas características, como:

- prever que um caso possa contribuir em diversos tipos de problemas;
- representar o objetivo dos casos, podendo ajudar na detecção de similaridade;
- possuir características que permitam seu reconhecimento e utilidade no futuro, caracterizando-se como concretos;
- possibilitar que um caso seja flexível a fim de ser útil em diversas situações, ou seja, que o caso possa ser abstrato.

Indexar casos é como fazer um índice para um livro, são escolhas de partes que correspondem a uma descrição; portanto, um índice depende de seu conteúdo, ou seja, seu significado e ao que ele leva (recuperação). Precisam possuir conteúdo para permitir procedimentos de diagnóstico, recuperação e planejamento. Tal conteúdo deve incluir registro de situações, soluções e resultados, isto é, devem representar o contexto do caso em questão.

Estudos definem que os índices a serem utilizados em sistemas CBR podem ser criados de modo automático ou manual [KOL 93], [WAT 97]. O modo manual exige que o programador analise todos os casos, classifique-os e determine o que servirá como índice, tendo em mente a que o caso se propõe e em que circunstâncias ele será utilizado, obrigando um grande conhecimento sobre o domínio de problema.

O modo automático inclui métodos computacionais simples que requer pouco conhecimento de programação, mas é responsável por grandes números de índices coletados para um mesmo caso. Kolodner [KOL 93] aponta dois tipos de métodos heurísticos para seleção de índices automáticos, que são:

- *cheklist* provido pelo construtor do sistema, onde é perguntado ao sistema, que tipo de características poderiam ser utilizadas como índices. O sistema gera uma lista que expõe as melhores dimensões para definir o caso visando a que estas sejam soluções e resultados preditivos;

- índices gerados através de processos que extraiam as diferenças entre os casos (um novo contra um da base).

Watson [WAT 97] cita, ainda, dois métodos automáticos:

- generalização baseada em explicação e similaridade, que produz um conjunto apropriado de índices para abstração entre casos;

- aprendizagem indutiva, a qual identifica características preditivas que serão utilizadas como índices através de outros processos.

Os índices podem ser selecionados manualmente - quando são analisados caso a caso para determinar que características variam sobre as conclusões - ou automaticamente - onde se quantificam as diferenças entre os casos e os relacionamentos entre feições do problema e as soluções adotadas. Na prática, os que são selecionados manualmente tendem a ter um melhor desempenho.

As técnicas utilizadas para selecionar índices são, segundo Ian Watson [WAT 94] e Mara Abel [ABE 96]:

- baseada em explicação - identificam-se os elementos do problema que são utilizados como índices para construir a solução;

- baseada em conhecimento do domínio - são obtidos os índices das correlações entre os elementos e conclusões;

- análise matemática - elementos e suas dimensões são analisados numericamente;

- baseada nas diferenças entre casos - casos similares são indexados pelas suas diferenças;

- métodos de generalização - casos abstratos são definidos a partir de elementos compartilhados;

- método de aprendizado indutivo - identifica-se os elementos que determinam as conclusões para serem utilizadas como índices.

De posse dos índices e da forma de armazenamento (memória) dos casos na base, fica-se dependente de procurar a melhor forma de representar estes casos na base.

2.1.3.4 Recuperação

Dada uma descrição de um problema ou situação, a recuperação consiste em fazer uma busca na memória de casos, selecionando aqueles que poderão ser aproveitados. Esta seleção é feita por algoritmos que determinam similaridade com o problema a ser resolvido. É uma tarefa bastante complexa, pois envolve julgamento, heurísticas e avaliações sobre feições dos casos.

Recuperar casos é algo mais do que procurar casos em uma base previamente estruturada e aplicar, sobre ela, algoritmos de recuperação. Recuperação compreende que o raciocinador deve focar uma dimensão apropriada dos casos, a qual deve guiá-lo até seus objetivos. A recuperação de casos é o processo responsável pela resposta correta, no tempo certo, para o usuário de um sistema de CBR. Inicia com a descrição de um problema e termina quando um caso similar é encontrado.

Outro ponto importante no processo de recuperação é a aplicação de algoritmos de recuperação que compreendem um processo responsável pela procura na biblioteca de casos para encontrar casos com o potencial de ser o melhor caso (*best matching*) para a nova situação.

A escolha por um algoritmo depende da estrutura organizacional dos casos. Segundo Watson [WAT 94], os principais métodos de recuperação de casos são:

- algoritmo de vizinhança - faz uma comparação entre um novo caso e aqueles armazenados no banco, utilizando uma soma ponderada das suas características, sendo que o maior problema é determinar o peso das características. Um algoritmo típico para este problema e que é utilizado pelo *ReMind* [MEN 99] é o demonstrado abaixo:

$$\frac{\sum_{i=1}^{n} w_i \times sim\left(f_i^I, f_i^R\right)}{\sum_{i=1}^{n} w_i}$$

onde: w – é o peso de uma feição i qualquer que descreve o caso; sim – é a função de similaridade e f_i^I e f_i^R são os valores da feição i, para o novo caso e o caso recuperado, respectivamente. É um bom método para banco de casos relativamente pequenos, devido aos problemas de convergência de soluções;

- indução guiada por conhecimento - emprega conhecimento da aplicação para induzir o processo a fim de que, manualmente, se identifiquem características dos casos conhecidos ou considerados por afetarem a decisão. Geralmente, utilizado em adição a outras técnicas;

- algoritmo de indução - determina quais características fazem o melhor trabalho de descrever o caso e gera uma estrutura de decisão em forma de árvore para organizar os casos na memória. Suficiente para casos comparados com apenas uma característica que determina a solução;

- recuperação de padrões - similar às procuras feitas em consultas SQL, recupera todos os casos que se encaixam com certas características já definidas. É muito eficiente em grande volume de casos e pobre em selecionar o melhor caso.

Pode-se organizar o processo de recuperação através de alguns passos considerados como sub-tarefas, que são: identificação, similaridade, métrica de similaridade, recuperação de casos e seleção de *best match*.

Identificar o problema compreende relatar as entradas, segundo um método intensivo de conhecimento, ou seja, proporcionar um entendimento do problema no contexto do mesmo, podendo haver participação do usuário.

Similaridade é a essência do raciocínio baseado em casos, pois o fundamento deste paradigma é solucionar um problema atual reutilizando uma solução de uma experiência passada similar que está contida na sua base de casos. Contudo, identificar e avaliar a similaridade entre duas experiências pode não ser tão simples. Ela depende do domínio de conhecimento da aplicação. A similaridade pode ser :

- sintática - mais superficial, onde os atributos são comparados em termos de sua semelhança sintática. Whitaker [WHI 89] propõe algumas categorias que são: sinônimos, categorias ordinais, análise de perfil, clusterização e qualificadores.

- semântica - propõe uma avaliação mais profunda por abranger o significado dos casos e compará-los. A busca da avaliação automática da similaridade semântica pode ser considerada como um dos gargalos da IA.

Quando a recuperação busca a similaridade diretamente pela comparação de índices, uma métrica de similaridade é usada para obter a resposta desejada. Estabelecer esta métrica equivale a uma medida da similaridade entre dois casos, muitas vezes utilizada para estabelecer o grau de similaridade dos casos e ordená-los. Conforme a importância de um caso para o contexto do domínio, este receberá um valor de métrica, ficando valorado em importância para o contexto.

O resultado da busca, em uma recuperação, é chegar a um conjunto de casos, cujas características combinem com a do problema de entrada, de acordo com as definições de similaridade estabelecidas. Finalmente, um caso, dentre os encontrados, deve ser escolhido como sugestão do sistema para solução do problema de entrada. O caso escolhido é normalmente tratado na literatura por *best match*. Pode ser encontrado por heurística, regras e inclusive com a participação de usuários.

Um ponto importante a ser observado no processo de recuperação, segundo Aamodt [AAM 94], é que os casos devam ser recuperados apenas por suas características de entrada, ou por características deduzidas destas.

Como pode-se concluir, a recuperação é muito dependente da indexação e da organização da BC, pois estas envolvem uma constante avaliação e troca de dimensões que levam a atingir o objetivo do caso consultado. Assim, segundo Kolodner [KOL 93], casos podem ser considerados similares quando apresentam dimensões similares que auxiliam o sistema a resolver os objetivos propostos.

2.1.3.5 Adaptação

Sistemas que se utilizam do paradigma de CBR são úteis e funcionais por serem capazes de procurar, dentro de antigas experiências, respostas de soluções para os problemas atuais, isto é, perfazendo a fase de adaptação de casos, os anteriormente resolvidos servem como índices de lembrança ou sugestões.

Apresentado um problema ao sistema, toda a procura por uma solução vai trazer ao usuário a que mais se adapta ao problema atual, mas normalmente não é exatamente aquela que ele necessita.

A adaptação procura por proeminentes diferenças entre o caso recuperado e o caso atual e aplica regras específicas para compensar estas diferenças, desde que os casos armazenados tenham alguma relação com o domínio do problema e estejam uniformemente distribuídos pelo domínio da aplicação. Em geral, existem dois tipos de adaptações em CBR [WAT 97]:

- adaptação estrutural - as regras de adaptação são aplicadas diretamente na solução armazenada com o caso;

- adaptação derivacional - reutiliza-se de algoritmos, métodos ou regras que geraram a solução original para proceder uma nova solução ao problema corrente. Assim, nesse método, a seqüência que construiu a solução original deve ser armazenada juntamente com o caso na memória de casos.

Derivar soluções exige uma perfeita compreensão dos casos armazenados e da forma como as soluções associadas foram geradas. Um conjunto de adaptação, se bem selecionado, atenderá a solução completa de um problema.

Algumas técnicas de adaptação têm sido desenvolvidas para adaptação em sistemas de CBR como as apresentadas por Watson [WAT94]:

- adaptação nula - o sistema fornece a solução ao melhor caso recuperado, sem qualquer modificação, útil para soluções simples mas que exigem mecanismos complexos de raciocínio para atingi-la. Utilizada no sistema CHEF [KOL 93];

- ajuste por parâmetros - o algoritmo compara parâmetros pré-determinados, entre o caso recuperado e o novo, para modificar a solução armazenada na direção correta. O sistema chamado JUDGE [KOL 93] fez sua adaptação por este método;

- abstração e reespecialização - utiliza regras gerais para fazer modificações simples sobre soluções anteriores e mecanismos de abstração complexos quando é necessário gerar soluções totalmente novas e criativas;

- adaptação baseada em crítica - utiliza informações do domínio para buscar combinação de feições que podem causar problemas na solução e corrige essas distorções;

- reinstanciação - cria uma outra instância dos casos recuperados com valores adequados ao problema do usuário. Complementou a adaptação do sistema CHEF;

- substituição derivacional - repete o método ou parte do que gerou uma solução armazenada similar, substituindo atributos distintos. Praticado no sistema BOGART [WAT 97];

- reparo guiado por modelo - utiliza um modelo causal para adaptar as soluções armazenadas ao problema do usuário. Indicada para sistemas sobre equipamentos, usado no sistema CASEY [KOL 93];

- substituição baseada em casos - usa casos para sugerir a adaptação da solução aos novos problemas.

A maioria dos produtos comerciais que usam CBR não implementa nenhum tipo de adaptação, como pode ser verificado em [MEN 99] quando do estudo de algumas ferramentas.

2.1.3.6 Aprender em Sistemas CBR

A parte central de um sistema raciocinador é o aprendizado. Aprendizado é um comportamento emergente que cresce de funções normais de um raciocinador baseado em casos. Um raciocinador não aprende por apenas refazer as situações, em tudo é preciso lógica.

Desde o nascimento, ou mesmo antes, as pessoas vão acumulando conhecimentos que lhes permitirão agir de modo a mostrar que são seres inteligentes. Isto comumente é denominado de aprendizado.

Em CBR, aprender significa conseguir colocar novos casos nas bases de casos. Através da identificação de similaridade, um sistema consegue discernir se um caso é diferente de outro e, assim, aprender que eles podem caracterizar novas situações que devem ser acrescentadas a sua BC. Este processo é tido como um raciocínio sobre os casos e resulta no aprendizado pelo sistema. A combinação do raciocínio com o aprendizado determina o comportamento do sistema baseado em casos.

A ênfase nos casos como a forma preferida de conhecimento e a habilidade de casos em guardar experiências adquiridas, associadas ao conhecimento, conduzem a diversas conclusões sobre a capacidade e aplicabilidade do raciocínio baseado em casos.

2.1.3.7 Estrutura Organizacional do Sistema

Trabalhar com sistema CBR requer a produção de um sistema que responda ao usuário em tempo e conteúdo aceitáveis.

Para atingir estes objetivos, torna-se necessário que, além do cumprimento das etapas do ciclo de vida deste tipo de sistema, tenha-se um contexto todo elaborado para funcionar adequadamente.

Estruturar um sistema compreende definir uma forma de organizar os casos para facilitar sua recuperação, tendo em vista que a procura por similaridade dentro deste

tipo de paradigma é encontrada mediante um casamento parcial de campos. Assim combinar muitos com muitos acabaria por comprometer o desempenho do sistema; o importante é garantir que nenhum casamento seja perdido e as comparações sejam limitadas. Tal garantia é dada pela correta organização dos casos e pela aplicação de algoritmos utilizados na recuperação, bem como pela alimentação dos casos na estrutura.

A forma de armazenamento influi diretamente no planejamento eficiente de um sistema CBR, pois cabe a ela responder por uma estrutura que suporte eficientes métodos de procura e recuperação na BC. Esta estrutura depende de um equilíbrio entre o armazenamento de métodos que preservam os índices e características dos casos, com os que proporcionam acesso simples aos casos relevantes para uma recuperação. Na literatura de IA, estes métodos são conhecidos como modelos de memória dinâmica de Schank e Kolodner [KOL 93] e categoria-exemplar de Porter e Bareiss [AAM 94].

O armazenamento dos casos corresponde à montagem da memória dos casos, onde deve estar representada a forma de raciocínio sobre os casos, possibilitando uma lembrança do que cada feição de um caso pode representar ao novo. É formada pela base de casos e os procedimentos para acessá-la .

Lewis [LEW 95] aponta quatro princípios, utilizados ultimamente em CBR, para organizar a estrutura de uma biblioteca de casos:

- memória seqüencial - os elementos da estrutura são organizados como uma lista;

- memória hierárquica - elementos organizados como classes e sub-classes, onde cada elemento existe, pelo menos, uma classe;

- memória *meshed* - as estruturas são organizadas como tipos e subtipos, onde um elemento pode ser de vários tipos;

- caso mestre - é uma estrutura em que uma experiência pode ser resumida dentro de um caso individual, ou seja, possibilita que se faça contribuições nos casos, quanto a solução ou descrição, quando estes perfizerem o mesmo contexto, não necessitando que se gere mais um caso.

Este capítulo proporcionou um estudo detalhado sobre os componentes de um sistema que se utiliza do paradigma de CBR, além de apresentar o seu ciclo de vida.

3 Ferramentas Existentes

Inicialmente, cada sistema especialista era criado a partir do nada; em geral, em LISP. Mas, depois de vários sistemas terem sido desenvolvidos, ficou claro; que esses sistemas tinham muito em comum, só se diferenciavam no que diz respeito ao conhecimento armazenado. Com base nesta semelhança entre os sistemas, criaram-se as *Shells*.

Uma *Shell* consiste de todas as partes constituintes de um sistema especialista, com exceção da base de conhecimento. Assim, cabe ao engenheiro de conhecimento colocar no sistema apenas o conhecimento adquirido junto ao especialista, já que a *Shell* possui o mecanismo de inferência para manipulá-las. O objetivo principal da *Shell* é permitir ao próprio especialista entrar com suas regras [RIC 93], [WAT 86].

Diversas companhias oferecem *Shells* para construir sistemas CBR e sistemas baseados em regras. Estas *Shells* possibilitam que se desenvolva uma aplicação rapidamente devido à utilização de flexibilidade de representação, padronização de raciocínio e métodos de aprendizagem.

Das ferramentas para CBR disponíveis no mercado, foram, neste trabalho, selecionadas apenas três, por serem, estas, as que mais se adaptam ao domínio de problema aqui exposto. A seleção foi baseada em descrições feitas na literatura, pelas propostas dos próprios fornecedores, pela verificação de uso comprovado por *softwares* já desenvolvidos no mercado e pela aplicação nos problemas do domínio específico de estudo. Portanto, a escolha se deu pelas ferramentas *ReMind*, *CaseAdvisor* e *CBR3*.

Todas estas ferramentas são indicadas como sendo ideais para sistemas de HD (*Help Desk*) [RAB 95]. Neste trabalho, elas são analisadas por um padrão sugerido por Althoff [ALT 95] para análise de sistemas de CBR, tendo em vista aspectos diferentes, em que a principal idéia é combinar diferentes tipos de critérios de avaliação, como:

Critério orientado à aplicação e domínio – critérios que são usados para descrever as exigências que surgem de um certo domínio e da tarefa de aplicação, e que devem ser satisfeitas se uma aplicação deseja resolver o problema;

Critério orientado tecnicamente e ergonomicamente – critérios usados para descrever as capacidades de um sistema CBR e que permitem comparações entre diferentes sistemas;

Critério de engenharia de conhecimento – critérios para uso de metodologias que guiam o usuário ao seu objetivo de desenvolvimento de sistemas CBR, baseado no relacionamento de critérios domínio/tarefa e critérios técnico/ergonômico.

A intenção é avaliar as ferramentas pelo critério de engenharia de conhecimento que, praticamente, engloba os outros dois critérios e também faz com que sejam medidos critérios qualitativos, dependentes do domínio, e quantitativos, que são independentes do domínio.

Os critérios qualitativos são os que dão informações de representação de casos e de medidas de similaridade, por isso são mais precisos e mais úteis para um sistema

CBR, enquanto que os quantitativos são critérios usados para avaliar performance, consistência e tratamento de dados incompletos.

Seguindo esta linha de avaliação, as ferramentas estudadas a seguir são analisadas quanto ao objetivo de estudo, à representação de casos, à organização de memória de casos, às subtarefas de raciocínio de recuperação de memória e à adaptação de casos.

Antes da análise de *shells* para sistemas CBR, que sirvam para o domínio de redes, é imprescindível que se exponha uma tentativa de padronização e auxílio aos desenvolvedores de sistemas CBR, que está sendo montado na Europa pelo grupo *ESPRIT*, o qual teve início no projeto *INRECA* e hoje avançou para o *INRECA*-II, tendo suas principais características registradas a seguir.

3.1 *INRECA*

INduction and REasoning from CAses (*INRECA*) é um padrão criado para suportar decisões e para resolver problemas de diagnósticos, formado pelo grupo *ESPRIT* (contrato P6322). Oferece ferramentas para desenvolvimento, validação e manutenção de sistemas de suporte à decisão. Trabalha sob duas tecnologias básicas: raciocínio baseado em casos, usada na sua ferramenta de nome S^3*-Case*, produto baseado no *PATDEX* [AL T91]; e indução, usada na ferramenta *Kate-Induction* [MAN 90].

O *INRECA* se propõe a utilizar as duas tecnologias integradas ou separadas, retirando de cada uma o que for mais conveniente para a solução de um determinado domínio de problema. Ele disponibiliza a estrutura de dados como uma árvore chamada de *k-d tree* [WES 94], a qual pode ter um comportamento de árvore de decisão, bem como de um árvore indexada para suportar CBR. Dentro do *k-d tree*, atributos numéricos e simbólicos são representados, assim como valores desconhecidos e não ordenados. A recuperação é feita de forma seqüencial ou relacional, sendo a relacional, baseada em valores indexados .

Com a crescente necessidade de se criar uma metodologia que suporte desenvolvimento e manutenção de aplicações CBR, o projeto *INRECA* começou a construir um novo padrão chamado de *INRECA*-II (*ESPRIT* contrato 22196). Este, tenta ser mais moderno e acompanhar as dificuldades dos usuários e empresas de sistemas CBR, apresentando como no projeto anterior, definições orientadas objeto para os casos, expressas na sua ferramenta *CASUEL*.

O *INRECA*-II é um padrão metodológico baseado em duas áreas recentes da engenharia de *software* que são: a produção de conhecimento [BAS 94] e modelos de processos de *software* [ROM 95].

Este padrão trabalha de forma semelhante ao padrão CBR capturando experiências de um desenvolvimento de CBR e armazenando-os em um pacote de experiências (PE) [MEN 99], um termo do padrão fábrica de experiências, o qual se assemelha a um banco de casos. As entidades sendo armazenadas no PE são modelos de processos de *software,* ou fragmentação destes em processos, produtos ou métodos. O

PE é organizado em três níveis de abstração: um nível genérico comum, no alto; um nível de guia (*cookbook*), intermediário e um nível específico do projeto na base.

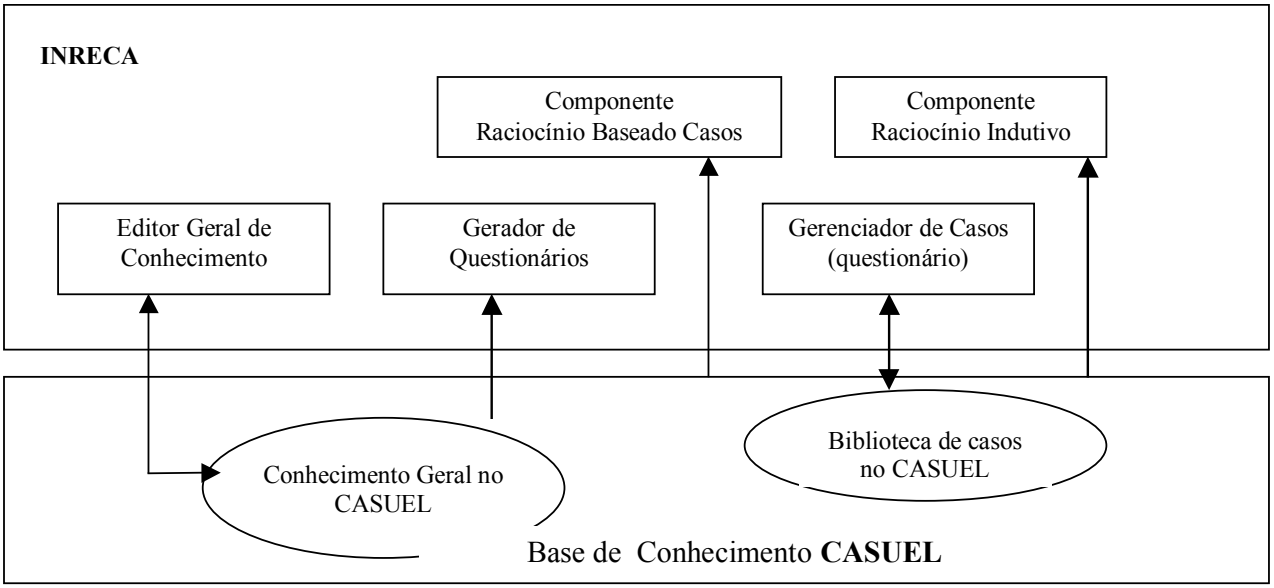


FIGURA 3.1 - A arquitetura do *INRECA*.

3.2 CASUEL

O *CASUEL* é uma das muitas ferramentas que fazem parte do projeto *INRECA*. É uma linguagem comum de representação de casos, faz *interface* entre todos os sistemas que compõem o projeto *INRECA*, e o integra com o mundo externo. Permite, ainda, um padrão de troca de informações entre sistemas de diagnóstico e de classificação que usam casos. Não foi construído para desenhar ou planejar tarefas. É considerada uma linguagem orientada a objeto para armazenamento e trocas de modelos descritivos e bibliotecas de casos em arquivos *ASCII*. É destinado a modelar naturalmente casos reais complexos.

defclass hub a_kind_of class;

slots sysdescr; infindex (elementos de uma MIB padrão e/ou de MIB proprietária);

rules verifica_status

num_portas := ?x (o valor da regra retorna na variável ?x)

adaptation_rules adapta_sysdescr

FIGURA 3.2 – Exemplo de objeto para o INRECA.

Os objetos são descritos por: uma classe hierárquica que utiliza herança; um *slot* usado para descrever o objeto; tipos de restrições relatadas no valor do *slot* e diferentes tipos de relacionamentos entre objetos. Suporta uma regra de formalismo para completar a adaptação de casos por regras, e um mecanismo para definição de pesos de similaridade (*sim*).

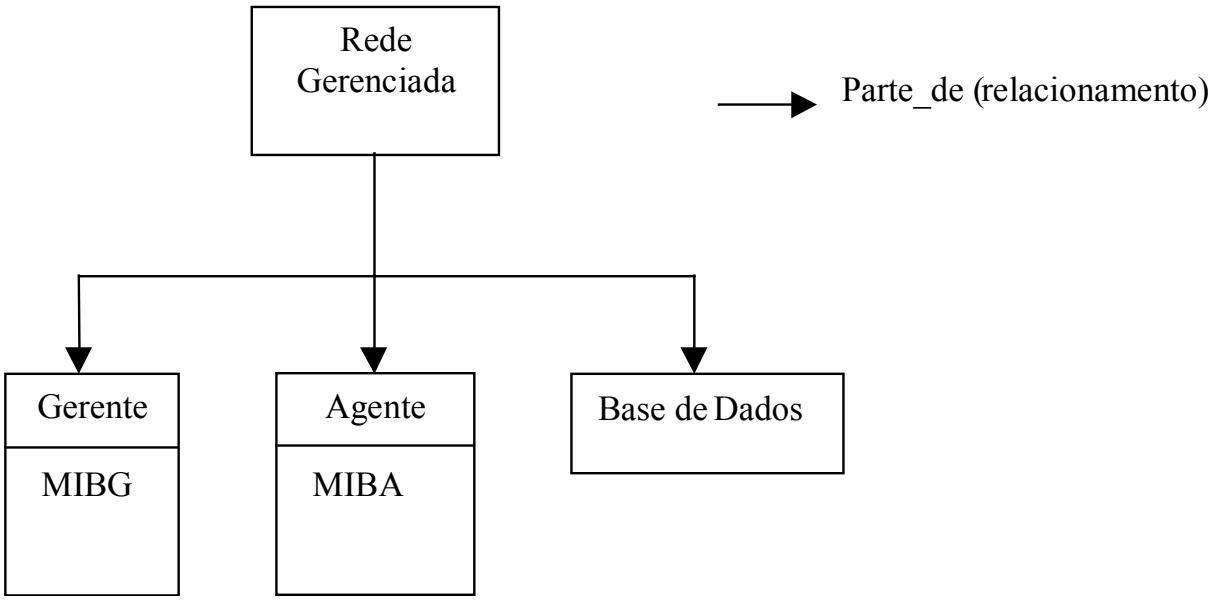


FIGURA 3.3 - Herança e Hierarquia no *CASUEL*.

A ferramenta *CASUEL* apresenta uma maneira particular de adquirir novos casos e tratá-los quando não são resolvidos. Quando ele percebe a existência de um novo problema, cria um novo caso que é armazenado em um banco de dados temporário. Todas as informações providas pelo cliente são ali armazenadas, até que o problema possa ser resolvido. Um mecanismo de recuperar casos não resolvidos pode ser ativado recuperando os casos deste BD, o qual, periodicamente, deve ser analisado por um especialista como modo de manutenção. Quando um novo caso resolvido é encontrado, passa a fazer parte do banco de casos principal, o qual vai receber todos os tratamentos de indexação característicos ao sistema.

O CASUEL consegue planejar suas decisões em diversos níveis e dependendo do problema, pode ser direcionado a uma decisão por árvore utilizando a ferramenta *Kate,* ou por raciocínio baseado em casos, utilizando a ferramenta S^3-Case, ou mesmo pela utilização de ambas ao mesmo tempo.

O CASUEL permite, como observado na figura 3.2, a adaptação pela criação de regras feita pelo engenheiro de conhecimento. Estas regras são determinadas pelo engenheiro de conhecimento durante o desenvolvimento do modelo do domínio; ele descreve como um caso pode ser recuperado e adaptado para se encaixar na consulta corrente.

A ferramenta possibilita a utilização de recursos via Web e de todas as vantagens de recursos multimídia.

3.3 *ReMind*

O objetivo da *shell ReMind* é a funcionalidade. Desenvolvida com o suporte do programa U.S. DARPA, sob a marca da *Cognitive Systems Inc*., ela oferece um ambiente interativo para aquisição de casos, um domínio de vocabulários, índices e protótipos.

3.3.1 Representação e Conteúdo dos Casos

O usuário deve definir relações hierárquicas através de atributos e uma medida de similaridade baseada nestas relações. O caso é representado como um *template* que utiliza pares de atributo/valor, sendo possível compartilhá-los com outros casos.

3.3.2 Aquisição

Para a aquisição, a ferramenta possui um analisador de palavras que elimina erros de grafia e redundâncias. As consultas podem ser feitas de forma textual e a entrada dos casos se dá pelo preenchimento de valores dos campos já determinados conforme a sua representatividade dentro do domínio de problemas. Esta alimentação é realizada utilizando-se um *Case Editor* do *ReMind*, o que muito se assemelha a qualquer tabela de preenchimento de um banco de dados relacional, sempre conservando os pares atributo/valor.

TABELA 3.1 - Disposição dos dados na inclusão de caso proposto na interface do *ReMind* pelo *Case Editor*.

Field Name	**Field Value**	**Field Type**
Is the Hub working?	No	Boolean
Is the Server working?	No	Boolean
Is the Cable transmiting?	Yes	Boolean
Is the network card working?	U	Symbol

3.3.3 Mecanismos de Aprendizagem

O *ReMind* aprende pela inclusão de novos casos de maneira simples no sistema. A sua capacidade de criar o Q-*model* e a maneira hierárquica de associar os atributos de seus casos permitem que ela associe um maior número de problemas a um determinado caso por conseguir avaliar melhor a similaridade entre eles.

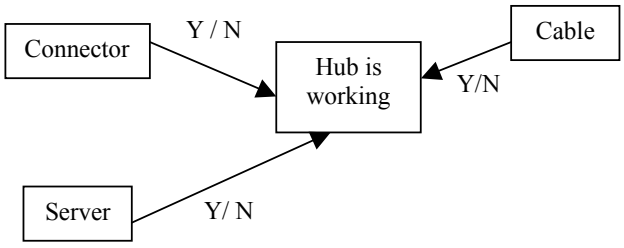


FIGURA 3.4– Um Q-*model* do *ReMind*.

3.3.4 Indexação e Similaridade

A indexação é feita indutivamente pela construção de uma árvore de decisão, permitindo ao usuário, graficamente, editar a importância dos atributos que serão responsáveis pelos cálculos de similaridade. Diversos métodos de recuperação são suportados: (1) recuperação indutiva, (2) recuperação por algoritmo de vizinhança e (3) recuperação de padrões, similar às procuras feitas em consultas SQL, onde se recuperam todos os casos que possuam uma determinada característica previamente definida.

Para a indexação dos casos, utiliza-se de uma árvore de decisão. Quando feita pelo método de recuperação indutivo, esta tarefa pode não envolver o usuário ou o usuário pode criar um modelo qualitativo para guiar o algoritmo de indução a fim de extrair raciocínio do conhecimento adquirido (indução guiada por conhecimento).

O modelo qualitativo (Q-*model*) é uma forma gráfica de se atribuir relacionamentos entre características de casos, para que se formem diferentes teorias sobre o domínio em estudo. Pode-se atribuir pesos a estas características que, na hora do raciocínio são responsáveis pela formação de árvores de decisão.

TABELA 3.2 – Comparação de casos feita pelo *ReMind*.

Input Case: 11	**Similarity: n/a**		
Field Name	Field Value	Field Value	Field Type
Is the Hub working?	No	No	Boolean
Is the Server working?	No	No	Boolean
Is the Cable transmiting?	Yes	Yes	Boolean
Is the network card working?	Yes	U	Symbol

3.3.5 Adaptação

A adaptação de casos se dá por fórmulas que ajustam valores baseados na diferença do caso recuperado versus o novo caso. Ainda tem a capacidade de representar relacionamentos casuais usando modelos qualitativos e de armazenar os passos que levaram a encontrar um caso, ou seja, possibilita a construção do método derivacional para a adaptação.

3.3.6 Comunicação com outros Mecanismos de Raciocínio

Não permite que sejam incluído outros mecanismos de raciocínio; a criação de modelos qualitativos para extração de conhecimento seria o único mecanismo alternativo da ferramenta para melhorar o raciocínio, o qual não envolve outros métodos de raciocínio.

Também apresenta uma grande facilidade de importar dados de banco de dados existentes.

3.3.7 Apresentação ao Usuário

Possibilita a construção de *interfaces* para o usuário do sistema; a apresentação da ferramenta ao construtor de sistemas é em modo gráfico. Não oferece, por enquanto, suporte à *Internet*.

3.4 *CaseAdvisor*

CaseAdvisor é um sistema inteligente de resolução e diagnóstico de problemas, desenvolvido pelo grupo de RBC da *Simon Fraser University.*

3.4.1 Representação e Conteúdo dos Casos

A representação de um caso fica assim determinada nesta ferramenta:

Keywords – Pequenas descrições dos casos;

Questions – Características típicas de um caso, representadas de forma relacional. As questões com múltiplas respostas recebem um mecanismo de indexação por logaritmos de recuperação de casos;

Case Description – Uma forma mais textual de descrição do caso, usada para confirmar a área geral do problema. Recursos de multimídia podem ser utilizadas neste campo, bem como recursos de hipertextos.

Case Solution – Gera uma solução para o caso em forma textual ou formato multimídia.

Um segundo componente da solução de casos é a árvore de decisão. Ela pode ser anexada ao caso para fornecer diagnósticos e planejamentos de ações interativas.

3.4.2 Aquisição de Casos

Os casos na ferramenta *CaseAdvisor* são adquiridos de três formas, através de:

TABELA 3.3 – Exemplo de como é colocado um caso na interface do *CaseAdvisor*.

Case 1	
Case Name	No found server
Description	The computer was stoped, it was reseted and no found server. The cable is working. The hub is working. The network card is receiving and transmiting.
Solution	Check the server. If it is stop, reset the server. If problem resolves, generate TT to FSR. If problem continues, generate TT for check cable of the server until the hub.

Um processador de textos (*CaseAuthor*) - Permite que se relatem casos estruturados ou não estruturados de maneira simples e fácil, como ao se utilizar um editor de textos. É um simples importador de títulos de casos, uma pequena descrição em inglês de perguntas e soluções associadas ao caso; aceita recursos gráficos e permite ao usuário atribuir medidas na forma de pesos para a importância do par pergunta-resposta em um caso.

TABELA 3.4 – Casos detectados como redundantes que podem ser tratados pelo *CaseMantainer*.

Case 1
Case Name: Server don't accept command.
Solution: Server is not working. Reset it. Keywords:server accept command working reset
Case 2
Case Name: Server is not working.
Solution: Server's screen is black and the keyboard whistle because it isn't working. Reset it.
Keywords:server working whistle keyboard reset screen

Um agente recuperador (*CaseMaintainer*) – Permite a resolução de problemas, um *runtime* para recuperação de casos, apresenta a capacidade de converter uma forma

textual em um caso, sugerindo palavras e perguntas para que sejam anexadas aos casos. Também checa redundância e inconsistência na base de casos, podendo o usuário descartar casos redundantes e combinar duas bases de casos eliminando redundâncias. Como no exemplo de redundância da tabela 3.4, o agente recuperador procura identificar as palavras-chave do problema, procedimento que pode se dar por meio de regras algébricas que se utilizam da capacidade de resolver algoritmos de *trigram.*

Tem-se dois casos, um deles é composto por p_i s_i; sendo que p_i representa um conjunto de palavras que indicam um problema e s_i uma solução (podendo ser composta por uma seqüência de passos) e o outro composto por p_2 s_2, como no exemplo abaixo:

Case 1 : Problems (p_1) Solution (s_1)

Case 2 : Problems (p_2) Solution (s_2)

Acha-se o valor de q_1, que é uma palavra-chave, ou um conjunto de palavras com algumas chaves e procura-se no caso 2 o mesmo conteúdo de q. Se encontrado, ambos são semelhantes e pode-se dizer que o caso 2 é subconjunto de caso 1. Assim, pode-se considerar o caso 2 como redundante.

Achados dois casos que se assemelham (p_1, p_2), ambos são apresentados ao usuário, o qual pode decidir se um pode ser apagado, por estar com as mesmas premissas do outro caso. Não significa, necessariamente, que um possa desaparecer. Desta forma, a responsabilidade é passada ao usuário;

Um serviço *online* – Permite a obtenção de problemas de casos não resolvidos e, assim, sua inclusão em uma base de casos, onde posteriormente podem ser transformados em novos casos com suas devidas soluções. Estas bases podem ser periodicamente limpas para que se obtenha um uso mais eficiente na recuperação e na utilização de métodos algébricos de detecção de redundância.

3.4.3 Mecanismo de Aprendizado

As bases de casos nesta ferramenta vão aprendendo por dois níveis; um nível de casos e um de pesos. O primeiro se dá pelo serviço de armazenar problemas não resolvidos, descrito acima na aquisição de casos.

O segundo nível existe na determinação de similaridade. A medida de similaridade (*sim*) entre casos é determinada por pesos anexados ao par pergunta-resposta. Este peso é determinado pelo usuário, a menos que se desenvolva um mecanismo que possa ajustar os pesos automaticamente, mas a ferramenta não oferece este mecanismo. Estudos da *Simon Fraser University* têm se encaminhado para este fim.

3.4.4 Indexação e Similaridade

A indexação aponta para pesos da base de casos. Os casos são indexados pelas suas características de problemas. Quando a descrição em inglês do problema é recebida, o sistema restaura os casos que possuam as palavras-chave mais semelhantes às descrições e distinções dos casos. Os casos com o maior valor *n* são recuperados. Quando *n* é um valor muito grande, o usuário pode optar por responder algumas

questões a fim de se aproximar mais dos casos procurados. As questões servem como estruturas logarítmicas para indexação, as quais dinamicamente se diferenciam de todos os casos recuperados. O tratamento da similaridade é feito pela utilização do algoritmo de vizinhança, o qual é baseado no peso da pergunta e no peso da resposta das perguntas, sendo que cada caso tem diferentes pesos para diferentes respostas da mesma pergunta.

3.4.5 Adaptação

O *CaseAdvisor* vem com uma API (*Aplication Programming Interface*) que pode ser utilizada para se construir a adaptação dos casos, que seria possível pela estipulação de um tempo de resposta para adaptação. Esta resposta é conseguida através de respostas fornecidas pelo usuário ao sistema, portanto o passo da adaptação não está pronto na ferramenta.

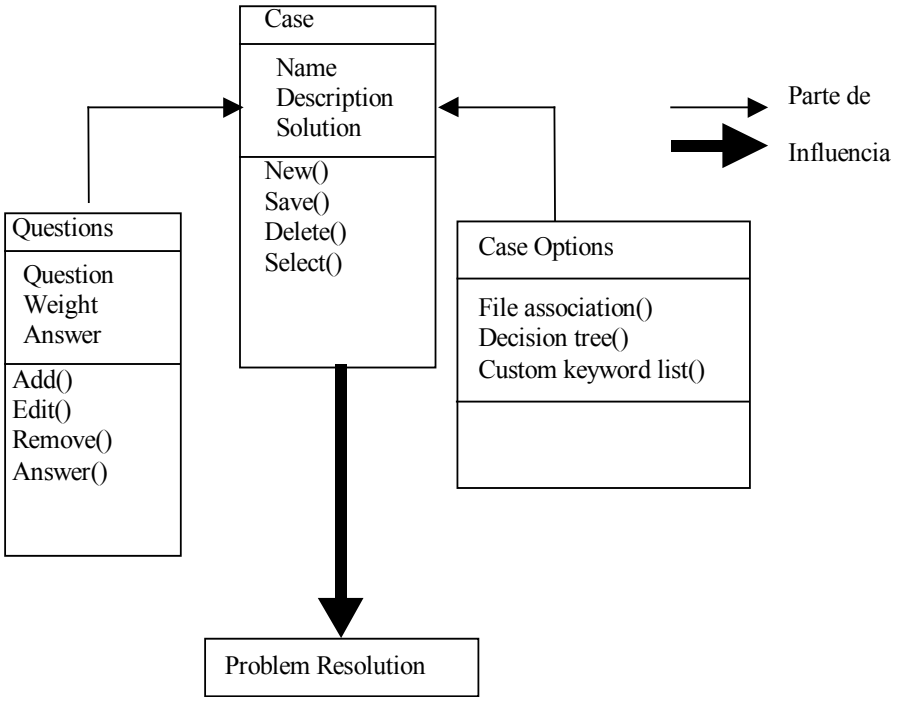


FIGURA 3.5– O mecanismo do CaseAdvisor

Ao apresentar um caso, ele se comunica com o usuário através de perguntas que, além de receberem as respostas, podem receber pesos para o atributo importante da pergunta no contexto do caso, Além disso, é possível se fazer associar o caso com outros arquivos de casos, bem como chamar a árvore de decisões do sistema para conduzir o algoritmo de indução através de conhecimento adquirido.

3.4.6 Comunicação com outros Mecanismos de Raciocínio

CaseAdvisor é muito conhecido pela sua integração com tecnologias que utilizem árvores de decisões, o que proporciona uma significativa estrutura de recuperação e organização das informações, onde as questões começam a ser organizadas antes de se chegar à procura pela resposta, o que resulta em um ganho de *performance*.

Outro grande poder de comunicação se dá pelo fato desta ferramenta demonstra grande facilidade de comunicação com bases de regras, o que, dependendo da maneira como se monta um sistema, poderia reduzir em muito o tempo de respostas para os casos propostos.

3.4.7 Apresentação para o Usuário

O *CaseAdvisor* possui um subsistema *WebServer,* o qual proporciona o uso da base de casos na *Web*. Isto é conseguido por uma *interface* CGI ou *Java* para a base de casos que integra padrões de servidores de *Web* com *softwares* que rodam em *Windows* NT e *Unix,* como é o caso do navegador *Netscape*.

Sendo assim, a sua *interface* com o usuário se dá por meio de *softwares* utilizados pela *Internet*.

3.5 *CBR3*

É uma família de produtos de RBC que alcançaram o maior sucesso de mercado, especialmente fabricados para clientes que mantêm *help desk*. *CBR3* é uma ferramenta composta por *CBR-Express, CasePoint, Generator, Tester e CasePoint WebServer*, produzida pela *Inference Corp*.

3.5.1 Representação e Conteúdo dos Casos

Utiliza-se de uma simples estrutura de registros de casos que são armazenados em um banco de dados relacional; tem como padrão o BD *Raima*, mas aceita os bancos de dados mais difundidos no mercado.

Um caso é composto por duas partes: a descrição e a solução do problema, esta última chamada de ação, que nada mais é do que os procedimentos para corrigir o problema, que são criados pelo *Action Editor*, como uma inclusão em separado.

Dentro destas partes encontram-se um título; uma maneira como o problema seria chamado ou identificado; uma descrição, onde se registra o que aconteceu através de texto em inglês; um conjunto de pesos para as perguntas, ou seja, pares de atributo-valor que corresponderão às respostas dadas às perguntas feitas pelo sistema, e um conjunto de ações, que são procedimentos que podem ser tomados para resolver o problema.

No registro das ações, pode-se anexar textos através de um campo que os comporta, ou adicionar um custo para as ações, o qual é utilizado pelo *CasePoint* como

índice nas indexações ou, ainda, considerar uma ação como prioritária ou como não aceitável pelo usuário do *CasePoint*.

CBR-Express considera as perguntas e ações como recursos que podem ser compartilhados por vários casos.

3.5.2 Aquisição

Os casos são adquiridos pela alimentação do módulo *CBR-Express Case Editor*. A entrada de casos na base é muito fácil e simples, basta ter claramente divididas as noções de título, descrição, questão e ação. A única exigência é que se tenha os casos bem estruturados antes da inclusão na base. Não há necessidade de preocupação com caracteres especiais como parênteses, vírgulas, etc., dentro do tratamento das palavras pelo sistema.

Outra maneira de aquisição, se dá pela alimentação de casos sem solução no sistema, onde se chama o *CasePoint*, que proporciona uma realimentação (*feedback*) que é enviada à base de casos. Este aviso de realimentação pode ser também ser enviado por *e-mail* para qualquer pessoa pré-determinada, útil para bases que estão em uma rede de computadores e que são gerenciadas somente em um determinado local.

O aviso de *feedback*, ao chegar para o engenheiro de conhecimento, recebe um tratamento para ganhar uma solução e é então incluído como um novo caso no sistema. Portanto, o *CBR3* possui um método de aquisição de novos casos.

3.5.3 Mecanismos de Aprendizagem

O *CBR3*, como visto acima, aprende pela alimentação do sistema na inclusão de novos casos ou de casos que entram no sistema e que no momento não têm solução, os quais são tratados pelo engenheiro de conhecimento e passam a fazer parte da base de casos. A aprendizagem é simples e feita como uma inclusão normal no sistema.

O *CBR3* ainda possui um módulo chamado *Generator* que é uma aplicação capaz de analisar gramaticalmente arquivos de textos, deduzindo seus conteúdos para criar um novo caso e já indexá-lo na base de casos. Este tipo de criação de casos é muito útil para sistemas de documentação técnica.

3.5.4 Indexação e Similaridade

Uma vez terminada a inclusão dos casos que pode representar situações diversas, é necessário preparar a base de casos para ser utilizada. Assim, é necessário que esta receba os tratamentos de indexações. *CBR-Express* indexa os casos para que possam ser recuperados eficientemente. A própria ferramenta possui uma opção de indexação, que pode levar mais ou menos tempo, dependendo do tamanho da base. Esta indexação é um arranjo na base de dados realizado através dos graus indicados nas entradas dos casos e que servem de índices para recuperação dos mesmos.

O usuário do sistema, ao consultar um problema, pode apresentá-lo com suas próprias palavras. Estas sofrem uma profunda avaliação feita pelo *CasePoint*, que é um

eficiente mecanismo de procura pela base de casos existente no *CBR-Express*, onde é realizada uma análise nas palavras relevantes, bem como, nas correções de escritas. *CasePoint* pode fazer uso de sinônimos e representar palavras por um conjunto de *trigrams* (junção de três caracteres na ordem em que as letras se apresentam na formação de uma palavra. Exemplo para servidor: ser, erv, rvi, vid, ido, dor). Tal característica ajuda a resolver problemas com padrões de escrita.

O resultado da entrada do problema é o surgimento de uma lista ordenada de soluções e uma lista de perguntas apresentadas ao usuário com o objetivo de ajudar a encontrar o *best match*. Esta lista é gerada por uma espécie de árvore de decisão. Assim que o usuário começar a responder as perguntas, que não requerem ordem, o *CasePoint* começa a utilizar um algoritmo próprio de recuperação de casos (podendo também utilizar-se do algoritmo de vizinhança, o que tornaria o sistema um pouco mais lento), fornecendo uma base pequena como resultado da procura. Conforme o usuário vai respondendo as questões, ele vai contabilizando a importância da resposta para o caso e comparando-o com casos que apresentam tais ítens como índices, até que encontre um com maior nota (*score*). Esta é a maneira como o sistema trata o problema de similaridade, ou seja, através de conhecimento guiado.

A procura pelo *best match* dentro do *CBR3* pode ainda contar com a ajuda de regras que utilizam *forward chaining*, o que pode vir a encurtar, em muito, a procura de ações para os problemas mais simples, comuns e melhor definidos.

3.5.5 Adaptação

Quando um problema consultado num sistema *CBR3* não tiver uma resposta ou apresentar uma que não seja correta, como comentado anteriormente, é feita uma atualização na base de casos através do ativamento da opção de *feedback*, que passa a adaptação para um especialista resolver. Este, no futuro, seguindo passos de um *workflow*, passa o registro da solução do caso para o engenheiro de conhecimento que a incluí como um novo caso.

A adaptação neste tipo de sistema é dada, portanto, pela realimentação do sistema após a interferência de um especialista, não exigindo nenhum procedimento automático que caracterize adaptação.

3.5.6 Comunicação com outros Mecanismos de Raciocínio

Como já comentado anteriormente, *CBR3* consegue se comunicar muito bem, inclusive incorporar, em seus módulos, raciocínio baseado em regras. Todos os passos que *CBR3* dita, durante seus procedimentos, para avaliar a situação do problema, têm a finalidade de achar um estado em que o caso proposto se identifique com algum do passado.

Então, de posse deste caso recuperado, o mais natural seria, para resolver o problema, utilizar-se de regras que indiquem que a situação atual é a mesma do problema recuperado e então proceder as mesmas ações (se algo aconteceu assim, faça assim – *If* servidor travado, *if* tela preta *then* desligue-o).

3.5.7 Apresentação ao Usuário

A *interface* da ferramenta tem sofrido várias modificações em seus diversos módulos, inclusive no *CBR-Express*, que apresenta um caso com o estrutura de uma pasta de arquivos.

Editores são abertos para que se preencham os dados de um caso. Todas as telas e módulos do *CBR3* são apresentados ao usuário de forma gráfica, ainda podendo dispor dos recursos fornecidos pelo módulo *CasePoint WebServer*, que é uma CGI para a base de casos, a qual proporciona toda uma *interface* baseada em hipertextos, inclusive de módulos como o *CasePoint*. O *WebServer* pode ser integrado à maioria dos *softwares* servidores para *Web* (*Solaris, NT, HP-UX* e *OS/2*) e ainda proporciona o gerenciamento de bases de casos globais.

É importante comentar que *CBR3* também possui um módulo de testes, chamado *CBR-Express Tester Module*, que ajuda o engenheiro a planejar a base de casos, provendo análises estatísticas e dinâmicas desta.

3.6 Avaliação das Ferramentas *Remind*, *CaseAdvisor* e *CBR3*

Todas as ferramentas apresentam formas próprias de representar um caso, obedecendo à formação de um caso por feições significativas para determinadas situações. O *ReMind*, apresenta a forma mais simples de se representar casos, feita pela utilização dos pares atributo-valor, porém as outras ferramentas apresentam a facilidade de se descrever um caso de forma textual.

Quanto ao item aquisição, as ferramentas *CaseAdvisor* e *CBR3* apresentam tratamentos de casos que não tiverem resposta de solução pela consulta de casos passados, sendo que o *CBR3* ainda oferece a possibilidade de enviar *feedbacks* aos engenheiros de conhecimento, enquanto o *CaseAdvisor* cria uma base de casos temporários para o tratamento destes casos.

O fato da *ReMind* possuir o módulo Q-*model* faz com que ela apresente um ótimo aproveitamento no que diz respeito a obter conclusões sobre um determinado número de dados, devido à maneira hierárquica de associar atributos, enquanto as outras aprendem pela atribuição de pesos de similaridade ou análise gramatical.

O processo de recuperação da ferramenta *CBR3* merece destaque por ter desenvolvido um algoritmo próprio de recuperação de casos que além de ser mais eficiente, consome menos tempo de recuperação em relação à utilização do algoritmo de vizinhança. O algoritmo já vai arrumando a árvore de decisão à medida em que as opções do caso proposto são incluídas no sistema. Todas as ferramentas podem guiar os algoritmos de recuperação pelo conhecimento que o sistema vai adquirindo.

A ferramenta *ReMind* é um bom exemplo de uma ferramenta que permite ajustes de parâmetros, pois o usuário pode alterar pesos na recuperação por algoritmo de vizinhança, tornando possível a adaptação de casos. Consegue, ainda, proporcionar o método derivacional para a adaptação, através dos relacionamentos permitidos pelo Q-m*odel*. Já a *CaseAdvisor* apresenta a possibilidade de construção de adaptação por

CGI, mas não apresenta nada pronto. Algo parecido pode ser percebido na *CBR3*, onde a adaptação é feita de modo manual, passando pelo usuário.

Tanto a ferramenta *CaseAdvisor* quanto a *CBR3* possibilitam comunicação com raciocínio baseado em regras, sendo que a segunda proporciona, inclusive, a incorporação de módulos de regras aos módulos de casos. Esta propriedade pode ajudar muito a *performance* dos sistemas, tendo em vista que sistemas RBC prezam pelo consumo de memória de trabalho.

Das três ferramentas analisadas, a única que ainda não apresentou um módulo para *Web* é a *ReMind*. Todas se comunicam através de várias plataformas e sistemas operacionais.

Pelos pontos em que foram analisadas as ferramentas, como pode ser verificada na tabela 3.5, tendo sempre em mente sua utilização especificamente para o domínio de redes de computadores, a ferramenta que apresentou melhores resultados foi a *CBR3*, principalmente pela sua possibilidade de comunicação com a maioria dos tipos de bancos de dados líderes de mercado. Este fato é importante quando ainda não se tem definida a base de casos ou quando se quer aproveitar bases existentes, pois uma característica necessária ao sistema RBC é que já exista uma base de casos quando este começa a ser operado. Pode-se citar ainda as características de possuir um módulo de testes, útil para montagem de protótipos, não só do sistema como da base de casos e, também, pela boa performance com grandes volumes de dados e transações e pela sua modularidade.

Sua maior vantagem sobre as outras ferramentas, porém, encontra-se no fato de possuir um analisador de casos, onde da inclusão textual pode-se obter feições de casos, montando-se automaticamente um novo, sem a interferência do EC, ponto importante para o domínio de problema em estudo. A avaliação das ferramentas foi feita sem considerar-se o padrão *INRECA,* por ser este, um padrão para construções de sistemas que se utilizam de RBC, o qual não poderia ser analisado sobre os mesmo enfoques de uma ferramenta.

TABELA 3.5 – Comparação das Ferramentas

Função	***ReMind***	***CaseAdvisor***	***CBR3***
Forma Própria de Representar um Caso	X	X	X
Módulo para Casos não Solucionados		X	X
Criação de Bases Temporárias		X	
Algoritmo Próprio para Recuperação		X	X
Algoritmo de Vizinhança	X		
Comunicação com bases de Regras		X	X
Apresentação Via *Web*		X	X
Módulo Automático para Adaptação	X		
Recuperação Indutiva por Árvore de Decisão	X	X	X
Módulo Próprio para Aquisição de Casos		X	X
Uso de Banco de Dados Relacional			X
Módulo de Testes			X
Algoritmos de Trigram		X	X
Relacionamento Hierárquico de Características	X		
Similaridade por Pesos de Características		X	X
Analisador de Casos			X

4 Gerenciamento de Redes

As organizações que, cada vez mais, procuram equipamentos e facilidades, tanto *hardware* quanto *software*, de diferentes fabricantes e fornecedores, estão encontrando a solução para o gerenciamento integrado e eficiente de seus sistemas de padronização. Segundo CEREDA [CER 97], até 1990, a maioria dos Sistemas de Gerenciamento de Rede (SGR) poderia ser considerado ilhas de informação, fornecendo capacidade para monitorar e controlar um limitado conjunto de equipamentos e, usualmente, não operando com equipamentos de outros fabricantes. Com o crescimento das redes de computadores nas organizações, isto tornou-se um fato crítico. O fato dos sistemas não interagirem forçava os operadores a se locomoverem de um console para outro a fim de resolver falhas isoladas. Esta situação serviu de motivação para o surgimento de vários grupos interessados no processo de desenvolvimento de padrões para o gerenciamento de redes. O objetivo principal destes grupos era fornecer mecanismos para a interconexão daquelas ilhas de informação. Atualmente, os sistemas de gerência aderem com mais ou menos força ao padrão Internet [SOU 95].

Gerenciamento de rede compreende o controle e monitoração de uma rede, feito por uma equipe de especialistas no assunto, os quais fornecem o melhor desempenho possível para os usuários da rede. Um sistema de gerência de rede e as informações por ele geradas são de grande importância porque estão relacionadas com todas as funções e operações sobre os recursos disponíveis através dela.

Gerência de redes é uma aplicação distribuída, onde processos de gerência (agentes e gerentes) trocam informações com o objetivo de monitorar e controlar a rede. Os processos de gerência se relacionam com objetos gerenciáveis, conforme suas atribuições, sendo eles:

- Gerente: obtém informações a partir dos agentes sobre os objetos gerenciados e os controla, emitindo operações (primitivas de monitoramento) para os agentes;

- Agente: executa operações de gerenciamento sobre objetos gerenciados e transmite as notificações emitidas pelos objetos gerenciados ao gerente;

- Objeto gerenciável (ou gerenciado): representação do recurso do sistema que está sujeito ao gerenciamento. As informações referentes aos recursos estão em uma base de informação de gerenciamento (MIB- *Management Information Base*), acessível somente pelo agente.

A dinâmica dos processos de gerência pode ser assim descrita: o processo gerente envia solicitações ao processo agente que, por sua vez, responde às solicitações e também transmite notificações referentes aos objetos que residem na MIB.

O serviço de gerência de redes atua, segundo definição da ISO, em cinco áreas funcionais de gerenciamento, que são [LEI 96]:

Gerência de Configuração: exerce controles sobre a configuração física e lógica da rede. Deve fornecer informações como o inventário dos tipos de equipamentos de uma rede, com seus detalhes e informações de mudanças ocorridas, bem como um

mecanismo fácil e rápido de busca por máquinas na rede. O fornecimento de tais dados compreende três etapas para esta área, que são: obtenção de informações, modificação de configuração de equipamentos e armazenamento de dados [SWI 96].

Gerência de Desempenho: analisa e controla o desempenho e as taxas de erros da rede, incluindo informações históricas sobre o funcionamento da rede. Compreende coletar os dados de utilização dos equipamentos, analisar os dados relevantes e estabelecer limiares de utilização da rede para tentar maximizar seu desempenho [SWI 96].

Gerência de Falhas: responsável pela detecção, isolamento e controle de procedimentos anormais da rede; quando possível, deve permitir à administração se antecipar às falhas para corrigir os problemas antes que elas ocorram. Vem a contribuir para aumentar a confiabilidade da rede por poder proporcionar informações necessárias sobre o estado corrente da rede. Consiste de detecção e aviso do problema, que podem se dar por alarmes ou eventos e por registros de problemas (*log*), seguidos pelo isolamento do problema e, por fim, pela correção através de métodos manuais ou automáticos.

Gerência de Contabilização: faz a coleta e o processamento dos dados referentes ao consumo de recursos na rede, permitindo um controle mais adequado quanto aos custos da rede. Compreende as seguintes etapas: obter dados da utilização da rede, estabelecer quotas de uso e verificar o uso que o usuário faz da rede e o modo de fazê-lo.

Gerência de Segurança: controla e monitora os mecanismos de segurança dos processos da rede, compreendidos como sendo acessos a pessoas não autorizadas a determinados locais da rede. A gerência de segurança deve detectar estes pontos, tornando-os seguros.

O gerenciamento de rede de computadores é uma atividade que desenvolve um grande número de tecnologias utilizadas simultaneamente, denominadas proprietárias ou padronizadas.

As tecnologias proprietárias são oferecidas em produtos comerciais classificados como plataformas de gerenciamento de redes, muitas vezes proibitivas devido aos elevados custos de aquisição.

As tecnologias padronizadas, utilizadas para gerenciar redes de computadores, estão presentes no que se pode chamar de arquiteturas de gerenciamento de rede, cuja utilização em redes TCP/IP é facilitada, principalmente, pelo livre acesso a diferentes pacotes implementados.

4.1 Importância de um Gerenciamento

A utilização de redes de computadores como meio de comunicação vem se tornando indispensável no desempenho de diversas atividades, tanto na educação como nas empresas. Manter a qualidade e disponibilidade dos serviços oferecidos à comunidade de usuários torna-se, cada vez mais uma tarefa de extrema complexidade. Administradores de rede necessitam automatizar suas rotinas de trabalho, permitindo o controle e monitoramento da rede de forma mais amigável. O gerenciamento de redes é utilizado como forma de suprir tais necessidades.

O número de profissionais capacitados para o desenvolvimento de aplicações de gerenciamento de redes é bastante reduzido. Considerada uma atividade complexa e com rápidas transformações tecnológicas, propicia o desenvolvimento de projetos de pesquisa, que busquem a capacitação de recursos humanos para o desempenho destas atividades.

O gerenciamento de redes é um tipo de serviço que depende da existência de uma efetiva coleção de informações manipuláveis e disponíveis para tomadas de decisões. Quando gerentes ou agentes se comunicam, eles utilizam um protocolo particular de gerenciamento de redes. Este protocolo define o tipo de mensagem que eles podem enviar, a estrutura das mensagens e do banco de dados de gerenciamento, as variáveis disponíveis no banco de dados, dentre outros. Cada agente possui um banco de dados (BD) que contém, entre outros, informações para solicitações de gerenciamento, o qual é vasculhado sempre que o gerente solicita uma informação. O BD gerenciável é composto por uma lista de objetos gerenciáveis dos tipos padrões (MIB padrão) e proprietários (MIB proprietária - implementada pelo vendedor). O protocolo de comunicação entre o gerente e o agente pode ser o SNMP (*Simple Network Management Protocol*) ou o CMIP *(Common Management Information Protocol*). O SNMP é da comunidade *Internet* e utiliza-se de IP, UDP, TCP e IPX; já o protocolo CMIP da ITU (*Internacional Telecommunications Union*) se propõe a trabalhar em qualquer rede ou dispositivo.

Devido a problemas causados pelo tráfego de informações gerados pelo SNMP, criou-se um monitoramento remoto (*probe*), onde o gerenciamento só repassa informações do agente para o gerente, em forma de *frames*, quando solicitado. Estes monitoramentos remotos apresentam um padrão chamado de MIB RMON (*Remote Network Monitoring*), podendo se dar, também, por protocolos analisadores que são similares aos *probes*, porém com mais capacidade de analisar os *frames* capturados. Esta análise pode significar percorrer os cabeçalhos dos *frames* para identificar em que camada o problema ocorreu.

4.2 Gerenciamento de Segurança

Todas as áreas de gerenciamento, discutidas acima, merecem atenção especial, mas, ultimamente, uma vem ganhando evidência em todas as organizações: é o gerenciamento de segurança, o qual compreende estabelecimentos e monitorizações de atividades que garantam a existência de sistemas e redes com algum nível de segurança. O objetivo não reflete o contexto utópico de implantação de procedimentos que os tornem completamente seguros, mas que garantam níveis de segurança nos quais a dificuldade para quebrá-los seja tanta, que o trabalho não justifique os resultados.

As informações tecnológicas são críticas e estão tendo peso crescente no sucesso das organizações, assim como a segurança de computadores. A tecnologia necessária para fornecer alguma forma de proteção está ficando mais complexa. Esquemas de encriptação, por exemplo, que antes eram considerados excelentes, podem ser agora apenas satisfatórios, ou inadequados, pela capacidade e quantidade dos micros existentes.

Tecnologia, entretanto, não é a resposta para todos os problemas de segurança de computadores, muitos são os tópicos a serem considerados. Para gerenciar a

confiabilidade, integridade e acessibilidade dos dados tem-se, hoje, contribuído para a criação de um ambiente onde a maior arma de segurança das informações é a própria informação.

O crescimento de redes corporativas e a grande necessidade de comunicação entre organizações, tem transformado-as em excelentes alvos de ataques por pessoas não autorizadas, mediante os mais variados objetivos e motivações. Uma quantidade cada vez maior de atividades essenciais é desenvolvida por intermédio das redes como a Internet, tornando fundamental seu funcionamento correto e confiável, onde serviços e dados trafegam pelos equipamentos de uma rede e seus protocolos.

Ao mesmo tempo em que, ocorreu um enorme avanço na comunicação entre redes, tornando possível o compartilhamento globalizado de seus recursos e informações, a segurança dos dados passou a ser uma área crítica, pois quanto maior for a facilidade ao usuário para acessar essas informações, maior é a probabilidade de pessoas intrusas compartilharem esses acessos.

Estes ataques podem ocorrer de várias maneiras, utilizando-se de diferentes procedimentos e ferramentas, podendo implicar na paralisação de segmentos da rede (ataques de *denial of service*) ou de sistemas servidores (ataques à *hosts*). Assim, técnicas especiais de segurança tornam-se indispensáveis nos sistemas computacionais modernos. Muitas informações e procedimentos de detecção, verificação e correção de tais problemas são desenvolvidos tanto no setor acadêmico quanto comercial, com o objetivo de prevenir ataques e auxiliar reparos, caso eles venham a ocorrer.

O gerenciamento de segurança lida com paradas nos elementos, *interfaces* e meios de transmissão dos equipamentos de rede, bem como, alterações em configurações e dados. As falhas estão, quase sempre, relacionadas diretamente aos componentes de infra-estrutura de uma rede, dentre os quais pode-se citar:

- *Hubs* e concentradores;
- Switches, bridges, routers;
- Modens, muxes ;
- Servidores, *mainframes*;

As alterações de configuração ou dados relacionam-se, geralmente, às configurações de um *host* e de uma base de dados. O termo *host* é empregado a qualquer computador que se comunica com outro semelhante, através da rede Internet, independente do tipo de sistema operacional e *software* utilizado, ou do serviço disponibilizado [CAN 96].

O objetivo das funções de gerenciamento de segurança é garantir proteção à rede e aos sistemas, através de recursos que, efetivamente, garantam medidas de segurança para equipamentos, assim como integridade e privacidade a dados disponibilizados na rede. Para que isto seja possível, as funções de gerenciamento devem prover:

- contínua análise de risco para identificar riscos de segurança;

- desenvolvimento de estratégias de risco e de um guia para minimizar ou eliminar riscos identificados;
- implementação de um plano de segurança;
- monitoração da redução dos níveis de risco da rede.

Para alcançar tais objetivos, torna-se necessária a instalação mínima de alguns mecanismos de segurança, como:

Manipulação de eventos - métodos para se relatar e manipular eventos relevantes à segurança;

Auditoria de segurança - testar a adequação dos sistemas de controle, para assegurar o cumprimento de políticas estabelecidas e procedimentos operacionais e para recomendar quaisquer mudanças no controle, política e procedimentos;

Recuperação de segurança – formas de facilmente recolocar em funcionamento o que foi atingido;

Gerenciamento de chave - é a geração, armazenamento, distribuição, deleção, arquivamento e aplicação de chaves, de acordo com a política de segurança do sistema.

Criptografia - ciência utilizada para codificar arquivos, de maneira que somente quem possua uma chave de entrada poderá decodificar o mesmo.

Controle de Acesso - registrar, alterar e apagar informações de acessos;

Integridade dos dados - garantir que os dados não foram alterados ou destruídos de forma não autorizada;

Autenticação - garantia de que quem envia uma determinada mensagem é realmente quem diz ser;

Enchimento de tráfego - utilizado para testes como, por exemplo, de performance da rede;

Controle de roteamento - obter o estabelecimento de uma rota segura, alterar a designação de segurança de uma rota, apagar a designação de segurança de uma rota;

Registro (*logs*) - registro em banco de dados e/ou arquivos de tudo o que acontece na rede;

Assinatura digital - algoritmos criptográficos que permitem ao remetente de uma mensagem, utilizando chaves criptográficas, assinar eletronicamente suas mensagens e fornecer aos usuários da rede uma outra chave que verifica a autenticidade da assinatura de suas mensagens.

Segundo o Comitê Gestor de Internet no Brasil [CGI 2000], os níveis de segurança podem ser estabelecidos ou melhorados pelo estabelecimento e cumprimento de políticas de segurança onde deve prevalecer o uso do bom senso e senso comum.

4.3 Política de Segurança

As decisões tomadas relacionadas à segurança dirão o quanto uma rede é segura ou não, mas tais decisões só podem ser tomadas mediante a determinação de metas de segurança chamadas de política. Política de segurança é, portanto, a expressão formal das regras pelas quais é fornecido acesso aos recursos tecnológicos de uma empresa, tendo como principal objetivo informar aos usuários, equipe e gerentes suas obrigações para a proteção da tecnologia e do acesso à informação.

O principal ponto para garantir segurança é proporcionar a visão de que existem riscos e ameaças. Segurança em rede é somente uma parte de um planejamento maior que inclui segurança física e recuperação em desastres.

Quando surgiram os primeiros microcomputadores e sistemas, o costume dos usuários era descrever, em etiquetas colocadas nas máquinas, os procedimentos de entrada aos sistemas, onde ficavam disponibilizadas inclusive as senhas dos mesmos. Com a proliferação do uso de computadores, a partir do ano de 1990, estes vícios começaram a desaparecer. Os sistemas e redes começaram, efetivamente, a sair de salas e depois dos prédios das empresas, fazendo parte da vida das pessoas, acompanhando-as por onde fosse necessário. Hoje, a maioria dos sistemas está disponível na Internet atendendo às exigências de crescimento das organizações, que acabam por colocar em teste a eficiência e eficácia dos sistemas e redes construídos. Este teste dá-se pela constante disponibilidade dos dados e pela ameaça vinda de pessoas não autorizadas para o compartilhamento destas informações (*cracker*).

As dimensões dos problemas causados por um ataque podem atingir vários níveis, desde uma inclusão de arquivos em contas ou trocas de senhas, até alterações ou substituições de arquivos de importantíssimos como o *passwd*.

Tendo em vista os crescentes problemas de segurança, órgãos de segurança, como o CERT (*Computer Emergency Response Team*) [CER 2000], CAIS/RNP (Centro de Atendimento a Incidentes de Segurança) [CAI 99] e NIC (*Network Information Center*) [NIC 2000] dedicam-se ao estudo e registro de falhas e correções no que tange à segurança das redes e sistemas, ajudando a definir as políticas necessárias a uma rede de sistemas abertos. De tais políticas surgem os tipos de serviços e equipamentos que se deseja proteger e do tipo de intrusão surgem procedimentos e avaliações do grau de comprometimento dos sistemas. É importante considerar que estão disponíveis aos ataques não só os dispositivos de uma rede, como também os sistemas construídos por programadores de alguma organização.

Segundo Garfinkel [GAR 94], uma política de segurança para rede ou sistemas deve garantir três requisitos: disponibilidade de recursos, confiabilidade e integridade dos dados.

Mecanismos de proteção são planejados para aumentar tais requisitos através de serviços que incluem autenticação, controle de acesso e não rejeição à conexão. Diferentes níveis de acesso e modelos de confiança devem ser estabelecidos de acordo com os serviços oferecidos. Assim, uma política de segurança deve ser prevista mediante acessos à *hosts* da rede interna versus *hosts* da rede externa.

Para verificar a veracidade de uma suspeita, torna-se necessário obter um retrato instantâneo do sistema. Como muitos incidentes podem desencadear inúmeros eventos, para validá-los são necessários arquivamentos de *logs*. Junto com a identificação do incidente deve ser feita uma análise sobre o impacto dos problemas procurando delimitar os limites atingidos para desta forma conduzir à reparos necessários. Muitos são os sintomas de um acidente portanto, todos devem receber atenção, como em casos de *crashes* de sistemas, novas contas de usuários, tentativas de escrever no *system*, negação de serviços (*denial of service*) e anomalias.

Os objetivos de uma política de segurança devem ser estabelecidos a partir de determinantes como os demonstrados na tabela 4.1e gráfico 4.1:

TABELA 4.1 - Tabela de Relação de Segurança com Serviço, Uso e Custo.

X	Segurança
Serviço	Cada serviço tem seu próprio risco; para alguns serviços o risco é superior ao benefício de seu uso.
Uso	A facilidade de uso está relacionada diretamente a segurança, o estabelecimento de senhas de acesso torna o uso menos amigável, mas mais seguro.
Custo	O custo de uma segurança deve ser contra-balanceada com a extensão de seus riscos em casos de perdas, que podem compreender diversos níveis como perda de privacidade, de dados e de serviços.

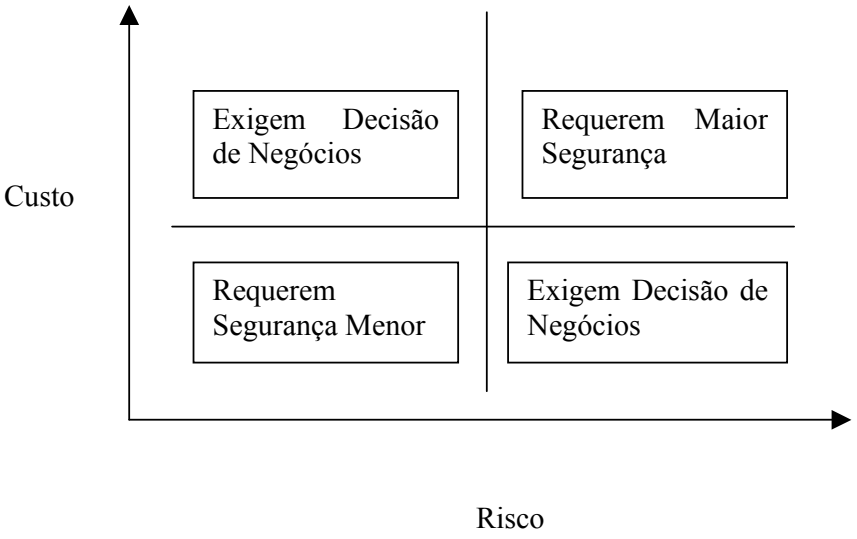


GRÁFICO 4.1 - Relação Custo e Risco de uma Política de Segurança.

Para que uma política de segurança se torne apropriada e efetiva, ela deve ter a aceitação e o suporte de todos os níveis de empregados dentro da organização . Deve possuir características como:

- procedimentos administrativos para uso da política;
- ferramentas de segurança e prevenções onde estas não se aplicam;
- auxílio e estipulação de procedimentos legais;
- áreas de responsabilidade bem definidas para usuários, administradores e gerentes.

4.4 Gerenciamento e Sistema de Apoio à Decisão

O gerenciamento de redes de comunicação e sistemas vem ao encontro das necessidades de seus usuários quanto à disponibilidade, ao desempenho e à estabilidade, os quais são responsáveis pela inicialização de funções de controle ou o chamado gerenciamento. Os fatores críticos para o sucesso de tais pontos são a metodologia, as ferramentas e os recursos humanos utilizados [TER 87], que vão diretamente ao encontro dos objetivos de uma organização.

A funcionalidade e desempenho em tempo real são relevantes no controle operacional da rede. As informações necessárias são extraídas da comunicação existente na rede e trabalhadas em uma base de dados de gerenciamento que processa e analisa tais características. Para chegar à determinação e correção de problemas, um grande número de atividades é desenvolvida visando-se a apurar as dimensões e durações dos problemas, bem como traçar um planejamento para correções.

Uma prática comum ao se trabalhar em uma área de gerenciamento é, ao se encontrar uma solução para um problema existente, jogar-se fora os diagnósticos e procedimentos tomados para se chegar à solução, o que caracteriza um grande erro. Só solucionar os problemas não é o suficiente, é importante saber se prevenir contra eles e poder obter relações de experiências para diagnósticos mais rápidos e eficientes no futuro. Um grande objetivo de estudos é produzir uma ferramenta que possa, rapidamente, especificar o tipo de falha e onde esta ocorreu, dando um preciso diagnóstico do problema.

Para realização de tais tarefas, o mais indicado é o uso de *trouble ticketing systems* (STT) ou *help desk systems* (HDS) que são sistemas usados para suporte técnico. Provêm, ferramentas para rápida e facilmente gerenciar problemas de redes e serviços solicitados. São alimentados pela geração de *trouble tickets* (TT), que podem ser criados manualmente ou automaticamente através da coleta de ferramentas de *e-mail,* de gerenciamento ou via telefone [JOH 92].

Os TT devem ser atualizados sempre que um procedimento com sucesso foi executado. Estes tipos de sistemas devem ser acompanhados de uma correta organização e de uma manutenção via um setor de *help-desk.* Compreendem planejamento das mais coerentes técnicas, consertos, alternativas de interrupção, seleção de ferramentas e acompanhamento de tarefas administrativas.

O relato de problema pode-se dar por três níveis, citados por [LEN 84]. O primeiro compreende o relato do problema, que pode ser realizado por telefone, fax, *e-mail* ou alarme de sistema. Tal relato é verificado por um operador do sistema que, de

um console, analisa as condições e ferramentas de sua rede, a fim de obter um diagnóstico do problema e resolvê-lo. Se solucionado em primeira instância é designado como incidente operacional, senão, o problema é passado a um especialista (técnico). Este fará o segundo nível de diagnóstico, que compreende testes em circuitos e outros equipamentos envolvidos no problema, podendo acionar a equipe de manutenção apropriada. Já o terceiro nível compreende o envolvimento de programadores e analistas. É muito raro um relato de problema chegar ao terceiro nível.

Os serviços de gerência de redes prezam pela eficiência e eficácia nas respostas a problemas. Poucos gerentes de rede têm o conhecimento completo para diagnosticar problemas de intrusão em redes e sistemas. Portanto, evidencia-se a necessidade de prover apoio ao gerenciamento de segurança nas redes, especialmente no que tange à detecção de intrusão e avaliação do grau de comprometimento de sistemas invadidos, bem como recomendações de cursos de ações corretivas que venham a apoiar a tomada de decisão dos gerentes.

A chave para um perfeito gerenciamento de redes encontra-se nas informações que possam ser obtidas da mesma. Assim, precisa-se ter a possibilidade de visão e manipulação destas informações, além de poder compartilhá-las com outras pessoas que trabalhem nas mesmas funções e/ou em colaboração, a fim de se obter fundamentação e evolução nos trabalhos realizados.

4.5 Sistemas de *Trouble Ticketing* (STT) ou Sistemas *Help Desk* (SHD)

Trouble Ticketing Systems ou *Help Desk Systems* são sistemas usados para suporte técnico. Um STT provém ferramentas para rápida e facilmente gerenciar problemas de redes e serviços solicitados.

Quando as pessoas pensam sobre gerenciamento de redes, diversas coisas vem em mente, como por exemplo; incluir protocolos de roteamento e tabelas, gerenciamento de estações SNMP, cabos, dentre outros.

Em problemas existentes com redes, os usuários sempre necessitam da assistência de um especialista. Quando a máquina do usuário não se conecta a um servidor de arquivos, o usuário necessita de alguém para ajudá-lo. Ele então se comunica com o setor responsável e, mesmo que o problema não seja reportado a uma central de atendimento ou a um sistema de *help desk*, o atendimento é feito, o problema é resolvido e o registro feito dentro das experiências do especialista que o atendeu. Nas próximas chamadas, é mais do que normal que o usuário se refira diretamente a quem o atendeu por último, não importando o nível de seus problemas e, se o especialista não puder ajudá-lo, será causa de frustrações.

A garantia de qualidade profissional nos problemas de rede requer algum tipo de sistema descobridor de problema, chamados de sistemas de *trouble ticket.* Sistemas básicos deste tipo agem como uma lista de hospital, coordenando o trabalho de muitas pessoas, as quais necessitam atuar no problema. As informações nos *tickets* podem ser usadas para produzir estatísticas. A eficiência de operação pode ser melhorada e se tornar mais rápida se incluídas informações da rede nos *trouble tickets* automaticamente. Também podem ser utilizados os sistemas de alerta para monitorar o progresso dos *trouble tickets*, que podem ser úteis para ajudar na comunicação entre

administrações de redes, bem como com outros setores da empresa, como o pessoal de vendas ou compras.

Um sistema de *trouble tickets* deve garantir muitos propósitos:

Uma lembrança de casos de problemas

Deve ser o primeiro objetivo deste tipo de sistema. O *ticket* deve registrar o problema por completo, a fim de que qualquer operador possa extrair dele várias informações e que o mesmo, possa servir de ponto inicial para problemas mais complexos ou semelhantes.

Listagem e transmissão de trabalhos

O STT deve prever e atender a prioridade de problemas simultâneos. Principalmente, para problemas que são lançados aos operadores em tempo real, estes já deveriam vir ordenados por prioridade, não esquecendo de acrescentar prioridades a problemas de dias anteriores.

Segurança no descuido do engenheiro e responsáveis pelo cliente

Se algum problema está em aberto e repetidamente encontra-se no sistema, deve-se prover um mecanismo que envie avisos para diversos engenheiros ou responsáveis pela parte onde o problema se encontra pendente, bem como prover mais detalhes através dos diversos registros do problema.

As informações em um *trouble ticket* podem ser disponibilizadas em campos livres e/ou campos indexados, os quais serviriam para verificação de dados ou entradas no sistema. Este tipo de campo funciona perfeitamente para ambientes uniformes, bem entendidos e estabelecidos, mas para STT muito grandes, os campos são costurados para o problema específico. Se um operador for tentar colocá-los indexados, adicionando estruturas e validando-os, ele certamente se sentirá muito confuso. Este tipo de tratamento acarreta manutenções mais seguidas e adições de novas entradas, bem como podem levar a forçar autorizações que pelo funcionamento não seriam corretas.

Quando relatos estatísticos de operação são o propósito principal para o sistema, diversos campos indexados são apropriados; se, no entanto, o principal objetivo do sistema for manter registros individuais de problemas e facilitar a comunicação entre operadores, os campos indexados devem ser evitados.

O relacionamento entre a estrutura do *ticket* e o problema proposto é muito útil se for possível identificar diferentes *tickets* para classes de problemas diferentes. Este ponto é importante para NOC's que trabalham com várias funções como por exemplo: manutenção de *mainframe*, operação de *workstation*, procedimentos de segurança, funções de *help desk* ou qualquer outra operação de resposta em tempo real. Portanto, seja qual for o STT adquirido ele necessariamente deverá ser adaptado aos padrões de trabalho do NOC que o adquiriu.

4.5.1 Estrutura de um Trouble Ticket

A estrutura de um *trouble ticket* assemelha-se a padronização de um *frame*, como pode ser observado na figura 4.1, deve possuir três divisões essenciais:

- cabeçalho – um campo de indexação;
- descrição do problema – inclui desde o relato do problema feito pelo requerente, onde o problema não estará claramente definido, passando por atualizações de cada um dos operadores que forem verificar ou contribuir para a solução do problema, todos os relatos atualizados por datas;
- dados da solução – uma vez resolvido o problema, é útil que se faça um resumo dos procedimentos adotados para futuras análises.

Ticket: A923	
Entrada	
Data: 10/12/98	Hora: 10.28
Fechamento	
Data: 10/12/98	Hora: 11:02
Recebido por: Jaina	
Chamado por: Carlos	
Fone: 1215	
Email: carlos@udesc.br	
Transmitido para: Ana	
Descrição: Jaina informa que sua máquina ao ligar está indicando a seguinte mensagem: "Servidor não encontrado: Verifique o cabo ou a placa de rede" Ela não esta conseguindo dar *logging* na rede.	
Status: Comunicado para Ana	
Atualizações: Após verificação feita em outro computador se constatou que o servidor ao qual Jaina estava tentando se ligar encontrava-se travado. Foi então feito o procedimento de recolocá-lo no ar e comunicado ao usuário que dentro de 20 minutos ela poderá ligar seu micro que este poderá entrar na rede.	
Soluções: Desligar o micro do usuário e executar procedimentos de *reboot* do servidor.	

FIGURA 4.1 – Exemplo de um *trouble ticket*.

Um STT, para ser bem sucedido, deve atender o usuário, o problema e a equipe de redes. Os dados são úteis para a procura com entradas de formato consistente. Este tipo de sistema necessita ajudar os operadores a preencher corretamente os campos. Deve, ainda, ter integração com a maioria das ferramentas utilizadas dentro do centro de operações de redes, onde deve ser incluído um ambiente de janelas de operação, um sistema de monitoração de alerta, conexão com banco de dados, coleção de dados sobre máquinas na rede, *e-mail*, passar problemas adiante e notificar o sistema, comunicação com outros STT, acesso à rede, *interface* para sub-rotinas (criação de subrotinas que

podem ser adicionadas no sistema futuramente), sistemas especialistas, além de privacidade e segurança, arquivamento de antigos casos e sistemas de *back-up*.

Uma rede bem definida e planejada terá menos problemas e permitirá mais tempo para se trabalhar na sua expansão e melhoramento.

4.5.2 Estabelecendo um Sistema de Trouble Ticket para Redes

Quando uma empresa possui um serviço de atendimento ao usuário (serviço de *help desk)*, as considerações devem ser voltadas para o ponto de formalidade que vai ser adotado. Se a rede é complexa e grande deveria ter pessoas fixas no atendimento; caso contrário, isto poderia ser trocado por um telefone móvel ou revezamento de pessoas no atendimento. As pessoas que estariam no atendimento deveriam possuir uma lista das perguntas e problemas mais comuns; podendo ser livros ou até mesmo páginas em um servidor Web, as quais poderiam ser atualizadas facilmente, além de mostrar diagramas, contendo múltiplas fontes.

Um dos maiores benefícios da implantação de um serviço de atendimento (*help desk)* é o de se estabelecer uma habilidade de tratar os problemas com mais detalhes, conseqüentemente com maior rapidez. Estas informações são irrelevantes quando se trata de estimar o custo de uma rede ou quando se analisa os padrões de problemas ao longo do tempo, mas muito útil no caso de se estabelecer um sistema de *trouble ticket*.

Finalmente, um atendimento técnico deve ter um claro conjunto de processos para incluir chamados e dar-lhes os devidos destinos dentro da equipe técnica e de apoio. Uma maneira eficiente de estabelecer estes procedimentos é a montagem de um sistema de *trouble tickets* ou sistema de *help desk*.

Assim, quando um usuário, ou o agente - no caso de gerenciamento de redes- reporta um problema para a central de atendimento, um caso é registrado em formato de *ticket* (segue um padrão de um *frame,* figura 4.1), o qual recebe um índice. Este caso, para estar correto, deve documentar o problema, informando quem é o usuário solicitante, quando o problema ocorreu e quem será destinado para verificar o problema. Todas as pessoas que trabalharem com o referido problema devem adicionar informações que consigam coletar, bem como o trabalho desenvolvido para resolver o problema. Ao se solucionar o problema, o *ticket* deve ser atualizado com a resolução final, bem como o tempo gasto para se chegar a ela e o usuário comunicado do final do problema. Feito isto, o *ticket* pode ser armazenado para futuras consultas, estatísticas, anotações.

Para qualquer problema encontrado, deve-se abrir um *ticket.* Isto permite a geração de estatísticas para análises posteriores, como no caso de se ter problemas mais complexos.

Ao se procurar por sistemas de *trouble tickets* deve-se observar as seguintes características:

- facilidade de uso;
- armazenamento permanente;

- acesso *on-line*;
- provimento de estatísticas;
- reaproveitamento de referências de problemas e soluções de um usuário para outro com problemas semelhantes.

É importante lembrar que um STT não pode nunca ser utilizado para medir eficiência e serviço de funcionários; o único objetivo deve ser o de resolver problemas de redes, bem como ver a direção dela, não dos empregados. Portanto, é preciso cuidado na emissão de estatísticas e seus propósitos.

A construção deste tipo de sistema envolve a escolha de uma ferramenta adequada, pois de sua flexibilidade depende o correto raciocínio sobre os casos (problemas) e, conseqüentemente, a qualidade do STT ou SHD para redes.

4.6 Tendências para Soluções ao Gerenciamento

Existe, à algum tempo, a tendência do uso de sistemas independentes para gerência de redes heterogêneas. Esta tendência vem acrescentando às tarefas de gerenciamento, procedimentos de inteligência artificial, gerando, assim, sistemas especialistas de apoio à gerência de redes.

Rede de computadores transformou-se em um elemento-chave para as organizações. Compreende um conjunto de diversas informações que usam inúmeros meios para se propagar, acompanhada de uma mistura de fornecedores de equipamentos e prestadores de serviços, distribuídos geograficamente e utilizando-se de diversos protocolos de comunicação. Desta maneira, os gerentes de redes estão optando por utilizar-se de ferramentas desenvolvidas a partir de uma técnica promissora, ou seja, o uso de técnicas de sistemas especialistas, para resolverem seus problemas.

O processo de informar as falhas ocorridas pode ser organizado através de (1) chamadas ao *Help Desk* (HD) ou alarmes da estação gerente da rede; (2) geração de *trouble ticket* (TT) da aplicação, que fornece informações iniciais; (3) passagens dos problemas para pessoas mais especializadas e (4) testes na rede, que podem ser feitos usando-se *probes* ou outras ferramentas.

O uso desta técnica em gerenciamento de redes traz alguns benefícios como os citados por Terplan [TER 87]:

- velocidade na determinação do problema;
- decisão baseada em uma base de conhecimento intensiva;
- decisão baseada em uma quantidade de dados intensiva;
- segurança;
- uso de pequeno número de pessoal;
- estabilidade;

- dependência decrescente de pessoal específico;
- flexibilidade;
- integração de ferramentas;
- não interpretação humana das regras operacionais.

O gerenciamento de rede deve monitorar, coordenar e controlar os recursos da rede de forma eficaz e eficiente, para suprir as operações realizadas sobre ela. Deve, também, envolver análise e planejamento de atividades para alcançar e manter níveis de desempenho aceitáveis.

A utilização de um Sistema de Registro de Problemas (*trouble ticket system)* auxilia os controladores de redes a diagnosticar problemas e permite criar bancos de dados de experiências com problemas, viabilizando a utilização de sistemas especialistas na solução dos problemas. É importante sempre lembrar de atualizar as respostas no TT antes de dar a tarefa como encerrada.

A integração de STT com sistemas especialistas pode prover soluções a problemas complexos, sendo capaz de substituir um especialista humano. Estes sistemas são denominados de sistemas que se utilizam da metodologia de sistemas especialistas baseados em conhecimento ou *Knowledge-Based Expert System* (KBES).

As técnicas identificadas nas metodologias de sistemas especialistas que têm importância para o gerenciamento de redes incluem:

- representação e aplicação de heurística para a solução de problemas matematicamente intratáveis, assim como, de problemas associados a listas ou sistemas de configuração;
- raciocínio na ausência da informação completa ou na presença de resultados ambíguos ou dados incorretos;
- automatização de idéias: replicação de situações e restrições, que são típicas de processos padrões de operações desenvolvidos nos cenários de emergência.
- ferramentas desenvolvidas por sistemas especialistas: criam um ambiente que incluí uma poderosa linguagem de representação e uma eficiente engenharia de inferência. Propiciam, também, um ambiente que facilita a sua comunicação com outros *softwares* e com os usuários.

Um sistema especialista desenvolvido para o gerenciamento de redes deve adicionar ou aumentar funcionalidades de gerenciamento de rede, provendo informações rapidamente para auxiliar a decisão do usuário do sistema e satisfazer todas as responsabilidades da rede.

Visando a este objetivo, diversas pesquisas têm sido desenvolvidas em torno de gerências de redes de computadores para que sejam produzidas ferramentas que, aliadas às facilidades permitidas por sistemas especialistas [KOL 93], forneçam auxílio e

rapidez aos gerentes e usuários das redes. A otimização das respostas sistêmicas fornecidas por eles, é traduzida na agilidade e precisão nos diagnósticos de falhas; já os recursos humanos podem utilizá-los para treinamentos ou para que operadores pouco experientes possam realizar tarefas sem o comprometimento da segurança dos sistemas.

Os sistemas especialistas construídos utilizando-se de *Case Based-Reasoning* (CBR) [KOL 93] são capazes de realizar este tipo de procedimento, ficando, portanto, evidentes as vantagens de sua utilização para emular o raciocínio de um especialista.

4.7 Sistemas Especialistas para o Gerenciamento de Segurança

O gerenciamento de falhas é a área de gerenciamento de redes que mais tem recebido ferramentas que utilizam a metodologia de sistemas especialistas, como pode ser percebido em Lundy Lewis [LEW 95].

Atualmente, a área de gerenciamento de segurança vem atraindo bastante atenção e sendo base para diversos trabalhos que envolvam as áreas de rede de computadores com IA. Este tipo de gerenciamento tem suas informações utilizadas para as seguintes atividades corretivas, que vêm a ser as próprias funções do gerenciamento de segurança:

- monitoração - constante análise sobre o comportamento da rede;
- diagnóstico - inventário sobre as informações da rede, obtenção de um retrato instantâneo da rede onde apareçam dados com níveis diferentes dos estipulados para um correto funcionamento da rede;
- detecção - determinação das informações de um problema, como: onde ele está ocorrendo, com que intensidade e que pontos da rede ou sistema está comprometendo;
- correção de falhas de segurança - parte final do problema, é a determinação da causa provável e recomendações para reparos, conseguidas através da provisão de informações detalhadas para análise de características, muito bem suportadas por ferramentas e procedimentos.

Assim, para facilitar o serviço de gerenciamento de segurança, pode-se priorizar tarefas como [TAR 90]:

- recepção de indicadores de falhas geradas automaticamente;
- reconhecimento de condições anormais por dedução devido ao acúmulo de relatos de erros;
- permanente localização, identificação e registro de anomalias;
- monitoração de desempenho através de dados históricos e correntes;
- diagnóstico para alarmes recebidos;
- rastreamento do tráfego para priorizar o início de atividades aconselháveis.

Os sistemas especialistas tem proporcionado uma automatização nos tipos de análises complexas. As técnicas que eles incluem compreendem: inferência de causas para conhecer efeitos (*backward chaining*), inferência de respostas para conhecer as ações (*forward chaining*), um mecanismo de fusão de dados (*blackboards*) e automação de processos de diagnósticos que precisam de uma certa quantidade de passos realizados para chegar a um resultado imediato (*data-driven-rule sets*).

Desta forma, as atividades corretivas do gerenciamento de segurança recebem algumas alterações nos procedimentos para chegar à solução dos problemas encontrados, conforme pode ser observado na figura 4.2.

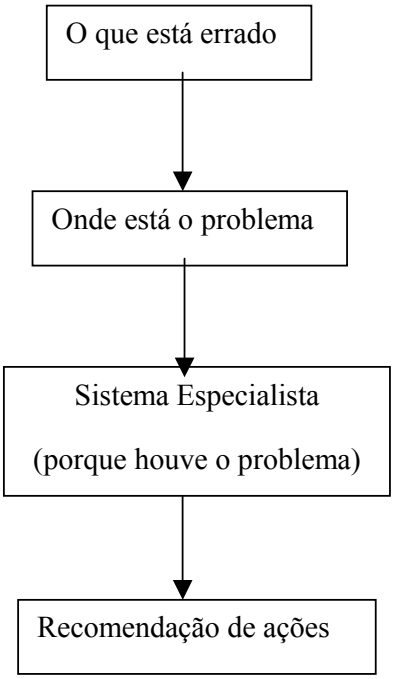


FIGURA 4.2 - Procedimentos para solução de problemas com uso de Sistema Especialista.

5 Segurança em Redes TCP/IP

O perfeito nunca é constante, por isso todas as coisas têm um tempo finito de vida, também acontecendo com o funcionamento de uma rede. Seria muito bom se tudo sempre estivesse funcionando perfeitamente, mas acontecem falhas em equipamentos e processos ou problemas provocados por intrusos.

O gerenciamento de redes, portanto, sempre envolve dois tipos de componentes; dependendo de como é realizado o primeiro, o segundo será ou não necessário:

- prevenção de problemas;
- procura por solução de problemas (*troubleshooting*).

A prevenção de problemas é considerada como o primeiro passo de um administrador de rede depois que ela se encontra funcionando. Como os problemas que podem ocorrer são diversos, envolvendo inúmeros equipamentos e das formas mais inconstantes, não há como planejar uma prevenção que cubra todos os tipos, mas pode-se trabalhar em relação aos mais prováveis.

Já a solução de problemas para redes pode ser iniciada a partir do momento em que se faz uma análise do perfil da rede, onde se pode identificar mudanças que poderiam ultrapassar limites pré-estabelecidos e, desta forma, comprometer o funcionamento da mesma.

A quebra de segurança em uma rede pode tornar-se uma monstruosa dor de cabeça para o administrador do sistema; portanto, é necessário que este tome as devidas providências para que invasões ou ataques de *crackers* não ocorram, quando do contrário, podem causar destruição, interrupção, corrupção e divulgação de dados.

Problemas com segurança não compreendem apenas ataques na infra-estrutura da rede mas, também, em seus serviços, como no caso de servidores Web e de *e-mail*, onde o que poderá ser violado são os dados e informações contidos nas mensagens que se encontram trafegando.

Ao se tratar com problemas de segurança é importante levar em conta que não há como prover segurança sem que seja possível a detecção da ocorrência ou não de um ataque à rede ou sistema. Desta forma, a preparação da rede para coletar tais informações é essencial. A preparação envolve usuários e ferramentas para detecção de intrusão, políticas de segurança, processos de suporte, rede e sistemas.

5.1 Identificação

Hoje em dia, existe uma generalização que confunde os conceitos de termos de segurança; a mesma confusão que existe em torno de *hacker* e *cracker* ocorre com ataque e invasão. No ataque, não se obtém acesso à maquina; consegue-se apenas tirá-la do ar ou prejudicar algum serviço remotamente. Já a invasão ocorre quando o *cracker* consegue acesso à máquina como superusuário.

Intrusos podem estar em locais internos ou externos à rede, estão sempre procurando por aberturas na rede ou sistemas que possam lhes fornecer um acesso às informações e recursos. Para conseguir isto, fazem uso de ferramentas e recursos diversos que podem explorar vulnerabilidades em operações de sistemas, infra-estrutura de rede ou protocolos.

Se uma rede ou sistema não está adequadamente preparado para detectar sinais de intrusão, será quase impossível determinar se houve um comprometimento dos sistemas; não havendo níveis para comparar, não há o que avaliar.

Segundo o CERT [CER 99], prover preparação para detecção de intrusão requer dois procedimentos fundamentais:

- definir níveis necessários para alcançar os objetivos de trabalho;
- implementar passos que treinam a equipe e sistemas na tarefa de detectar sinais de intrusão ou invasão. Como exemplo pode-se citar: mecanismos de *loging* e ferramentas detectoras de invasão.

Já o trabalho de detecção inclui:

observar a rede, sistemas e usuários para apurar qualquer atividade incomum;

investigar qualquer atividade incomum;

iniciar uma reação à intrusão, desde que as atividades incomuns não sejam justificadas por nenhum procedimento autorizado.

Estes procedimentos requerem atenção especial sobre cinco áreas, que são:

- integridade do software usado para detecção de intrusão, se for o caso;
- integridade contida no sistema de arquivos e arquivos de dados;
- operação e tráfego sobre a rede;
- formas físicas de intrusão nos computadores da rede;
- armazenamento de *backup*;
- investigação de ações incomuns realizadas por usuários da rede.

Mesmo que um administrador de rede avalie cada segmento da mesma e seus riscos potenciais e, em seguida, execute etapas para minimizar, eficazmente, tais riscos, o grande problema para se chegar à solução ainda estará centrado no entendimento de qual problema está realmente acontecendo. Após o encontro da causa real do problema, providenciar a solução pode ser uma tarefa corriqueira.

A obtenção de um retrato da rede no momento do problema, proporcionado pela constante monitoração sobre os dados fornecidos por ela, pode, portanto, traduzir a identificação de causas levando à diferenciação entre ser uma simples falha e um ataque real.

Para que seja possível delimitar o problema que está ocorrendo, alguns pontos devem sempre ser pesquisados ao se tratar de suspeitas de intrusão, como:

- que evidências há no sistema de que houve um ataque? Verificar se o problema está caracterizando um ataque propriamente dito ou um erro no funcionamento da rede;

- a origem do problema, o número IP ou o nome da máquina de onde partiu o registro de erro;

- o resultado do problema, as dimensões sobre quais o problema está limitado e se ele está afetando alguma máquina em específico ou a rede como um todo;

- informações dos *logs* dos sistemas, registrando os comandos que foram executados e quem sofreu alterações com estes, e os processos que estão sendo executados;

- os usuários que tiveram acesso à rede são validados? Especialmente os que executaram procedimentos no horário em que se registrou o início do problema;

- o nível e o tipo de tráfego que está passando pela rede. A determinação do tipo de tráfego na rede no momento ou depois da detecção do problema pode ser decisiva para determinação do tipo, em caso de ataque;

- os tipos de erros que o sistema está emitindo. Muitas vezes o sistema consegue coletar importantes informações sobre o que está acontecendo;

- houve alteração nos tipos e números de programas já instalados? A instalação de algum programa pode caracterizar que algum tipo de vulnerabilidade possa ter sido instalada;

- há algum arquivo desconhecido ou alterado em um lugar suspeito? Muitas vezes, os atacantes instalam arquivos de senhas no sistema, ou arquivos que possam desencadear processos que quebrem a segurança;

- houve a criação de alguma conta suspeita para usuário? O intruso pode garantir seu retorno ao sistema através da criação de uma autorização de entrada, ou seja, uma conta legítima do sistema.

Como descrito anteriormente, vulnerabilidades de segurança podem caracterizar problemas em três aspectos: de conectividade, de meio físico e de configuração.

5.2 Formas e Tipos de Ataques

Suspeitas de intrusão ou ataque, às vezes, podem caracterizar, na verdade, uma falha de funcionamento da rede escondida em algumas de suas camadas. Portanto, a seguir, são descritos problemas que, se corretamente analisados, compreenderão problemas com segurança. O levantamento é baseado em estatísticas e informações

fornecidas pelos órgãos de segurança como CERT, CAIS e NIC, pelo cotidiano de um centro de operações de redes e pela literatura em geral.

Segundo Neumann [NEU 89], existem quatro tipos de comportamentos para invadir uma rede ou sistemas, que são:

ataque simples - o intruso usa uma máquina na rede para acessar outra;

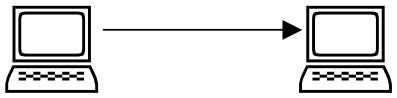


FIGURA 5.1- Ataque simples

ataque *doorknob* - as tentativas partem de uma única máquina para acessar outras;

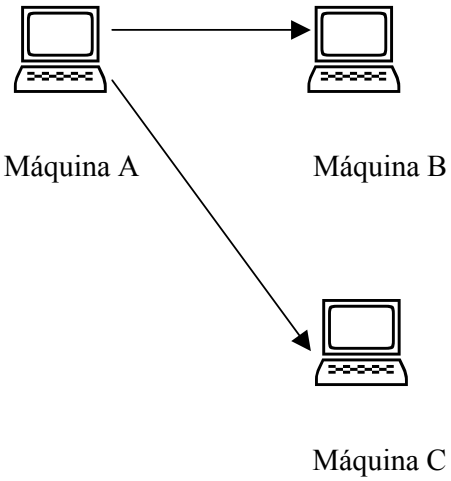


FIGURA 5.2 - Ataque *doorknob*.

ataque em cadeia - o intruso passa por diversas máquinas para esconder sua origem;

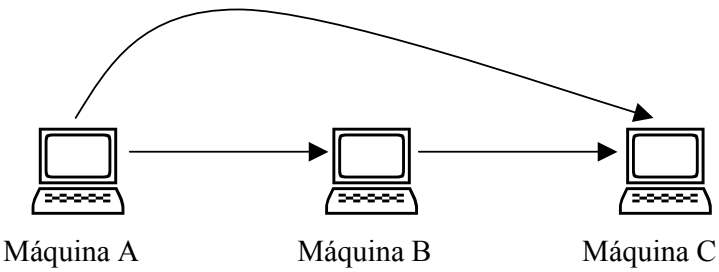


FIGURA 5.3- Ataque em cadeia.

ataque em *looping* - a máquina atacante é a mesma atacada.

Máquina A Máquina A

FIGURA 5.4 - Ataque *looping*.

Ataques à segurança de um sistema de computador ou rede são caracterizados pelas funções que desempenham, onde o fluxo da informação vai de uma origem para um destino. Os ataques exploram este tipo de comunicação, categorizando-se, segundo William Stalling [STA 95], em dois tipos de ameaças, que são:

- as passivas: interceptação de mensagens (obtenção de conteúdo e análise de tráfego). Não envolvem alteração de dados ou protocolos;

- as ativas: envolvem modificações nos dados originais ou criação de dados falsos.

Estes dois tipos de ameaças envolvem, segundo o mesmo autor, quatro classes de ataques, que são:

Interrupção – compreende um ataque nas funções disponibilizadas por um sistema de computador ou rede, resultando na interrupção do funcionamento. Como exemplo, pode-se citar: a destruição de uma parte física ou desabilitação dos sistemas de gerenciamento de uma rede.

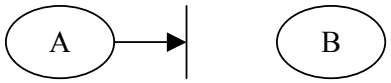


FIGURA 5.5 – Interrupção.

Modificação – é um ataque na integridade de um sistema ou rede. Um acesso não autorizado é conseguido e resulta em uma falsificação ou adulteração dos dados que trafegam ou depositados em uma rede ou sistema.

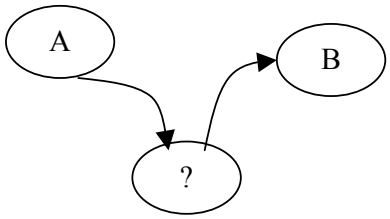


FIGURA 5.6 – Modificação.

Interceptação – consiste de um acesso a dados ou componentes de uma rede ou sistema, sem haver autorização necessária para esta tarefa. Atinge o sigilo de dados, ou seja, a confidência destes. Podem ser exemplificados por: a cópia indevida de dados ou programas e a captura de dados transmitidos por uma rede.

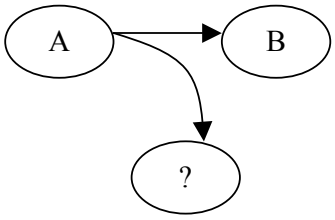


FIGURA 5.7 – Interceptação.

Fabricação – abrange acessos não autorizados, resultando na inclusão de objetos em um sistema; é um ataque de autenticidade. Como exemplo, pode-se citar: adição de registros em arquivos ou banco de dados e inclusão de mensagens falsas na rede.

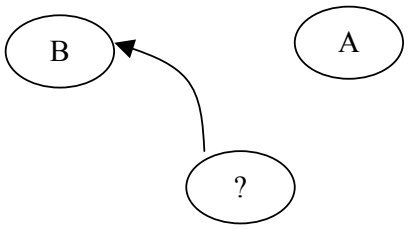


FIGURA 5.8 – Fabricação.

5.3 Localização do Ataque

Ataques seguem basicamente as mesmas técnicas para se desenvolver, as quais se caracterizam pela ausência do intruso dentro da rede, ou seja, na maioria das vezes, ele está fisicamente distante do sistema sob ataque, fazendo uso de algum ponto de rede durante sua tentativa. Mas é importante não se descuidar do fato de que o intruso possa estar dentro da própria rede; sendo assim, ele teria acesso direto aos componentes da rede e, até mesmo, acesso físico às máquinas.

"Um ataque ou uma vulnerabilidade de segurança não podem estar ocorrendo na camada física da rede, este tipo de problema caracteriza-se como problema de falha nos componentes de rede".

Esta é a forma como usuários e técnicos de rede aprenderam a pensar esquecendo-se de que um ataque pode, também, ser procedido *in loco,* ou seja, na própria máquina como no caso de: retirada de cabos de rede, *flat cable* ou periférico, alteração de mapa de caracteres de teclado, dentre outros.

5.4 Técnicas mais Utilizadas

Todo dia, novas formas de ataques são descobertas; muitas vezes, o seu conhecimento vem acompanhado de alguns estragos. A Internet favorece o uso de ferramentas por parte de *crackers*, tendo em vista o uso de portas para realizar seus

serviços. Estas portas geram caminhos abertos para ataques e invasões. Assim, torna-se essencial que um administrador de rede saiba como identificar os serviços e as portas que estão abertas em seus sistemas.

O trabalho de um *cracker*, ao tentar invadir uma máquina, consiste em saber quais portas estão em *listen*, ou seja, aceitam conexões de um IP externo, e, posteriormente, quais são as versões dos programas utilizados. Em seguida, ele sai em busca de ferramentas de ataque, muitas delas disponíveis na Internet.

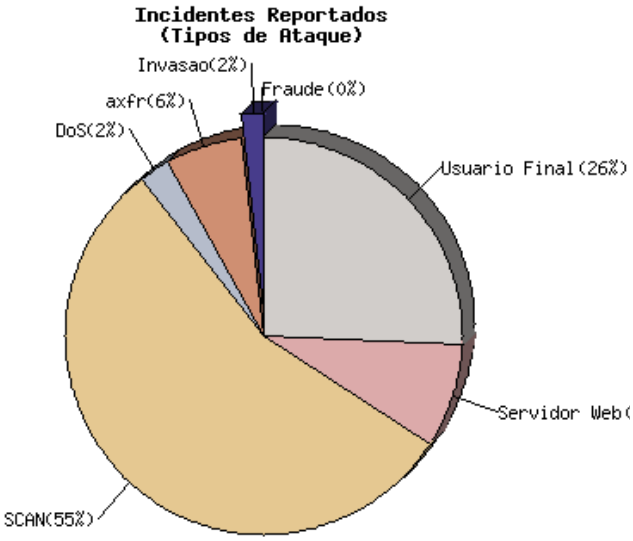
Em uma rede, as portas TCP são pré-estabelecidas para a comunicação entre um serviço e seu *browser*. Na tabela 5.1 pode-se verificar as principais portas TCP e o respectivo serviço associado a elas.

TABELA 5.1 - Tabela de portas TCP e respectivo serviço.

Porta	**Serviço**
7	*Echo*
20	ftp-*data*
21	ftp
22	Ssh
23	*Telnet*
25	Smtp
53	*Domain*
80	www
110	Pop3
389	Idap
443	https
3306	MySQL

Segundo o órgão de segurança NIC-BR [NIC 2000a], alguns tipos de ataques se proliferam mais do que outros. O gráfico 5.1, produzido pelos relatos deste órgão, compreende os tipos de ataques mais executados no Brasil no período de janeiro a junho de 2000.

Vários ataques foram criados explorando as deficiências existentes na arquitetura TCP/IP. O grande problema é que, nesta arquitetura, a maioria dos serviços e protocolos usa endereços IP como base para autenticação. Entre estes serviços e protocolos, pode-se citar o NFS (*Network File System*) e o DNS (*Domain Name System*), por exemplo. Pacotes IP podem ser forjados, ou seja, o cabeçalho IP pode ser alterado da forma que o atacante queira (como as modificações do endereço IP da origem do pacote, por exemplo). Desta forma, é possível para um atacante personificar qualquer máquina na rede (ataque por IP *Spoofing*, descrito a seguir). Somam-se, a estes ataques, outros que usam *bugs* de segurança nas implementações dos serviços e protocolos TCP/IP e falhas nos *softwares* em geral.



Legenda:

Axfr: tentativa de obter/atualizar mapas de DNS.
DoS: Denial of Service.

GRÁFICO 5.1 – Incidentes Reportados para o NIC-BR de janeiro a junho de 2000.

A seguir, é feita uma descrição do que são e como acontecem estes tipos de ataques, bem como dos mais freqüentemente utilizados na Internet. A descrição tem por objetivo mostrar suas execuções sob o aspecto técnico, para que administradores de rede dêem atenção a tais pontos de funcionamento da rede.

5.4.1 Scan

São técnicas que têm por objetivo procurar serviços e falhas que possam comprometer uma máquina. Esta é a primeira etapa de um ataque. Assim, um atacante pode coletar todas as informações possíveis sobre o seu alvo e, conseqüentemente, descobrir falhas na sua segurança.

5.4.2 AW (Ataque a Servidor Web)

Um servidor Web foi projetado basicamente para atender a pedidos anônimos, de usuários não autenticados, a fim de disponibilizar as informações requisitadas de uma maneira eficiente. Por outro lado, servidores Web, por natureza, são programas bastante complexos e, em muitos casos, o código fonte destes programas é amplamente divulgado. Conseqüentemente, tornam-se susceptíveis de serem analisados em busca de vulnerabilidades que possam ser exploradas, especialmente quando se utilizam de algumas funcionalidades do serviço Web, como as oferecidas pelo CGI ou Java *Script.*

5.4.3 Bug de Software

São brechas em um determinado software, criadas pelos próprios programadores. Normalmente, elas são elaboradas para análise do código ou, em alguns casos, os desenvolvedores simplesmente as esquecem no programa.

5.4.4 Backdoors

Um *backdoor* é um meio ou uma porta escondida que possibilita, a um *cracker*, retornar à máquina invadida posteriormente quantas vezes desejar. Um bom *backdoor* que satisfaça a maioria dos *crackers* deve permitir acesso à máquina atacada nas seguintes situações:

- mesmo se o administrador tentar torná-la segura mudando senhas;
- com a menor visibilidade possível; a maioria dos *backdoors* provê alguma forma de evitar que sejam gerados *logs* do acesso e, em muitos casos, a máquina não mostra se há alguém conectado nela durante o acesso do invasor;
- da forma mais rápida possível, sem repetir explorações de falhas de segurança.

A existência de um *backdoor* possibilita o mascaramento de processos, os quais tornam as atividades de ataque não percebíveis pelos administradores de rede como no caso do estabelecimento de *shell* TCP ou ICMP.

5.4.5 Vírus

Os vírus são programas que interferem no fluxo normal de processamento de um programa, eles mudando arquivos e configurações. Sempre necessitam de uma ação para dispará-los e depois eles mesmo ativam o que necessitam de si próprios.

Geralmente penetram na rede através de *softwares* piratas, *downloads* de *shareware* ou outro *software* suspeito. Sua instalação pode originar *backdoors*, como descrito na seção 5.4.3 .

5.4.6 Denial of Service (DoS)

O ataque de negação de serviço ocorre quando um *cracker* consome todos os recursos da máquina, impossibilitando que outros usuários possam utilizá-la. O resultado é a degradação ou perda de serviço.

5.4.7 Distributed Denial of Service (DDoS)

Apresenta a mesma característica do DoS, mas é mais eficiente. O DDoS utiliza várias máquinas no ataque, causando um fluxo de requisições bem maior. Este ano, ataques deste tipo tiraram do ar os *sites* como a CNN, o Yahoo e, no Brasil, o UOL.

5.4.8 Spoofing

O *Spoofing* é considerado como uma tentativa de invasão; não é uma técnica de ataque pois consiste, basicamente, do envio de um pacote comum disfarçando o IP origem real. Para produzir alguma ação, transformando-se em um ataque, necessita ser acompanhado de outras técnicas, como por exemplo DoS ou *SYN Flood*.

5.4.9 Sniffers

Sniffers ou farejadores são técnicas que exploram o fato de o tráfego de pacotes das aplicações TCP/IP não usar nenhum tipo de cifragem nos dados. Captura, assim, todos os dados que estiverem trafegando na rede, permitindo que senhas sejam obtidas facilmente.

5.4.10 Buffer Overflow

É uma técnica que se utiliza de códigos que tenham problemas em sua programação. O princípio é sobrescrever partes da memória que supostamente o programa não deveria acessar, fazendo com que o código sobrescrito possa executar um comando arbitrariamente.

5.5 Protocolo, Infra-estrutura, Serviço e Procedimento Utilizado para Ataque

Como comentado na seção 5.4 os serviços, estruturas e protocolos, juntamente com as portas que utilizam, ajudam a criar ambientes propícios a ataques e invasões. Todo o ataque tem como objetivo deixar um caminho livre para procedimentos a pessoas não autorizadas, quantas vezes forem desejadas. Uma invasão caracteriza-se por tentar fixar *backdoors* nos sistemas e *host* invadidos.

Assim, nesta seção, descreve-se em detalhes o funcionamento dos principais serviços utilizados por um acesso à Internet, centrando-se na forma como problemas de segurança se desenvolvem por meio deles.

5.5.1 ICMP (Internet Control Message Protocol)

O protocolo IP não é orientado à conexão e não é confiável. Desta forma, muitos erros podem ser reportados neste nível através do que pode ser colhido pelo protocolo ICMP da mesma camada. Assim, este protocolo é uma maneira eficiente de se coletar dados de falhas.

O ICMP está definido na RFC 792 e faz parte da camada TCP *(Transfer Control Protocol)* no nível de rede. Este protocolo faz uso de datagrama IP para o envio de mensagens de controle de fluxo, sendo que estas compreendem dois tipos: mensagens de solicitação (*query*) e mensagens de erro.

Ele tem como objetivo dar um *feedback* sobre os problemas no ambiente de comunicação. Permitindo que *gateways* enviem mensagens de erro ou de controle para outros *gateways* ou máquinas, provê comunicação entre o protocolo Internet (IP) em uma máquina e o IP em outra, mas não ajuda a localizar onde está o erro, pois ele responde apenas para a máquina que originou o pacote errôneo, e o erro pode estar em algum *gateway* no caminho, o que pode ocasionar perda de outros pacotes IP.

Tanto o protocolo ICMP como o UDP (*User Datagram Protocol*), permitem a utilização de endereços falsos de origem do pacote. Assim, o atacante poderá enviar mensagens que inviabilizem operações de uma rede ou respondam indevidamente para outras redes.

Pode-se produzir *floods*, através de pacotes ICMP (*echo*), pelo emprego de *ping,* resultando na saturação de um ou mais *links* de um *backbone*, incluindo impressoras de rede. São os chamados *big ping*.

Pacotes ICMP podem ser transmitidos, também através do *broadcast* de roteadores e *gateways* de suas redes internas, fazendo com que ocorra uma saturação de mensagens colocando os equipamentos fora de serviço. Esta técnica de ataque é denominada *Smurf* [CER 98].

5.5.2 TCP (*Transfer Control Protocol)*

TCP é um protocolo orientado à conexão com detecção e correção de erros fim-a-fim. Utiliza-se de número nas mensagens, janelas deslizantes, temporização e controle de fluxo para garantir confiabilidade. Assim como o protocolo UDP, o TCP passa dados entre as camadas de aplicação e Internet e utiliza-se de portas de protocolos para identificar os processos comunicantes de maneira unívoca.

O TCP trata uma mensagem como um datagrama, o que garante que alguém deverá complementar o gerenciamento do envio de mensagem para garantir a confiabilidade. Então, ele provê uma negociação (*handshake*) entre as extremidades envolvidas para estabelecer um canal lógico de comunicação e o envio de um *checksum* do destinatário, caso o segmento tenha chegado com sucesso e ele esteja aguardando o próximo segmento.

O estabelecimento normal de comunicação TCP, entre dois *hosts*, envolve trocas de *flag*, onde o *host* cliente seleciona e transmite um número de seqüência inicial chamado de *SYN* e o *host* servidor envia a sua seqüência, que é o *SYN ACK*, para o

cliente. Após este recebê-lo, envia sua resposta *ACK*. Está estabelecida uma conexão embrionária remota, que é encerrada pela troca da mensagem *FIN ACK*, que ocorre quando a origem da mensagem não detecta chegada de mais nenhum datagrama. Havendo, durante a comunicação, qualquer *checksum* não recebido pela origem, ela completa a capacidade de seu *buffer* de recebimento, espera um determinado tempo e retransmite ou encerra a conexão.

Através do estabelecimento de comunicação entre máquinas remotas, o protocolo TCP pode proporcionar um ambiente de ataque, ou seja, após a resposta de um *SYN,* o computador remoto cria uma conexão embrionária e aguarda um certo intervalo de tempo pela resposta do computador local. Caso isto não ocorra, essa conexão é descartada. Um atacante se aproveita deste tempo para enviar um grande número de pacotes com o *flag SYN* ligado, até estourar o limite de aceitação de novos pedidos e, dependendo do tempo que fica bombardeando o sistema remoto, pode deixá-lo inoperante, como pode ser ilustrado na figura 5.9 caracterizando o chamado ataque de *denial of service SYN Flood*. O atacante, neste caso, para não ser identificado, falsifica o endereço de origem das mensagens, através da técnica de *Spoofing* [BEL 89].

A capacidade de manter conexões pendentes (*backlog*) é limitada nos sistemas. Por exemplo, o sistema Linux suporta até seis *backlog*; já sistemas tipo BSD suportam até cinco *backlog*.

Em 1994, pôde-se sentir mais um problema provocado pela vulnerabilidade do protocolo TCP: o chamado problema de predição TCP ou *Sequence Number Attack*, que já havia sido percebido, anteriormente, por Morris [MOR 85] e Bellovin [BEL 89]. Este ataque se caracteriza por explorar a predicabilidade do *Initial Sequence Number* (ISS), utilizado por um computador durante o *handshake* na seqüência de início de conexão estabelecida através do protocolo TCP.

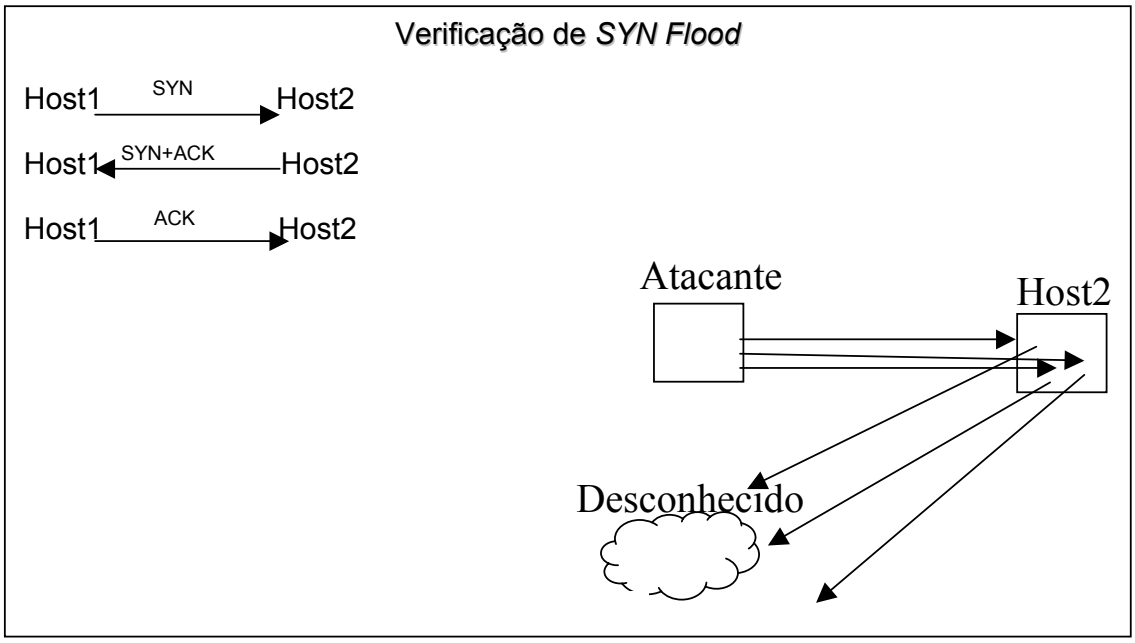


FIGURA 5.9 - Desenvolvimento de um ataque por *Syn Flood*

A determinação do número de seqüência é aleatória, mas algumas implementações do TCP/IP utilizam mecanismos bem previsíveis, o que vem a possibilitar o ataque.

Um intruso percebe uma comunicação entre dois *hosts* (*host1*-origem e *host2*-destino) e inicia uma comunicação com o *host2*, por exemplo através da porta SMTP, assim recebe um ISS para se comunicar, a partir daí pode desenvolver formas para adivinhar o ISS que a outra conexão estaria utilizando, se acertar ele passa a responder para o *host2* como se fosse o *host1*, podendo captar informações e receber os privilégios que seriam de direito do *host1*. Se o ISS for bem adivinhado o *host2* nunca perceberá a interferência, enquanto que, o *host1* receberá, simplesmente, um aviso de desconexão ou um ataque de *denial of service*. O ataque de predição TCP pode também, ser entendido, analisando-se a figura 5.10.

Correto

host1 ----------SYN: ISS1----------→ host2
host1 ←---------SYN+ACK (ISS1) : ISS2 -- host2
host1----------ACK (ISS2) ----------→ host2

Invasor

X ----------SYN: ISSx ----------→ host2

X ←--------SYN+ACK (ISSx) : ISS2' -- host2

X ---------- ACK (ISS2') ----------→ host2

X ---------- RST (ISS2') ----------→ host1

FIGURA 5.10 - Ataque por predição TCP.

Outro ataque muito parecido com o anterior é o chamado TCP *Splicing*, onde o invasor localiza a comunicação entre os dois *hosts* (*host1* e *host2*), monitora tal comunicação e consegue gravar o ISS1 e o ISS2, utilizados durante o *login*. A seguir, providencia alguma forma para o *host1* não se comunicar com *host2* e assume o papel do *host1* com o correto ISS, sem o *host2* tomar conhecimento.

A partir de 1997, muitos administradores de rede passaram a ter muitas preocupações pois, sem nenhuma lógica ou problema aparente, alguns de seus sistemas operacionais abortavam a sua execução e travavam ou, simplesmente, reinicializavam. Foi, então, que se descobriu a existência de uma técnica chamada de *Nuke*, a qual é executada através do envio de determinados *flags* TCP ativos.

5.5.3 UDP (User Datagram Protocol)

UDP é um protocolo não orientado à conexão e não-confiável. Como o protocolo TCP, passa dados entre as camadas de aplicação e Internet e utiliza-se de portas de protocolos para identificar os processos comunicantes.

Trata uma mensagem como um pacote e não carrega nenhuma informação além das necessárias para a entrega da mensagem, que são os endereços das aplicações origem e destino (*ports*). Não fornece nenhum tipo de confirmação de recebimento correto do que foi enviado.

Os serviços que *echo* e *chargen,* utilizados normalmente para testes e depuração, utilizam-se do protocolo UDP. Um invasor pode transformar um pacote UDP em um *small service*, ou seja, através da falsificação dos pacotes (cabeçalho) conectar uma porta *echo* de um equipamento a uma porta *chargen* de outro, provocando, assim, um *flood* no caminho entre ambos.

5.5.4 IP (Internet Protocol)

Pacotes de roteamento que, para transitarem por uma rede, se utilizam do protocolo IP, obedecem à regra de que devem retornar à sua origem utilizando o caminho reverso de rotas indicado pelo protocolo. Assim, no estabelecimento da comunicação entre dois *host*, ocorre um estabelecimento de confiança entre eles. Ataques exploram este tipo de relacionamento, originando a técnica de *Spoofing*, comentada na seção 5.4.8 deste capítulo.

O atacante toma posse do número IP de um dos *host*, desativando-o de alguma forma (*Syn flood, nuke*), enquanto transforma o outro em seu alvo, obtendo relação de confiança com este último. Pode, então, disparar pacotes com informações de *source routing* para o computador alvo o qual pensa estar conversando com o verdadeiro dono do número IP.

Uma solução bem eficiente e simples, para este tipo de problema, é não confiar em autenticações baseadas em endereços.

5.5.5 RIP (Routing Information Protocol)

O protocolo RIP é utilizado pelos *gateways* para troca de informações sobre novas redes e *gateways*. As redes locais geralmente utilizam-se do protocolo nas versões 1 e 2, o qual, não possui qualquer forma de autenticação. Um atacante pode fabricar um pacote RIP, que dita, erroneamente, quem é o *gateway* responsável por uma determinada rede e envia-o aos roteadores de uma organização, enganando-os. Daí em diante, o tráfego dos pacotes destinados a esta rede será enviado ao *gateway* aprendido através do RIP falso.

5.5.6 Serviço de FTP (File Transfer Protocol)

Um servidor FTP localizado atrás de um *firewall* pode funcionar como uma porta de entrada para invasores com destino a um determinado número IP. Através de um serviço de FTP que possua alguma área *incoming* com direitos de escrita aberta, o atacante pode chegar até um alvo, mesmo sem ter acesso direto a ele.

O atacante pode registrar na área de *incoming* um arquivo contendo algum tipo de diálogo de um determinado protocolo. Depois, através de mais um ftp, envia um comando de execução do serviço, o número da porta e um número IP da rede no qual deseja executar as ações de ataque. Assim, o servidor FTP abre uma conexão com o alvo e autentica o atacante, o qual envia um comando que executa o conteúdo do arquivo que anteriormente foi depositado na área de *incoming*. O servidor FTP envia este arquivo para a porta e o computador informados pelo atacante. Este tipo de procedimento contorna regras de acesso direto impostas por *firewalls* e *packet filters*.

5.5.7 DNS (Domain Name Service)

O serviço de DNS pode apresentar uma série de vulnerabilidades, facilitando explorações de invasores.

Um atacante, ao conseguir acesso a um servidor DNS, pode alterar os mapas de nomes do servidor, o que lhe proporciona autenticação como o nome de um computador que tem relação de confiança com um determinado computador alvo. Podem vir a resultar em execuções de várias das técnicas de ataque descritas na seção 4.4, ou ainda, por respostas de servidores falsos de WWW, o que pode ser muito perigoso quando se deseja acessar URL's com máximo sigilo e com senhas, como o caso de *sites* de bancos.

5.5.8 Proxies e Servers

Um computador qualquer em uma rede que possibilite uma autenticação ou acesso livre a um atacante, fornece, a este, condições para que possa instalar um serviço de *proxy*, sem que o administrador da rede tome conhecimento. De posse de um serviço deste nível, um atacante pode penetrar na rede e chegar aos servidores de seu interesse, pois as máquinas internas têm acesso livre a alguns servidores, o que contorna os controles de acesso existentes.

5.5.9 Internet Firewall

Firewalls são sistemas ou programas que barram conexões indesejadas na Internet, por proporcionar um perímetro de segurança bem definido da rede externa, considerada perigosa e "não confiável" [CER 2000]. Este sistema é tido como uma proteção para redes e sistemas, mas, algumas vezes, pode se transformar no meio de acesso necessário aos *crackers*, que se utilizam de portas altas que um servidor *firewall* não consegue filtrar, para entrar nas redes protegidas. Este é o caso de execução na porta SMTP, que é liberada na maioria dos *firewalls*, ou *shell* ICMP, onde o invasor insere dados nos pacotes ICMP do *ping* e faz um túnel de *shell* entre as máquinas, tornando a ação não percebida a menos que se abra o conteúdo dos pacotes *ping*.

Como o serviço DNS pode ser realizado pelo protocolo UDP, alguns *firewalls* são configurados para deixar passar tais pacotes. Assim, o invasor pode colocar o *backdoor* com shell UDP nesta porta e conseguirá atravessar o *firewall*, sem problemas.

5.5.10 SMTP (Simple Mail Transfer Protocol)

O serviço de SMTP é o serviço responsável pelo envio das mensagens de *mail* por um servidor. Este serviço estabelece uma conexão direta entre dois servidores de SMTP: um que enviará a mensagem e outro que a receberá; ou seja, dois servidores são conectados para o mesmo serviço de transporte.

Uma vulnerabilidade de segurança é percebida no fato de que este tipo de serviço não faz autenticação de usuários que estão enviando mensagens. Isto possibilita que um *cracker* dispare procedimentos que inviabilizam operações de SMTP do servidor receptor do *mail*, através do recebimento de uma enxurrada de mensagens.

O único controle que pode ser estabelecido é aceitar o SMTP da rede que está fornecendo o acesso à Internet para o usuário. Isto não impede que um atacante se utilize de diversos endereços de origem e, ao mesmo tempo, diversos servidores de SMTP, para proceder seus ataques, dificultando a filtragem e bloqueio, das origens atacantes.

5.6 Resolução de Problemas

Garfinkel [GAR 94] propõe duas regras básicas e simples para responder a incidentes, que são:

- não entrar em pânico;
- documentar.

As soluções para problemas de segurança, como em qualquer área de problema, são garantidas por atividades pró-ativas e de recuperação (reativas).

Um comportamento pró-ativo para segurança de rede e sistemas deve compreender a determinação dos seus níveis de vulnerabilidades. Desta forma, procura-se pelas correções que se fazem necessárias. Existem várias ferramentas comerciais que auxiliam na varredura e auditoria de sistemas buscando vulnerabilidades, sendo que alguns *patches* de segurança são disponibilizados gratuitamente pelos vendedores, bastando apenas instalá-los.

Outro ponto importante na segurança pró-ativa é a inclusão de linhas de base MD5 [CERT 98] nos sistemas, as quais obrigatoriamente devem ser criadas ANTES que um sistema seja atacado pela primeira vez.

Como surgem novas idéias todos os dias em vários assuntos, o mesmo acontece no campo da segurança, o que possibilita a proliferação rápida de novas vulnerabilidades que, muitas vezes, não são acompanhadas da mesma forma por maneiras preventivas, o melhor remédio, portanto, para uma tecnologia de segurança efetiva é a vigilância.

Quando os únicos procedimentos a serem tomados são os de uma política reativa, as atividade mais adequadas passa a ser resolver o problema de forma a restaurar os serviços dentro de um intervalo de tempo aceitável; caso contrário, o

gerenciamento da rede não está sendo feito de forma eficiente. Em um modelo básico para resolver problemas, devem estar incluídos os seguintes pontos:

- garantir a informação;
- desenvolver um plano de ataque;
- isolar o problema e executar o plano de ataque;
- documentar o que foi feito;

Com o objetivo de aplicar este modelo, alguns procedimentos, como os descritos abaixo, podem ser executados para solucionar furos de segurança:

- estabelecer regras que possam fiscalizar qualquer tipo de erro disparado pelo usuário;
- checar a parte física da rede para saber se tudo está dentro dos padrões de normalidade,
- restringir o tempo para uso da rede;
- requerer trocas freqüentes de senhas;
- requerer senha única para uso na rede;
- manter sempre atualizadas as cópias de dados dos servidores (*back-up*);
- obter informações detalhadas do que acontece, identificando e eliminando suas causas potenciais.

Uma vez que o ataque tenha sido realizado, os caminhos a seguir são únicos, como descritos a seguir e demonstrados na figura 5.11:

- determinar de onde veio o ataque;
- determinar as características do ataque;
- determinar a intensidade do ataque;
- verificar a integridade do sistema em relação as cópias dos originais;
- analisar arquivos de todos os usuários, procurando por conteúdo suspeito;
- verificar a integridade dos *filesystems* exportados.

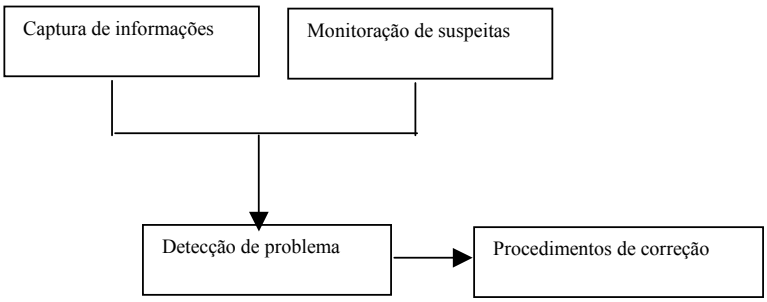


FIGURA 5.11 - Procedimentos básicos de combate a ataques e invasões.

Uma regra que deveria ser obedecida dentro dos NOC's é a de que tempo significa dinheiro, ou seja, a disponibilidade de um profissional especializado para determinadas funções básicas custa muito. Sendo assim, formas de amenizar este tipo de serviço devem ser criadas, como é o caso dos sistemas especialistas voltados para o domínio de problema.

6 União de Sistema Especialista, Sistema de Registro de Problemas e Segurança em Redes de Computadores

A maioria dos profissionais em diversas áreas, principalmente as voltadas à informática, buscam soluções para seus problemas em experiências. As características de um problema podem ser alteradas sob alguns aspectos mas, mesmo em domínios mais complexos como a gerência de redes de computadores ou a medicina, algo que foi aprendido sempre tem utilidade em novas situações (experiência).

Baseando-se nestes procedimentos, surgiram diversas pesquisas para unir, ao dia-a-dia, facilidades de raciocínio. Desta forma, começaram a ser desenvolvidos projetos que aliassem teorias de IA, como raciocínio baseado em casos, com sistemas de registros de problemas e, mais recentemente, gerenciamento de redes de computadores, sobre qualquer uma de suas cinco áreas funcionais descritas no capítulo 4.

Atualmente a área de maior concentração de esforços e que tem dado maior preocupação a profissionais de redes, tem sido o gerenciamento de segurança. Há algum tempo, os ataques a pontos de redes de computadores tornaram-se sinônimo de capacidade e esperteza. O crescimento acelerado e sempre contínuo da Internet tem despertado sentimentos de desafios nos chamados *cracker's,* levando-os a desenvolverem atividades ilegais em máquinas e redes que não lhes pertencem. Isto vem a exigir, dos técnicos dos centros de gerenciamento de redes, respostas tão rápidas quanto os ataques possam ocorrer. Tarefa esta, nem sempre considerada fácil, rápida e normal, pois este tipo de profissional convive sempre com muita pressão e com nível de problemas bem maior do que a capacidade de resposta, além de seu tempo para treinamento ser praticamente inexistente.

Vê-se, assim, um importantíssimo campo que é o de prover ferramentas que possam encurtar o caminho entre treinamento, problema e solução. Para consolidar este ambiente é necessário o uso de sistemas mais inteligentes e próximos do raciocínio humano. Isto pode ser conseguido com a união de tecnologias que são os sistemas especialistas, os sistemas de registro de problemas e os problemas propriamente ditos.

Na busca por um método de sistema especialista, várias são as opções baseando-se em histórias de sucessos em trabalhos realizados. Sistemas de raciocínio baseado em casos (CBR) têm obtido imensa popularidade nos últimos anos em ferramentas de solução de problemas. Embora este tipo de sistema seja muito dependente da estruturação da base de casos, como visto no capítulo 2, ele se sobressai para domínios onde os problemas são mais complexos devido a sua capacidade em aprender com a experiência e por evitar manutenções excessivas.

Ao unir-se sistema CBR com registro de problemas (STT) ganha-se outra facilidade encontrada na semelhança quanto à estrutura para representar seus objetos de trabalho (caso, problema e solução).

Pela teoria de ambos os sistemas, descrita nos capítulos 2 e 4, a construção de um sistema que trabalhe essas metodologias requer uma arquitetura que una os processos tradicionais de registros de problemas com uma forma de raciocínio.

Tais tarefas têm sido trabalhadas por diversos projetos e sempre com sucessos, como o caso do sistema DUMBO [MEL 98] e o CANASTA [LEW 95]. Já especificamente para a área de segurança; alguns desenvolvimentos de trabalhos têm-se utilizado de técnicas de inteligência artificial abrangendo basicamente dois grupos [SPA 93]:

- detecção de anomalias e detecção de violação.

Tais exemplos de trabalhos geraram o desenvolvimento de um protótipo que usa a proposta de união das metodologias descritas acima, e os aplica sob a crescente necessidade dos domínios de gerência de segurança em redes de computadores, chamado de SABER (Sistema Apoiador Baseado em Raciocínio), que foi conceitualmente criado para deixar a base de casos do sistema, estrutura necessária aos sistemas baseados em casos, independente do contexto a ser trabalhado. Isto proporciona que facilmente se possa representar sistemas de *helpdesk* para diversos domínios de problemas, por sua capacidade em deixar o raciocinador independente das feições de um caso ou problema. Ele é um protótipo para treinamento que se utiliza de experiências que envolveram os dois grupos citados por Eugene Spafford [SPA 93], porém não utiliza nenhuma destas abordagens, apenas auxilia a tomada de decisão e aprendizagem.

6.1 Estrutura do Sistema SABER - SISTEMA APOIADOR BASEADO EM RACIOCÍNIO

A estrutura de implementação caracteriza-se como um *framework* que contém duas interfaces para usuários distintos, uma base de casos contida em um banco de dados relacional e um módulo de buscas onde estão garantidos os procedimentos de recuperação, adaptação, evolução e controle de registros de problemas. As duas interfaces compõem dois módulos distintos do protótipo que interagem com outros módulos e, através destes, com a base de casos.

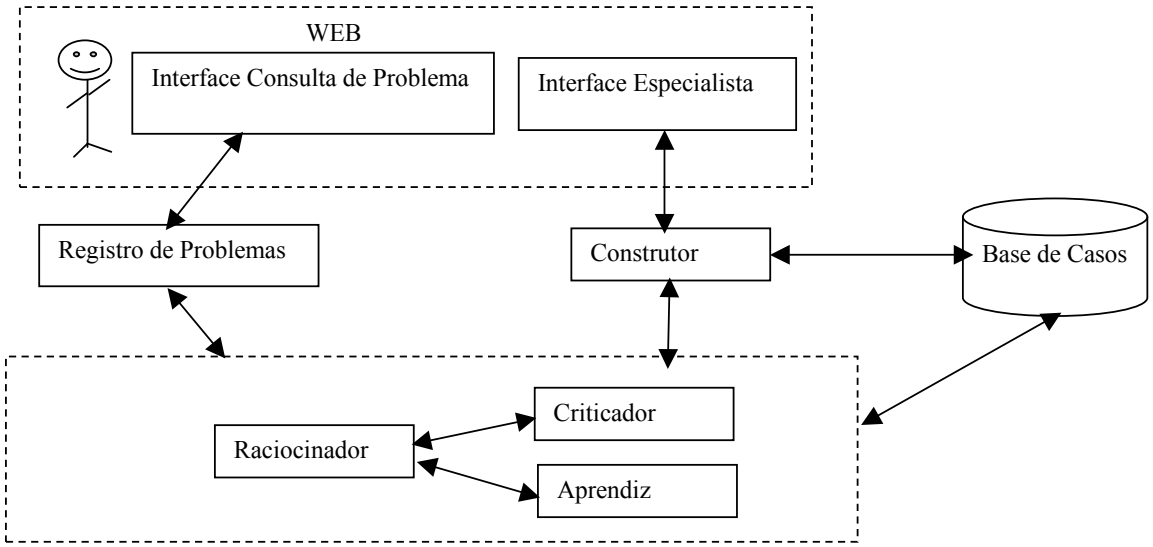


FIGURA 6.1 - A estrutura do Sistema SABER

Para que o protótipo possa ser construído respeitando-se as metodologias que serão utilizadas, alguns parâmetros básicos precisam ser acertados. Assim, os tópicos seguintes compreenderão as definições dos principais aspectos que comporão o protótipo em seus módulos, visando a manter uma relação entre caso, registro de problema e mecanismos de raciocínio com independência da base de casos.

6.1.1 Definições e Aquisições do Conhecimento

Os sistemas baseados em metodologias de CBR têm como principal objetivo a extração de soluções para problemas, buscando-as em casos já ocorridos, os quais devem representar um ambiente de problema e não uma ocorrência. São caracterizados por possuírem uma estrutura bem definida na base de conhecimento, um mecanismo de inferência e uma *interface* para tratamento com o usuário, o que os torna donos de procedimentos muito dependentes da estrutura e contexto da base de casos para que possam realizar corretas comparações.

Para um caso, restrições e características de uma situação são importantes para relatar os sintomas e condições de um problema. Não se faz a inclusão de todos os casos que possam ocorrer em um domínio, o que seria representação de raciocínio por modelos, mas de todos os que possuam feições que possam caracterizar uma situação (caso) diferente. Assim, tem-se um caso como representante de um ambiente de problema e não uma ocorrência, o que possibilita o uso de métodos de *clustering* entre casos, montados mediante o casamento das respostas dadas pelo usuário do sistema, que gera uma biblioteca agrupada por casos similares formando a organização hierárquica de memória.

Já para um sistema de registros de problemas (STT), cada relato corresponde a um problema sem haver correlações. As tais características específicas dos casos ou estão ausentes ou descritas em campos de texto livre, embora a melhor maneira de se delimitar o problema seja descrevendo-o. Isto resulta em tratamentos difíceis e trabalhosos para os processos de raciocínio.

Esta diferença de conceito de representação transforma a modelagem do domínio em um ponto crucial ao sucesso do protótipo, pois dela sairão as instâncias dos casos.

Lewis [LEW 95] afirma que há uma forma de impor restrições para a linguagem de descrição de um problema, que é fazer o usuário ser preciso sobre suas informações. Isto, ainda segundo ele, pode ser conseguido pela apresentação de um campo livre para sua descrição de problema e várias perguntas com respostas fixas sobre como se deseja que a investigação percorra uma rede semântica, ou qualquer representação de fluxo de raciocínio.

Tratando-se uma descrição desta forma, obtém-se uma característica importante para a determinação de uma causa real de um problema, que é o verdadeiro tipo de problema. Geralmente o que o usuário descreve não representa a correta definição do que ele vê; é assombrosa a lacuna de comunicação entre um técnico e um usuário. É muito importante se cercar de fatos que contornem estes espaços, e obter respostas fixas é uma ótima alternativa.

Uma descrição convencional de um problema feito em um STT pode ser observada na figura 6.2.

Caso 10

Descrição do Problema

O *host* não está ativo / Os componentes estão ativos

Ações

Checar os últimos pacotes enviados e coletas dados do atacante;

Restringir acesso ao serviço atacado;

Restringir a entrada da rede atacante.

Avaliação

Nenhum ataque registrado oriundo da rede atacante.

Criação: 10/08/99 **Atualização**: 11/08/99

FIGURA 6.2- Descrição de um problema em um STT.

Com o incremento de informações, conforme descrição de Lewis [LEW 95] ter-se-ia o acréscimo das questões mostradas abaixo, que guiariam o usuário para um determinado nodo de decisão de um problema.

TABELA 6.1 – Perguntas para guiar o problema.

Questões	**Resposta**	**Valor**
O host responde a ping?	Não	Default
Há muitas conexões pendentes?	Sim	Default
Que tipo de pacotes recebeu em maior número?	TCP	Default
	ICMP	_____

Como pode ser observado, ao unir-se estas duas metodologias é necessário ajustar alguns pontos que estabeleçam relações entre estes conceitos, como por exemplo a representação de problemas que, embora semelhantes, precisam de um refinamento para servirem a ambas. Assim, o sistema proposto tem como domínio de problema a união de registros de problemas para domínios desconhecidos, com técnicas de raciocínio que utilizam a metodologia de CBR.

Este sistema necessita ser criado com controle na forma de descrever um problema, como tratado acima, e uma independência entre a base de casos e o domínio.

Tais objetivos são conseguidos através do armazenamento dos casos na base, de maneira que os índices de busca não fiquem relacionados com características específicas de um caso e sim, atrelados à forma de raciocínio, proporcionando que para aumentar a base não sejam necessárias alterações de programação.

No protótipo SABER, optou-se por manter uma base de casos separada da estrutura de um TT com o objetivo de registrar um maior número de informações a respeito de um caso e obter-se um sistema de propósito geral, o qual usará a mesma máquina de inferência com diferentes bases de conhecimento, onde estarão incluídas, também as rotinas necessárias para criar as bases e os procedimentos de perguntas sugeridos por Lewis [LEW 95].

Como citado por Janet Kolodner [KOL 93], um caso pode ser considerado como uma representação textualizada do conhecimento sobre uma experiência. Ele comporá todo o conhecimento de um sistema de CBR, podendo ser representado de diversas maneiras, sem que a comunidade de IA tenha chegado a um consenso sobre a forma ideal. Seja qual for esta representação, o importante é que ela conserve duas partes principais, que são [KOL 93]:

- as lições que ensinam,
- o contexto onde tais lições são aplicáveis.

Stefick [STE 95] afirma que a melhor forma de se representar e trabalhar um caso é pela hierarquização de suas características, que poderão ser herdadas por nodos abaixo durante processos incidentes.

Seguindo-se por este conceito de representação, um caso é representado graficamente, com determinada facilidade, por uma rede semântica que é composta por nodos e arcos capazes de demonstrar relações hierárquicas entre um objeto e seus atributos. Nesta rede os nós (vértices) representam objetos, situações ou conceitos, sendo os elementos pertencentes à rede, enquanto os arcos (arestas) exprimem as relações entre estes elementos. A rotulação dos nós deve conseguir traduzir a semântica do que se quer representar; senão, não se está diante de uma rede semântica.

Ao selecionar-se um problema de segurança em redes de computadores, como por exemplo através de um ataque a uma máquina pela execução de procedimentos conhecidos por *Syn Flood* [BER 96] – estouro de *backlog*, tem-se a descrição feita através de uma rede semântica, conforme demonstrado na figura 6.3. Por atender perfeitamente a uma descrição gráfica de problemas, reproduzindo hierarquias tanto para casos como para registros, o protótipo trabalhará sob a representação de problemas feita por redes semânticas.

A base de casos de um sistema CBR, como descrito anteriormente, é responsável por armazenar os casos. É ela quem determina o que fará parte ou não de um caso, através da definição de sua estrutura, indexação e trabalho eficiente realizado pelos processos envolvidos no raciocínio. Uma base bem estruturada e com um perfeito conhecimento sobre o domínio de problema deverá contar com o maior número possível de informações que possam situar um problema. É ela quem dita o grau de raciocínio de um sistema.

Como o objetivo é produzir um protótipo que forneça diagnósticos eficientemente servindo a vários domínios, o ponto central deste trabalho está em modelar o domínio de problema, o qual não pode ser fixo. Portanto, o que deve ser determinado é o fluxo de informações a serem trabalhadas e obtidas pela realização de um conhecimento heurístico do domínio. Isto porque não há como extrair um modelo de alguns destes domínios, como em diagnósticos médicos ou de redes, onde não existem muitas formas exatas de se determinar diagnósticos, pois não se tem respostas precisas dos objetos afetados para os vários problemas e seus agentes causadores.

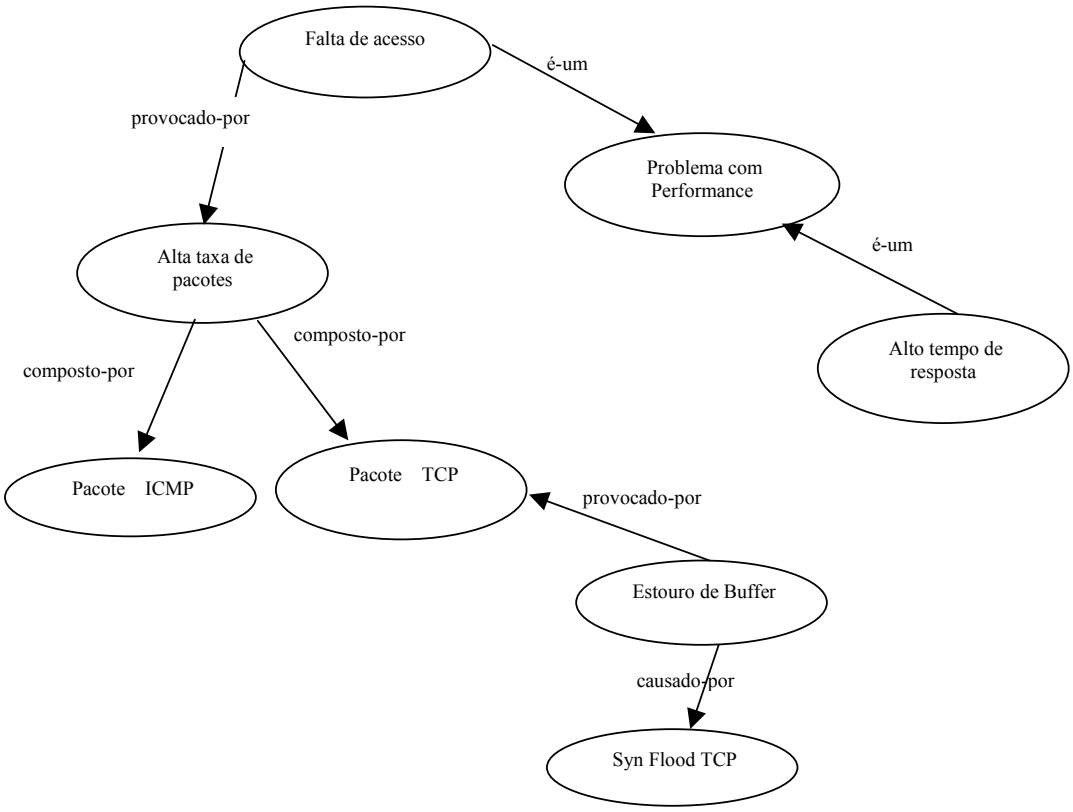


FIGURA 6.3 – Rede semântica de um problema de *Syn Flood*

O protótipo faz uso exclusivamente de uma engenharia de conhecimento para realizar a tarefa de elicitar e codificar o conhecimento do domínio especialista. As informações são extraídas por entrevistas e/ou dos *trouble tickets* dos STT e passados ao protótipo por um especialista cadastrado, onde são utilizadas em seus processos de raciocínio.

A forma de obtenção das informações, considerada, é, portanto, através de especialistas humanos que são perguntados para expor seus conhecimentos na forma de princípios qualitativos, que são características previamente cadastradas e com limites valorados no sistema. Tem-se, então, o modelo de diagnóstico do sistema raciocinando da experiência humana e o comportamento do sistema não explicitamente modelado.

Toda a inclusão de conhecimento é baseada no engenheiro de conhecimento e suas redes semânticas de problemas. Este profissional deve sempre considerar que um estudo de casos não deve ser restrito a problemas; ele pode mostrar soluções e em que situações fazer uso delas, mostrar diversos problemas e quais as melhores maneiras de resolvê-los.

Tem-se notado, através de estudos de sistemas baseados em casos, que os melhores aprendizados de casos são aqueles que possuem ambos: as devidas e as não devidas soluções para o problema. Assim, já se teria registrado o que não se pode considerar para o problema. O protótipo permite estas regras, pois seu módulo Construtor possibilita ao engenheiro entrar com suas características livremente e valorá-las.

6.1.2 Módulo de Interface Especialista

O módulo Interface Especialista é responsável pela entrada de definições vindas de um usuário privilegiado (engenheiro de conhecimento) que alimenta a base de casos e provém o sistema com todo o conhecimento necessário. O engenheiro de conhecimento constrói um caso para apresentá-lo de forma interativa ao usuário. Este módulo responde pela intervenção do protótipo com o usuário e é composto por: uma interface para com o *browser*, um módulo Construtor que recebe as métricas definidas pelo EC e interage com o módulo de Busca fornecendo o raciocínio desejado quando realizada uma consulta pela interface Consulta de Problema. Os módulos que o compõem são descritos a seguir:

6.1.2.1 Módulo Construtor - Representação e Conteúdo dos Casos

Como comentado no capítulo 2, o modo como será feita a representação de um caso não é importante, mas sim que ela atenda à forma de raciocínio do sistema. A responsabilidade quanto ao que coletar e armazenar está nas mãos do engenheiro ao alimentar o sistema pelos processos de aquisição.

A inclusão de conhecimento no protótipo se dá através do módulo Construtor. Ele recebe o conhecimento do engenheiro e o transforma para apresentar aos seus usuários; representa a interação entre engenheiro e protótipo.

Tem como processos principais definir a forma de armazenamento de casos e feições que, ditadas pelo engenheiro, farão com que o sistema fique independente do domínio de problema; responde pela hierarquização do sistema.

Para a representação dos casos, o engenheiro utiliza uma simples estrutura de registros de casos que são armazenados em um banco de dados relacional, como feito pelo *shell* CBR3, descrito no capítulo 3.

O engenheiro deve definir relações hierárquicas através de atributos e um peso que representará uma medida de similaridade baseada nestas relações. O caso é representado como um *template* de um objeto que utiliza pares de atributo-valor, possíveis de serem compartilhados com outros casos. Neles, o objeto é a conclusão definida por características associadas e o atributo é uma qualidade específica que, com sua regra, ajuda a definir o objeto. Assim, aplica-se a um objeto uma regra onde ele tem ou não tem um atributo.

Esta decisão baseou-se na possibilidade de indexação por características não específicas dos problemas, sempre respeitando funções, comportamentos e estruturas das descrições hierárquicas feitas nas redes semânticas dos problemas. Tais procedimentos vêm a facilitar o trabalho do mecanismo de inferência sobre a base, trazendo rapidez nos processos de busca, o que é o grande problema de um sistema especialista.

Assim, tem-se para a descrição em rede semântica feita acima, quanto ao problema de *Syn flood,* o armazenamento na base como demonstrado na tabela 6.2.

O ideal para este tipo de sistema seria colocar próximo de seu topo os atributos que causam a maior parte das podas em uma árvore de decisão mas, quando a base é muito grande isto só seria possível através de muito uso e geração de estatísticas. O protótipo gera estatísticas mas visando medir a utilização das soluções.

Um caso, para o módulo Construtor, é composto por duas partes criadas como inclusões em separado que são:

- a descrição - como pode ser observado na figura 6.3, inclui um título e características de como o problema (objeto) seria chamado ou identificado;

- a solução do problema - um conjunto de ações que compreendem procedimentos os quais podem ser tomados para resolver o problema.

TABELA 6.2 – Descrição do problema de *Syn Flood*, segundo uma OVL.

OBJETO	ATRIBUTO	VALOR
Ataque por *Syn Flood*	Taxa de pacotes	Alta
	Pacote	TCP
		ICMP
	Estouro de Buffer	Sim
	Host, máquina, servidor, gateway...	

Outra função importante deste módulo é a inclusão, feita pelo engenheiro, quanto a perguntas que guiam o usuário para decisões, as quais possuem um conjunto de pesos além de pares de atributo-valor que correspondem às respostas dadas às mesmas. Estas perguntas serão apresentadas pelo protótipo através do módulo de Busca, e são as características de descrição de um caso que, ao serem respondidas, recebem valores predeterminados pelo engenheiro e compõem o algoritmo de similaridade, descrito a seguir.

TABELA 6.3 – Estrutura completa de uma instância de problema para o protótipo SABER.

Objeto: **TCP *SYN flood***	
Atributos: • requisições de conexão (*SYN*), • conexões pendentes (*backlog*), • sem comunicação (*time-out*),	Valores: • Sim • Sim • Sim
Soluções: • Rejeitar requisições externas pelos *gateways*, • Somente permitir acessos autorizados,	
Perguntas: 1. Existem conexões pendentes? Sim/Não - 9 *2.* A máquina está com alta taxa de pacotes? Sim/Não - 9	
Respostas: 1. Sim – 10 Não - 0 2. Sim – 10 Não - 0	

As perguntas ajudam o protótipo a decidir e preencher lacunas deixadas na descrição do problema. O engenheiro deve gerá-las tendo em mente sua rede semântica, ou seja, perguntando o que poderia podar nodos ou guiar a decisão por outros, até que o raciocínio possa ser completado e o melhor casamento realizado entre problema proposto e casos candidatos. Portanto, a estrutura completa para que o problema de ataque por *Syn Flood* passe a constituir um caso da base de casos do protótipo deveria compreender as características detalhadas na tabela 6.3.

Estruturas, estados e funções hierárquicas são usadas como domínio de conhecimento levando à construção de uma estrutura de memória que indexa os casos e otimiza a utilização de espaços. A esta estrutura dá-se o nome de base de conhecimento, onde ficam armazenados os casos.

6.1.2.2 Módulo de Busca – Consulta de Casos

Quando o processo de inclusão de casos estiver terminado, podendo representar situações diversas, é necessário preparar a base de casos para ser utilizada através de tratamentos de indexações. Como o sistema proposto não pode fixar seus índices em caraterísticas específicas, todos os atributos podem representar índices, padrão utilizado nos sistemas CASECAD e CADSYN [MAH 96]. Cabe ao engenheiro, ao incluir suas perguntas, optar pelas que definirão os caminhos (nós) da rede que o raciocínio deve seguir, procedimentos tratados pelo módulo de Busca.

Este módulo é responsável por todo o processo de consulta e montagem de raciocínio e é utilizado pelas duas interfaces do protótipo. Para a interface de consulta, o módulo responde pelos procedimentos de busca e apresentação de resultados. Para a interface especialista, ele realiza diversos procedimentos, que incluem:

- atribuições de índices,
- cálculo de similaridade,
- seleção, aceitação e alteração de soluções,
- cálculo do grau de aceitação das soluções,
- contabilização de perguntas mais respondidas.

A fim de alcançar seus objetivos, conta com o trabalho do módulo Raciocinador que completa seus processos usando dois submódulos: o Criticador e o Aprendiz. Cada submódulo é responsável por procedimentos exclusivos que resultam na apresentação das soluções para o usuário.

6.1.2.3 Módulo Raciocinador – Indexação

O módulo Raciocinador possui os parâmetros de comparação, bem como os graus de relevância de cada feição de um caso, predeterminados pelo engenheiro, tendo como responsabilidade a elaboração, a recuperação de casos relevantes e a adaptação dos casos, procedimentos realizados por seus módulos complementares. Tem por função preparar a entrada de informações para o processamento.

Todo procedimento de busca deve retornar no mínimo um pequeno grupo de casos com maior potencial de similaridade para a situação descrita. Os problemas propostos raramente possuem a descrição exata dentro de uma BC. Assim, pode-se efetuar busca em dois estágios: o primeiro retorna casamentos parciais entre caso e problema e o segundo, um casamento completo.

A indexação de características é a forma com que o protótipo trata o processo de busca que constitui um arranjo na base de dados, realizado através dos graus indicados nas entradas dos casos e que servem de índices para recuperação dos mesmos. Seus valores obedecem à atribuição do par atributo-valor e são fixados pelo EC.

A escolha de índices deve ser uma tarefa cuidadosa, eles influem diretamente na comparação de *slots* e consequentemente tornam a recuperação mais lenta. Um exemplo de atribuição de índices pode ser observado na tabela 6.4. Os índices correspondem às

características dos objetos que vão sendo selecionados e a indexação aponta para pesos na base de casos.

TABELA 6.4 – Atribuição de índices para casos

		Valor do índice em caso armazenado		
		Sim	Não	Desconhecido
Valor do índice no caso proposto	Sim	5	1	0
	Não	1	5	0
	Desconhecido	0	0	0

Os índices são utilizados quando uma semelhança é detectada entre as características do problema proposto com um caso. Outra forma de utilização dá-se pelo uso de perguntas e respostas que servem como estruturas logarítmicas para indexação, as quais dinamicamente diferenciam os casos recuperados, se forem corretamente cadastradas. Assim, os índices definidos passam a guiar o raciocinador por uma decisão.

6.1.2.4 Módulo Criticador - Similaridade

O módulo Criticador faz a crítica entre os parâmetros de entrada e os recuperados no Raciocinador, ou seja, faz a comparação entre o que entra pela descrição e pelas respostas do usuário com o que há cadastrado na base de dados. Utiliza-se de cálculos de métricas de similaridade para fornecer os casos candidatos que serão apresentados ao usuário, o qual pode aceitá-los ou não, ou ainda aperfeiçoá-los.

Quando o módulo Criticador recebe a descrição do problema, ele procura por casos que possuam as palavras semelhantes às descrições e características do problema proposto. Os casos com o maior valor do somatório de *a* (valor do índice) são recuperados, resultando em duas listas, uma de soluções ordenadas e outra de perguntas apresentadas ao usuário com o objetivo de ajudá-lo a encontrar o *best match*. Se a descrição do usuário não gerar casamento algum, então apenas as perguntas serão montadas para o usuário que, ao respondê-las, aproxima mais seu problema dos casos candidatos.

As primeiras perguntas apresentadas correspondem às que possuem o maior peso cadastrado e que têm relação com a descrição do usuário. Caso nenhuma relação possa ser identificada, o protótipo apresenta perguntas selecionadas de um *ranking* das perguntas que mais receberam respostas, ou seja, as com maior representatividade em problemas segundo o usuário. Esta lista é uma valoração estatística que procura relacionar problema proposto com os casos cadastrados no sistema. Desta forma, o protótipo aprende quais as perguntas que mais se relacionam aos casos, corrigindo possíveis erros nas descrições dos casos, desobrigando o engenheiro de responder pela forma ideal de resolver os problemas. Então, através das respostas do usuário as perguntas apresentadas é gerado um valor y_i . Ou seja:

y_i = valor total das características ou atributos obtidos por conhecimento guiado.

Adquire-se assim, o que se chama, neste trabalho, de grau de confiança. Este coeficiente é determinado pela razão entre y_i e z_i , onde z_i é um valor pré-calculado de acordo com as características cadastradas na base de casos.

$$GrauConf_i = \frac{y_i}{z_i}$$

Na apresentação das soluções pelo grau de similaridade ($Simil_i$) e, também na apresentação das soluções pelo grau de confiança ($GrauConf_i$), o usuário pode aceitar ou rejeitar as soluções propostas, assim, através destas informações pode-se calcular o índice de aceitação e, consequentemente, o índice de rejeição:

$$Aceitação = \frac{\sum Quant.Casos \text{ Aceitos}}{Quant.\text{Casos Apresentados}}$$

Rejeição = 1 – Aceitação

Dois tipos de similaridade são, portanto considerados no protótipo. Um é chamado de similaridade e obtido pelo primeiro processo de casamento (casamento parcial), onde uma porcentagem de similaridade ($Simil_i$) é calculada através da comparação de palavras que compõem descrições e características entre casos armazenados e o problema proposto. Cada palavra semelhante encontrada em um caso tem seu peso acumulado para este. O segundo tipo compreende a similaridade chamada de confiança, onde o usuário guia o raciocinador montando sua rede semântica através de respostas a perguntas sugeridas e assim, um peso ($GrauConf_i$) é contabilizado para o caso que apresentar relação com a pergunta respondida e com a resposta. As duas medidas são mostradas ao usuário quando solicitado um refinamento de procura; se esta não for selecionada apenas a medida chamada similaridade será apresentada, se for possível de ser encontrada.

Aamodt [AAM 94], entre outros autores, afirma que os casos recuperados devem ser conseguidos pelas características de entrada ou deduzidas destas. Assim, as representações de casos para CBR, geralmente, propõem que um caso seja composto por um conjunto de características e sua solução. Tal proposta levou a criação do algoritmo de vizinhança demonstrado na seção 2.1.3.4.

Segundo Cain e Pazzani [CAI 91], os métodos numéricos mais utilizado para calcular o grau de casamento entre dois casos similares têm sido o algoritmo de vizinhança ou adaptações dele.

Este protótipo, realiza o tratamento de similaridade através da utilização de uma derivação do algoritmo de vizinhança produzindo um algoritmo próprio, o qual é

baseado no peso das características, da pergunta e da resposta a perguntas relacionadas com um caso, sendo que cada caso possui diferentes pesos para respostas da mesma pergunta, mas são relações e valores determinados pelo engenheiro, como descrito anteriormente e sugerido por Aamondt [AAM 94].

Conforme as respostas são coletadas, o que não requer ordem, o módulo realiza etapas que constituem seu algoritmo de recuperação, fornecendo uma pequena base de casos como resultado da procura. Toda resposta é contabilizada para o problema proposto e para casos que apresentem tais ítens como índices. Ao final de todas as inclusões os casos com maior nota (*score*) são recuperados. Esta é a maneira como o sistema trata o problema de similaridade, ou seja, através de conhecimento guiado e uma adaptação do algoritmo de vizinhança.

O processo que envolve o cálculo de similaridade e o grau de confiança são descritos pelas seguintes representações numéricas:

confiança (p,z) = x_i, z_i

Onde: p = problema proposto;
z = caso candidato;
x_i = valor total das características ou atributos obtidos por similaridade sintática

$$x_i = \sum_{n=1}^{nce} a_n c_n$$

Onde *nce* representa o número de características encontradas no problema proposto, sendo que a cada característica é atribuído um peso *a* (constante) e *cn* são as características encontradas no problema proposto. Ou seja, através da análise sintática é determinado o valor *x* para o problema proposto, sendo que os pesos são pré-definidos na base de dados para cada característica.

A métrica de similaridade utilizada é determinada pelo quociente entre x_i e z_i. Onde x_i é obtido através da análise sintática de z_i é o valor total das características de cada caso armazenado. Então:

$$Simil_i = \frac{x_i}{z_i}$$

Como resultado final, o usuário tem a apresentação das soluções encontradas para sua descrição de problema. Várias soluções podem ser apresentadas e ele pode, então, verificar quais as que lhe são úteis ou quais poderiam receber alterações, processos efetuados pelo submódulo Aprendiz.

6.1.2.5 Submódulo Aprendiz – Aprendizagem e Adaptação

O módulo Aprendiz interage com o Raciocinador para responder pela aprendizagem do sistema, a qual é armazenada na base de conhecimento (BC). É o

módulo que responde pela aquisição de novos casos, procedimentos executados pelo EC.

A entrada de casos na base é muito fácil e simples, basta ter claramente divididas as noções de título, característica e solução. A única exigência é que se tenha os casos bem estruturados nas redes semânticas antes da inclusão na base.

Os casos são inseridos no protótipo pelo EC, que os descreve seguindo as definições feitas anteriormente, ou seja, obedecendo que todo caso possua título, descrição e solução e, além disto, perguntas que possam aproximá-los dos problemas propostos, ajudando nos processos de raciocínio.

Outra maneira de aquisição dá-se por cadastro de problemas sem solução. Problemas, os quais, quando consultados, não realizaram casamento com nenhum caso cadastrado, sendo armazenados em uma tabela temporária onde o EC deve realizar verificações constantes e achar soluções devidas. Isto possibilita que tais descrições passem ou não a constituir novos casos. Esta é a forma como o protótipo realiza a sua aquisição de novos casos e trata os problemas não solucionados.

A base temporária onde ficam os registros abertos deve, também, ser verificada periodicamente por um engenheiro para coletar novos casos para o sistema, pois, se estes estão em aberto, o raciocinador não conseguiu identificar casamentos para eles ou o usuário não aceitou o que lhe foi recuperado.

Quando um caso novo é cadastrado, passa a fazer parte do banco de casos principal, o qual vai receber todos os tratamentos de indexações característicos ao sistema, procedimentos adotados pelo *shell* CASUEL [MEN 99]. Assim, a aprendizagem é simples e feita como uma inclusão normal no sistema.

Quando um problema consultado no protótipo não obtiver uma resposta ou obtiver uma que, possivelmente, não é correta aos olhos do usuário, como comentado anteriormente, é permitida uma atualização na base de casos, onde o usuário que registrou o problema pode fazer adaptações nos passos que precisou utilizar para chegar ao encerramento do mesmo. Passos, estes, que não constavam no sistema e passam a constituir sua solução nas próximas buscas para problemas semelhantes.

A adaptação neste tipo de sistema é dada, portanto, pela realimentação do sistema após a interferência de um especialista ou engenheiro, não possuindo nenhum procedimento automático.

O protótipo apresenta uma estatística própria que contabiliza eficiência ou fracasso, toda vez que uma solução recuperada for aceita ou não pelo usuário. Este processo garante o aproveitamento das soluções, contribuindo com o usuário quanto a informar o grau de sucesso do que lhe está sendo oferecido, deixando-o mais confiante em suas tomadas de decisões.

6.1.3 Interface Consulta de Problema

O módulo Interface Consulta de Problema destina-se ao usuário comum de um registro de problema. Neste módulo, ele encontra as funcionalidades de um STT e pode registrar suas consultas e problemas. A entrada de um problema no protótipo é

armazenada em uma tabela temporária onde recebe um identificador e, somente após encontrada sua solução, passa para uma tabela definitiva e adquire representividade para o contexto do protótipo. Seus processos são executados pelo módulo Registro de Problemas, o qual interage com o módulo de Busca.

6.1.3.1 Módulo Registro de Problemas

O módulo Registro de Problemas possui todos os componentes de um STT, onde, para cada problema, deve ser criado um novo registro que terá um identificador único, todos os dados sobre o problema e ações efetuadas ao longo de sua criação até o seu encerramento.

Os dados a serem registrados compreendem, além do identificador, uma estrutura sugerida por Johnson [JOH 92], que consiste de cabeçalho, atualizações e dados da resolução. O cabeçalho é composto pelas informações de abertura do problema, as quais, para o protótipo, compreendem:

- um título,
- descrição com pequeno texto livre,
- nome do reclamante,
- nome do responsável pelo problema,
- data de abertura e última alteração,
- severidade do problema.

As atualizações podem ser feitas já na inclusão, através de um campo de texto livre onde expõem-se detalhes do problema, que servirão como dados históricos do mesmo. Já os dados da resolução dizem respeito ao encerramento do problema, onde constam data de encerramento e dados das soluções adotadas.

Assim, um registro com relação ao caso de *Syn Flood* pode ser encontrado neste módulo com as descrições feitas na tabela 6.5.

O usuário do protótipo, ao consultar um problema, pode apresentá-lo com suas próprias palavras. Estas sofrem uma avaliação feita pelo módulo de Busca, ativando mecanismos de procura pela base de casos, onde é realizada uma análise nas palavras relevantes que podem vir a coincidir com as características de algum(ns) caso(s). Palavras cadastradas em um arquivo podem ser excluídas de buscas exaustivas, como o caso de preposições, conjunções ou qualquer outra que o engenheiro entenda como não relevante ao seu domínio de problema.

O módulo Registro de Problemas compõe um processo onde o usuário faz o registro de problema de um cliente e/ou usuário e interage com o módulo de Busca à procura de casos que tenham características semelhantes. Se estas não forem detectadas pela descrição do usuário, elas poderão ser conseguidas por perguntas auxiliares no encontro por similaridade. Este módulo serve ao Raciocinador através do fornecimento de características que podem compor um novo caso. Este acréscimo na base de casos

fica dependente da intervenção do engenheiro, armazenado em uma tabela temporária como um registro de problema temporário.

TABELA 6.5 – Abertura de um registro de problema.

ID: 10	Prioridade: Máxima
Título: O *host* não responde	
Descrição: Está na rede mas ninguém acessa	
Detalhes: O host 200.18.6.25 não responde, mas não apresenta problemas de parte física com a rede. Diversas requisições estão sendo recusadas.	
Reclamante: Pedro	
Responsável: Cinara	
Data de Abertura: 10/10/2000	

7 Implementação do Protótipo

Tendo como objetivo validar a união das metodologias aplicadas neste trabalho, desenvolveu-se um protótipo apresentado neste capítulo. Nele é feita uma análise do ambiente utilizado para desenvolvimento e implementação, bem como a descrição de cada componente do protótipo, suas dificuldades e vulnerabilidades.

7.1 Plataforma de Hardware

Optou-se pela instalação do protótipo na estação Sparc – ULTRA 1, possuidora de 64 MB de memória, instalada na rede da UFRGS (Universidade Federal do Rio Grande do Sul), com o endereço IP 143.54.1.30, de nome penta2.ufrgs.br.

Foi desenvolvido para ser utilizado em qualquer tipo de configuração de ambiente possuidor de acesso à Internet através dos *softwares* mais comuns a tal procedimento, como navegadores Netscape 4.x ou superior ou Internet Explorer 4.x ou superior.

As restrições de uso são feitas mediante acessos por senhas e níveis de usuários criados por uma tabela de consulta do próprio protótipo.

7.2 Recursos de Software

O protótipo foi desenvolvido utilizando uma estação de trabalho com o sistema operacional Linux, e posteriormente transportado para uma estação SUN (penta2.ufrgs.br). Necessitou dos seguintes softwares:

- Perl 5;
- Sistema de Banco de Dados Relacional MySQL 3.22 [MYS 99], multi-usuário e multi-*thread*, que é uma implementação cliente/servidor, de domínio público.

7.3 Arquitetura do Sistema

O protótipo, por ser desenvolvido sob o banco de dados relacional MySQL, é projetado em uma arquitetura cliente-servidor, onde os clientes comunicam-se com o módulo servidor através de módulos CGI e o próprio módulo servidor é uma junção de CGI responsável pelos procedimentos de raciocínio do sistema.

Para que o protótipo seja implementado em um banco de dados MySQL, primeiramente é necessária a criação de uma base de dados chamada 'conhecimento,' na qual o instalador deverá criar a tabela SENHA e nela cadastrar os usuários do sistema. Este é um módulo à parte, onde os procedimentos são executados diretamente no banco de dados, sem que tenha sido desenvolvida interface de inclusão. Os comandos encontram-se descritos no Anexo 1.

A identificação do usuário (*username* e senha) é cadastrada com dois tipos de níveis: um ‘adm’ que compreende acesso total a todas as rotinas, acesso permitido aos engenheiros de conhecimento que alimentarão a base de dados, e acesso ‘user’ para o usuário que apenas incluirá e consultará problemas.

A inicialização do protótipo dá-se pela escolha de qual rotina o usuário gostaria de obter, conforme demonstrado na figura 7.1, sendo possível inserir ou consultar conhecimento.

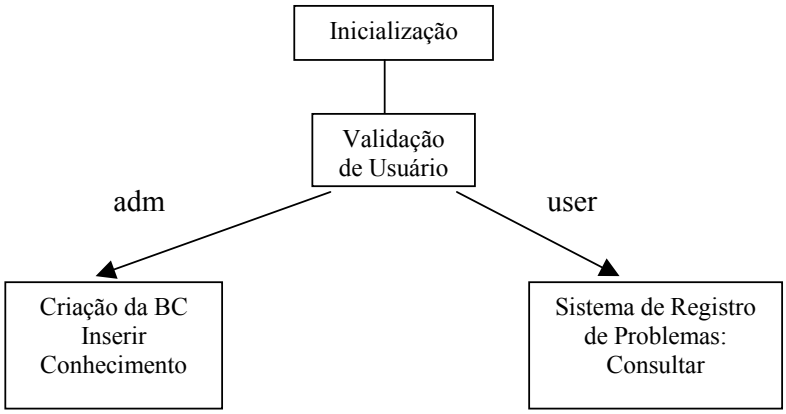


FIGURA 7.1 - Inicialização do protótipo.

Como os acessos são nivelados e permitem rotinas diferenciadas, descreve-se a seguir a implementação para o usuário com permissão de inclusão de conhecimento e, em seguida, para usuários com acesso apenas ao registro de problemas. Quando um usuário não é autorizado, uma mensagem de erro é apresentada.

7.4 Acesso para Engenheiro de Conhecimento

Com o objetivo de evitar que um usuário não especialista influa no raciocínio do protótipo, o acesso permitido ao engenheiro de conhecimento (EC) é autorizado por senha e dá direitos de montagem de casos e de árvores de decisão.

Após a validação, o EC faz uma seleção do domínio de problema a ser criado. Ao nome escolhido por ele corresponderá a criação de uma base de dados dentro do MySQL que possuirá, ao final de todos os processos do protótipo, as seguintes tabelas:

Caso o nome da base digitada pelo EC já exista, o protótipo apenas a selecionará para uso nas tarefas subseqüentes. As tabelas acima são criadas, alteradas ou excluídas conforme os processos vão sendo executados. Para cada base de dados com nome diferente, o que caracteriza, pela concepção do sistema, um domínio diferente, todas estas tabelas são criadas independentemente para cada domínio.

TABELA 7.1 - Tabelas do banco de dados do protótipo para um domínio de problema.

Tabela	Descrição
ATRIBUTO	Características específicas dos casos.
CANDIDATO1	Tabela temporária com os casos "candidatos " para um problema.
OBJETO	Características associadas gerando um caso e dados do caso.
PERGUNTA	Cadastro das perguntas definidas pelo EC, conforme sua rede semântica, com os pesos para cada caso.
PESO1	Tabela temporária de ordenação dos pesos dos casos "candidatos".
RECLAMANTE	Cadastro de usuários do sistema com nível 'user'.
REL_MONITORA_SOL	Relação entre as vezes que a solução aparece e é aceita pelo usuário.
REL_OBJETO_ATRIBUTO	Relação de características que um caso apresenta.
REL_OBJETO_PERGUNTA	Relação de perguntas descritas pelo EC para um determinado caso.
REL_OBJETO_SOLUCAO	Relação de soluções para os casos.
REL_OBJ_TICKET	Relação entre um ticket e um caso.
REL_PERGUNTA_ATRIBUTO	Relação de opções de respostas para as perguntas cadastradas.
REL_PERGUNTA_RESPOSTA	Relação de resposta para uma pergunta.
REL_PERGUNTA_RESPOSTA_ OBJETO	Relação entre uma opção de resposta para uma pergunta para, com o caso, proporcionar reaproveitamento de perguntas para vários casos.
PALAVRAS	Cadastro de palavras que serão excluídas durante a busca por similaridade sintática.
RESPOSTA	Cadastro de respostas ideais para cada pergunta armazenada, com o peso que tal atributo acarreta na árvore de decisão.
SOL1	Tabela temporária de montagem das soluções a serem apresentadas ao usuário, conforme respostas recebidas pela decisão guiada para cada *best match*.

SOLUCAO	Cadastro das diversas soluções de cada caso.
SOMAPER	Tabela de armazenamento do ranking de perguntas que mais apareceram e receberam respostas nas procuras por problemas.
TEMPO1	Tabela temporária de respostas dadas às perguntas apresentadas ao usuário.
TICKET	Cadastro de registros de problemas.
TICKETT	Cadastro dos registros de problemas enquanto estes não podem ser tratados pelo sistema por não se possíveis de "casamento".

A inclusão de conhecimento é dada seguindo a definição de que um caso é um objeto composto por vários atributos, conforme representação na figura 7.2, tal afirmação ajudará o módulo Raciocinador na extração de raciocínio indutivo.

O engenheiro deverá proceder o cadastramento dos objetos considerados como conclusões de vários atributos. Esta tarefa pode ser realizada de duas formas: vários objetos e depois suas características (atributos) ou um objeto e todas as suas características. Para o segundo procedimento deve-se selecionar um objeto que terá relação com os atributos a serem cadastrados.

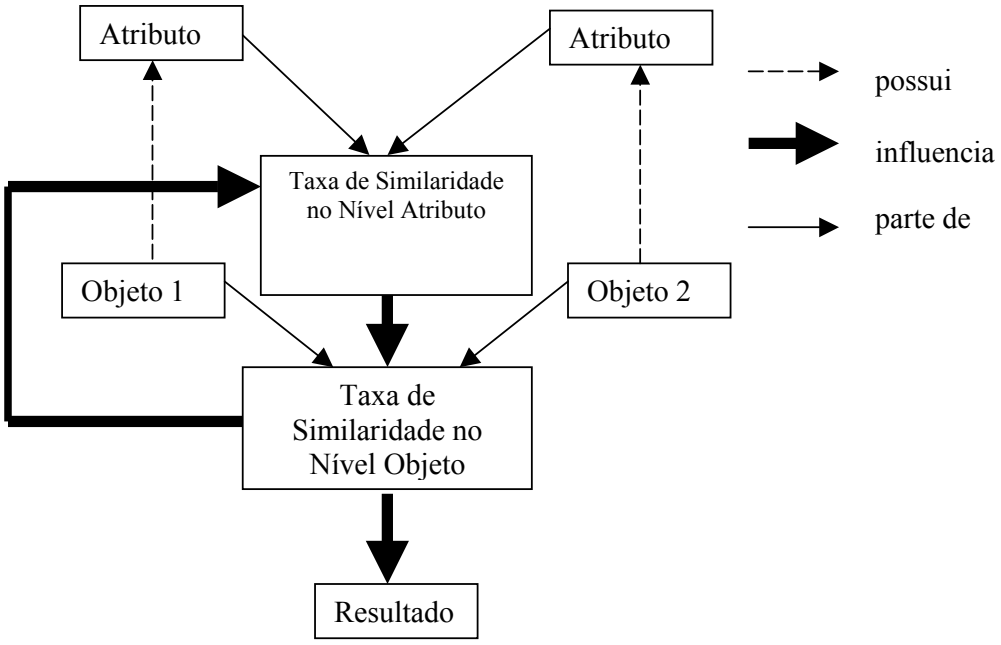


FIGURA 7.2 - Busca por similaridade no protótipo SABER.

De posse de um objeto, pode-se proceder o cadastro dos atributos e partir deste à inclusão de perguntas, respostas para estas perguntas e soluções para o determinado objeto de problema. O cadastramento de conhecimento no sistema obedece à ordem

descrita na figura 7.3 . Após a realização dos cadastros 1 e 2, somente precisará ser dada atenção à ordem entre os cadastros 3 e 4.

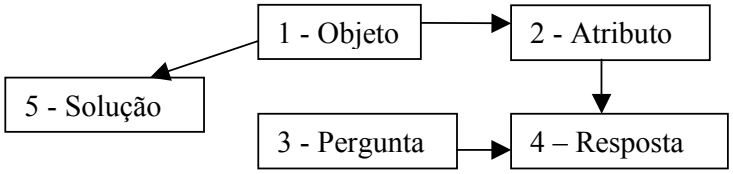


FIGURA 7.3 - Ordem de cadastramento do protótipo SABER.

Deve-se estar atento ao fato de que, nenhum processo de cadastramento será realizado corretamente se um objeto não estiver pré-selecionado.

O cadastro de perguntas deverá ser feito pela análise da rede semântica do objeto a ser encontrado, responsabilidade do EC. As perguntas recebem pesos e serão utilizadas para guiar o protótipo para uma decisão; portanto, uma mesma pergunta poderá servir para diversos objetos. Ela poderá ter como resposta várias características de outros objetos e ser mais adequada a um deles.

Para representar esta relação, basta selecionar o atributo desejado e classificá-lo como a melhor resposta para a referida pergunta, o que implica em atribuir-lhe um peso maior caso ela seja selecionada pelo usuário como resposta à pergunta em questão. Esta inclusão pode ser observada na tela representada na figura 7.4.

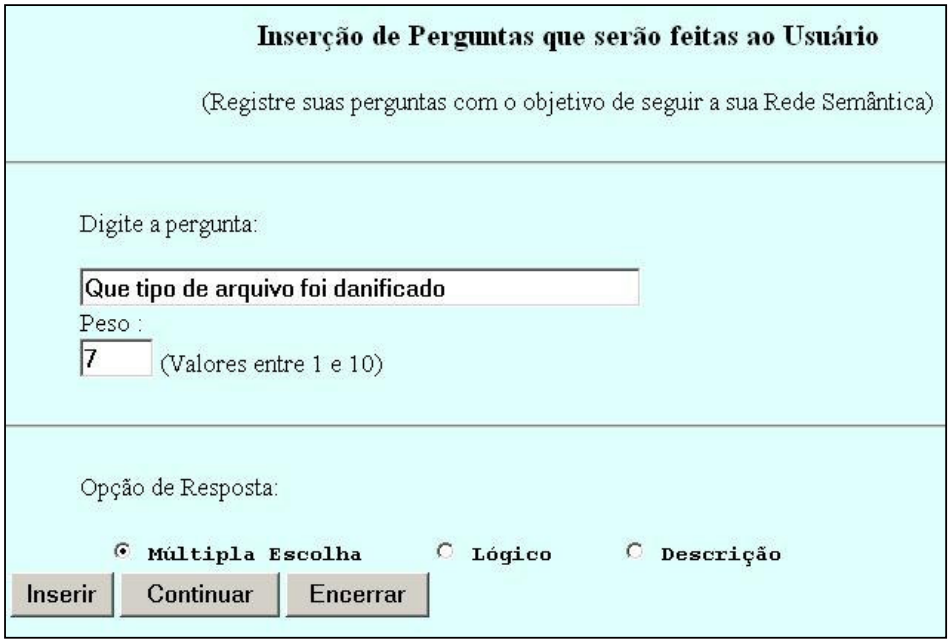


FIGURA 7.4 - Inclusão de uma pergunta no protótipo SABER.

Os pesos das perguntas e respostas são estipuláveis por importância durante a análise do EC sobre o seu domínio de problema.

As perguntas apresentam três formas de receber respostas que são: por múltipla escolha (fixa), por descrição (descritiva) ou por resposta lógica. As duas primeiras correspondem às características que foram descritas nos atributos de um caso e os valores lógicos compreendem respostas SIM ou NÃO. A forma de se obter respostas foi atrelada aos atributos para facilitar buscas por similaridades exaustivamente, conforme apresentado por Lendy Lewis [LEW 95].

As opções de respostas para a pergunta cadastrada pode ser feita quando da inclusão da mesma ou após, mediante seleção de uma delas na opção Operar Resposta, onde também pode ser alterado o peso e a forma de resposta. Os pesos de perguntas e respostas são de responsabilidade do EC que analisará sua importância na rede de problemas a ser descrita. Estes pesos serão utilizados no cálculo de similaridade; portanto, sua atribuição poderá levar a uma busca mais rápida ou a um casamento mais completo.

As soluções, como comentado anteriormente, podem ser cadastradas desde que selecionado um caso para relacioná-las.

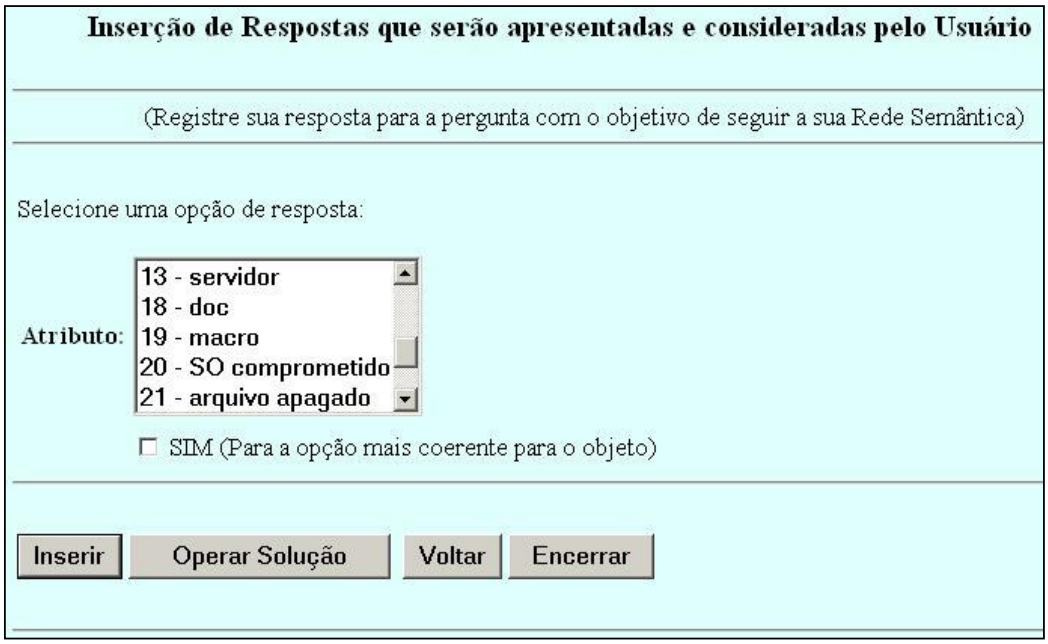


FIGURA 7.5 - Inclusão de respostas para uma pergunta.

Com os procedimentos acima, encerra-se a inclusão de casos no protótipo obedecendo à definição de que um caso deve possuir um título, uma descrição e uma solução. A responsabilidade pela correta construção da árvore de decisão, como colocado anteriormente, é depositada no cadastramento dos casos por parte do EC e suas atribuições de pesos, que guiarão os cálculos de similaridade.

O EC pode também partir para o registro de problemas retornando à rotina inicial de acesso ao protótipo.

7.5 Acesso a Usuário

Com a devida identificação, o usuário receberá autorização para consultar uma base de problemas. Devido às limitações de comandos do banco de dados escolhido para o desenvolvimento do trabalho, todas as bases existentes na máquina servidora serão expostas para o usuário, o qual somente poderá ter acesso às bases que foram construídas para o protótipo, ou seja, onde as tabelas a serem trabalhadas correspondem às que foram criadas e que serão utilizadas. A escolha por uma base refere-se à seleção do domínio de problema a ser consultado.

No módulo de Consulta, tanto engenheiro quanto usuário comum terão apenas acessos compreendidos como registros de problemas. Desta forma, o conhecimento dos domínios não poderá ser alterado. As opções de ações permitidas podem ser verificadas na tela da figura 7.6.

FIGURA 7.6 - Tela de opções de registros de problemas ao usuário do protótipo.

O usuário faz uma solicitação de recuperação de casos similares por descrever seu problema, conforme demonstrado na tela da figura 7.7.

FIGURA 7.7 - Tela de inclusão de registro de problemas no protótipo.

O registro de um problema é compreendido por uma tabela chamada Ticket, com o seguinte formato:

id_ticket	título	desc_ticket	id_rec	id_res	prioridade	dt_modify	dt_close	dt_open	det_ticket

Onde:

Id_ticket : valor seqüencial identificando internamente o problema (*ticket*);

Título: título do problema,

Desc_ticket: pequena descrição do problema;

Id_rec: valor seqüencial identificando internamente o reclamante. São os usuários cadastrados;

Id_res: valor seqüencial identificando internamente o responsável. São os usuários com nível de administrador de conhecimento;

Prioridade: nível de prioridade do problema,

Dt_modify: data da última alteração do problema,

Dt_close: data de encerramento do problema,

Dt_open: data em que o problema foi registrado no sistema,

Det_ticket: descrição do problema em detalhes. Tem representatividade histórica.

Todos os processos de casamento entre um caso e um problema apresentado pelo usuário através de um título e descrição podem ser acompanhados pela figura 7.8 .

Feita a solicitação de pesquisa por um caso similar, primeiramente é realizada uma quebra nas frases digitadas e são analisadas somente as palavras dos campos Título e Descrição, que não estão registradas na tabela Palavras.

Esta tabela existe como alternativa para pular uma busca sintática por palavras não representativas em um registro, como do uso preposições, conjunções e pronomes, sendo que, quem define o que poderá ser excluído, é o próprio EC.

A primeira busca é feita por comparação de palavras (análise sintática) dentro das tabelas representativas das descrições de casos e seus atributos. Se algum casamento for detectado, o próprio objeto ou o objeto relacionado com o atributo encontrado passa a ser registrado em uma tabela temporária, onde ficará acrescentado seu peso de similaridade sempre que alguma relação sua com a descrição do usuário possa ser identificada. Desta forma, disputa colocação com os outros objetos candidatos.

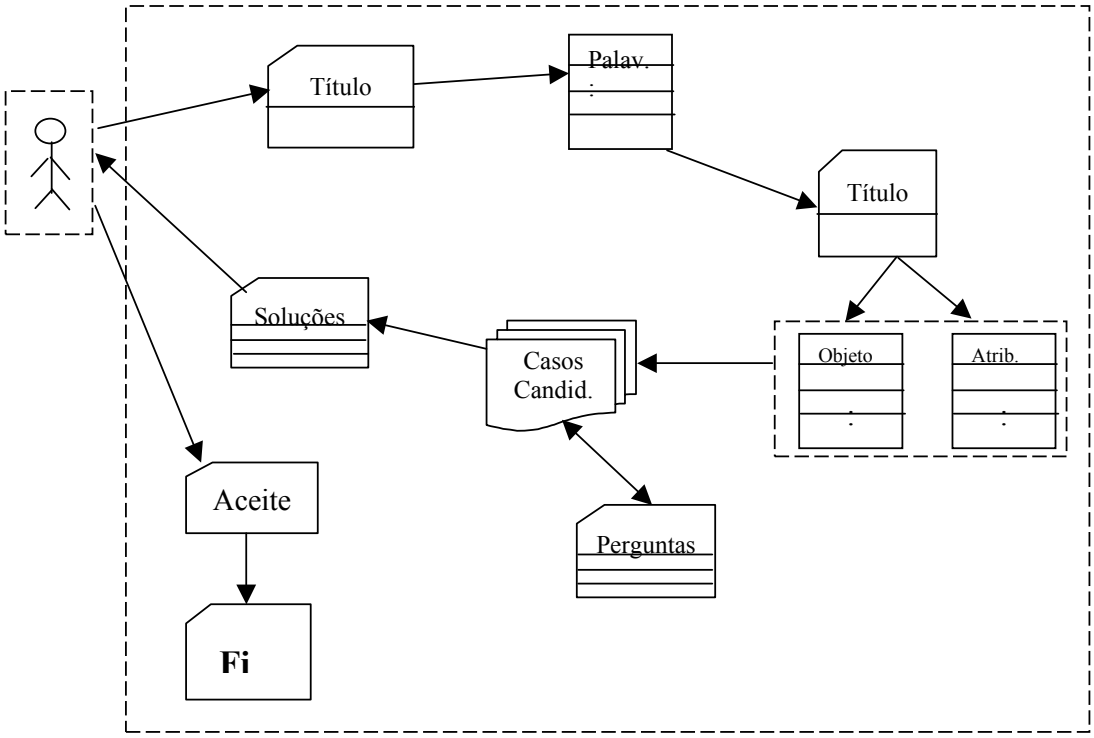


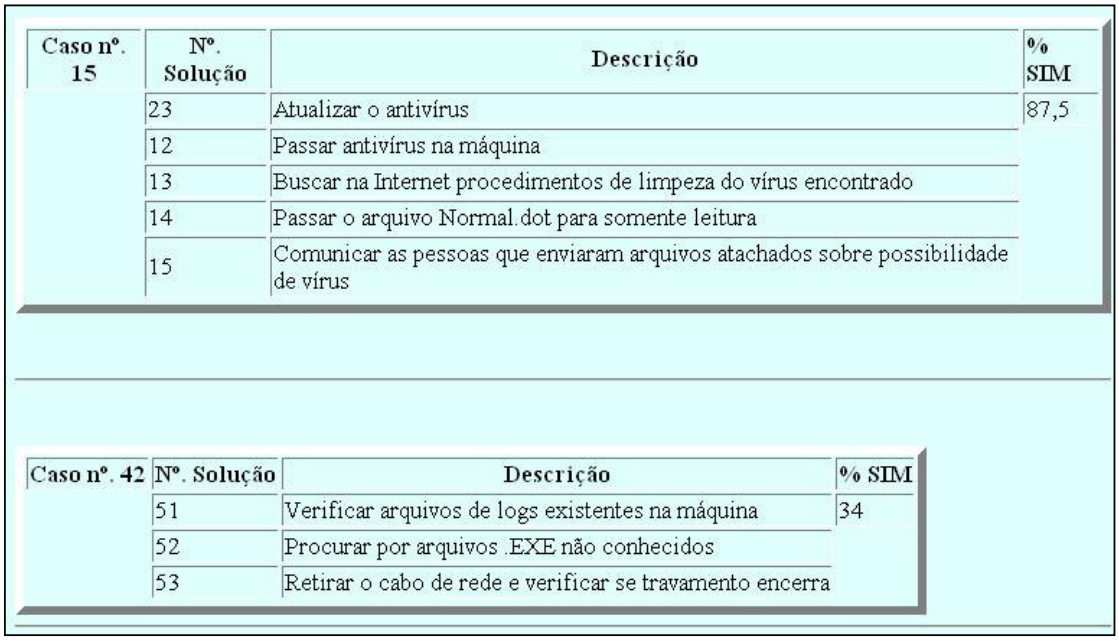
FIGURA 7.8 - Processo de consulta de registro de problema.

Se os casamentos neste estágio não forem nulos, serão buscadas as soluções para os objetos candidatos e apresentadas ao usuário por ordem de somatório de pesos, conforme tela na figura 7.9. Caso ele não fique satisfeito com o que lhe foi sugerido, poderá refinar sua consulta através de respostas às perguntas que serão apresentadas, onde tem início a segunda forma de busca.

A utilização de perguntas é uma forma de acrescentar características ao problema proposto e torna a procurar por soluções mais rápidas e simples. Isto pois, fornecem índices capazes de guiar o raciocinador através da hierarquia da base, por manterem uma relação direta de suas respostas com as características de um caso.

Para evitar que perguntas não necessárias sejam apresentadas e, conseqüentemente, não respondidas, as perguntas são dispostas respeitando algumas ordens de classificação:

- ordem de pesos por casamento parcial com um determinado caso;
- ordem de perguntas que obtiveram maior número de respostas quando apresentadas ao usuário;
- ordem de pesos determinados pelo EC. São as perguntas que não se classificaram nas duas opções acima.



Caso nº. 15	Nº. Solução	Descrição	% SIM
	23	Atualizar o antivírus	87,5
	12	Passar antivírus na máquina	
	13	Buscar na Internet procedimentos de limpeza do vírus encontrado	
	14	Passar o arquivo Normal.dot para somente leitura	
	15	Comunicar as pessoas que enviaram arquivos atachados sobre possibilidade de vírus	

Caso nº. 42	Nº. Solução	Descrição	% SIM
	51	Verificar arquivos de logs existentes na máquina	34
	52	Procurar por arquivos .EXE não conhecidos	
	53	Retirar o cabo de rede e verificar se travamento encerra	

FIGURA 7.9 - Resultado de busca por similaridade na descrição do usuário.

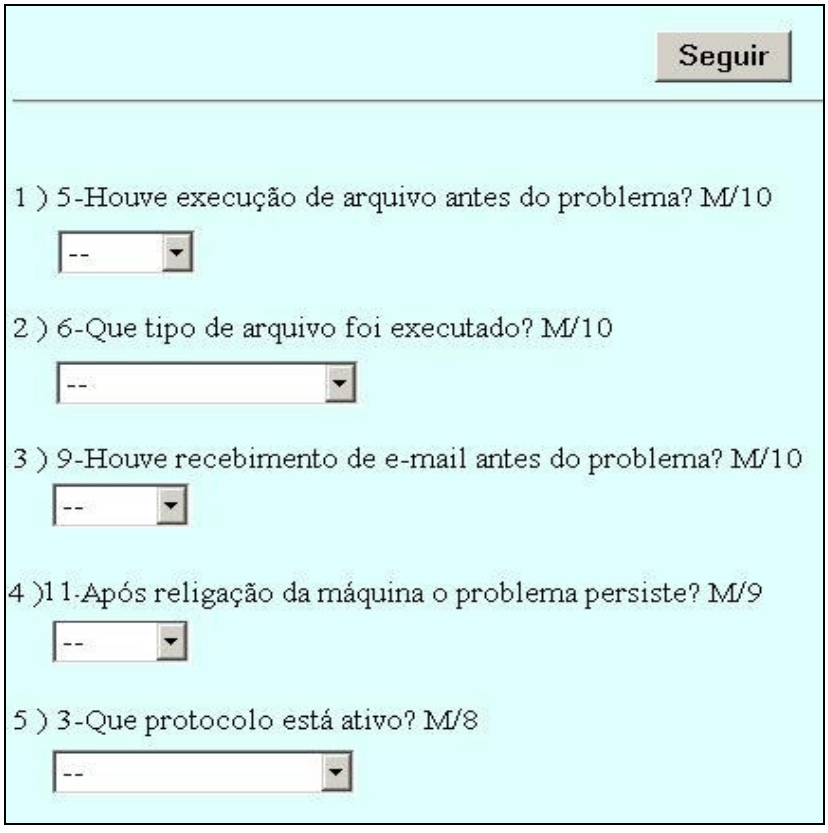


FIGURA 7.10 - Relação de perguntas expostas ao usuário.

O usuário responde às perguntas sem ordem ou obrigatoriedade, com respostas fixas, lógicas ou descritivas, que terão relevância se compreenderem atributos de casos. A tela de perguntas pode ser vista na figura 7.10.

Cada pergunta, ao ser respondida, recebe um acréscimo de peso, tem sua relação com um caso avaliada e recebe os mesmos tratamentos que a primeira forma de busca; somente não contabiliza os pesos no mesmo percentual que a anterior. As respostas e pesos acrescidos ficam registrados em uma tabela com a seguinte descrição:

Id	id_resp	id_atrib	atrib	Peso_resp	tipo_resp

Id : valor seqüencial identificando internamente o registro de resposta;

Id_resposta: valor seqüencial identificando internamente a resposta;

Id_atrib: valor seqüencial identificando internamente o valor da resposta na tabela de características;

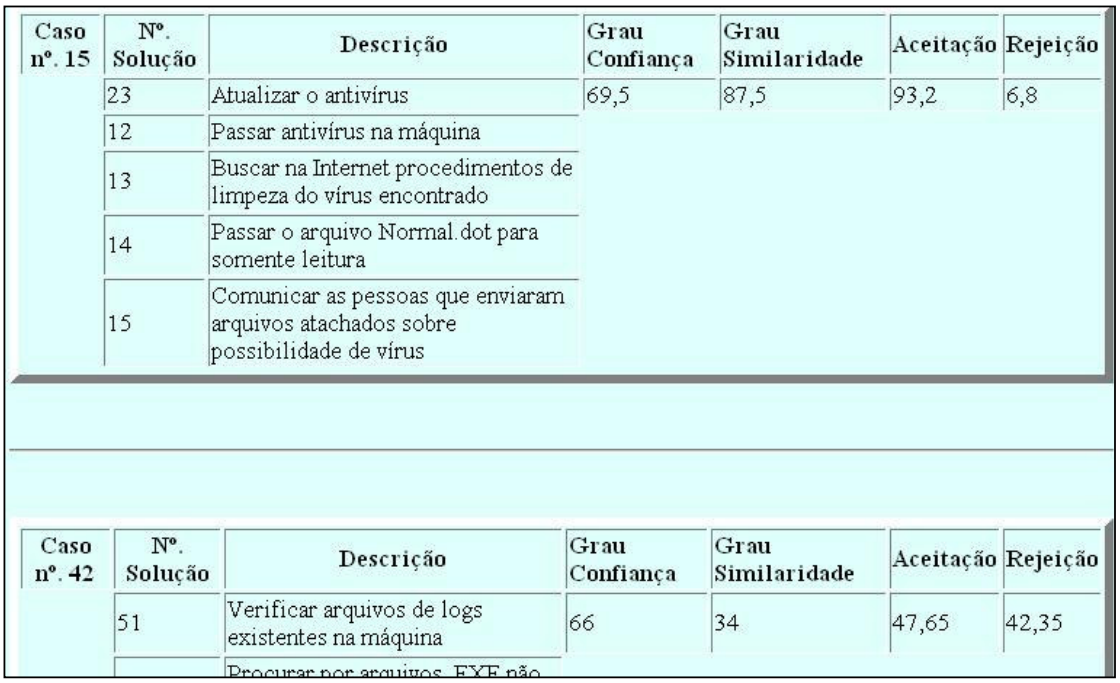
Atrib: descrição da resposta;

Peso_resp: peso da resposta para aquela pergunta;

Tipo_resp: tipo da resposta, se lógica, descritiva ou múltipla escolha.

Os pesos das respostas são denominados pelo EC ao incluir as características de um caso e as perguntas com suas respostas. Para cada resposta obtida, ocorre um acréscimo de cinco pontos no peso de um objeto (caso), com excessão de uma resposta lógica, onde cinco pontos são dados para respostas NÃO e dez pontos para respostas SIM. Considera-se, assim, que todo encontro de características deverá somar no mínimo cinco pontos ao peso total de um caso.

Embora o objetivo das duas formas de procura pelo melhor casamento entre problema proposto e caso armazenado localize-se em encontrar as melhores soluções para um tipo de problema, ambas recebem classificações e nomes diferentes, como pode ser observado na tela da figura 7.11. Para a primeira busca, é dado o nome de busca por similaridade e contabiliza o grau de similaridade resultante da descrição do usuário. A segunda recebe o nome de grau confiança, onde o melhor caso é encontrado por análise semântica através das respostas fornecidas às perguntas.



Caso nº. 15	Nº. Solução	Descrição	Grau Confiança	Grau Similaridade	Aceitação	Rejeição
	23	Atualizar o antivírus	69,5	87,5	93,2	6,8
	12	Passar antivírus na máquina				
	13	Buscar na Internet procedimentos de limpeza do vírus encontrado				
	14	Passar o arquivo Normal.dot para somente leitura				
	15	Comunicar as pessoas que enviaram arquivos atachados sobre possibilidade de vírus				

Caso nº. 42	Nº. Solução	Descrição	Grau Confiança	Grau Similaridade	Aceitação	Rejeição
	51	Verificar arquivos de logs existentes na máquina	66	34	47,65	42,35
		Procurar por arquivos .EXE não				

FIGURA 7.11 - Resultado de uma busca refinada.

Sempre que um processo de busca é disparado para um determinado registro de problema e alguma solução encontrada, esta pode ser aceita ou rejeitada.

Cabe ao consultor de um problema aceitar ou não as soluções expostas pelas duas formas de busca. Se forem rejeitadas, o problema permanece em uma tabela temporária com o *status* "aberto", não mantendo relação com nenhum caso da base.

Se as soluções forem aceitas, o problema adquire data de encerramento e *status* "fechado", passando a fazer parte de uma tabela definitiva que representa a base de registros de problemas. Mas se, para corrigir o problema, alguns outros passos foram necessários e não são encontrados nas soluções recuperadas, pode-se optar por acrescentar soluções onde o usuário seleciona, dos casos que lhe foram recuperados, qual o que melhor adaptou-se a sua realidade e nele faz os acréscimos desejados.

Toda vez que um caso recuperado é aceito ou rejeitado, uma avaliação do aproveitamento das soluções é feita internamente e apresentada como índice de aprovação e rejeição quando uma solução é recuperada, como pode ser visto na figura 11. Esta medida serve para auto-avaliar o aproveitamento do protótipo nas soluções de problemas e dar uma maior certeza na tomada de decisão do usuário ao receber suas respostas. Assim, o conjunto de casos recuperados pode, muitas vezes, indicar os melhores casos dentre os casos do protótipo, embora possa não representar a melhor situação possível.

7.6 Avaliação do Protótipo

Segundo Watson [WAT 97], testar uma base de casos compreende dois processos:

- verificação – refere-se à correta construção do sistema, garante as respostas corretas;

- validação – refere-se à construção do sistema correto, o sistema é aquilo que o usuário espera dele.

Como o trabalho desenvolvido destina-se a, além de unir as metodologias descritas nos capítulos 2 e 4, a proporcionar que a base de casos fique independente do domínio de problema tratado, a avaliação do protótipo será centrada mais na verificação do que na validação. A avaliação sendo que a segunda é descrita com finalidades de comprovar a primeira avaliação, ou seja, o que será avaliado é a metodologia aplicada e não as representações completas de um domínio de problema.

As etapas de verificação foram divididas em três fases, as quais são essenciais para atender as exigências das metodologias aplicadas e por serem pontos propostos como resultados deste trabalho. Estas etapas são descritas abaixo e comentadas nas próximas seções:

- capacidade de representação formal dos casos;

- avaliação da máquina de inferência;

- uso do protótipo para vários domínios de problema.

Os domínios selecionados para estudo incluem essencialmente o domínio de problemas de segurança em redes de computadores, como justificado na seção 4.2; e um menos complexo, e comum a qualquer ser humano compreendido pelo nome de "frutas". São dois domínios cujas características não podem ser comparadas, sendo que o primeiro requer um estudo aprofundado de problemas e o segundo não.

As bases de casos podem ser divididas em: homogêneas, quando todos os casos compartilham a mesma estrutura de registros, e heterogêneas, quando os casos apresentam estruturas de registros variados, com diferentes atributos e valores. O domínio de segurança, por exemplo, é tido como um domínio possuidor de uma base heterogênea, pois a qualificação e diversificação de situações são inúmeras e certamente difíceis de serem todas previstas. Levando-se em conta o problema de tempo e a heterogeneidade de representação do domínio de segurança, somente uma pequena amostragem de problemas será tratada nesta validação. O segundo domínio de problema não terá um detalhamento maior, por compreender um domínio homogêneo e de conhecimento não determinístico de especialistas.

A instalação do protótipo consumiu aproximadamente 400 kbytes de espaço em disco do servidor Linux, onde foi instalado.

7.6.1 Capacidade de Representação Formal dos Casos

O objetivo de verificação deste item é centrado na inclusão das redes semânticas. Nele será avaliada a capacidade do protótipo em representar formalmente as descrições das redes de problemas.

Os casos inseridos compreendem seis casos e representam situações de problema, conforme descrição completa feita no capítulo 5, tendo suas redes semânticas mostradas no Anexo 2. Os tipos de problemas compreendem as classes de problemas de segurança citadas na tabela 7.2 e são selecionadas pelo grau de suas ocorrências em um domínio de segurança, segundo orientações do CERT e NIC-BR, como demonstrado no gráfico 4.1 e pela categorização dos tipos de ataques feito por Stalling [STA 95], comentados no item 5.2.

TABELA 7.2 - Relação das classes de casos testados.

Classe de Problema	Nome do Problema – Ataque	Quantidade
Axfr	Pacote BIND	1
Af (ataque ao usuário final) – *Backdoor*	*Back orifce*	1
Aw	Vulnerabilidade de CGI	1
DoS	*Land*, *Winnuke*	2
Vírus	Vírus	1

Inicialmente, obedecendo à metodologia de criação do protótipo, incluem-se as classes de problemas (objetos) definidas na tabela acima. Em seguida, foram incluídas as características em forma de atributo e valor para cada classe, sendo que para classes onde tais características se repetiram, a relação foi incluída sem necessitar novas inclusões de valores. Portanto, a avaliação do primeiro item de verificação foi avaliado como perfeitamente atendido, pois o protótipo apresentou ambiente propício para receber todas as definições feitas.

Para que as próximas avaliações possam ser realizadas, foram cadastrados os ítens que completam o conhecimento do domínio, como as perguntas e respostas a serem apresentadas em caso de refinamento de pesquisa e as soluções para os casos, extraídas da literatura citada acima.

7.6.2 Avaliação da Máquina de Inferência

O objetivo deste teste no protótipo dá-se pela necessidade de comprovar se as redes semânticas criadas são capazes de definir os problemas a serem representados e se o sistema é suficientemente desenvolvimento para raciocinar sobre as características descritas por uma consulta e dela retornar um casamento entre o problema proposto e um caso armazenado na base de casos.

Conforme orientações de Ian Watson [WAT 97], foi consultado um especialista na área de segurança para avaliar as representações de dos problemas de segurança demonstrados no Anexo 2. De posse das corretas descrições das redes semânticas, definem-se os graus de relevância para cada caraterística, também sob supervisão deste especialista.

Segundo o mesmo autor, uma correta verificação de uma base de casos bem cadastrada e de um sistema bem desenvolvido se dá por testes onde se deve perguntar

algumas questões após consultas realizadas e considerar uma resposta de 1 a 6, onde 1 = totalmente inaceitável e 6 = perfeito. Sendo as perguntas as seguintes:

- O sistema recuperou um caso satisfatório ou um conjunto de casos?
- A recuperação deu-se em um espaço de tempo aceitável?

Neste item de avaliação serão considerados, portanto, o grau de recuperação, tempo e recursos necessários a esta tarefa.

O funcionamento correto da máquina de inferência é dado pela avaliação da recuperação de problemas, tarefa extremamente influenciada pela correta atribuição de graus de similaridade às características relevantes de cada caso. Os graus estipulados podem ser identificados no Anexo 3, os quais foram gerados pelo conhecimento do EC no domínio de problema.

A forma mais coerente de saber se as características cadastradas e a máquina de inferência são suficientes para responder pela similaridade é através de consultas feitas ao protótipo. Para cada consulta buscou-se uma classe de problema diferente, como demonstrado na tabela 7.2, a fim de validar as corretas inclusões tanto da forma de cadastro quanto das características e seus pesos.

Como o domínio de problema é muito heterogêneo e somente alguns casos são tratados para a validação do protótipo, teve-se o cuidado de consultar problemas cujas classes já tivessem sido incluídas na BC, pois o que se está buscando neste teste é o comportamento da máquina de inferência.

Os problemas consultados estão registrados no Anexo 4, juntamente com os casos que a consulta recuperou.

A descrição do problema consultado número 12, por exemplo, relatava a interrupção de serviços em algumas máquinas em uma sub-rede. No mundo real, isto foi detectado como um problema de *winnuke* causado por maus usuários de um laboratório de informática.

O caso recuperado número 5 corresponde a um ataque da categoria de Interrupção, causado por uma falha de serviço do tipo DoS. Onde um sistema operacional *windows*, ao receber em uma de suas portas uma *strings* com OOB, não suporta o pacote, causando problemas na pilha de TCP/IP e tendo como resultado a paralisação de alguns serviços de rede. Conclui-se portanto, que a recuperação inicial foi satisfatória, tendo em vista a quantidade de informação disponibilizada na descrição do problema e o casamento efetuado.

A relação identificada na primeira consulta entre o caso candidato e o problema proposto após o refinamento ainda se manteve, resultando em um grau de confiança que garante que ambos compreendem a mesma classe de problema.

Após o refinamento, o caso candidato nº 6 não foi mais classificado, pois nenhuma pergunta respondida mantinha relação com suas características, perfazendo um total muito pequeno de confiança, o que resulta em descarte por parte do protótipo. Este descarte também é feito para muitos casos que poderiam compor os candidatos por

possuírem o atributo *windows,* como pode ser observado no Anexo 3, mas como seu grau de similaridade é muito baixo, não aparecem como candidatos.

Fazendo-se algumas alterações nas respostas às perguntas, observam-se alterações de raciocínio da máquina de inferência. Ao se responder à pergunta nº 6 , a qual não obteve resposta no primeiro refino, e alterando-se as respostas das perguntas de nº 11 e 6, outro caso é considerado como melhor casamento para o problema proposto. A máquina de inferência passa o contexto do problema de um DoS para um Vírus através das características informadas. Observa-se, claramente, que o contexto do problema é sensível às informações coletadas na descrição do problema proposto. Para a alteração deste exemplo, o caso de melhor casamento seria feito com o caso de nº 6 e não mais com o de nº 5 .

O exemplo de problema nº 13, por sua vez, corresponde a um problema com a performance do equipamento, causado por algum tipo de arquivo executado. A recuperação trouxe três tipos de casos, os quais obtiveram relação considerável com a descrição do problema. Ao analisar-se os casos candidatos da primeira busca, percebe-se que todos os recuperados tinham relação com execução de arquivos, ou seja, com uma característica marcante da descrição do problema, portanto recuperação válida. Após o refinamento observa-se que um dos candidatos, ou seja, o da classe de DoS, não é mais registrado, pois a medida de confiança dele passa a ser tão pequena que o sistema descarta-o, restando apenas os outros dois tipos iniciais, sendo que o que apareceu com maior cotação na primeira busca permaneceu em primeiro. Esta decisão baseou-se no fato de ter sido constatado que após a religação do equipamento o problema persistia, o que é uma característica observada em problemas de vírus, raciocínio condizente com informações disponibilizadas e armazenadas.

Conclui-se, portanto, que a máquina de inferência está reagindo conforme as características estão sendo registradas e, como pode ser verificado nos casos testados, as respostas foram condizentes com a realidade. Embora o conhecimento do domínio esteja pouco representado na BC, as consultas feitas recuperaram um satisfatório grupo de casos, todos com alguma relação real ao que estava sendo pesquisado ou descrito.

O tempo gasto nas consultas realizadas foi normal, nota-se que ele tende a ser maior quando realiza a primeira forma de busca. Este problema pode ser resolvido conforme são cadastradas palavras que não devam ser buscadas na BC e com descrições mais centralizadas.

Concluindo-se este item, as notas dadas as perguntas de Watson seriam:

- O sistema recuperou um caso satisfatório ou um conjunto de casos? **5**

- A recuperação deu-se em um espaço de tempo aceitável? **3**

Devido à pequena amostragem feita por motivos de tempo e complexidade do domínio, fica impossibilitado o teste de velocidade de recuperação versus tamanho da base.

7.6.3 Uso do Protótipo para Vários Domínios de Problema

O protótipo mostrou-se eficiente na representação e recuperação de problemas para o primeiro domínio de problema; portanto, nesta validação será testado outro domínio, para analisar se ele está servindo a vários domínios de problema sem que seja necessário fazer alterações na parte de programação.

Conforme citado no início desta seção, este segundo domínio, chamado de domínio de "frutas", não requer conhecimento específico acerca dos problemas registrados, por ser de conhecimento geral.

A representação deste domínio é feita, também, por redes semânticas, tendo suas características relevantes demonstradas no Anexo 3 e suas consultas no Anexo 4.

O cadastro de características para cada caso é feito da mesma forma que para o domínio de segurança, portanto não se faz necessário um detalhamento maior.

Para os problemas que serviram de exemplo para este domínio, buscou-se recuperar em primeira instância uma fruta que fosse gostosa e redonda. A BC tinha cadastrado apenas duas frutas que apresentam a característica "arredondada"; portanto, ambas foram recuperadas na primeira busca. Ao fornecer-se mais informações através do refino de pesquisa, indica-se que a fruta que se busca é cultivada no Norte. Como esta característica somente está presente na fruta Laranja, ela obtém o maior grau de confiança com o problema proposto, ficando o caso 1 (maçã) como segunda alternativa por possuir todas as outras características informadas no problema.

Seguindo-se as definições e metodologias propostas por este sistema, verificou-se que, alterando-se as características registradas nas tabelas que compõem a base de casos, pode-se, com o mesmo sistema, representar diferentes tipos de domínios, pois a relação de similaridade é dada pelas características que serão inseridas no sistema e não pela forma como este foi programado.

8 Conclusão

A transmissão de conhecimento otimiza a realização de tarefas. O problema é encontrar formas de replicar, fácil e rapidamente, o conhecimento de um especialista acerca de um domínio.

Um profissional torna-se especialista quando consegue adquirir informações suficientes que lhe forneçam aprendizado, capacidade de entender e solucionar problemas de um determinado assunto, ou seja, quando coleciona ensinamentos por suas experiências. Este especialista passa a ser referência em soluções a problemas e, conseqüentemente, muito solicitado. Ele, ao resolver um problema, ensina o que fazer, mas não como raciocinar para chegar à solução, muitas vezes não repassando os passos que originaram esta solução. Outra pessoa, ao realizar novamente a mesma correção, necessitará do especialista ou deverá aprender todo o caminho que possa levar a uma possível solução. Caminho, este, já percorrido pelo especialista.

Raciocínio baseado em casos é uma metodologia que busca raciocínio da mesma forma que especialistas, ou seja, pela recuperação de informações armazenadas que tenham alguma relação com o que está sendo procurado.

Diversos serviços ou tarefas de especialistas, para serem considerados eficientes e eficazes, devem responder por alguns fatores como: tempo de resposta a problemas, qualidade do serviço oferecido e preço. Alguns locais de trabalho sentem estes fatores mais fortemente que outros, como por exemplo os centros de gerenciamento de redes onde o enfoque maior é dado no tempo de resposta. Para estes locais, qualquer minuto é vital aos sistemas e comunicações em trânsito. Cada vez mais há a necessidade de ótimos serviços nas tarefas de configuração, manutenção e gerenciamento de falhas.

Fornecer segurança a uma rede de computadores e sistemas requer profissionais bem preparados e exige um conhecimento mais aprofundado sobre o funcionamento da estrutura de rede. Muitos problemas ocorridos na área de segurança podem ser evitados se os profissionais souberem onde estão centradas suas vulnerabilidades. Em muitos casos, isto somente será conhecido após uma notificação de ataque ou tentativa. Baseando-se no fato de que a maioria dos ataques e invasões seguem um determinado padrão de atuação, estes padrões podem ser amigavelmente armazenados em uma base de dados e utilizados dinamicamente na aprendizagem de atividades intrusas em uma rede e sistemas.

Fortes indícios para atingir este objetivo estão voltados à inclusão de sistemas que utilizem inteligência artificial, mais especificamente, raciocínio baseado em casos. Esta é uma área relativamente nova, mas já ganhou seu espaço junto aos pesquisadores de IA. Ela tem sido usada, com bom proveito, para resolver problemas nas mais diversas áreas do conhecimento humano. Como exemplos podem ser citados: mediações de disputa – MEDIATOR [KOL 93], planejamento de refeições – JULIA [KOL 93], monitoramento de poluição do ar – AIRQUAP [LEK 94] e planejamento de processos para fabricação de peças – PROCASE [YAN 94].

Este trabalho descreveu uma proposta de apoio à problemas que, independente do domínio, possibilita a aquisição e modelagem de conhecimento de um especialista para que, de maneira fácil e rápida, possa ser reutilizada por outras pessoas que tenham

problemas semelhantes no mesmo domínio. Os objetivos foram atingidos pela utilização da metodologia de raciocínio baseado em casos (CBR) aplicado sobre sistemas de registros de problemas (STT).

Primeiramente, foi estudada a metodologia de raciocínio baseado em casos – capítulo 2, o funcionamento de sistemas de registros de problemas – capítulo 3, e o que existe de ferramentas com objetivos semelhantes – capítulo 4. Este estudo necessitou ser aplicado sobre um domínio de problema; o escolhido foi o gerenciamento de segurança em redes de computadores e sistemas, por suas características e necessidades levantadas anteriormente. Assim, um estudo mais aprofundado sobre este ambiente foi exigido e demonstrado no capítulo 5.

Foi então apresentado um mecanismo chamado de SABER que, através de registros de usuários, reutiliza o conhecimento armazenado em sua base de casos, fornecendo casos já ocorridos que apresentam semelhanças com o que está sendo apresentado pelo usuário.

A implementação da proposta é feita através de um banco de dados relacional e um sistema de programação. No banco de dados é implementada a parte de características e suas representatividades para um objeto, enquanto a parte de programação implementa a parte operacional, a adaptação das soluções e a máquina de inferência. A inferência é guiada pelo grau de representatividade das características para um determinado caso na base de casos, obtida por cálculos de similaridade adaptados do algoritmo de vizinhança [KOL 93].

Desta forma, um domínio de problema estará provido de mecanismos complementares para auxílio ao usuário não especialista em tomadas de decisões ou em consultas para aprendizagem, onde se destacam algumas funcionalidades:

- A forma de representação de um problema. A extração de conhecimento através de características relevantes a um tipo de problema e a não inclusão de todas as ocorrências de problemas para um domínio facilitam a sua compreensão e representação;

- A atribuição de diferentes índices de significância, usados para realizar a identificação de um objeto, possibilita raciocinar sobre um problema, o que evita extensas descrições para demonstrá-lo, descrições repetidas e agilidade em sistemas de busca .

- A facilidade de representar domínios de problemas agindo na mesma máquina de inferência. A estrutura de representação dos problemas permite que após feito um levantamento e descrição em redes semânticas de um domínio de problema, este consiga ser armazenado em uma base de casos do sistema SABER, sem necessitar acertos na parte de programação;

- Funcionamento da máquina de inferência. Esta máquina apresenta seu bom funcionamento através dos testes realizados, embora em pequena quantidade devido ao tempo disponível, onde se mostrou que a alteração de informações fornecidas pelo usuário podem representar re-direcionamentos na identificação e solução de problemas ;

- Auto-avaliação de relações entre objetos e perguntas criadas pelo EC. As estatísticas do aproveitamento das perguntas nas buscas por problemas semelhantes identifica as relações bem sucedidas pelo EC ao relacionar uma característica com um objeto;

- Emissão de nível de aceitação das soluções para os casos recuperados. Contribui para que o usuário, ao receber a identificação do seu problema, antes de realizar as soluções sugeridas, saiba o nível de aceitação destas para um problema anteriormente resolvido.

- Fornecimento de meios de controle do aprendizado para usuários capacitados. O especialista ou EC podem, quando desejarem, ajustar ou alterar dados de representação dos problemas.

As principais dificuldades enfrentadas, além do tempo para desenvolvimento do trabalho integral, foi a dificuldade em manter as relações entre os objetos e suas características durante todo o processo de cálculo de similaridade. A escolha pelo banco de dados relacional dificultou este tipo de relacionamento, o que poderia ter sido solucionado com a utilização de um banco de dados orientado a objeto. Outro problema extremamente relevante é a complexidade e necessidade de conhecimento para se modelar um domínio de segurança em redes e sistemas. Este é um domínio muito heterogêneo e com poucos especialistas no assunto.

O início deste trabalho tinha como objetivo modelar um domínio de segurança em redes de computadores através de raciocínio baseado em casos, porém com um objetivo maior, que seria manter a base de casos totalmente independente do domínio. Durante o desenvolvimento, gastou-se muito tempo com o isolamento da máquina de inferência prejudicando a completa representação do domínio proposto.

Embora sem a obtenção de muitos dados relevantes sobre o sucesso do sistema proposto, devido à quantidade de testes a serem realizados, se corretamente realizadas suas etapas de aquisição e modelagem, pode-se avaliar que ele apresenta capacidade de auxiliar a aprendizagem e tomada de decisão quando as informações fornecidas indicarem a existência de um problema. Assim, o sistema encontra-se pronto para ser testado em um ambiente real.

8.1 Trabalhos Futuros

Alguns pontos poderiam aprimorar e eliminar pontos negativos avaliados neste trabalho, alguns não serão realizados por não constarem no escopo deste trabalho, mas ficam como propostas futuras, podendo-se citar:

- Conversão da base de casos para um banco de dados orientado a objetos;

- Criação de uma forma automática de aquisição para novos casos e para casos sem solução;

- Representação completa do domínio de segurança em redes e sistemas.

- Desenvolvimento de uma *interface* mais amigável para o usuário.

Outros pontos podem ser estabelecidos para desenvolvimentos futuros, como por exemplo:

- Criação de uma estrutura de sinônimos. Realizar inclusões em uma estrutura onde fique representado o significado de atributos de problemas, evitando a existência de ruídos durante a busca por similaridade. Estas estruturas serão baseadas na teoria de listas circulares duplamente encadeadas [TER 91];

- Inclusão de uma opção de *feedback* para o responsável e para o reclamante do registro de problema;

- Possibilidade de inclusão de arquivos ou links de páginas Web como soluções para problemas;

- Integração de domínios. Para cada domínio é criada uma base de casos em separado. Não foi testado o comportamento da máquina de inferência quando os problemas de vários domínios encontram-se descritos na mesma base. Fica como proposta de teste a modelagem do sistema DUMBO [MEL 98] implementada no sistema SABER.

- Realização de testes em um ambiente real com uma base de casos provida de uma grande quantidade de casos armazenados, assim poderá ser avaliado o item *performance*.

Anexo 1 Criação de Base e Tabela de Dados no MySQL

Os comandos descritos abaixo, têm por finalidade a criação de uma base de dados e uma tabela para início da implementação do sistema SABER

Conforme descrito no capítulo 7 (implementação do protótipo) esta criação deve ser feita de maneira manual. Executados os comandos, o sistema pode iniciar desde que, implementado em um servidor com as características de hardware e software descritas no mesmo capítulo. Segue abaixo os comandos necessários:

Base de Dados: conhecimento

Comandos:

Create database conhecimento;

Use conhecimento;

Tabela: SENHA (id_pessoa, usuario, senha) Descrição: Tabela de senhas;

Comandos:

CREATE TABLE SENHA (id_us INTEGER (3) NOT NULL AUTO_INCREMENT,usuario VARCHAR(13) NOT NULL,senha VARCHAR(8) NOT NULL,tipo VARCHAR(5) NOT NULL,mail VARCHAR(25) NOT NULL,tel INTEGER(8) NOT NULL,PRIMARY KEY (id_us));

Observação: tipo poderá ser: "user" ou "adm"

Comandos para inserções na tabela SENHA:

INSERT INTO SENHA VALUES ('1','cinara', 'cinara','adm', 'r4cnr@udesc.br', '23612');

INSERT INTO SENHA VALUES ('2', 'visita', 'visita', 'user', 'fulano@qualquer.br', '581');

INSERT INTO SENHA VALUES ('3','cristina','cris','adm','r4cnr@inf.ufrgs.br', '3486');

Anexo 2 Redes Semânticas do Domínio de Segurança

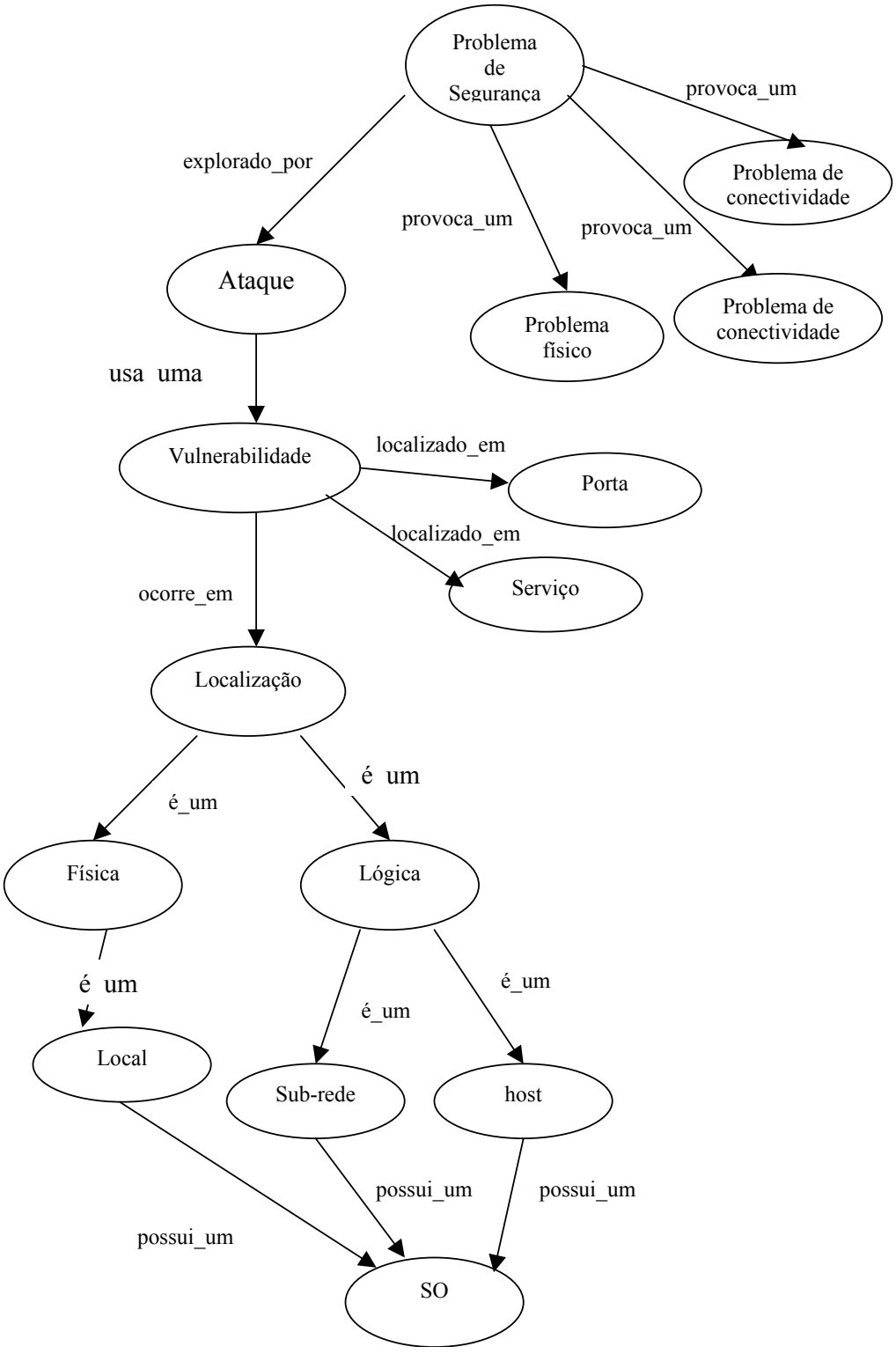


Figura A 2.1. - Rede Semântica Geral de Problemas de Segurança (1)

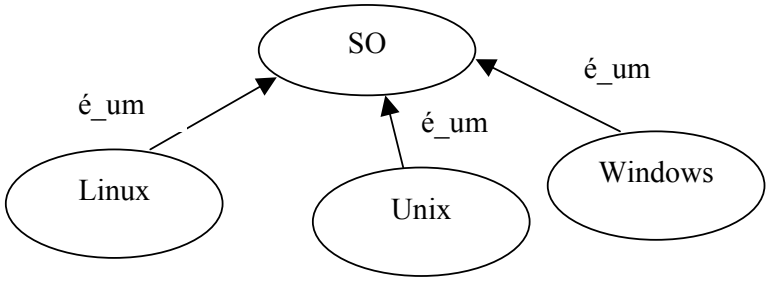


Figura A 2.2. - Rede Semântica Geral de Problemas de Segurança (2)

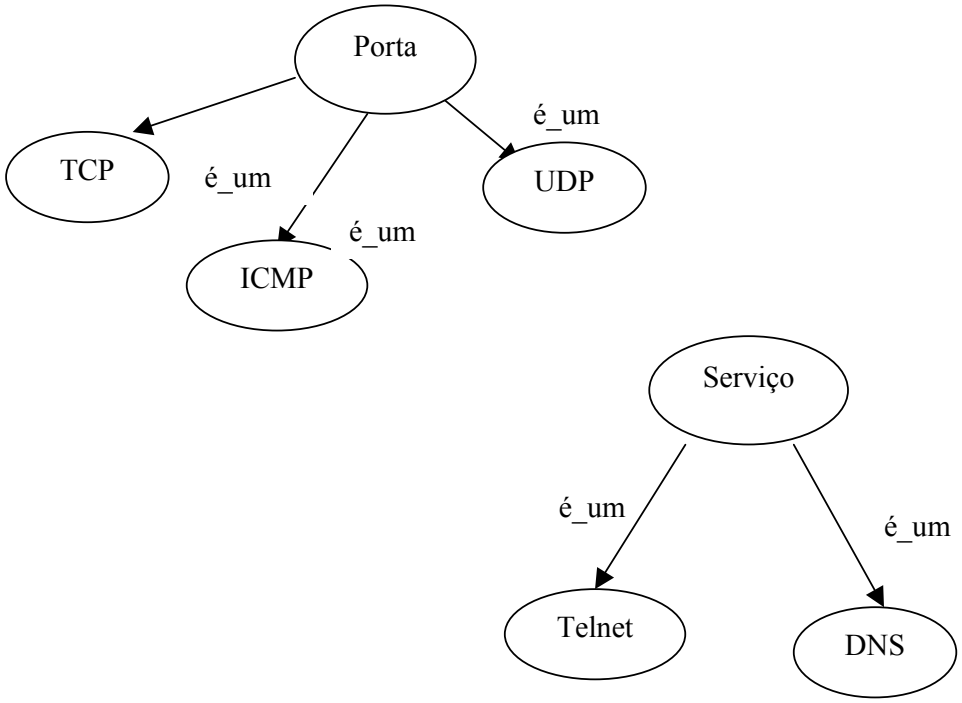


Figura A 2.3. - Rede Semântica Geral de Problemas de Segurança (3)

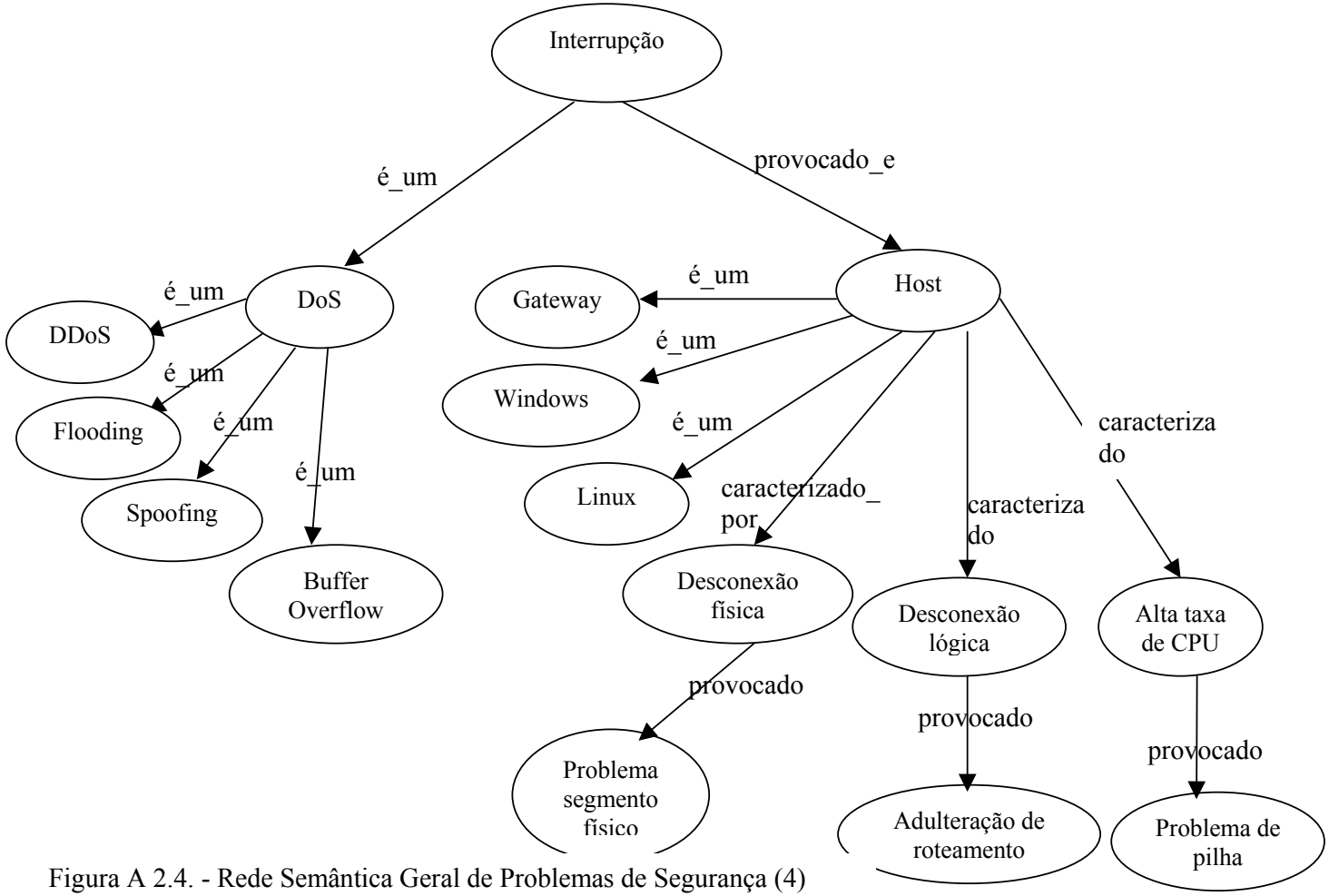


Figura A 2.4. - Rede Semântica Geral de Problemas de Segurança (4)

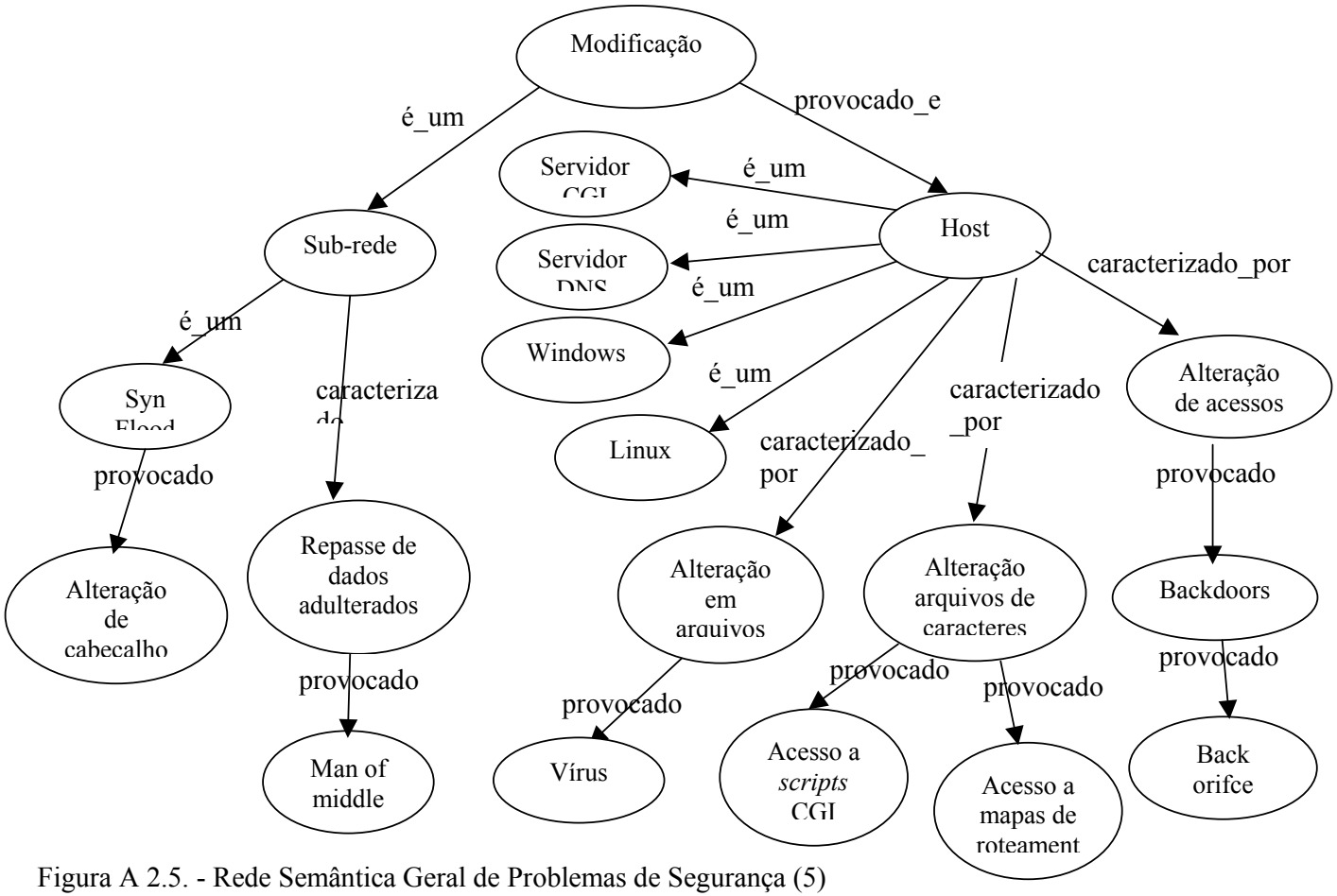


Figura A 2.5. - Rede Semântica Geral de Problemas de Segurança (5)

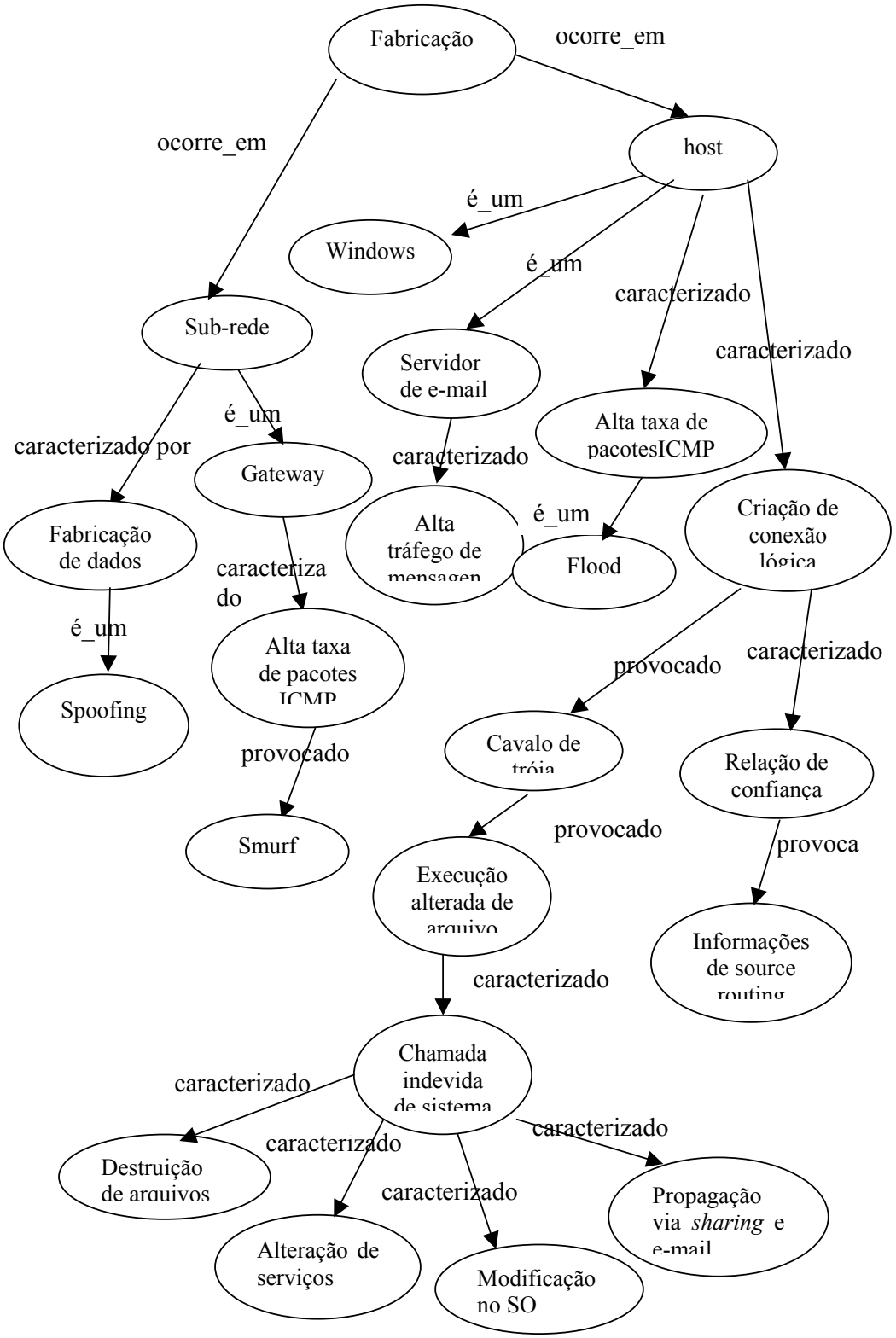


Figura A 2.6. - Rede Semântica Geral de Problemas de Segurança (6)

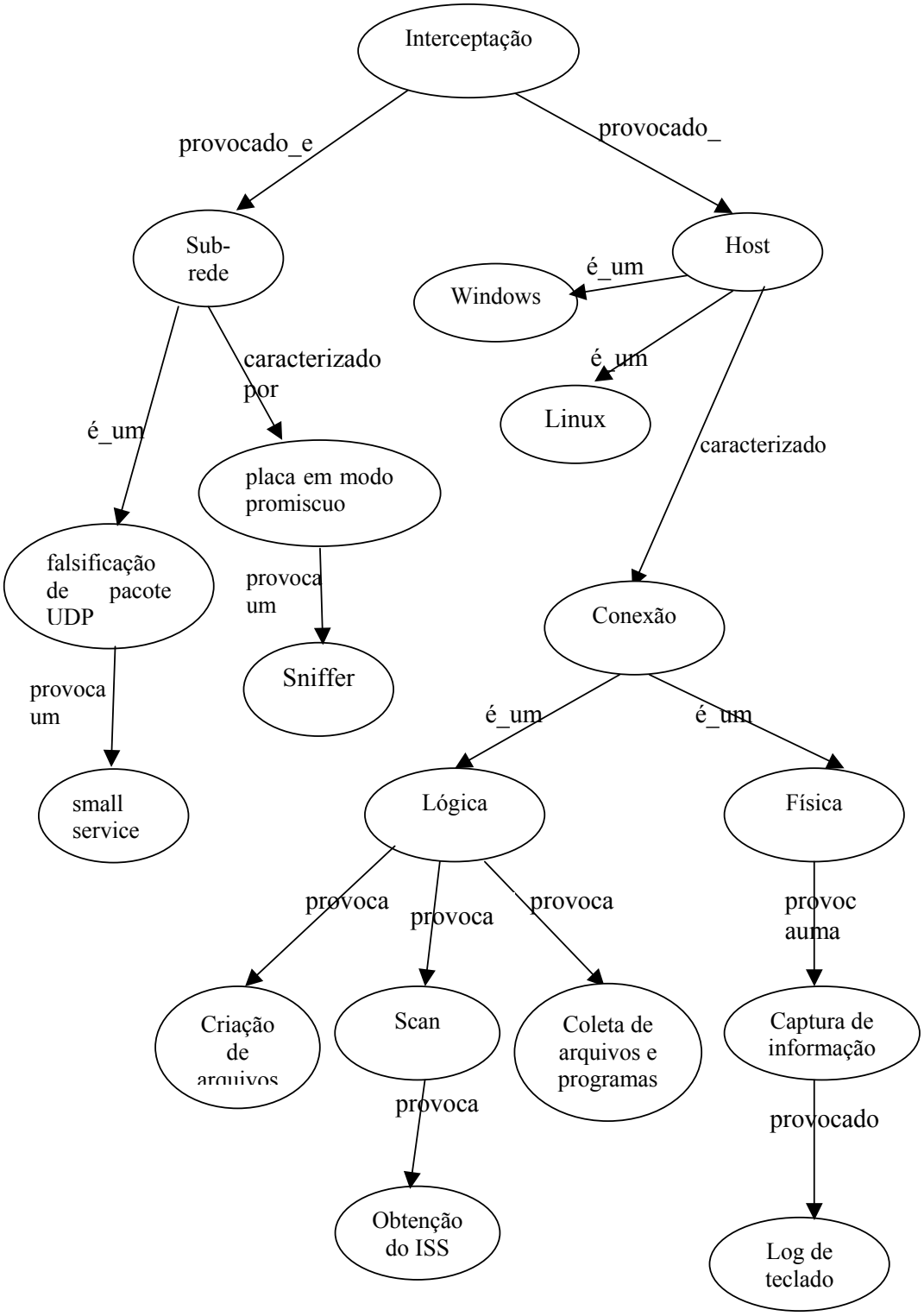


Figura A 2.7. - Rede Semântica Geral de Problemas de Segurança (7)

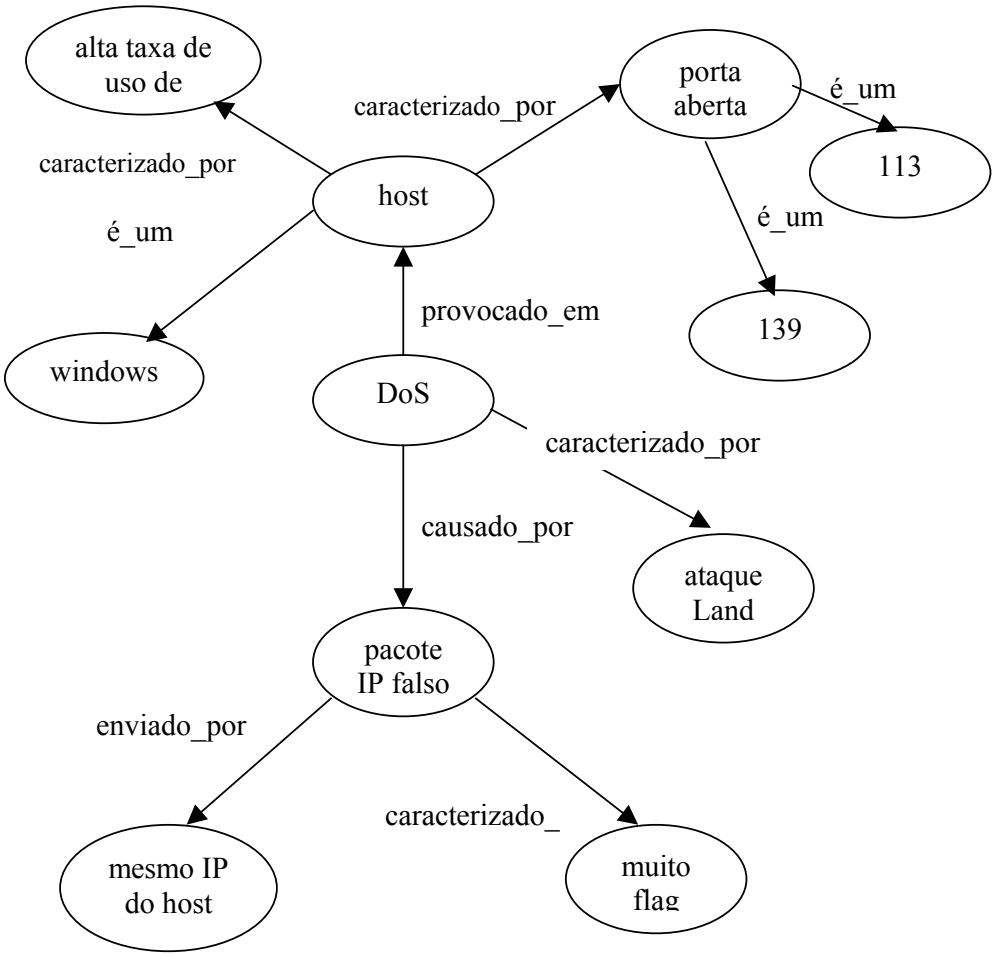


Figura A 2.9. - Rede Semântica Geral de Problemas de Segurança -Land

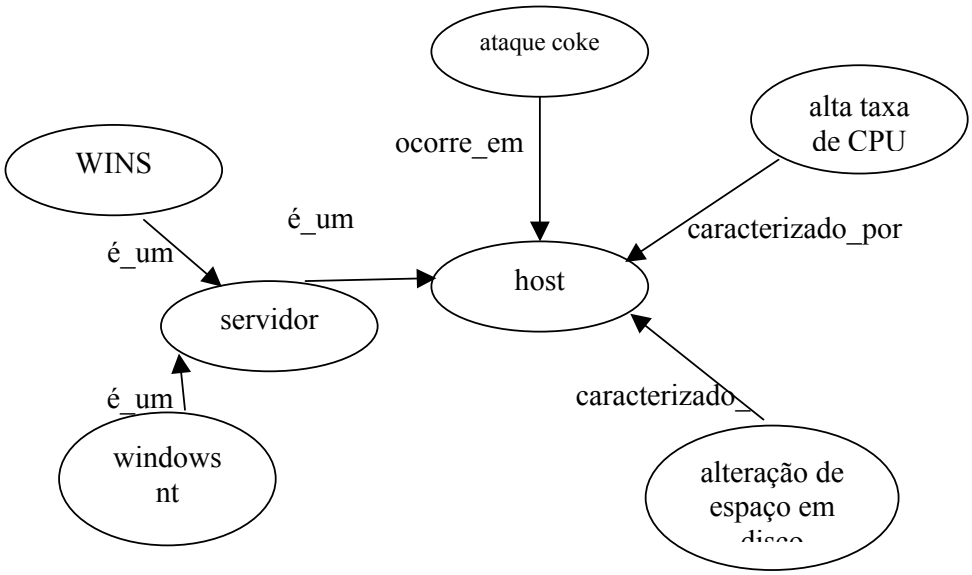


Figura A 2.8. - Rede Semântica Geral de Problemas de Segurança - Coke

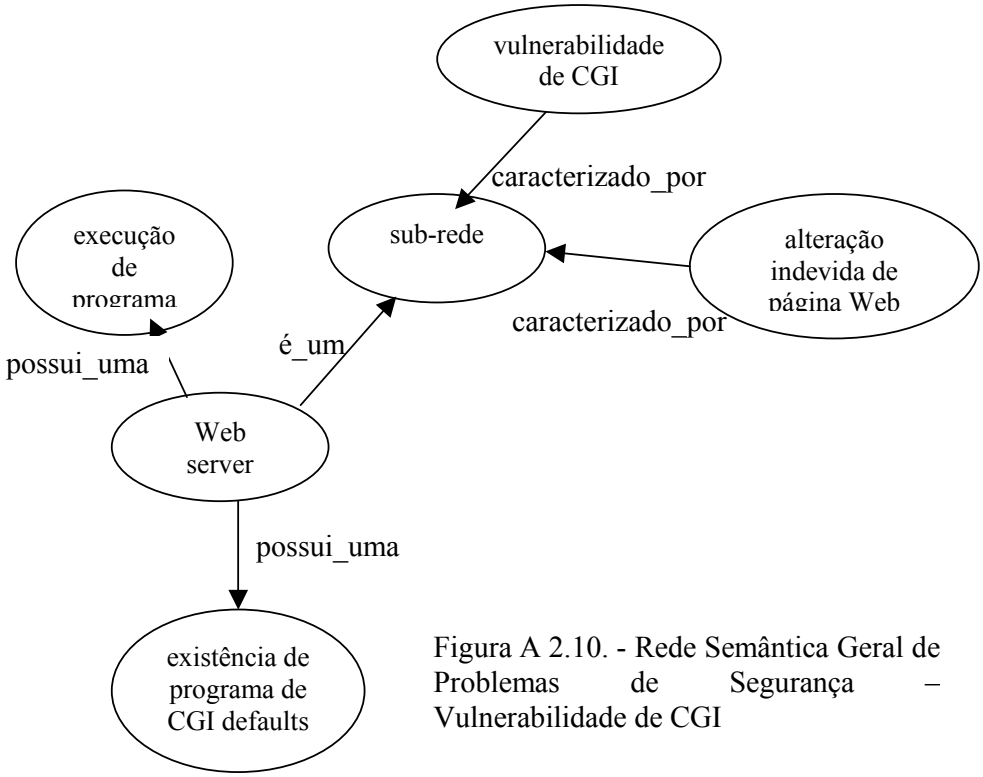


Figura A 2.10. - Rede Semântica Geral de Problemas de Segurança – Vulnerabilidade de CGI

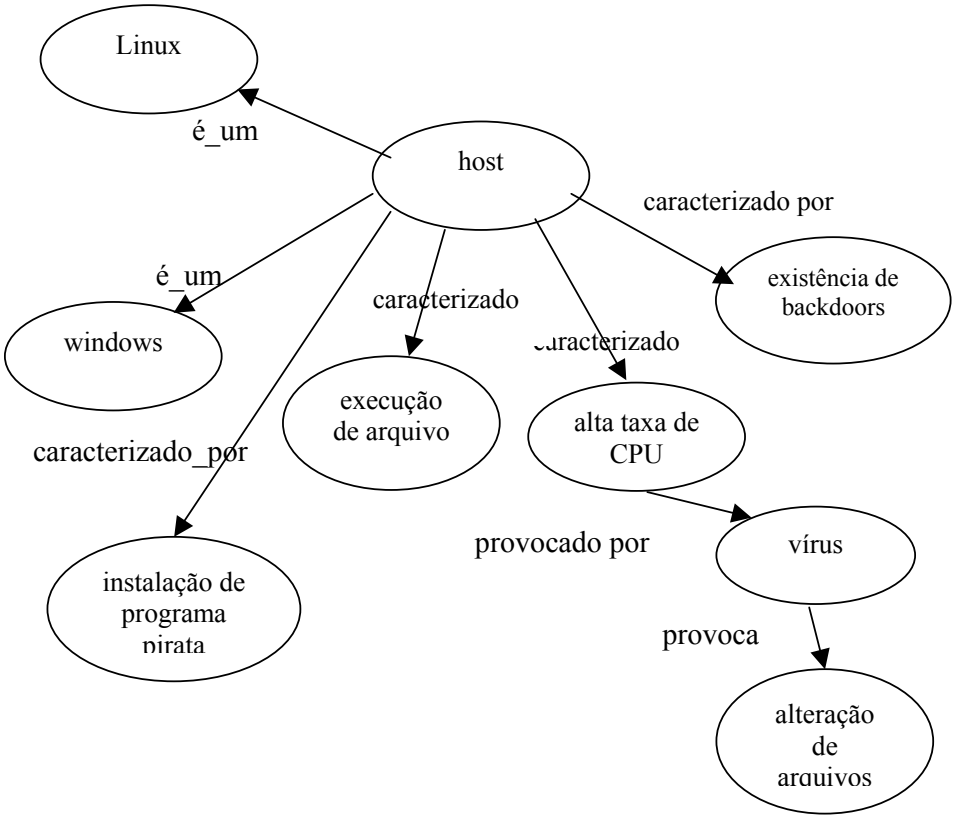


Figura A 2.11. - Rede Semântica Geral de Problemas de Segurança - Vírus

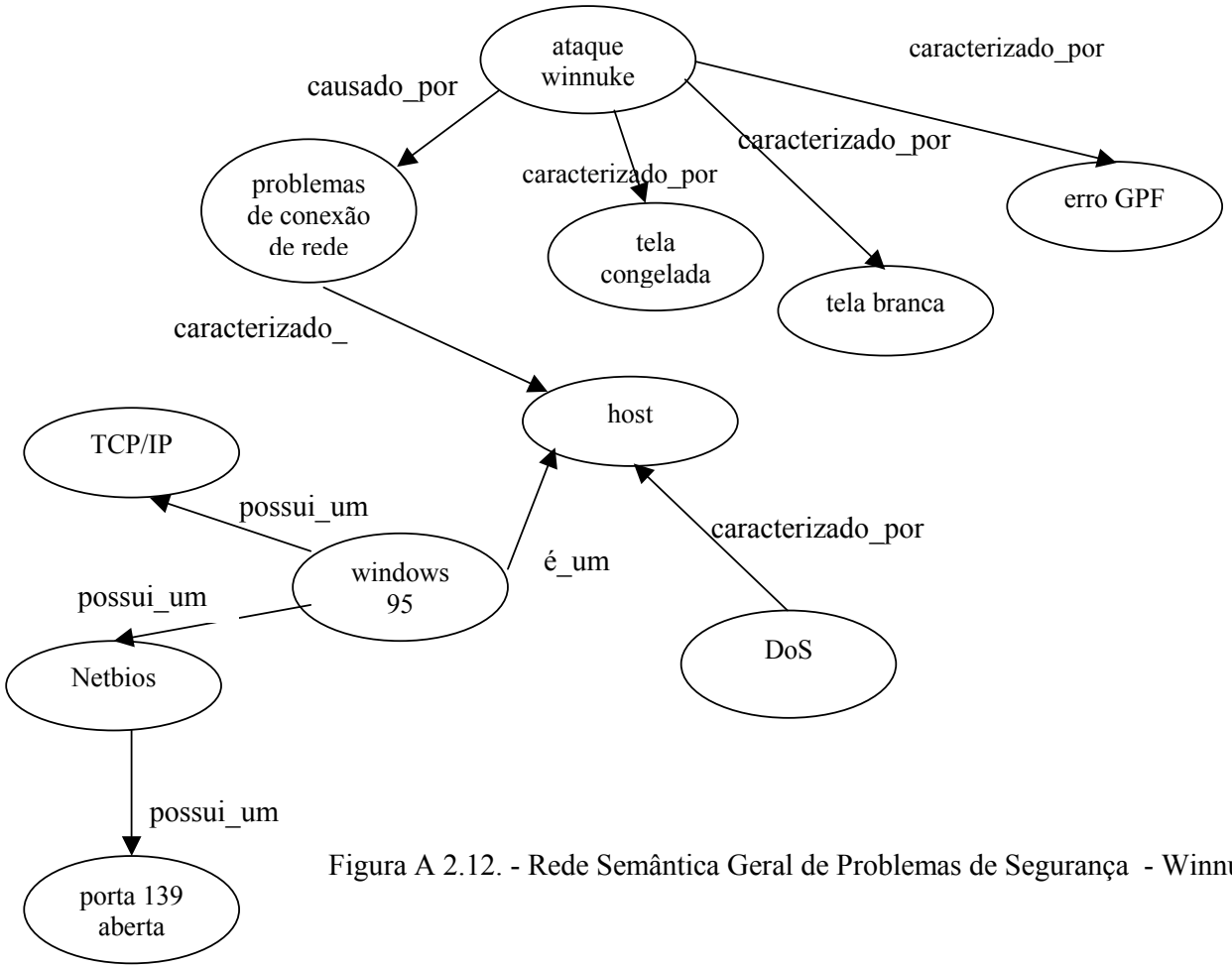


Figura A 2.12. - Rede Semântica Geral de Problemas de Segurança - Winnuke

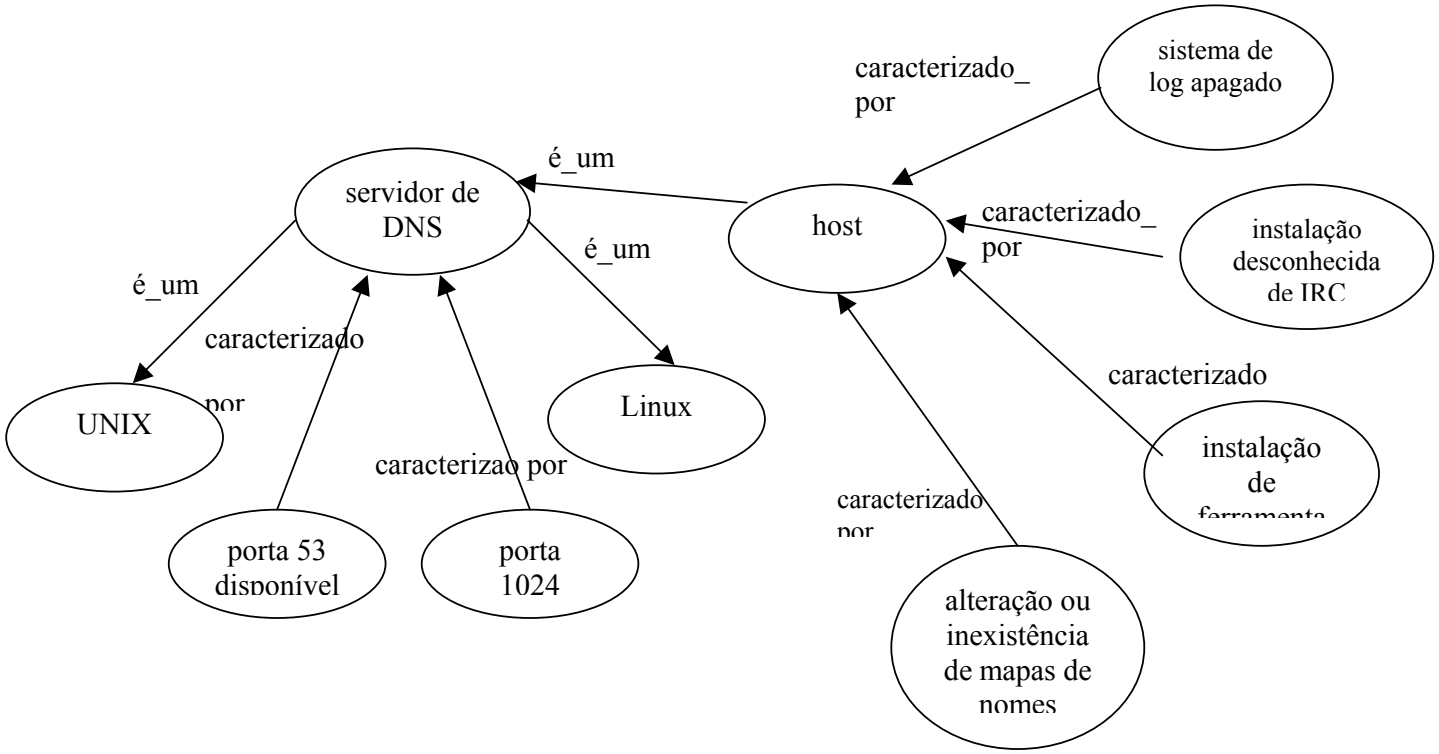


Figura A 2.13. - Rede Semântica Geral de Problemas de Segurança – Vulnerabilidade de DNS

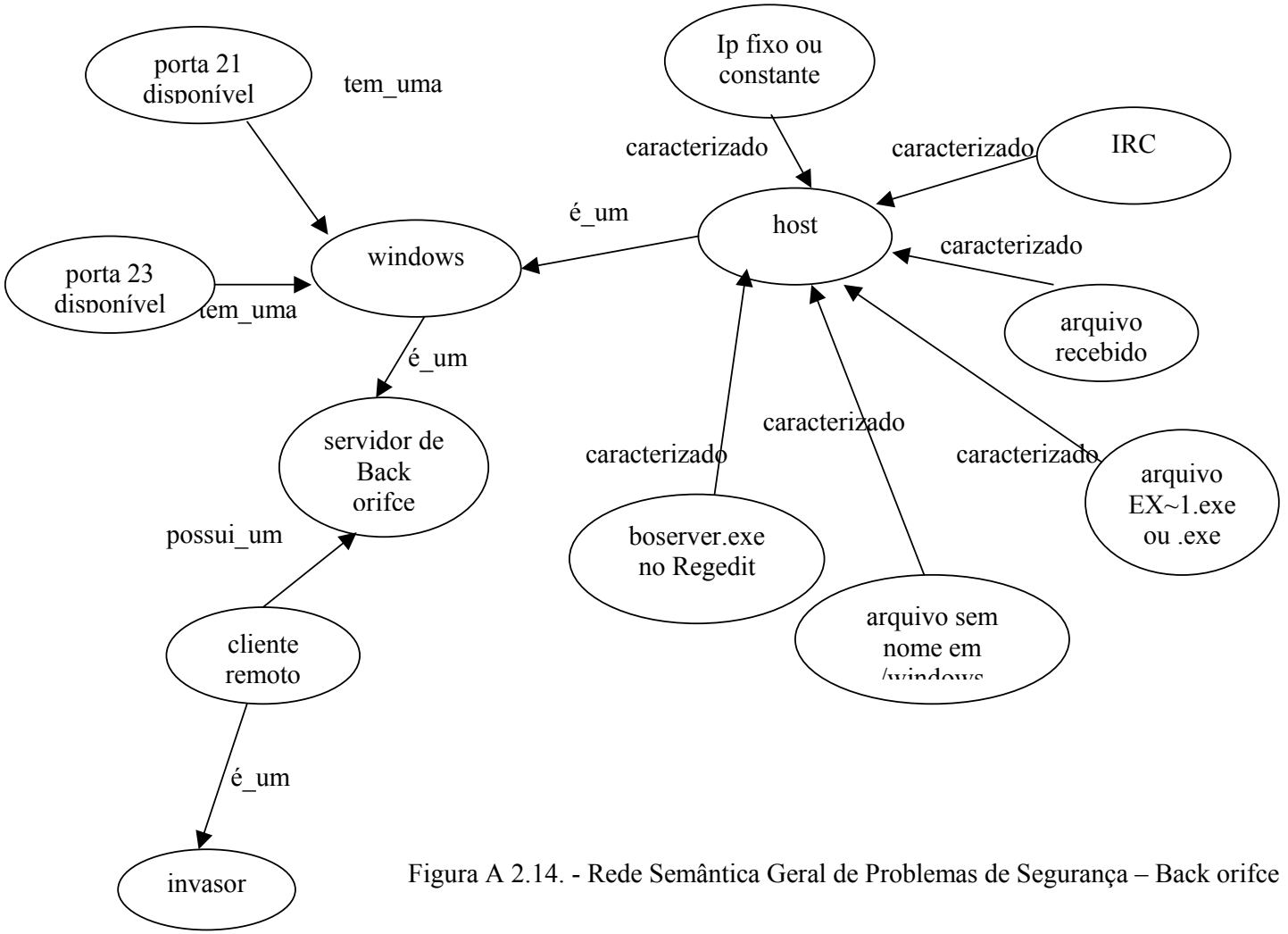


Figura A 2.14. - Rede Semântica Geral de Problemas de Segurança – Back orifce

Anexo 3 Relação de Características de Casos Cadastrados

Domínio de Segurança

1 – DoS (Ξ)

Tabela A 3.1. – Características de DoS

Atributo	Valor	Peso
pacote IP ♦ 6	falso	8
IP origem 6	= destino	8
flag SYN	sim	5
porta σ ♦ 6	aberta	9
porta 6	113	9
porta 6	139	9
host σ λ ♦ 6	sim	5
taxa de uso de CPU σ λ ♦ 6	alta	7
windows σ λ ♦ 6	sim	5
land	sim	10
servidor σ λ ♦ 6	sim	5
windows σ λ ♦ 6	nt	6
WINS σ λ ♦ 6	sim	9
espaço em disco σ λ ♦ 6	alterado	8
coke	sim	10
tela λ 6	congelada	6
tela λ 6	branca	6
erro λ 6	GPF	6
conexão de rede λ ♦ 6	ruim	7
windows σ λ 6	95	6
protocolo σ λ ♦ 6	TCP/IP	6
protocolo σ λ ♦ 6	Netbios	6
winnuke	sim	10

2 – AW (σ)

Tabela A 3.2. – Características de AW

Atributo	Valor	Peso
sub-rede	alterada	6
web server	sim	8
programas CGI default	sim	8
págia Web	alterada	9
vulnerabilidade de CGI	sim	10
arquivo	doc	9
religar	persiste	9

3 – Vírus (λ)

Tabela A 3.3. – Características de Vírus

Atributo	Valor	Peso
vírus	sim	10
vírus de macro	sim	10
backdoors	sim	9
linux 6	sim	6
e-mail 6	sim	7
programa pirata 6	instalado	8
execução de arquivo 6	recente	8
sistema operacional SO 6	trava	8
arquivo executado 6	exe	9

4 – Axfr (♦)

Tabela A 3.4. – Características de Axfr

Atributo	Valor	Peso
servidor	sim	6
ser viço de DNS 6	sim	9
UNIX 6	sim	6
porta disponível 6	53	8
porta disponível 6	1024	8
mapa de nomes	alterado	10
mapa de nomes	inexistente	10
arquivo de log 6	apagado	9
instalação de IRC	sim	9
ferramenta de Scan 6	sim	10

5 - Af – Backdoors (6)

Tabela A 3.5. – Características de Af

Atributo	Valor	Peso
Ip	fixo	7
IRC	sim	8
arquivo recebido	executado	9
boserver.exe no regedit	sim	10
arquivo sem nome	sim	10
diretório system	alterado	9
arquivo EX~1.exe	sim	10
arquivo .exe	sim	10
back orifce	sim	10
sharing	desconhecido	9
porta disponível 21	sim	9
porta disponível 23	sim	9
porta aberta	sim	9

execução de arquivos	sim	9
MIRC	sim	9
Real Player	sim	9
windows nt	sim	9
service pack	não	9
SP 1	sim	9
SP 2	sim	9
SP 3	sim	9
SP 5	não	0

Domínio de Frutas

1 – Maçã

Tabela A 3.6. – Características de Maçã

Atributo	**Valor**	**Peso**
àrvore	sim	8
arredondada	sim	7
extremo sul	sim	8
vermelha	sim	6
amarela	sim	6

2 – Uva

Tabela A 3.7. – Características de Uva

Atributo	**Valor**	**Peso**
trepadeira	sim	8
pequena	sim	7
roxa	sim	8
espinhos	não	0
cores	várias	7
vinho	sim	10

3 – Laranja

Tabela A 3.8. – Características de Laranja

Atributo	**Valor**	**Peso**
árvore	sim	8
arredondada	sim	7
norte	sim	8
laranja	sim	10
espinhos	sim	8

Anexo 4 Problemas Consultados e Casos Recuperados Domínio de Segurança

Problema Consultado nº. 12

Título: *As máquinas com windows estão travando.*

Descrição: *Os windows estão com tela azul ou erro de GPF.*

Detalhes: *O usuário está trabalhando e derrepente as máquinas estão travando, quatro máquinas da sub-rede 200.18.6 já reclamaram que não conseguem executar nenhuma conexão de rede.*

Resultado da Primeira Recuperação

Problema Detectado: *DoS - Winnuke*

Caso: *5* Similaridade: *0,17*

Solução Recomendada:

- *Religar o computador.*
- *Procurar na rede por envio de pacotes para a porta 139 das máquinas que estão emitindo problemas, se encontrado monitorar o tipo de pacote enviado para ver se há envio de string OOB (Out of Band), problema de Winnuke.*

Problema Detectado: *Vírus*

Caso: *6* Similaridade: *0,12*

Solução Recomendada:

- *Instalar um bom antivírus.*
- *Orientar o usuário a NUNCA executar arquivos enviados por e-mail por origens não conhecidas.*

Resultado para Recuperação Refinada

Problema Detectado: *DoS - Winnuke*

Caso: *5* Similaridade: *0,17* Confiança: *0,49*

Solução Recomendada:

- *Religar o computador.*
- *Procurar na rede por envio de pacotes para a porta 139 das máquinas que estão emitindo problemas, se encontrado monitorar o tipo de pacote enviado para ver se há envio de string OOB (Out of Band).*

Perguntas Apresentadas e Respostas Fornecidas:

3 – Que protocolo está ativo ?	*TCP/IP*
1 – Existe portas em modo promíscuo ?	*Sim*
7 – Que porta está sendo usada para receber pacotes ?	*139*
8 - O espaço em disco está satisfatório ?	*Sim*
11 – Após a religação da máquina, o problema persiste ?	*Não / Sim*
5 – Houve execução de arquivo antes do problema ?	*Não / Sim*
6 – Que tipo de arquivo foi executado ?	*---------- / doc*
15 – Ao executar algum anti-vírus, algum vírus foi encontrado ?	*---------*
16 – Que tipo de vírus foi encontrado ?	*---------*
17 – Houve execução de algum programa pirata ?	*----------*

Problema Consultado nº. 12 – Alteradas respostas de Perguntas

Observação: Modificando as respostas das perguntas 11-5-6, como pode ser observado acima, altera-se, significativamente, o caso recuperado após o refinamento, passando a ser melhor classificado o caso 6:

Problema Detectado: *Vírus*

Caso: *6* Similaridade: *0,12* Confiança: *0,60*

Problema Consultado nº. 13

Título: *A máquina windows extremamente lenta*.

Descrição: *Após a execução de um arquivo a máquina está super lenta.*

Detalhes: *Uma execução de arquivo na máquina 200.18.6.171 tem deixado o sistema operacional super lento, observa-se alta taxa de uso de CPU.*

Resultado da Primeira Recuperação

Problema Detectado: *Vírus*

Caso: *6* Similaridade: *0,26*

Solução Recomendada:

- *Instalar um bom antivírus.*
- *Atualizar o anti-vírus.*
- *Comunicar as pessoas que enviaram arquivos atachados sobre possibilidade de vírus.*
- *Orientar o usuário a NUNCA executar arquivos enviados por e-mail de origens não conhecidas.*

Problema Detectado: *Backdoor*

Caso: *4* Similaridade: *0,12*

Solução Recomendada:

- *Religar o computador;*
- *Atualizar as versões de programas como IRC,ICQ, MIRC, Real Player e de e-mail .*
- *Atualizar as versões de correção do sistema operacional em uso.*
- *Instalar programas de Firewall na rede em uso.*
- *Procurar por portas em modo promiscuo.*
- *Procurar por serviços e arquivos desconhecidos no host.*
- *desinstalar qualquer programa desconhecido limpando registros no sistema, como regedit para o caso de windows.*
- *Buscar informações via Internet para correçao o problema de backdoor.*

Problema Detectado: *DoS*

Caso: *5* Similaridade: *0,11*

Solução Recomendada:

- *Religar o computador;*

- *Procurar na rede por envio de pacotes para a porta 139 das máquinas que estão emitindo problemas, se encontrado monitorar o tipo de pacote enviado para ver se há envio de string OOB (Out of Band).*

Resultado para Recuperação Refinada

Problema Detectado: *Vírus*

Caso: *6* Similaridade: *0,26* Confiança: *0,79*

Solução Recomendada:

- *Instalar um bom antivírus.*
- *Atualizar o anti-vírus.*
- *Comunicar as pessoas que enviaram arquivos atachados sobre possibilidade de vírus.*
- *Orientar o usuário a NUNCA executar arquivos enviados por e-mail de origens não conhecidas.*

Problema Detectado: *Backdoor*

Caso: *4* Similaridade: *0,12* Confiança: *0,43*

Solução Recomendada:

- *Religar o computador;*
- *Atualizar as versões de programas como IRC,ICQ, MIRC, Real Player e de e-mail .*
- *Atualizar as versões de correção do sistema operacional em uso.*
- *Instalar programas de Firewall na rede em uso.*
- *Procurar por portas em modo promiscuo.*
- *Procurar por serviços e arquivos desconhecidos no host.*
- *desinstalar qualquer programa desconhecido limpando registros no sistema, como regedit para o caso de windows.*
- *Buscar informações via Internet para correçao o problema de backdoor.*

Perguntas Apresentadas e Respostas Fornecidas:

5 – Houve execução de arquivo antes do problema ? Sim

6 - Que tipo de arquivo foi executado ? exe

9 – O arquivo executado foi recebido por e-mail ? Sim

15 - Ao executar anti-vírus, algum vírus foi encontrado ? ---------

16 – Que tipo de vírus foi encontrado ? ---------

17 – Houve execução de algum programa pirata ? Não

20 – Arquivo boserver.exe no regedit ? ----------

21 – Acesso a disco constante ? Sim

22 – Que tipo de programa tem instalado na máquina ? IRC

3 - Que protocolo está ativo ? TCP/IP

1 – Existe portas em modo promíscuo ? ----------

7 - Que porta está sendo usada para receber pacotes ? ----------

8 - O espaço em disco está satisfatório ? Sim

11 – Após a religação da máquina, o problema persiste ? Sim

Domínio de Frutas

Problema Consultado nº. 1

Título: *A fruta é gostosa.*

Descrição: *Tenho uma fruta redonda.*

Detalhes: *Queria muito descobrir que fruta é esta redonda e gostosa*.

Resultado da Primeira Recuperação

Problema Detectado: *Laranja*

Caso: *3* Similaridade: *0,21*

Solução Recomendada:

- *Sua fruta é uma laranja, cresce em árvores com ou sem espinhos, é arredondada, de cor laranja, muito cultivada no Norte do Brasil, possui vários nomes.*

Problema Detectado: *Maçã*

Caso: *1* Similaridade: *0,20*

Solução Recomendada:

- *Sua fruta é uma maçã, cresce em árvores, é arredondada, de cor vermelha ou verde, muito cultivada no Extremo Sul do Brasil.*

Resultado da Recuperação Refinada

Problema Detectado: *Laranja*

Caso: *3* Similaridade: *0,75*

Solução Recomendada:

- *Sua fruta é uma laranja, cresce em árvores com ou sem espinhos, é arredondada, de cor laranja, muito cultivada no Norte do Brasil, possui vários nomes, algumas de suas árvores tem espinhos.*

Problema Detectado: *Maçã*

Caso: *1* Similaridade: *0,42*

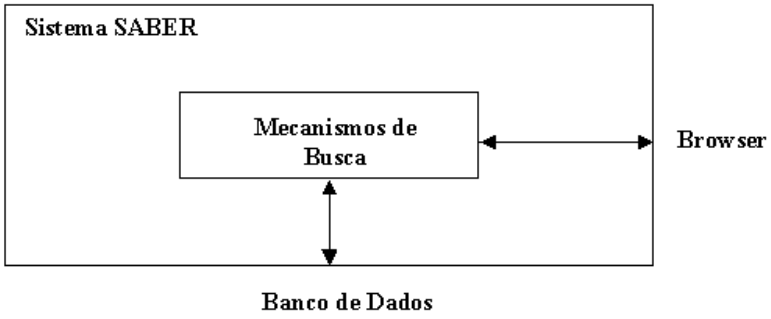
Solução Recomendada:

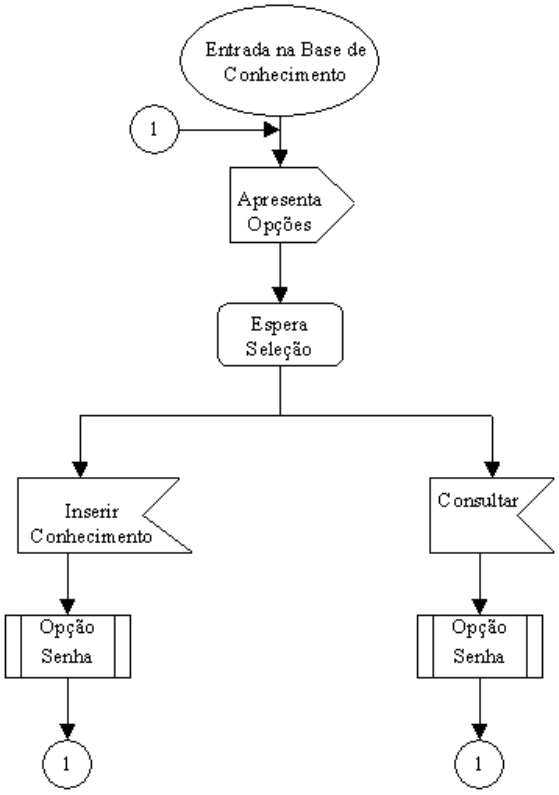
- *Sua fruta é uma maçã, cresce em árvores, é arredondada, de cor vermelha ou verde, muito cultivada no Extremo Sul do Brasil.*

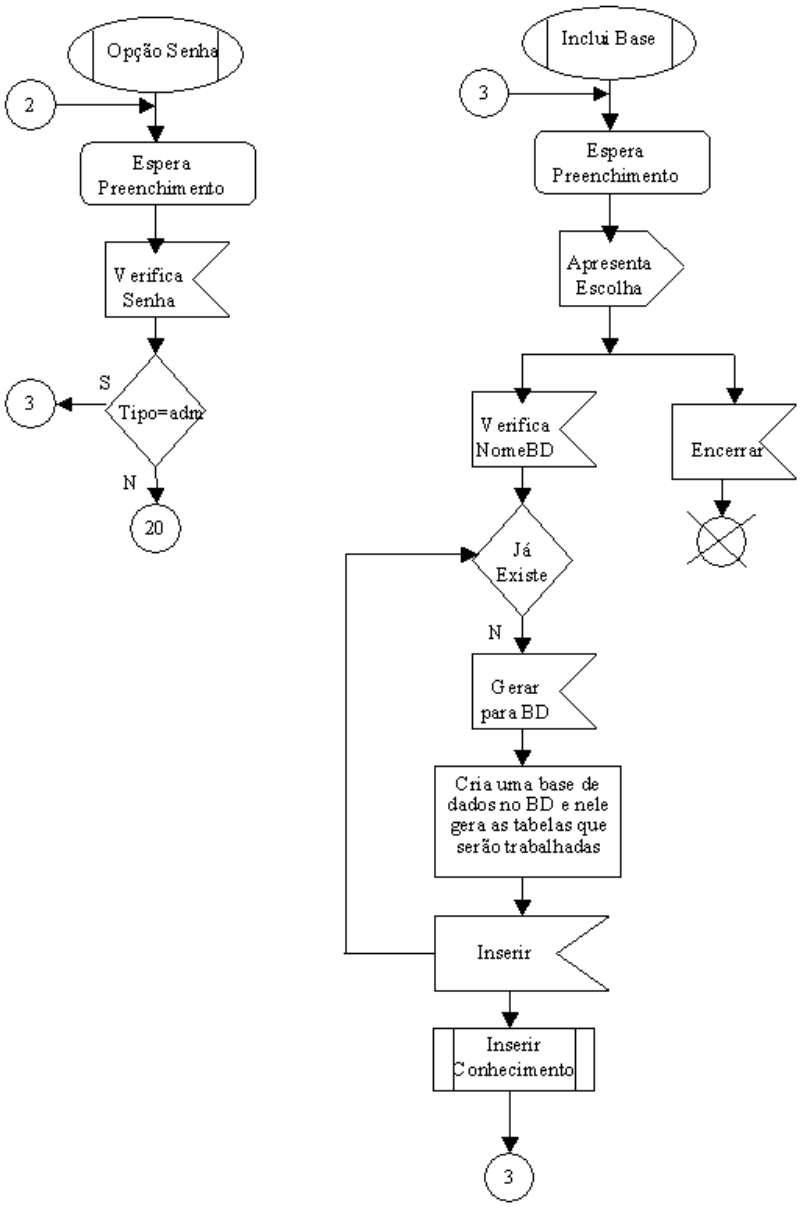
Perguntas Apresentadas e Respostas Fornecidas:

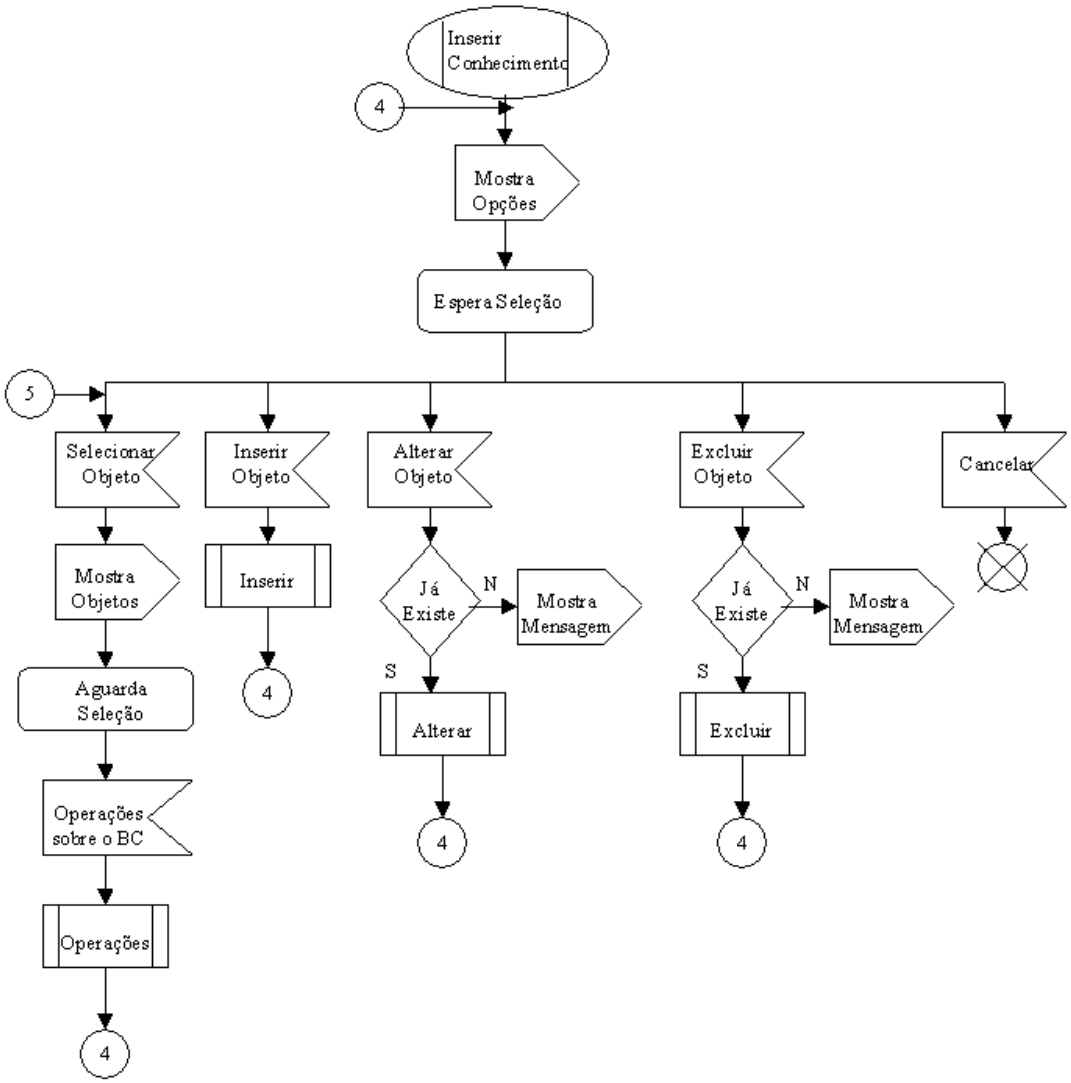
1 – Como cresce esta fruta ?	Árvore
4 - Que formato possui ?	Arredondada
3 – Qual sua cor exata ?	Laranja
5 – Qual sua região de maior cultivo ?	Norte
2 – Possui espinhos ?	-------
6 – Serve para fazer vinho ?	-------

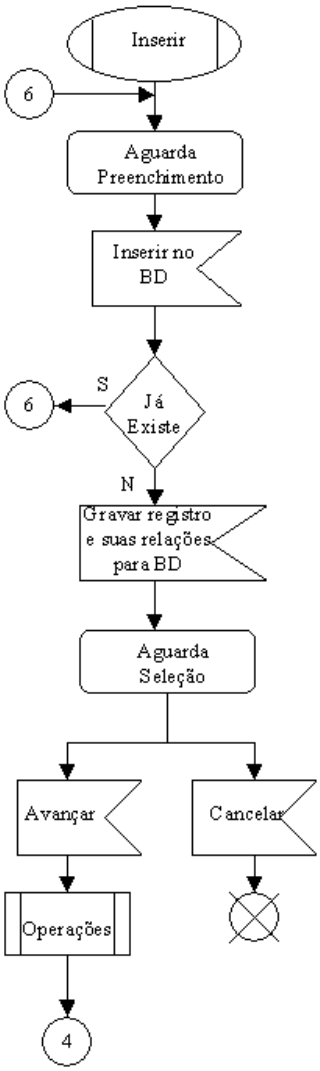
Anexo 5 Especificação do Sistema SABER em SDL

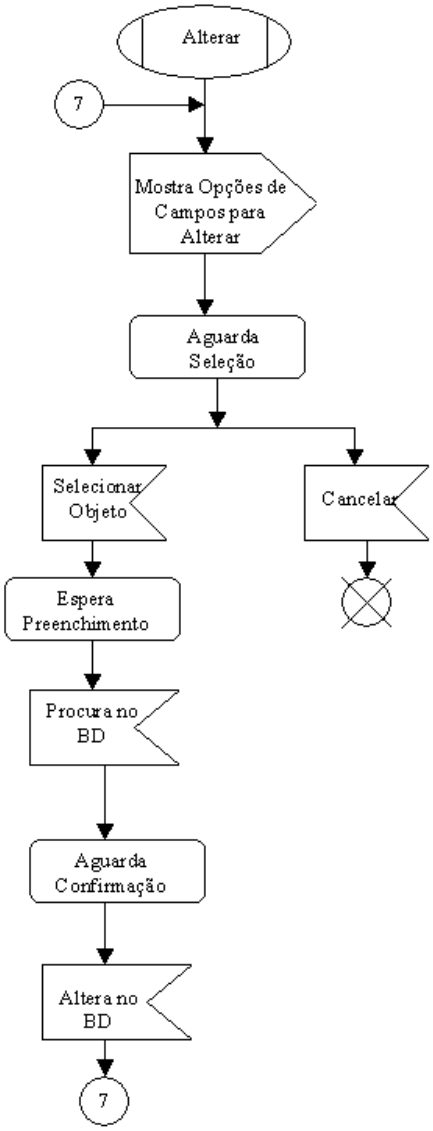


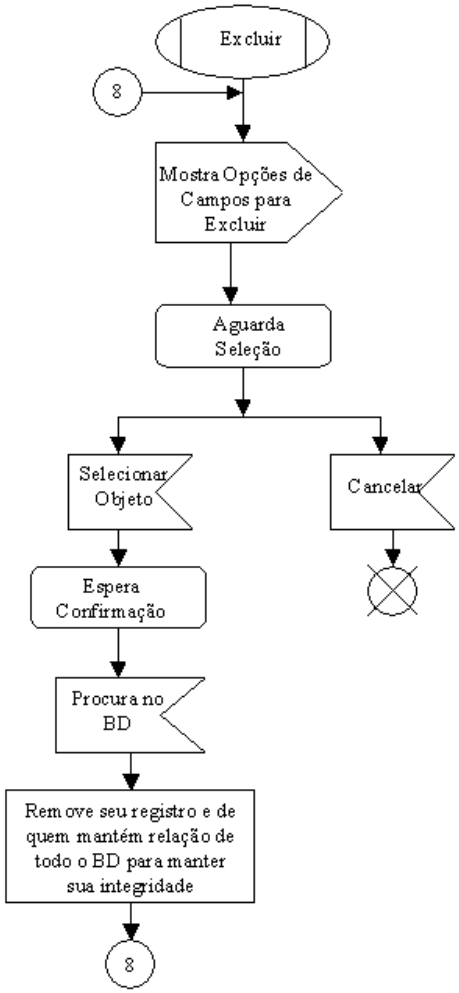


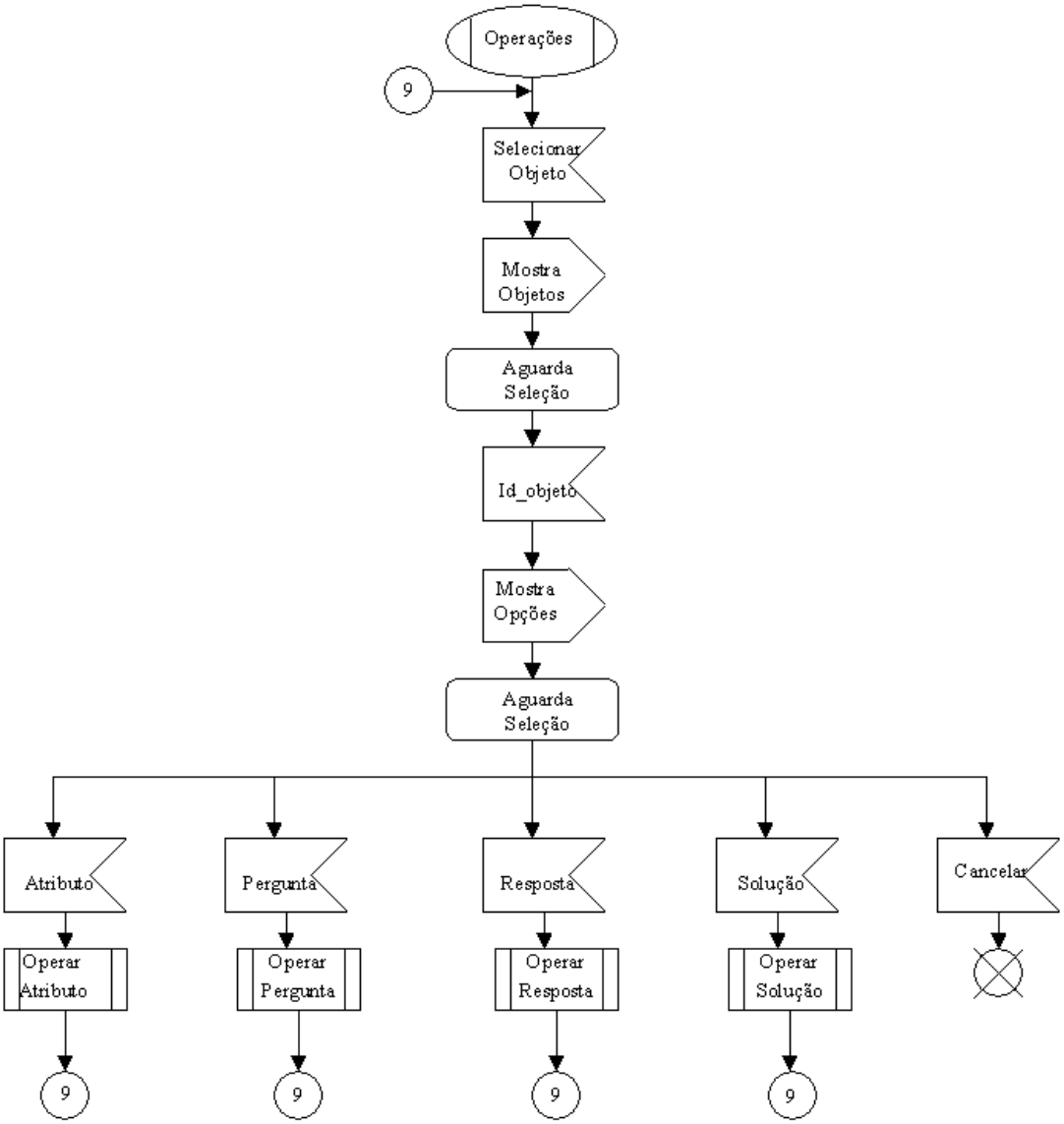

Opção Senha
2
Espera Preenchimento
Verifica Senha
S
3
Tipo=adm
N
20
Inclui Base
3
Espera Preenchimento
Apresenta Escolha
Verifica NomeBD
Encerrar
Já Existe
N
Gerar para BD
Cria uma base de dados no BD e nele gera as tabelas que serão trabalhadas
Inserir
Inserir Conhecimento
3

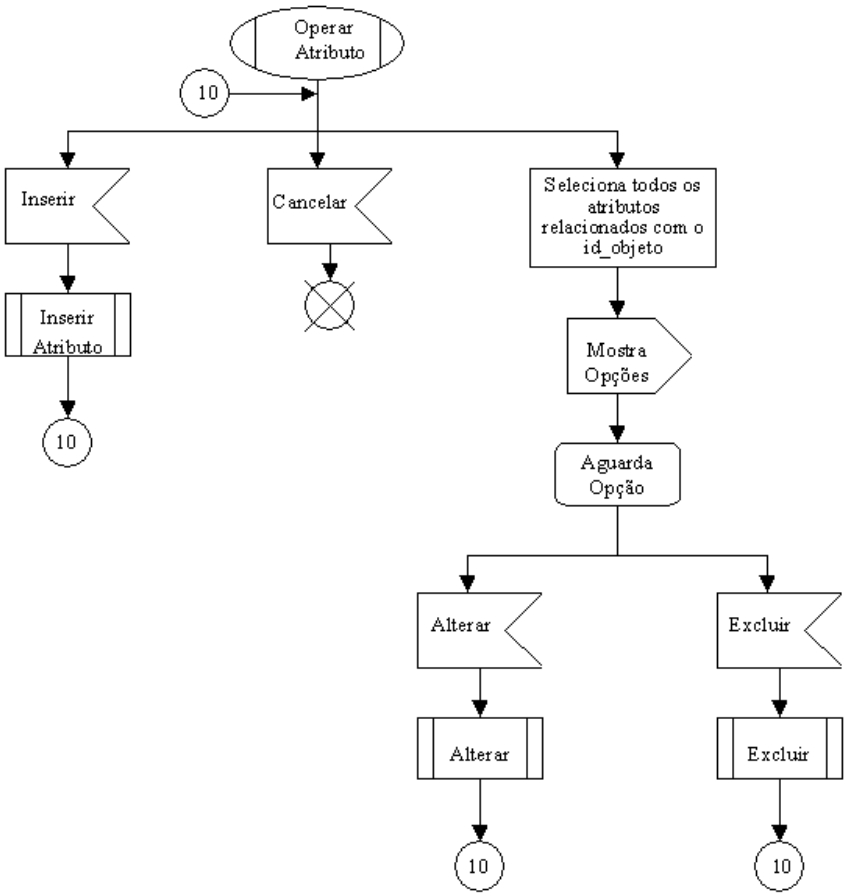

Inserir Conhecimento
4
Mostra Opções
Espera Seleção
5
Selecionar Objeto
Inserir Objeto
Alterar Objeto
Excluir Objeto
Cancelar
Mostra Objetos
Inserir
Já Existe
N
Mostra Mensagem
Já Existe
N
Mostra Mensagem
Aguarda Seleção
4
S
Alterar
S
Excluir
Operações sobre o BC
4
4
Operações
4


Inserir
6
Aguarda Preenchimento
Inserir no BD
S
6
Já Existe
N
Gravar registro e suas relações para BD
Aguarda Seleção
Avançar
Cancelar
Operações
4


Alterar
7
Mostra Opções de Campos para Alterar
Aguarda Seleção
Selecionar Objeto
Cancelar
Espera Preenchimento
Procura no BD
Aguarda Confirmação
Altera no BD
7


Excluir
8
Mostra Opções de Campos para Excluir
Aguarda Seleção
Selecionar Objeto
Cancelar
Espera Confirmação
Procura no BD
Remove seu registro e de quem mantém relação de todo o BD para manter sua integridade
8


Operações
9
Selecionar Objeto
Mostra Objetos
Aguarda Seleção
Id_objeto
Mostra Opções
Aguarda Seleção
Atributo
Pergunta
Resposta
Solução
Cancelar
Operar Atributo
Operar Pergunta
Operar Resposta
Operar Solução
9
9
9
9


Operar Atributo
10
Inserir
Cancelar
Seleciona todos os atributos relacionados com o id_objeto
Inserir Atributo
Mostra Opções
10
Aguarda Opção
Alterar
Excluir
Alterar
Excluir
10
10