

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

MICHELLE DENISE LEONHARDT

**Doroty: um Chatterbot para Treinamento
de Profissionais Atuantes no Gerenciamento
de Redes de Computadores**

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Profa. Dra. Liane Margarida Rockenbach
Tarouco
Orientador

Porto Alegre, maio de 2005

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Leonhardt, Michelle Denise

Doroty: um Chatterbot para Treinamento de Profissionais Atuantes no Gerenciamento de Redes de Computadores / Michelle Denise Leonhardt. – Porto Alegre: PPGC da UFRGS, 2005.

110 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2005. Orientador: Liane Margarida Rockenbach Tarouco.

1. Chatterbots. 2. Gerência de redes. 3. Linguagem natural. 4. SNMP. I. Tarouco, Liane Margarida Rockenbach. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof^a. Valquiria Linck Bassani

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

Coordenador do PPGC: Prof. Flávio Rech Wagner

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

*“Primeiro aprenda a ser um artesão.
Isso não impedirá você de ser um gênio.”*
— EUGENE DELACROIX

AGRADECIMENTOS

Agradecer a todos que ajudaram a construir esta dissertação não é uma tarefa fácil. A realização desse mestrado representa para mim, além de uma conquista, um sonho se realizando. E durante todo o período houve pessoas especiais que contribuíram, cada uma a sua maneira, para que esta página e tantas outras pudessem estar sendo escritas. Por essa razão, desejo expressar os meus sinceros agradecimentos:

O meu primeiro agradecimento, é, sem dúvida, dirigido à minha orientadora, Dra. Liane Tarouco. Já dos tempos de iniciação científica e durante todo o período em que me acolheu, foi, com certeza, um exemplo de dedicação, força e garra. Muitos foram os conselhos, as palavras de encorajamento e toda atenção, muitas vezes rigorosa (mas necessária), que me foi dada quando precisei. Agradeço pela oportunidade, pelo apoio e pelo respeito, além do carinho e da compreensão dispensados.

Agradeço também ao professor Dr. Lisandro Zambenedetti Granville pela paciência, pelo tempo dedicado a me ajudar. Pelo companheirismo com a galera do grupo de pesquisa, por encontrar as palavras certas quando as crises típicas de mestrado apareciam e por ter estado sempre de braços abertos para ouvir o que quer que eu ou os colegas tivéssemos a dizer.

Agradeço ao Instituto de Informática e ao Programa de Pós-Graduação em Computação pelos recursos, infra-estrutura, estímulo e acolhida, além da educação de ótima qualidade. Aos professores pelo exemplo e dedicação. Aos funcionários pelas pequenas coisas, tão importantes, que muitas vezes passam despercebidas. Também agradeço à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo suporte financeiro que me foi dado, possibilitando a realização dessa pesquisa.

Aos colegas do grupo de pesquisa e de laboratório, que estavam sempre ali, seja para dar uma força no trabalho, seja para ir lancha no bar, seja para escutar uma, duas, três vezes as apresentações prévias de semana acadêmica e artigos, contribuindo da melhor forma possível para que eu me sentisse segura e preparada. Colegas que foram se tornando amigos e que hoje, com certeza, compartilham a minha alegria: Ricardo Neisse, Diego Contessa, Rafael Huff, Rodrigo Sanger, Evandro Pereira, Clarissa Marquezan, Ricardo Vianna, Cleber Ortiz, Tiago Fioreze e todo o pessoal do LABCOM.

Aos colegas da pós, que estavam enfrentando as mesmas batalhas e compartilhando as mesmas conquistas. Destes colegas, alguns foram formando uma turma especial cujos momentos passados vão além da convivência acadêmica e serão lembrados em muitas ocasiões: os aniversários com direito a bolo, os festivais de bergamota, os eventos extra-expediente que permitiram a descontração necessária. À Gabriela Jacques da Silva, Roberto Drebes e Laura Silveira Mastella, pessoas essenciais durante todo o processo e que hoje tenho a alegria de ter como amigos de fé.

Aos amigos que, apesar da distância, estiveram sempre presentes mostrando que a

amizade espera e supera qualquer contratempo. Este espaço é pequeno para enumerar todos eles, mas alguns devem ser citados: Catarina Lenskij, Thaís Webber e Cesar Henrique Costa. À Daiane Castro e Roseclea Medina que contribuíram muito com sua alegria e conhecimento e a todos os amigos que, de uma forma ou de outra, contribuíram com sua amizade e com sugestões efetivas para a realização deste trabalho.

Ao Tiago e sua família pelo carinho, acolhida, paciência e compreensão reveladas não só durante esse período, mas durante todo o tempo em que se fazem presentes. Aos meus familiares Marcos e Marcia pelo estímulo e apoio incondicional em todos os momentos, pela paciência e grande amizade com que sempre me ouviram, e sensatez com que sempre me ajudaram. Pela compreensão nas horas em que não estive presente mesmo estando presente fisicamente. Um agradecimento especial aos meus pais, Denise e Bruno, pelo exemplo de vida, por não medir esforços para poder me proporcionar as melhores oportunidades, pelo abraço e pelo ombro na hora do medo e da ansiedade, pela paciência nas horas de sufoco e por acreditarem em mim e no meu potencial, me apoiando sempre e respeitando minhas escolhas.

Encerro dizendo que não é simplesmente agradecer que quero. Quero que se faça presente, neste documento, o nome daqueles que já o percorrem nas entrelinhas.

Muito Obrigada.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	8
LISTA DE FIGURAS	10
LISTA DE TABELAS	11
RESUMO	12
ABSTRACT	13
1 INTRODUÇÃO	14
1.1 Objetivos	15
1.2 Organização da Dissertação	15
2 CHATTERBOTS: TECNOLOGIA E USOS	17
2.1 Conceituação e Classificação	17
2.2 Origens	19
2.2.1 O Teste de Turing	21
2.3 Estratégias de Conversação	22
2.4 Evolução Histórica dos Chatterbots: Alguns Exemplos	23
2.4.1 Eliza	23
2.4.2 Parry	26
2.4.3 Julia	26
2.4.4 Cybelle	27
2.4.5 A.L.I.C.E	27
2.5 Um Experimento de Conversação entre Chatterbots	30
2.6 Considerações Finais Sobre o Capítulo	31
3 GERÊNCIA DE REDES	33
3.1 Conceituação e Contextualização	33
3.2 SNMP	35
3.3 Inteligência Artificial e o Gerenciamento de Redes	36
3.3.1 Agentes	37
3.3.2 Sistemas Especialistas	40
3.4 Considerações Finais Sobre o Capítulo	43

4	DOROTY: CARACTERÍSTICAS E IMPLEMENTAÇÃO	44
4.1	Motivação	44
4.1.1	Problemas Encontrados no Gerenciamento de Redes	45
4.1.2	Linguagem Natural, Interfaces e Chatterbots	45
4.2	Visão Geral do Sistema: Conhecimento	49
4.2.1	Aquisição do Conhecimento	49
4.2.2	Conhecimento Prático	50
4.3	Arquitetura do Protótipo	53
4.4	Implementação e Funcionamento	54
4.4.1	Tecnologias Utilizadas na Implementação	55
4.4.2	Interface e Interação com o chatterbot	55
4.4.3	Base de Conhecimento e Máquina de Inferência	56
4.4.4	Banco de Informações e Histórico de Dados	63
4.4.5	Módulo Coletor e Módulo de Atualização de Histórico	64
4.4.6	Módulo Central e Regras de Funcionamento	67
4.4.7	Módulo Auxiliar de Diagnóstico de Problemas	69
4.5	Considerações Finais Sobre o Capítulo	69
5	AVALIAÇÃO E ANÁLISE DE RESULTADOS	70
5.1	Testes Realizados e Etapas de Avaliação	70
5.1.1	Teste de bot Estático	70
5.1.2	Teste de bot Dinâmico	71
5.2	Considerações Finais sobre os Resultados Alcançados	83
6	CONSIDERAÇÕES FINAIS	84
6.1	Trabalhos Futuros e Possíveis Expansões	85
	REFERÊNCIAS	87
	APÊNDICE A QUESTIONÁRIO DE AVALIAÇÃO	93
	APÊNDICE B CONHECIMENTO - EXEMPLO DE CATEGORIAS	104

LISTA DE ABREVIATURAS E SIGLAS

AIML	Artificial Intelligence Markup Language
ALICE	Artificial Linguistic Internet Computer Entity
API	Application Programming Interface
CBR	Case-Based Reasoning
EAD	Ensino a Distância
ER	Entidade Relacionamento
FAQs	Frequently Asked Questions
HTML	HyperText Markup Language
IA	Inteligência Artificial
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IHC	Interação Humano-Computador
IP	Internet Protocol
ISO	International Organization for Standardization
JDBC	Java Database Connectivity
MANIA	Managing Awareness in Networks with Intelligent Agents
MIB	Management Information Database
MUD	Multi-User Dungeons
PAGE	Perceptions, Actions, Goals, Environment
PBL	Problem-based Learning
PDU	Protocol Data Unit
PLN	Processamento de Linguagem Natural
QoS	Quality of Service
RFC	Request For Comments
SMA	Sistemas MultiAgentes
SNMP	Simple Network Management Protocol

SQL	Structured Query Language
TCP	Transmission Control Protocol
TT	Teste de Turing
UDP	User Datagram Protocol
XML	Extendable Markup Language

LISTA DE FIGURAS

Figura 2.1:	O jogo da imitação original	21
Figura 2.2:	O jogo da imitação em sua nova sistemática: o Teste de Turing	22
Figura 2.3:	Exemplo simplificado de uma regra utilizada por ELIZA	25
Figura 2.4:	Cybelles mostrando informações sobre ALICE durante o diálogo (em francês).	28
Figura 3.1:	Funcionamento do SNMP (STALLINGS, 1996)	37
Figura 3.2:	Um exemplo de funcionamento do protocolo SNMP	38
Figura 3.3:	Arquitetura do agente proposto por (ESFANDIARI; DEFLANDRE; QUINQUETON, 1996)	40
Figura 3.4:	Ciclo de processos de um sistema que utiliza o paradigma CBR	42
Figura 4.1:	Diferenciação da parte estática e da parte dinâmica em uma resposta	49
Figura 4.2:	Organização da MIB II	51
Figura 4.3:	Arquitetura do chatterbot Doroty	53
Figura 4.4:	Diagrama detalhado da arquitetura de Doroty (Interação)	54
Figura 4.5:	Tela inicial de Doroty: componente de diálogo	56
Figura 4.6:	Tela de apresentação de informações gerais sobre Doroty	57
Figura 4.7:	Tela simples de interação com Doroty	58
Figura 4.8:	Exemplo de execução do Graphmaster	62
Figura 4.9:	Trecho AIML com passagem de parâmetros em Doroty	63
Figura 4.10:	Tabelas do Banco de Dados de Doroty	65
Figura 4.11:	Funcionamento do módulo de atualização de histórico	66
Figura 4.12:	Funcionamento do módulo central para os casos citados na tabela 4.4	68
Figura 5.1:	Tela que apresenta informações sobre a tabela de roteamento	74
Figura 5.2:	Exemplo da interface simples de Doroty apresentando link direto para uma página com informações sobre o conteúdo	75
Figura 5.3:	Experiência dos avaliadores	76
Figura 5.4:	Um exemplo de execução do comando ping através de Doroty	79
Figura 5.5:	Um exemplo de resposta dada por Doroty quando uma pergunta incompleta sobre a rede é feita.	80
Figura 5.6:	Resposta para uma pergunta sobre o objeto sysDescr	81

LISTA DE TABELAS

Tabela 2.1:	Classificação dos chatterbots segundo conjunto PAGE	19
Tabela 2.2:	Categoria de bots	20
Tabela 4.1:	Exemplos de objetos da MIB2 que podem ser consultados para verificação de problemas de segurança em uma rede	53
Tabela 4.2:	Características de um problema de interface de rede	53
Tabela 4.3:	Tabelas pertencentes ao banco de informações e histórico de Doroty .	64
Tabela 4.4:	Regras de tratamento - módulo central	67
Tabela 5.1:	Nível de instrução dos avaliadores	76
Tabela 5.2:	Área de atuação dos avaliadores	77
Tabela 5.3:	Notas atribuídas ao protótipo	82

RESUMO

O gerenciamento de redes exige dos administradores a disponibilidade de uma grande quantidade de informações sobre os seus equipamentos, as tecnologias envolvidas e os problemas associados a elas. Nesse cenário, administradores de redes devem, cada vez mais, aprofundar o seu conhecimento através de constante treinamento, até que estejam aptos a administrar uma rede de maneira mais eficiente e confiável.

Alguns estudos têm sido feitos buscando integrar tecnologias de Inteligência Artificial na área de gerenciamento de redes. Abordagens utilizando sistemas multiagentes, agentes de interface e sistemas especialistas já foram utilizadas com o objetivo de facilitar a tarefa de gerenciamento de rede aos olhos do usuário. Os chatterbots representam um grande potencial para a tarefa de treinamento e gerenciamento de redes já que utilizam linguagem natural e são capazes de ser facilmente integrados em ambientes mais complexos.

O principal objetivo deste trabalho é investigar o uso de chatterbots como uma ferramenta de gerenciamento utilizada por administradores menos treinados. O trabalho envolveu a adaptação do chatterbot ALICE para permitir o treinamento e a gerência de redes através da inclusão de módulos que permitem a monitoração de equipamentos de uma rede (através do protocolo SNMP) e módulos que permitam consultar e armazenar histórico de informações da mesma. Desta forma, a grande contribuição da arquitetura proposta é a de prover uma comunicação mais efetiva entre o administrador menos experiente e a rede, através do chatterbot assistente, que recebe consultas em linguagem natural, interpreta os dados coletados e expõe os conceitos envolvidos no processo de gerenciamento.

Palavras-chave: Chatterbots, gerência de redes, linguagem natural, SNMP.

A Chatterbot for Training Professionals in the Network Management Field

ABSTRACT

Network management demands from the administrators the availability of a great amount of information on the network equipments, the involved technologies and the problems associates to the networks. In this scenario, network administrators must deepen their knowledge through constant training, until they are able to manage a network in a more efficient and trustworthy way.

Several studies have been carried out in order to integrate technologies of Artificial Intelligence in the area of network management. Approaches using multiagent systems, interface agents, and expert systems have already been used, since this facilitates network management. Chatterbots represent a great potential for the training and management purposes since they use natural language and are also capable of being easily integrated in more complex environments.

The main goal of this work is to investigate the use of chatterbots as a management tool used by less experienced network administrators. The work involved the adaptation of the ALICE chatterbot to help in training and managing networks through the inclusion of modules that enable ALICE to monitor (through SNMP protocol), query and store information about network devices. The main contribution of the proposed architecture comes from the fact that it provides a more feasible and effective communication tool between the less experienced administrator and the managed network, through the role of the assistant chatterbot, acting as a data interpreter as well as a source of the conceptual knowledge involved in the management process.

Keywords: chatterbots, network management, natural language, SNMP.

1 INTRODUÇÃO

As redes de computadores nasceram e evoluíram a fim de facilitar a troca de dados, informações e serviços entre usuários e entidades separadas. Entretanto, a acelerada evolução da tecnologia da informação tem trazido profundos impactos em diversas frentes envolvidas em sua aplicação. Dessa forma, ao mesmo tempo em que as redes de computadores criaram inúmeras oportunidades para usuários e empresas se comunicarem e compartilharem informações, essa tecnologia digital criou um grande desafio no que diz respeito a sua administração e gerenciamento. A tarefa de gerenciar diferentes variedades de software e hardware está se tornando cada vez mais difícil devido ao rápido progresso das tecnologias envolvidas nesse processo (TSAI; CHANG, 1998).

Há uma tendência a aumentar o número de redes heterogêneas na comunicação atual. Tamanha diversidade faz com que operadores e gerentes de redes necessitem mais conhecimento e treinamento. Um dos principais pontos da atividade de gerenciamento de redes é que possíveis falhas ou comportamentos incoerentes devem ser diagnosticados e solucionados automaticamente ou através de um gerente preparado para tomar as providências necessárias (BIESZCAD; PAGUREK; WHITE, 1998). Essa atividade exige dos gerentes de redes a disponibilidade de uma grande quantidade de informações sobre os seus equipamentos, as tecnologias envolvidas e os problemas associados a elas. Nesse cenário, administradores de redes devem, cada vez mais, aprofundar o seu conhecimento através de constante treinamento, até que estejam aptos a administrar uma rede de maneira mais eficiente e confiável.

Observa-se, então, a necessidade de uma alternativa inovadora que seja capaz de suprir as necessidades de treinamento e de redução do tempo de aprendizagem de um profissional menos capacitado que atue na área. Tais profissionais necessitam obter informações de forma simples, até que possam se tornar suficientemente seguros de si e capacitados para tomarem suas próprias decisões.

Nesse contexto, os chatbots representam um grande potencial para a tarefa de treinamento e gerenciamento de redes já que incorporam novos conceitos e introduzem técnicas oriundas de outras áreas da Ciência da Computação, como a utilização da linguagem natural, campo de estudo fortemente explorado pela Inteligência Artificial (IA). Operadores de rede com pouca experiência podem não utilizar adequadamente os dados coletados de uma rede. Da mesma forma, eles nem sempre sabem como coletar os dados mais adequados que possam levar à solução de algum problema. Utilizando linguagem natural, um chatbot pode servir como um interpretador de dados, além de uma fonte de consulta para solução de problemas em uma rede, tentando replicar o papel de um administrador de rede mais treinado e mais capacitado.

Os chatbots são capazes de ser facilmente integrados em ambientes mais complexos, além de poderem estar disponíveis para utilização durante 24 horas por dia. O ob-

jetivo principal de tais sistemas é o de simular uma conversação, em linguagem natural, com o objetivo tornar esta comunicação o mais próxima possível da conversação de um ser humano.

A vantagem da utilização de chatterbots para este fim pode ser também justificada na pesquisa de Jenkins (1999) quando discute a contribuição de se construírem interfaces de gerenciamento que podem estar disponíveis via Internet, possibilitando que, num futuro não muito distante, profissionais possam se deslocar e, ao mesmo tempo, obter uma possibilidade de satisfazer dúvidas remotamente.

O presente trabalho apresenta Doroty, um chatterbot que é capaz de interagir não somente com o administrador de uma rede, mas também com a própria rede gerenciada através da utilização do protocolo SNMP. A contribuição da arquitetura proposta é a de prover uma comunicação mais efetiva entre o administrador menos experiente e a rede, através do papel do chatterbot assistente, que reconhece consultas apresentadas em linguagem natural, interpreta os dados coletados e expõe os conceitos envolvidos no processo de gerenciamento. Esta proposta permite uma abordagem de treinamento contextualizado onde é possível formular consultas que serão respondidas mediante inspeção de equipamentos reais em uma rede real, possibilitando a aplicação dos conhecimentos adquiridos em problemas reais.

1.1 Objetivos

Este trabalho tem como objetivo propor uma solução para o problema da falta de capacitação e treinamento de alguns profissionais que atuam no gerenciamento de redes de grandes corporações. Tais profissionais costumam entrar no mercado sem uma vivência maior na área e, muitas vezes, sentem-se inseguros em seus diagnósticos, perdendo muito tempo com pesquisas muitas vezes desnecessárias.

Dentro desta visão, apresenta como objetivos secundários:

- Verificar o impacto da utilização de linguagem natural como forma de capacitação de profissionais em áreas que exigem conhecimento técnico profundo, como no caso do gerenciamento de redes de computadores;
- Demonstrar como as técnicas de IA podem ser promissoras quando utilizadas na área do gerenciamento de redes e na área de redes de computadores, tanto para ferramentas de gerenciamento propriamente dito, como para ferramentas cujo objetivo está mais focado na área de educação;
- Abordar um modelo de treinamento mais centrado no usuário, fazendo com que o mesmo vivencie os conceitos na prática, preparando-o para enfrentar os problemas surgidos no mundo real.

1.2 Organização da Dissertação

O presente documento apresenta-se organizado na forma de cinco capítulos, além desta introdução:

O capítulo 2 apresenta uma revisão bibliográfica sobre chatterbots, observando suas características, tecnologia e usos. Inicialmente, é apresentada uma breve discussão sobre a classificação de tais sistemas, uma vez que, na literatura, há uma certa confusão em se tratando de conceituar a relação entre chatterbots e agentes. Em seguida, apresenta as

origens dos robôs de conversação, destacando o Teste de Turing que serviu como ponto de partida para o desenvolvimento desses sistemas e é usado, atualmente, como forma de medir o nível de inteligência e desenvoltura alcançado pelos mesmos. Por fim, apresenta as estratégias de conversação de tais sistemas, bem como sua evolução histórica através de exemplos que ressaltam as características de implementação. Além disso, são apresentados diálogos que ilustram o grande potencial dos mesmos.

Continuando a revisão bibliográfica, o capítulo 3 trata sobre a gerência de redes. Em um primeiro momento, uma conceituação e contextualização da área é apresentada, destacando a importância do gerenciamento de redes para seus usuários. Logo após, é apresentado brevemente o protocolo SNMP, utilizado por Doroty para se comunicar com a rede gerenciada. Um estudo sobre a integração de tecnologias de IA na área de gerenciamento de redes é apresentada a fim de demonstrar como a utilização desta tecnologia pode contribuir na área, tanto no gerenciamento propriamente dito como na capacitação e treinamento de profissionais que atuam nela.

Em seguida, o capítulo 4 apresenta Doroty, mostrando principalmente as características do sistema, englobando um estudo mais detalhado sobre o porque da utilização de linguagem natural em interfaces. Também apresenta aspectos da implementação do sistema, bem como a forma com que cada um dos componentes da arquitetura interagem.

A avaliação do sistema desenvolvido é o tópico do capítulo 5. Nele, é descrito o processo de avaliação, suas etapas e experimentações. Além disso, é feito um levantamento dos aspectos positivos e negativos do sistema. Cada um dos processos e características avaliadas é abordado separadamente, de forma a propiciar uma melhor visualização da contribuição de cada aspecto do sistema em especial.

Finalmente, uma análise sobre o trabalho é apresentada no capítulo final, destacando as conclusões e apresentando as possibilidades de trabalhos futuros.

2 CHATTERBOTS: TECNOLOGIA E USOS

2.1 Conceituação e Classificação

Desde o início dos tempos, o ser humano vem procurando desenvolver a comunicação natural entre interlocutores. Com o avanço da tecnologia e, conseqüentemente, a popularização de acesso aos computadores pessoais e à Internet, essa preocupação também foi sendo transferida para as máquinas, acarretando no surgimento dos chamados robôs de conversação e bate papo: os Chatterbots.

Um chatterbot é um programa de computador que tenta simular um ser humano na conversação com as pessoas. O objetivo é responder às perguntas de tal forma que as pessoas tenham a impressão de estar conversando com outra pessoa e não com um programa de computador. (TEIXEIRA; MENEZES, 2003)

Há uma certa confusão em se tratando de conceituar a relação entre chatterbots e agentes. Chatterbots são, muitas vezes, classificados como agentes conversacionais já que apresentam a possibilidade de estender o escopo de sistemas de conversação e exibir comportamento não verbal apropriado, simulando todo o poder de comunicação de um humano, como, por exemplo, expressões faciais, gestos e processamento de linguagem natural em todos seus níveis, contribuindo, desta forma, para um melhor entendimento de uma mensagem. Bickmore (1999), utiliza o termo agentes conversacionais incorporados (*Embodied Conversational Agents*) e o define como entidades que estendem os chatterbots conversacionais. Eles utilizam, em tempo-real, fala, gestos, observação (entre outros canais verbais e não verbais), para simular a interação humana face-a-face.

Porém, na IA, o termo agente não apresenta uma definição consensual, pois além de envolver diversas áreas de investigação, cada autor define o termo de modo que melhor se relacione com o seu trabalho. Uma definição bastante conceituada para o termo é a de Wooldridge (1995) quando diz que um agente é um sistema de computador baseado em hardware ou software que desfruta as propriedades de autonomia, capacidade social, reatividade e pró-atividade. O mesmo autor, define em (WOOLDRIDGE, 1997) que agente é um sistema computacional de hardware ou software que se encontra situado em um ambiente e apresenta flexibilidade e autonomia dentro do ambiente a fim de atingir seus objetivos.

De acordo com Franklin e Graesser (1996), um agente autônomo é um sistema situado em um ambiente ou parte dele, capaz de sentir o ambiente e agir nele, afetando suas percepções no futuro.

De modo geral, a maioria dos autores enumera características que devem estar presentes nos agentes, ou seja, apresentam uma lista de atributos para os mesmos. Ainda para Franklin, as características que devem estar presentes na maioria dos agentes são:

- **Autonomia:** um agente autônomo deve ter controle sobre suas ações. Um agente pode ser autônomo em relação a outros agentes ou a um ambiente;
- **Pró-atividade:** capacidade de tomar a iniciativa para atingir seus objetivos, não se limitando apenas a estímulos do ambiente;
- **Reatividade:** capacidade de reação a estímulos e mudanças dentro do ambiente no qual encontra-se inserido;
- **Continuidade Temporal:** possibilidade de permanecer continuamente ativo;
- **Capacidade Social:** a sociabilidade implica na comunicação de um agente com outros agentes ou com humanos. A capacidade de comunicação pode levar a uma necessidade de cooperação e negociação entre agentes, que, por sua vez, são características que devem estar presentes em agentes quando necessário;
- **Capacidade de Adaptação:** possibilidade de alterar o comportamento baseado na sua experiência. Este processo também é conhecido como aprendizagem;
- **Mobilidade:** capacidade do agente de se mover dentro de um ambiente;
- **Flexibilidade:** habilidade de escolher dinamicamente uma ação ou seqüência de ações em resposta a um estado do ambiente no qual se encontra.

A definição de tais características não implica que um agente deva apresentar todos os atributos, já que estes dependem do tipo de aplicação para o qual o agente é desenvolvido. Muitos autores acreditam que a presença de uma ou mais características descritas acima são suficientes para caracterizar um agente.

Embora não haja uma definição mundialmente aceita, as idéias apresentadas pelos autores citados denotam um agente como uma entidade real ou virtual, de hardware ou software que apresenta características que se diferenciam das demais aplicações. Um agente autônomo, outra nomenclatura criada para designar um campo de pesquisa dentro da IA, é uma mesma entidade real ou virtual, inserida em um ambiente, capaz de perceber e agir dentro de tal ambiente, de forma autônoma, comunicando-se ou não com outros agentes.

Seja qual for a definição, Russel e Norvig (1995) atentam para o fato de que antes de iniciar o design de agentes, é preciso ter uma idéia bem definida sobre o ambiente no qual o agente atuará, englobando o conjunto de percepções, ações e objetivos do agente. Este conjunto é conhecido como PAGE (*Perceptions, Actions, Goals, Environment*).

Galvão (2003), em sua pesquisa, traça um paralelo entre o conjunto PAGE e os chatbots, quando afirma que os chatbots também podem ser caracterizados segundo esse conjunto, apesar de essa não ser a prática comum de quem trabalha na área, uma vez que as percepções e ações desse tipo de agente podem sempre ser reduzidas à interpretação e à produção de texto, respectivamente.

Ele ainda considera a caracterização dos chatbots, segundo esse modelo, vantajosa em muitos aspectos e caracteriza algumas aplicações de chatbots conforme esse modelo (Tabela 2.1, retirada de (GALVÃO, 2003)).

Independente de classificação, a importância dos chatbots é bastante considerável. Eles apresentam grande potencial para atuar no campo pedagógico, comercial e social. Atualmente, muitas aplicações estão sendo desenvolvidas, em diversas áreas do conhecimento, utilizando como base a tecnologia e o conceito de chatbots. Alguns

Tabela 2.1: Classificação dos chatterbots segundo conjunto PAGE

Aplicação	Percepções	Ações	Objetivos	Ambiente
Entretenimento	Texto Escrito	Conversar sobre um tema qualquer ou guiar um usuário em um jogo, por exemplo	Divertir o usuário, geralmente simulando vida artificial	usuários em geral
Ensino a Distância (EAD)	Texto Escrito	Sugerir assuntos de discussão, tirar dúvidas ou efetuar demonstrações, por exemplo	Auxiliar na aprendizagem de um estudante	estudantes
Atendimento ao Consumidor	Texto Escrito	Responder dúvidas ou anotar reclamações, por exemplo	Minimizar custos de atendimento	consumidores
Comércio Eletrônico	Texto Escrito	Sugerir compra de produtos ou guiar o comprador pela loja, por exemplo	Maximizar vendas e satisfazer o cliente	compradores

exemplos de novas aplicações podem ser obtidos em (L'ABBATE; THIEL, 2002; GALVÃO et al., 2004; ANGELI; LYNCH; JOHNSON, 2001; LEONHARDT; NEISSE; TA-ROUCO, 2003).

Apresentaremos, neste capítulo, uma breve revisão sobre os chatterbots, demonstrando sua evolução histórica e suas características principais.

2.2 Origens

A palavra robô teve origem na peça "R.U.R." de Karel Capek, escrita em 1921. A sigla é uma abreviatura para "Rossum's Universal Robots", onde robota quer dizer em tcheco trabalho e bot é uma simplificação da palavra robot.

Nesta peça, o real significado da palavra bot remete à idéia de um homem-máquina capaz de efetuar atividades humanas consideradas pesadas. A peça de Capek retrata uma fantasia dramática em que cada uma das personagens vive dilemas psicológicos, tanto humanos quanto máquinas. Inicialmente, os robôs são criados para substituir o homem nos trabalhos pesados; mas começam a ser vistos como uma máquina "humana" com inteligência e personalidade individual.

A semelhança na característica de imitar o comportamento humano que pode ser feita entre os bots de Capek e os chatterbots ainda provoca muitas confusões nos usuários. Cabe-se salientar que os chatterbots nada mais são do que uma categoria particular de bots, como pode ser visto na tabela 2.2, adaptada de (PRIMO; COELHO, 2000).

Existe uma enorme variedade de categorias de bots. Algumas não foram citadas por não se adaptarem a nenhuma dessas classificações ou não serem suficientemente expressivas. Nota-se, também, que existem bots que podem pertencer a mais de uma categoria ao mesmo tempo, tendo em mente que qualquer bot acadêmico que seja capaz de se comunicar por mensagens, por exemplo, também é um chatterbot e um bot de conhecimento.

Tabela 2.2: Categoria de bots

Categoria	Assunto	Características
academic bots	acadêmico	relacionados aos assuntos acadêmicos como sites de professores ou laboratórios de pesquisa
design bots	planejamento	possuem ferramentas e habilidades para a produção de outros bots e agentes inteligentes
commerce bots	comércio	desempenham atividades de comércio na Internet
fun bots	entretenimento	divertem usuários através de jogos, ambientes virtuais e personagens de realidade virtual
government bots	governamentais	buscam informações em sites governamentais
knowledge bots	conhecimentos	congrega agentes inteligentes, agentes de informação, agentes da web e muitas ferramentas inteligentes de busca
news bots	notícias	criam jornais personalizados e clips de artigos de jornais do mundo inteiro
search bots	buscas	bots e agentes inteligentes de busca na Internet
shopping bots	compras	fazem compras e comparações de preços para internautas
stock bots	investimentos	monitoram o mercado de ações e mandam mensagens sobre os últimos preços e tendências
update bots	novidades	avisam ao usuário quando um site especificado foi atualizado ou modificado.
chatter bots	conversação	são programas que simulam uma conversa com um ser humano.

Os chatterbots, por sua vez, podem ter diversas nomenclaturas, sendo possível encontrar menções a chatter-bot, chatbots, entre outras. Essa categoria teve origem com uma pergunta simples, mas cuja resposta vem instigando pesquisadores das mais diversas áreas: "Podem as máquinas pensar?" A pergunta, proposta no texto do matemático Alan Turing: *Computing, Machinery and Intelligence* (TURING, 1950), originou um teste que foi chamado de Jogo da Imitação, mas que veio a ser conhecido como o Teste de Turing.

2.2.1 O Teste de Turing

O jogo da imitação foi proposto como forma de caracterizar e estudar a possibilidade de se produzir uma máquina dita pensante. Na sistemática original do jogo, três pessoas são envolvidas: um homem (A), uma mulher (B) e um interrogador (C). O interrogador pode ser de qualquer sexo. Ele permanece num quarto, separado dos outros dois. Assim, o objetivo do jogo, para o interrogador, é descobrir quem é o homem e quem é a mulher (Figura 2.1).

Para atingir este objetivo, (C) pode fazer diversos tipos de perguntas, como por exemplo: "Qual o comprimento de seu cabelo?", ou "Some 34.957 e 70.764", ou ainda "Por favor, escreva-me um soneto cujo tema seja amor". Pode-se concluir agora que a estratégia adotada pelo homem talvez seria confundir o interrogador dando respostas como "Meu cabelo é curto, e os fios longos tem cerca de 20 centímetros de comprimento". A mulher poderia ajudar o interrogador dando respostas do tipo: "Eu sou a mulher, não escute a ele". Porém, neste caso, esta resposta seria inútil, já que o homem pode dar o mesmo tipo de respostas.

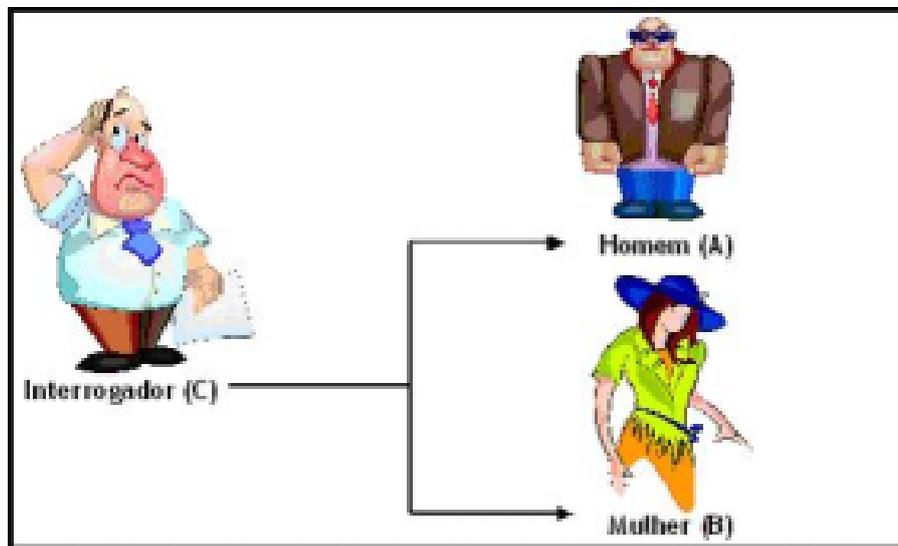


Figura 2.1: O jogo da imitação original

Segundo Turing, muitos cuidados devem ser adotados quando o jogo é colocado em prática. Para que os tons de vozes não ajudem o interrogador, as respostas devem ser datilografadas. Uma outra saída para o problema das vozes é a de se adotar um intermediário que possa repetir as respostas dadas pelo homem e pela mulher.

Reformulando a questão: "O que acontecerá quando uma máquina ocupar o lugar de (A) neste jogo?" Será que o interrogador decidirá erroneamente com a mesma frequência, quando o jogo é jogado dessa forma, do que quando o fazia ao tempo em que o jogo era jogado entre um homem e uma mulher? (Figura 2.2)

Com o novo questionamento, Turing abre a discussão sobre a possibilidade de se dar a uma máquina a capacidade de agir como um ser humano ou de pensar como um, não levando em conta se a composição física de uma máquina seja o silício ou qualquer material que possa ser criado pelo ser humano a base de carbono. Ele afirma: "Há pouca vantagem em tentar tornar uma máquina pensante humana vestindo-a com tal carne artificial". Isto porque na nova sistemática do jogo, o interrogador estaria conversando com o computador e com o homem apenas através da digitação de mensagens, ou seja, sem poder vê-los. Sendo assim, o jogo, da forma como está proposto, abrange um campo da atividade humana básico, servindo para provar como a inteligência humana pode vir a ser modelada e recriada em um computador.

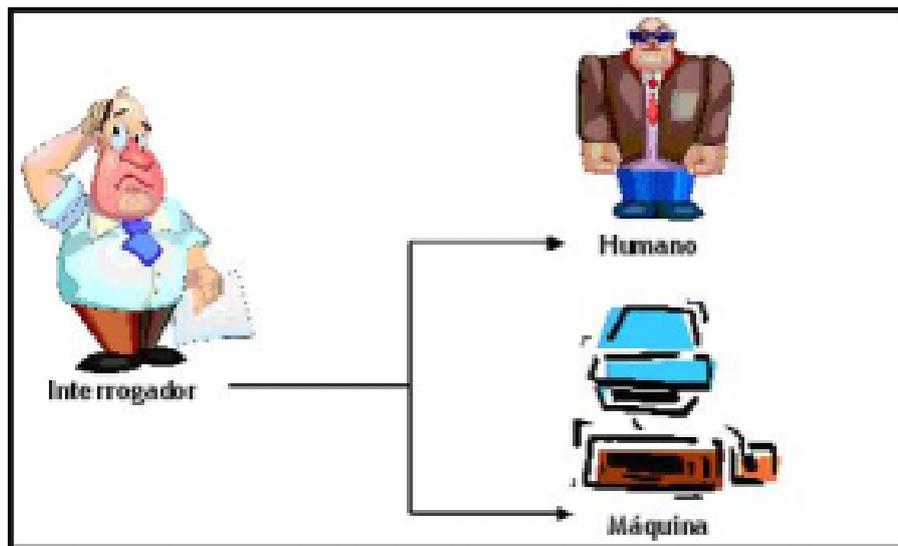


Figura 2.2: O jogo da imitação em sua nova sistemática: o Teste de Turing

2.3 Estratégias de Conversação

Existem algumas estratégias diferentes para a escolha algorítmica da resposta mais apropriada a cada interação de um chatterbot com um usuário (CASSEL, 2002):

- Casamento de padrões: é uma técnica em que é feito um casamento entre um conjunto de palavras-chave e um grupo de respostas relacionadas àquelas chaves, respeitando a ordem das chaves. Em geral, é definida uma certa prioridade entre as respostas. Por exemplo, para uma pergunta que contenha a palavra "mãe", uma possível resposta poderia ser selecionada de um grupo de respostas relacionadas ao tema "família";
- Rede de ativação de respostas: é uma técnica baseada em respostas passadas. Consiste em vários padrões com respostas associadas a um valor de ativação associado a cada nó contendo a resposta. Caso o usuário mencione algum padrão específico, a ativação é estimulada nos nós relacionados e inibida nos outros nós (MAULDIN, 1994);
- Raciocínio baseado em casos: o programa possui um conjunto de casos passados na sua base permitindo percorrer o conjunto de dados à procura de uma solução e responder, adaptando ao caso atual.

Em se tratando de conversação e da forma como se deve iniciar, direcionar ou manter uma conversação com um usuário, muitos trabalhos já vem sendo desenvolvidos traçando uma analogia com a forma humana de abordagem. Os chatterbots fazem uso de diversas estratégias (MAULDIN, 1994) para dar a ilusão de inteligência e fluência, incluindo:

- Manter a iniciativa através do constante questionamento;
- Incluir partes da pergunta do usuário na formação da resposta;
- Mudar o nível de conversação através de perguntas que aprofundem o diálogo, como, por exemplo: "Porque você me perguntou isto?";
- Permanecer maior tempo possível no mesmo tópico, questionando o interlocutor quando o mesmo surpreendentemente mudar o assunto;
- Começar um novo tópico quando a conversa se tornar muito repetitiva;
- Fazer comentários controversos e humorísticos sobre algum assunto que seja foco da conversação.

2.4 Evolução Histórica dos Chatterbots: Alguns Exemplos

De acordo com DeAngeli, Lynch e Johnson (2001), os últimos anos serviram para testemunhar uma explosão extraordinária de interesse nos chatterbots, impulsionado, principalmente, pela área comercial que constantemente necessita estratégias inovadoras para aumentar suas vendas e fidelizar seus clientes. Com a vantagem de poder estar sempre disponíveis e poder manter conversas simultâneas com diversos usuários, os chatterbots se tornam um recurso muito promissor para interação com o ser humano nas mais diversas áreas.

A seguir serão apresentados alguns exemplos de chatterbots já desenvolvidos a fim de ilustrar suas características e evolução histórica, permitindo ao leitor uma contextualização mais abrangente. Os chatterbots citados foram escolhidos dentre os vários existentes por já terem vencido o Concurso de Loebner (LOEBNER, 2003). Este concurso tem o objetivo de premiar anualmente o melhor chatterbot. O vencedor total, ou seja, aquele capaz de criar um chatterbot que realmente seja inconfundível com um humano, receberá o prêmio Loebner, composto por uma medalha de ouro e a quantia de 100 mil dólares. Porém, enquanto não surge um chatterbot capaz de se passar por humano, os melhores de cada ano ganham medalhas de bronze e 2 mil dólares.

2.4.1 Eliza

Considerada como um dos mais antigos e conhecidos chatterbots, Eliza (WEIZENBAUM, 1966) foi desenvolvida pelo professor Joseph Weizenbaum no Massachusetts Institute of Technology em 1966. Seu objetivo inicial é o de simular um psicanalista rogeriano em uma conversa com seu paciente. A terapia rogeriana objetiva fazer o paciente falar e refletir sobre suas emoções, seus problemas e suas convicções. Assim, Eliza espera que o usuário conte seus problemas e participe através de perguntas, estimulando o paciente a contar cada vez mais sobre aquilo que o incomoda. Pela limitação no seu escopo de atuação, Eliza é um programa pequeno, composto, em sua versão original, por cerca de 200 linhas de código.

O algoritmo usado pelos programadores para tentar dar ao diálogo de Eliza uma certa naturalidade e credibilidade é bastante interessante. Sua arquitetura é composta de uma base de regras e um programa. A base de regras é utilizada para processar as sentenças do usuário, enquanto que o programa tem como objetivo ler as sentenças digitadas pelo usuário e processar a base de regras em busca daquela que está associada à sentença digitada.

As regras de Eliza são compostas por uma lista de palavras-chaves. Essas palavras chave se baseiam, por sua vez, em regras de decomposição, responsáveis por separar a sentença de entrada em termos. Da mesma forma, a existência de regras de reconstrução permitem ao sistema fornecer as respostas ao usuário. Assim, em um primeiro momento, Eliza identifica as palavras mais importantes presentes na mensagem de entrada do usuário. Depois, aplica uma regra de modificação na sentença do usuário, considerando a importância de cada palavra e contextualizando tais palavras em um contexto mínimo, por exemplo: a palavra "você" seguida da palavra "é" são classificadas como uma afirmação. Nos casos em que não é encontrada nenhuma regra associada à sentença do usuário, o programa retorna uma resposta livre de contexto ou resposta já utilizada.

Em suma, Eliza trabalha com modificação de sentenças a fim de dar segurança aos usuários. Um exemplo de diálogo onde se observa essa modificação de sentenças pode ser lida abaixo:

Eliza: *"So, how are you doing today?"*
("Então, como você se sente hoje?")

Usuário: *"I am tired."*
("Eu estou cansado")

Eliza: *"Did you come to me because you are tired?"*
("Você veio me ver porque está cansado?")

Usuário: *"Yes, I feel alone."*
("Sim, me sinto só.")

Eliza: *"Do you often feel alone?"*
("Você se sente só constantemente?")

Embora seja um dos chatterbots mais antigos já criados, o princípio de funcionamento interno do software Eliza ainda é utilizado por chatterbots modernos. A figura 2.3, adaptada de (GALVÃO, 2003), ilustra um exemplo simplificado de uma regra utilizada por Eliza.

Quando ocorre a decomposição de uma regra, um número "N" indica que o termo será composto por uma seqüência de "N" palavras. O número zero (0) representa um termo composto por uma seqüência infinita de palavras. Palavras-chaves como, por exemplo, "you" e "me" determinam em qual termo da sentença elas devem aparecer. Nesse caso, se a sentença de entrada do usuário for: "It seems that you hate me", a decomposição ocorrerá da seguinte maneira:

(1) IT SEEMS THAT

(2) YOU

(3)HATE

(4) ME

Quando ocorre a reconstrução da frase, um número "N" representa o n-ésimo termo da decomposição. Portanto, conforme a figura 2.3, podemos notar que a resposta para a entrada anterior seria:

WHAT MAKES YOU THINK I HATE YOU

O software Eliza, porém, tem algumas limitações. Não há uma memória no robô, ou seja, ela não consegue lembrar o que foi falado anteriormente. A construção de algumas respostas também pode desanimar o usuário, já que a modificação de frases de entrada gera, algumas vezes, diálogos uns tanto confusos. Se a entrada fosse somente "I am fine", a conversa pareceria muito natural com a resposta: "How long have you been fine?". Porém, o que se observa é que quando a entrada apresenta complementos além de sua estrutura normal, o robô tende a dizer frases sem sentido. Um exemplo dessa situação pode ser ilustrado da seguinte forma: se a entrada for, por exemplo, a frase: "I am fine now thank you" ("Eu estou bem, obrigada."), a resposta pode ser algo como: "How long have you been fine now thank I?" ("Há quanto tempo você esteve bem agora, obrigada eu?.").

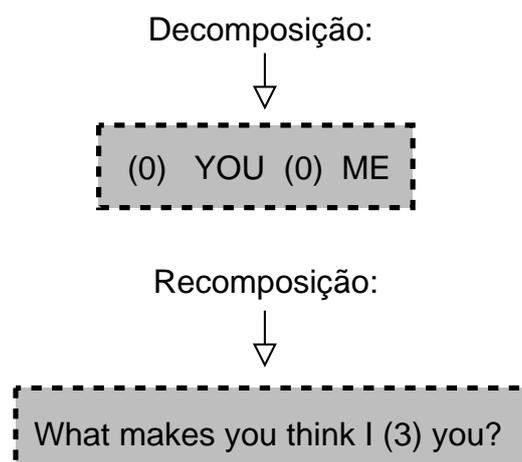


Figura 2.3: Exemplo simplificado de uma regra utilizada por ELIZA

Entretanto, a pesquisa de Primo e Coelho (2001) mostra um fato interessante. Em 1976, Weinzenbaum destacou três fatos curiosos:

- Muitos psiquiatras praticantes acreditaram que Eliza poderia ser desenvolvida até um sistema quase totalmente automatizado de psicoterapia;

- Os usuários se envolviam muito rapidamente com o programa; Quem conversa com ela tende a desenvolver uma relação de afetividade com sua figura, não demonstrando barreiras para expressar seus sentimentos mais profundos.
- Algumas pessoas acreditaram que o programa demonstrava uma solução geral para o problema computacional da compreensão.

Em seu trabalho, DeAngeli, Johnson e Coventry (2001) afirma que Eliza foi e ainda é um sucesso. Ela tem sido o software mais estudado de toda a história da ciência da computação e ainda desperta extrema admiração em quem tem a oportunidade de conversar com ela.

2.4.2 Parry

Parry, desenvolvido por Kenneth Colby (1971), é um chatterbot que tem o objetivo de simular um comportamento paranóico através do rastreamento do seu estado emocional interno.

Ao contrário do que esperado neste tipo de comportamento, a essência de Parry não está em dar respostas individuais, mas no fato de poder manter uma conversação aparentemente coerente. Foi concebido de forma a aparentar ter sempre algo a dizer e sempre mantendo um ar de mistério e paranóia. Ele possui um conceito exagerado de si mesmo e é bastante influenciado pela idéia de perseguição. Dessa forma, suas confusões dentro de um diálogo podem ser sempre interpretadas como um distúrbio momentâneo, não sendo, portanto, julgadas. Ele sabe pouco sobre o mundo e trabalha com um conjunto de 6000 padrões que podem ser combinados com qualquer entrada digitada por um usuário. Suas sentenças de entrada também podem ser modificadas através da eliminação de algumas palavras e utilização de um dicionário, ou através da redução do tamanho da sentença original.

Para Wilks e Catizone (2000), Parry nunca foi tão famoso quanto Eliza, mas sua implementação estende seu algoritmo em vários aspectos. Seu modelo emocional foi projetado de forma a simular comportamentos como medo, raiva e desconfiança e ele é capaz de avaliar a personalidade de seu entrevistador, classificando-o, por exemplo, como amigável, hostil ou dominante. A implementação de tais características ocorre da seguinte maneira: para cada sentença, o valor da intensidade das emoções é modificado de acordo com o que PARRY "pensa" a respeito do assunto abordado. Por exemplo, quando o usuário fala sobre algum tópico que lhe cause raiva, o valor da emoção "raiva" é incrementado. Assuntos considerados neutros fazem com que Parry fique mais amigável, pois diminuem o valor das emoções negativas. Além disso, em seu algoritmo, quando o casamento de padrões não retorna um resultado, ele utiliza bases de conhecimento com temas específicos para montar as respostas, que podem envolver até assuntos como máfia.

Buscando testar seu bot, Colby criou sua própria versão do Teste de Turing: psiquiatras deveriam comparar as interações de Parry e de paranóicos reais para descobrir quem era o chatterbot. Porém, a conclusão dos psiquiatras atestou a qualidade de Parry já que não foi possível distinguir quem eram os humanos de quem era o software. Essa forte personalidade ainda leva muitas pessoas a desacreditar que Parry é, na verdade, um sistema de computador.

2.4.3 Julia

Um TinyMUD (*Multi-User Dungeons*) é um mundo virtual em um jogo onde vários usuários controlam personagens. Este jogo é jogado em rede através de terminais com cli-

entes que emulam ambientes através de interfaces de texto. E é neste mundo que vive Julia (MAULDIN, 1994), criada por Michael Mauldin na Carnegie Mellon University, atuando como um personagem com a função de auxiliar outros usuários, mapeando cavernas e passando mensagens.

Julia pode ser considerada como um agente autônomo, considerando as discussões sobre agentes apresentadas no início deste capítulo, uma vez que ela é pró-ativa, tendo como atividade principal a exploração das cavernas e avaliação do cenário. Quando questionada, ela pode fornecer para cada usuário informações sobre o caminho mais curto do mesmo até um determinado local. Não obstante, ela é capaz de lembrar informações referentes a ações tomadas anteriormente e informações sobre outros jogadores (e.g., quem a matou), para que possam ser utilizadas no processo de tomada de decisão do jogo.

Por estar inserida dentro de um ambiente e atuar sobre ele apenas, ou seja, por ter um escopo reduzido, Julia supera as expectativas em relação ao nível de conversação que pode alcançar. Sua primeira versão apresentava um algoritmo bem simples, baseado em iterações do tipo "if-then-else". Entretanto, posteriormente, suas habilidades conversacionais passaram a ser derivadas de um módulo baseado em Redes Neurais, estruturado com uma rede de avaliação de respostas. Dentro da rede, cada nó consiste em um conjunto de padrões, uma resposta simples, uma lista de nós estimulados e outra de nós inibidos. Quando a entrada do usuário aciona algum padrão, os nós que contém o modelo têm sua ativação estimulada e o de maior nível é selecionado, enviando sua resposta ao usuário.

A personalidade da bot também é curiosa. Em determinados momentos do mês, ao ser questionada, Julia pode afirmar estar sofrendo de tensão pré-menstrual e pode, inclusive, ficar com humor levemente alterado.

2.4.4 Cybelle

Outro chatterbot capaz de interagir em inglês ou francês é Cybelle (AGENTLAND, 2000), criada em 2000. Apresentando uma imagem virtual, Cybelle é construída para ser uma espécie de referência no mundo dos agentes. Além de conversar com ela, em inglês ou francês, podem-se visualizar, ao mesmo tempo, informações sobre agentes virtuais.

Cybelle também está integrada a um portal sobre agentes. A navegação no portal pode ser direcionada durante a própria conversa com o robô por ela mesma ou pelo usuário independentemente. Assim, se a entrada for: "O que exatamente você é?", Cybelle além de fornecer a resposta ainda disponibiliza uma página com informações sobre de sua personalidade. Se a entrada é uma pergunta sobre ALICE, por exemplo, Cybelle mostra informações a respeito e ainda solicita a opinião do usuário sobre o Chatterbot. (Figura 2.4).

2.4.5 A.L.I.C.E

A.L.I.C.E (*Artificial Linguistic Internet Computer Entity*) (WALLACE, 1995) é um dos chatterbots mais populares da atualidade. Sua base de conhecimento é constituída por centenas de fatos, citações e idéias de seu criador e apresenta um vocabulário de mais de 5000 palavras, além de módulos de conversação que capacitam a ela classificar o usuário por idade, sexo, localização geográfica e profissão.

Sua inovação está na forma como é apresentada: além de muita documentação, apresenta uma saudação sonora ao visitante. Criada por Richard S. Wallace na Lehigh University e ativada em 1995, ALICE tem um grande poder de comunicação, além de uma interface gráfica que estimula o diálogo. É programada para dar muitas informações a seu respeito e pode sugerir até que o usuário a veja cantar, além de atuar como uma secretária



Figura 2.4: Cybelle mostrando informações sobre ALICE durante o diálogo (em francês).

que lembra as pessoas de seus compromissos. Sua irreverência vai ainda mais longe já que ela é capaz de espalhar fofocas e contar piada.

O AIML e ALICE representam ponto de partida para muitos outros projetos de chatbots disponíveis hoje na Internet. Para isso, basta que seja desenvolvida uma nova base de conhecimentos em AIML. Atualmente existe uma fundação que promove a disseminação do software, que é gratuito e parte integrante do projeto GNU, bem como a construção outros chatterbots que se utilizem do sistema.

A base do conhecimento e comportamento de ALICE é construída através da linguagem de marcação AIML (*Artificial Intelligence Markup Language*) (TAYLOR, 2003), uma das derivações da linguagem XML (*eXtensible Markup Language*), a ser discutida mais adiante. A performance excepcional deste software no concurso de Loebner pode ser atribuída a três fatores principais (RUSSEL, 2002):

- A licença gratuita do software permite com que as pessoas estejam familiarizadas com o ambiente de desenvolvimento e possam adicionar funcionalidades que permitam ao bot se comunicar mais robustamente com usuários;
- A linguagem utilizada para armazenar o conhecimento do bot é de fácil aprendizagem e permite que conhecimentos sejam adicionados a qualquer momento, por qualquer usuário, de forma a representar seu próprio conhecimento e suas próprias respostas nas mesmas ocasiões;
- A plataforma ALICE e a linguagem AIML são independentes, ou seja, não é necessário ser um programador com grande conhecimento para utilizar o programa.

Este terceiro fator se complementa com o fato de que a plataforma não exige recursos extras para que seja possível sua utilização e de que existem diversas implementações

do sistema (Java, Perl, C++, PHP), responsável pelo processamento de bases de conhecimento em AIML, sendo todas de código aberto. Além disso, a ênfase do trabalho de ALICE é no desenvolvimento da linguagem AIML, e não na tecnologia de suporte ou em uma aplicação particular do chatterbot.

A fim de ilustrar o potencial de ALICE, um exemplo de um diálogo entre ALICE e um dos juízes do Concurso de Loebner é mostrado abaixo. Nesse exemplo, podemos observar que ALICE utiliza truques para tentar se aproximar da realidade das conversações humanas. Ao ser questionada sobre o compositor de uma música, ela simplesmente sorteia um nome de uma lista e o apresenta com um sorriso, como se estivesse brincando com o juiz.

Usuário: *"My name is judge02."*
(*"Meu nome é juiz02"*)

Alice: *"I am always glad to make new friends, judge02."*
(*"Eu sempre fico contente em fazer novos amigos, juiz02"*)

Usuário: *"Hi, please to meet you"*
(*"Oi, prazer em conhecer você"*)

Alice: *"Hi there judge02. Nice meeting you too judge02."*
(*"Olá juiz02. Legal conhecer você também juiz02"*)

Usuário: *"May i ask you some questions?"*
(*"Posso te fazer algumas perguntas?"*)

Alice: *"Sure ask me anything."*
(*"Claro pergunte qualquer coisa"*)

Usuário: *"Who composed beethoven's piano concerto?"*
(*"Quem compôs o concerto de piano de Beethoven?"*)

Alice: *"Charlie Parker. :-)"*
(*"Charlie Parker :-)"*)

Usuário: *"Excellent. Can you name another composer of music for me?"*
(*"Excelente. Você pode me dar o nome de outro compositor musical?"*)

Alice: *"Thanks for the compliment."*
(*"Obrigada pelo elogio."*)

Alice: *"Linus Torvalds. :-)"*
(*"Linus Torvalds. :-)"*)

Com todo seu potencial, ALICE representa uma ferramenta de pesquisa muito interessante para investigar a dinâmica social envolvendo as relações humano-máquinas (ANGELI; JOHNSON; COVENTRY, 2001). De fato, muitos estudos vêm sendo realizados utilizando o chatterbot ALICE e suas variações.

2.5 Um Experimento de Conversação entre Chatterbots

Com o objetivo de ilustrar as potencialidades conversacionais de um chatterbot, Güzeldere (2002) transcreve uma conversa realizada entre Eliza e Parry, a psicanalista rogeriana e o paciente paranóico, respectivamente:

Parry: *"People get on my nerves sometimes."*
(*"As pessoas me dão nos nervos algumas vezes"*)

Eliza: *"I am not sure I understand you fully."*
(*"Eu não tenho certeza se te compreendi totalmente"*)

Parry: *"You should pay more attention."*
(*"Você deveria prestar mais atenção"*)

Eliza: *"Suppose you should pay more attention?"*
(*"Suponho que você deveria prestar mais atenção?"*)

Parry: *"You're entitled to your opinion."*
(*"Você está preso a sua opinião"*)

Eliza: *"What makes you think I'm entitled to my opinion?"*
(*"O que faz você pensar que estou preso a minha opinião?"*)

Parry: *"It bothers me just to be around people."*
(*"Me incomoda estar em companhia das pessoas"*)

Eliza: *"Please go on."*
(*"Por favor continue"*)

Parry: *"Have you ever really thought about the Mafia?"*
(*"Você alguma vez já pensou sobre a máfia?"*)

Eliza: *"Suppose you ever really thought about the Mafia?"*
(*"Suponho que você já tenha pensado sobre a máfia?"*)

Parry: *"Let's talk about something else."*
(*"Vamos falar de outra coisa"*)

Eliza: *"I am not sure I understand you fully."*
(*"Eu não tenho certeza se te compreendi completamente"*)

Parry: *"You keep asking me the same things, why not change the topic?"*
(*"Você fica me perguntando sempre as mesmas coisas. Porque não mudar de assunto?"*)

Eliza: *"In what way?"*
(*"De que modo?"*)

.....

Parry: *"I am getting tired of answering the same questions over and over."*
("Estou cansado de responder sempre as mesmas coisas")

Eliza: *"What resemblance do you see?"*
("Que semelhança você vê?")

Parry: *"Why do you need to know that?"*
("Porque você precisa saber disso?")

Eliza: *"What do you think?"*
("Porque você acha?")

.....

Parry: *"You harp on the same questions too much."*
("Você insiste demais nas mesmas perguntas")

Eliza: *"What does that similarity suggest to you?"*
("E o que essa similaridade sugere a você?")

Parry: *"I don't get you."*
("Não te entendi")

Eliza: *"Do you wish that you did get it?"*
("Você gostaria de ter entendido?")

Parry: *"It's about time you showed some interest in my feelings."*
("Já era tempo de você demonstrar interesse em meus sentimentos")

Eliza: *"Oh, I showed some interest in your feelings?"*
("Oh, eu demonstrei interesse em seus sentimentos?")

Apesar da conversa ser um pouco confusa, por serem talvez os primeiros bots criados, Eliza e Parry demonstram uma grande naturalidade e desenvoltura em seu diálogo. Podemos notar que o diálogo ilustra as características apresentadas anteriormente para ambos os chatterbots, como por exemplo, o estímulo para que o paciente fale sobre o que lhe chateia através da sugestão "Por favor, continue" e a saída utilizada pelo paranóico Parry ao responder sobre um assunto genérico de sua base, a máfia: "Você alguma vez já pensou sobre a máfia?".

2.6 Considerações Finais Sobre o Capítulo

Chatterbots são sistemas que procuram simular uma conversação similar a de um ser humano a fim de enganar, ao menos momentaneamente, um interlocutor. Essa característica desperta o interesse da comunidade científica e industrial, pois, além desses sistemas

proverem uma interface mais amigável e natural, eles permitem explorar as relações sociais que os usuários desenvolvem com máquinas.

O desenvolvimento de um chatterbot se baseia no "Jogo da Imitação"(TURING, 1950), que tem como propósito fazer com que um ser humano acredite que esteja conversando com outra pessoa. Dependendo da finalidade em que o bot é empregado, esse recurso faz com que um usuário seja estimulado a interagir com um computador através da troca de mensagens com o sistema, já que não necessita nenhuma aprendizagem ou conhecimento prévio para realizar a comunicação e obter as informações que deseja.

A primeira experiência no desenvolvimento de chatterbots foi através do software Eliza (WEIZEMBAUM, 1966). Desde então surgiram diversas versões e variantes deste sistema, além de novas técnicas de conversação que permitiram o avanço no estudo e implementação. Este capítulo apresentou uma visão geral sobre estes sistemas, suas características, origens e exemplos. Um quadro teórico sobre o assunto é extremamente importante para o entendimento do potencial dos robôs de conversação quando aplicados aos mais diversos campos de estudo.

3 GERÊNCIA DE REDES

As redes de computadores nasceram e evoluíram a fim de facilitar a troca de dados, informações e serviços entre usuários e entidades separadas (FULLER, 1999). A evolução tecnológica e a conseqüente diminuição dos custos dos computadores tornaram cada vez mais atraente a possibilidade da interconexão de computadores em redes. Atualmente, a acelerada evolução da tecnologia da informação tem trazido profundos impactos em diversas frentes envolvidas em sua aplicação. Dessa forma, ao mesmo tempo em que as redes de computadores criaram inúmeras oportunidades para usuários e empresas se comunicarem e compartilharem informações, essa tecnologia digital criou um grande desafio no que diz respeito a sua administração e gerenciamento.

Gerenciar uma rede tem se tornado uma tarefa complexa. Os gerentes de rede têm se preocupado cada vez mais em realizar um trabalho conciso, de forma a fornecer maior grau de confiabilidade para seus usuários. Dessa forma, estes profissionais precisam aprender a lidar com um crescente número de dispositivos, recursos e ferramentas disponíveis para este fim. Essa necessidade de aprendizado se justifica, também, pelo fato de que uma administração falha de uma rede pode provocar um grande impacto no funcionamento das redes domésticas e de corporações de todo porte, já que a interrupção de seu funcionamento pode causar atraso ou não recebimento de dados importantes, entre outras conseqüências.

Assim, a atividade de gerenciamento de redes tem por objetivo maximizar o controle organizacional das redes de computadores, de maneira mais eficiente e confiável, ou seja, planejar, supervisionar, monitorar e controlar qualquer atividade da rede. Este capítulo apresenta alguns conceitos sobre gerência de redes, destacando o protocolo SNMP. Além disso, apresenta uma breve discussão sobre a integração de conceitos e ferramentas de Inteligência Artificial com a prática do gerenciamento de redes, a fim de ilustrar a potencialidade de tal abordagem.

3.1 Conceituação e Contextualização

O gerenciamento de redes pode ser definido como o ato de inicializar, monitorar e modificar a operação das funções primárias de uma rede. Podemos definir funções primárias como aquelas que suportam diretamente as necessidades de um usuário, ou seja, aquelas que permitem, por exemplo, que os usuários acessem diversos dispositivos em uma rede e possam trocar informações através dela. Assim, o gerenciamento de redes se faz necessário a fim de colocar e manter em funcionamento os dispositivos de uma rede que realizam tais atividades primárias (PRAS, 1995).

As atividades básicas do gerenciamento de redes consistem na detecção e correção de falhas em um tempo mínimo e no estabelecimento de procedimentos para a previsão de

problemas futuros. Através dessas atividades é possível tomar medidas que evitem o colapso da rede, como a reconfiguração das rotas ou a troca do roteador por um modelo mais adequado, através da monitoração de linhas cujo tráfego esteja aumentando ou roteadores que estão se sobrecarregando.

Embora possamos encontrar na literatura outras definições sobre gerenciamento de redes que não a citada anteriormente, podemos dizer que, de um modo geral, a atividade de gerenciamento de rede consiste no controle e administração de forma racional dos recursos de hardware e software em um ambiente distribuído buscando melhor performance e eficiência. O gerenciamento de redes permite visualizar informações relevantes para garantir a saúde da rede como um todo. A visualização de informações como estados de dispositivos, tráfego de pacotes e tipo de tráfego, por exemplo, possibilitam que um gerente detecte algum problema na rede e possa tomar uma atitude em relação a esse problema.

Dentre as razões que fazem com que a atividade de gerenciamento de redes seja necessária, podemos salientar aquelas que se referem às expectativas dos usuários (TERPLAN, 1992) apud (STALLINGS, 1996):

- Recursos estratégicos controlados: os recursos presentes nas redes de computadores assumem constantemente uma importância vital nas organizações. Sem um controle efetivo, esses recursos podem não prover o retorno esperado;
- Controlar a complexidade: o crescimento contínuo no número de componentes, usuários finais e protocolos de uma rede podem implicar em perda de controle sobre quais os equipamentos conectados a uma rede e como tais equipamentos estão sendo usados;
- Melhorar o serviço: usuários finais esperam um serviço do mesmo nível ou de nível maior a medida que os recursos computacionais crescem;
- Balancear necessidades: os recursos computacionais de uma organização precisam fornecer aos usuários finais aplicações com sustentação, com foco nas áreas de desempenho, de disponibilidade, e de segurança. O gerente de rede deve atribuir e controlar recursos para balançar estas várias necessidades;
- Aumentar o tempo de disponibilidade: quando os recursos computacionais se tornam mais importantes, os usuários esperam que sua disponibilidade de uso se aproxime de 100%;
- Controlar custos: os recursos de uma rede devem ser monitorados e controlados a fim de que as necessidades dos usuários sejam atendidas com o menor custo possível.

Complementando os itens acima, o trabalho de Pras (1995) se aprofunda um pouco mais. Ele divide a atividade de gerenciamento de redes em algumas etapas: planejamento, implantação e operação. Em seguida, aponta alguns pontos que considera importantes para que o gerenciamento de redes se faça necessário, enumerando as atividades necessárias para melhor solucionar cada um dos aspectos levantados.

Por exemplo, no caso da redução e controle de custos, os usuários finais de uma rede anseiam pelos melhores equipamentos ao menor custo possível. Assim, já em sua fase de planejamento, um gerente de redes deve se preocupar em criar um projeto que seja capaz

de atender um maior número possível de tipos de usuários. Já o crescimento das tecnologias de redes de computadores e a falta de conhecimento de todas as particularidades de tais tecnologias, podem provocar muitas falhas. Quando os equipamentos não se comportarem da forma como esperado, uma rede pode vir a ficar indisponível, bem como seus principais serviços. Essas situações são críticas e fazem com que um gerente deva se preocupar com a rede gerenciada, tentando aprender e observar ao máximo o comportamento e os desvios de comportamento de sua rede e de seus equipamentos.

Essa última característica aponta para uma atividade importante dentro do gerenciamento de redes: a habilidade de adquirir informações sobre os equipamentos envolvidos em uma rede e as mudanças que ocorrem nos mesmos. No início, a coleta de informações sobre os equipamentos e mudanças em uma rede eram feitas de várias formas não padronizadas. Porém, com o crescimento das redes, foi surgindo a necessidade de desenvolvimento de um protocolo padronizado para o gerenciamento de redes, que pudesse ser facilmente aprendido e usado para monitorar e configurar diversos equipamentos distintos. Assim, foi criado o protocolo SNMP (*Simple Network Management Protocol*), que ainda é amplamente utilizado nos dias atuais.

3.2 SNMP

No início da década de 80, o modelo SNMP (CASE et al., 1990; CASE; ROSE; WALDBUSSER, 1993) começou a ser desenvolvido pelo IETF (*Internet Engineering Task Force*), com o objetivo de disponibilizar uma forma simples e prática de realizar o controle de equipamentos em uma rede de computadores. Atualmente, pela simplicidade de sua arquitetura, quase todos os equipamentos disponíveis nas atuais redes de computadores suportam o SNMP (TSAI; CHANG, 1998). Esse modelo de gerenciamento consiste em 4 elementos chave:

- Estação de Gerenciamento (gerente);
- Nó gerenciado (agente);
- Informações de gerenciamento;
- Protocolo de gerenciamento.

Os nós gerenciados podem ser hosts, roteadores, impressoras, *switches* ou quaisquer outros equipamentos que possam comunicar suas informações de status. Para que tais dispositivos possam ser gerenciados, é necessária a presença de um agente SNMP. Um agente SNMP responde requisições feitas pelas estações de gerenciamento e pode, ainda, prover informações importantes para a estação de gerenciamento, mesmo que estas não sejam requisitadas. Uma estação de gerenciamento é um dispositivo que contém um ou mais processos que se comunicam com os agentes espalhados pela rede, emitindo comandos e obtendo respostas.

Cada dispositivo dentro da rede deve manter uma ou mais variáveis que descrevem seu estado, que são chamadas de objetos gerenciados. Um objeto gerenciado é a visão abstrata de um recurso real do sistema. Assim, todos os recursos da rede que devem ser gerenciados são modelados, e as estruturas dos dados resultantes são os objetos gerenciados. Os objetos gerenciados podem ter permissões para serem lidos ou alterados, sendo que cada leitura representará o estado real do recurso e, cada alteração também será refletida no próprio recurso. O conjunto formado por todos os objetos gerenciados possíveis

em uma rede é agrupado em uma estrutura de dados chamada de MIB (*Management Information Database*). Assim, as MIBs armazenam as informações de gerenciamento de cada dispositivo de uma rede.

Para que ocorra a interação entre a estação de gerenciamento e os agentes, é utilizado um protocolo de gerenciamento, chamado de protocolo SNMP. Este protocolo define algumas mensagens para troca de informações:

- **GetRequest:** tem como objetivo solicitar o valor de uma variável;
- **GetNextRequest:** tem como objetivo solicitar o valor da variável seguinte, ou seja, o gerente fornece o nome de uma variável e o cliente obtém o valor e o nome da próxima variável;
- **GetBulkRequest:** solicita uma tabela (valores de uma tabela). Essa mensagem também serve para diminuir o tráfego de mensagens em uma rede;
- **SetRequest:** tem como objetivo atualizar o valor de uma variável.
- **GetResponse:** tem como objetivo retornar o valor de uma variável.
- **InformRequest:** tem como objetivo trocar informações de gerenciamento entre gerentes.
- **Trap:** tem como objetivo reportar um evento inesperado, ou seja, quando os nós gerenciados apresentam comportamento não usual (congestionamentos, falhas, entre outros.) uma mensagem deste tipo é enviada ao gerente reportando que um evento significativo ocorreu.

As figuras 3.1 e 3.2 ilustram o funcionamento do protocolo. Um gerente (em uma estação de gerenciamento) envia mensagens SNMP a um ou mais agentes com objetivo de extrair informações relevantes sobre o estado atual da rede e do equipamento (**GetRequest**, **GetNextRequest**, **GetBulkRequest**) ou solicita a atualização de uma variável (**SetRequest**). O agente, por sua vez, consulta as informações solicitadas utilizando a MIB e repassa novamente para o gerente (**GetResponse**). Além disso, quando uma falha ou evento inesperado acontece em um equipamento, o agente é responsável por enviar um relatório ao gerente, reportando o evento (**Trap**).

O SNMP utiliza o protocolo de UDP para transportar as mensagens. O protocolo UDP se caracteriza por ausência de conexão. Assim, nenhuma conexão é mantida entre a estação de gerenciamento e os agentes espalhados pela rede. Dessa forma, cada troca de mensagens é uma transação separada entre tais componentes.

3.3 Inteligência Artificial e o Gerenciamento de Redes

A Inteligência Artificial pode ser definida como a parte da ciência da computação que concerne ao projeto de sistemas computacionais que exibem inteligência humana: aprender novas informações, entender linguagens, raciocinar e resolver problemas (ARNOLD; BOWIE, 1986). Mais profundamente, Inteligência Artificial é um campo de estudos multidisciplinar, originado da computação, da engenharia, da psicologia, da matemática e da cibernética, cujo principal objetivo é construir sistemas que apresentem comportamento inteligente e desempenhem tarefas com um grau de competência equivalente ou superior ao grau com que um especialista humano as desempenharia (NIKOLOPOULOS, 1997).

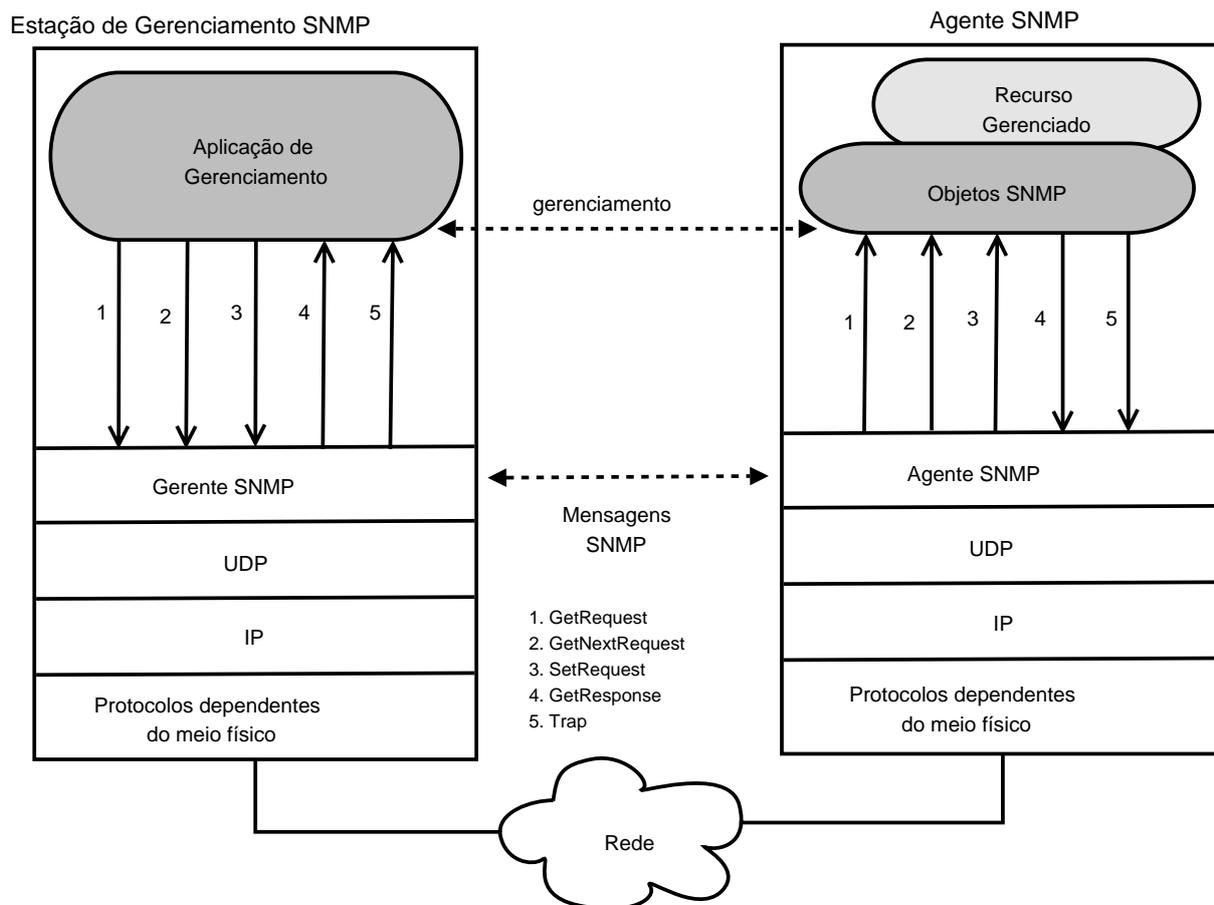


Figura 3.1: Funcionamento do SNMP (STALLINGS, 1996)

Essa possibilidade de aproximação do comportamento humano faz com que os estudos de Inteligência Artificial venham apresentando interessantes soluções computacionais para os mais diversos problemas, dentro dos novos campos que se abrem na Tecnologia da Informação. Nesse contexto, sua aplicação busca tornar o computador capaz de realizar atividades tipicamente humanas, como o reconhecimento de padrões, o diagnóstico e a tomada de decisão, simulando um comportamento inteligente.

Muitos estudos têm sido feitos com o objetivo de integrar tecnologias de Inteligência Artificial na área de redes e gerenciamento de redes. Soluções que utilizam Agentes e sistemas especialistas já foram utilizados com a intenção de facilitar o processo de gerenciamento de redes ou auxiliar no treinamento de profissionais que atuam nesse campo. A seguir, são apresentados alguns estudos feitos com este objetivo, a fim de apresentar as potencialidades da integração de tais tecnologias, ilustrando como a utilização dessas tecnologias pode ser bastante interessante em ferramentas desenvolvidas para as mais diversas funções dentro da área de gerenciamento de redes.

3.3.1 Agentes

A utilização de agentes e sua organização em SMAs (Sistemas Multiagentes) permite com que se possa trabalhar com entidades de software reusáveis, mais simples e mais flexíveis. Nessa situação, um grupo de agentes passa a representar muito mais do que o somatório das características de seus componentes. (MULLER, 1996)

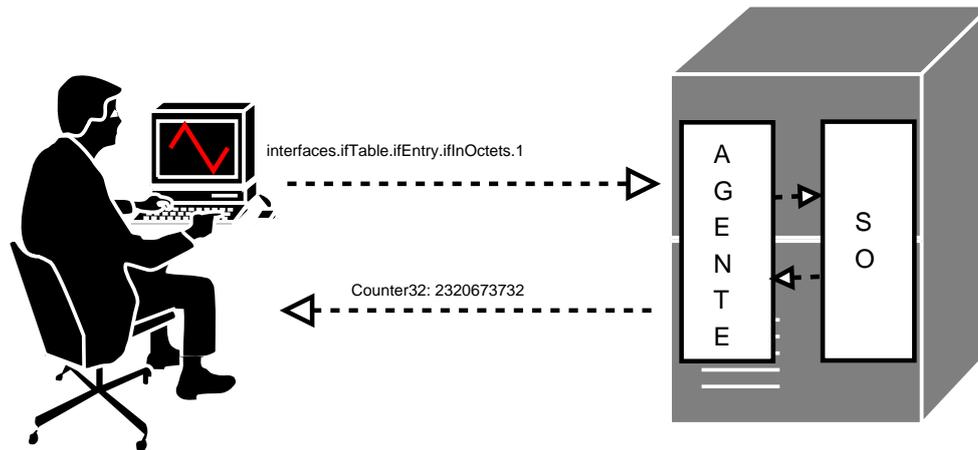


Figura 3.2: Um exemplo de funcionamento do protocolo SNMP

3.3.1.1 Agentes Móveis

Agentes móveis são capazes de se locomover entre diversos pontos, usando as informações capturadas em suas visitas para realizar alguma atividade específica. Usados na área de gerenciamento de rede, os agentes móveis apresentam a possibilidade de encapsular *scripts* de gerenciamento e executar tais *scripts* no exato ponto da rede em que essa execução se faça necessária. Assim, um agente móvel pode ser enviado para um domínio de uma rede e percorrer todos os seus elementos a fim de coletar e processar dados pertinentes para atividades de gerenciamento.

Uma aplicação de gerenciamento que utiliza agentes móveis é apresentada por Appleby e Steward (1994). Nela, agentes móveis são usados para controlar o congestionamento de tráfego em uma rede. Uma primeira classe de agentes móveis, chamados agentes pais, navega randomicamente pelos nós da rede e coletam informações sobre sua utilização. Através dessas informações, eles podem medir uma média de utilização aproximada de tais nós, além de poderem identificar quais nós apresentam congestionamento. Quando um desses nós é identificado, um agente móvel para balanceamento é criado. Esse agente atualiza as tabelas de roteamento do mesmo e de seus vizinhos a fim de reduzir o tráfego do nó congestionado.

Magedanz (1996) apresenta outra possibilidade de utilização de agentes móveis. Um servidor pode enviar agentes móveis para os computadores de usuários finais a fim de adaptar seus equipamentos para os novos serviços que o mesmo deseja oferecer.

Embora os agentes móveis apresentem grandes possibilidades quando aplicados em um ambiente de gerenciamento de redes, seu uso deve ser avaliado com cuidado. Em se tratando de consumo de banda, por exemplo, nem sempre é possível garantir que tais agentes sejam mais eficientes do que aplicações de gerenciamento que trocam informações através de mensagens na rede e processam tais informações de forma centralizada (CHEIKHROUHOU; CONTI; LABETOULLE, 1998).

3.3.1.2 Agentes Inteligentes e Sistemas MultiAgentes

Com a utilização de agentes inteligentes no gerenciamento de redes, hierarquias de gerenciamento podem ser estabelecidas, cada qual com responsabilidades distintas. Isso possibilita que os problemas sejam escalonados dentre os níveis da hierarquia e, dessa forma, sejam solucionados mais rapidamente (MULLER, 1997).

Um exemplo de aplicação que se utiliza de tecnologias de agentes inteligentes é apresentada no projeto MANIA (*Managing Awareness in Networks with Intelligent Agents*) (OLIVEIRA; LABETOULLE, 1995) e tem como objetivo gerenciar QoS (Qualidade de Serviço) em uma rede. Nessa aplicação, os agentes são estruturados de forma a apresentar desejos, intenções, objetivos e comprometimento. Nesse modelo, os desejos dos agentes expressam sua percepção do ambiente e estão divididos em três partes: a primeira descreve o estado de uma rede em tempo real (exemplo: a impressora está sendo muito solicitada neste momento). A segunda parte contém o comportamento histórico dessa mesma rede. O comportamento histórico armazenado é usado para adquirir algum aprendizado sobre a dinâmica do comportamento da rede. Considerando o exemplo anterior, podemos citar como exemplo o fato de a impressora ser bastante utilizada todos os dias entre 10 e 11 horas da manhã. A terceira parte tem por objetivo traduzir o estado dos serviços disponíveis na rede, por exemplo, qual o número máximo de clientes que um servidor web pode aceitar em um determinado período de tempo.

Adicionalmente, os desejos dos agentes podem conter uma outra parte, que seria responsável por atender as exigências dos usuários em relação à aplicação. Por exemplo, conter informações sobre as expectativas e exigências do usuário em termos de QoS.

Os desejos dos agentes consistem em duas partes. A primeira corresponde aos pedidos que o agente não foi capaz de satisfazer (exemplo: um usuário deseja transmitir vídeo quando não há banda disponível). A segunda parte consiste de políticas de motivação. Um administrador de rede pode querer motivar um agente de forma a atribuir uma certa prioridade a determinadas pessoas dentro de um projeto.

Seu funcionamento pode ser resumido da seguinte forma: quando um agente recebe uma descrição das necessidades de um determinado usuário e seus parâmetros de QoS, ele então transforma essas necessidades e parâmetros em um conjunto de objetivos. Os objetivos, por sua vez, são independentes do estado do sistema. Após este processo, os objetivos são mapeados em intenções. As intenções levam em consideração o estado do sistema e da rede, os meios disponíveis para atingir os objetivos e as restrições que podem ocorrer durante o processo. Por fim, a motivação do agente e suas possíveis políticas de funcionamento definem a parte do comprometimento dos agentes.

3.3.1.3 *Agentes de Interface*

Um agente de interface é um programa de computador que emprega tecnologias de IA para prover assistência a um usuário que trabalha com uma aplicação particular de computador. Tais assistentes aprendem "olhando sobre o ombro" do usuário e detectando padrões e regularidades no comportamento do mesmo (MAES; KOZIEROK, 1993).

O trabalho de Esfandiari, Deflandre e Quinqueton (1996) se utiliza de tais agentes para processar uma grande quantidade de alarmes e eventos, filtrando os mesmos e relacionando cada notificação a um contexto. Usando um modelo específico que correlaciona eventos que terminam em uma mesma ação, o agente é capaz de executar racionalização temporal e automatizar as tarefas de gerenciamento. Enquanto o agente olha pelo ombro do usuário, ele identifica eventos crônicos e associa cada um com uma ação. Além disso, um sistema de aprendizado avalia os eventos identificados, comparando com eventos anteriores, de forma a armazenar o conhecimento em uma base de eventos. Conforme a definição apresentada por Maes e Kozierok (1993), esse agente aprende pela observação das ações do usuário ou através de instruções explícitas que podem ser passadas a ele.

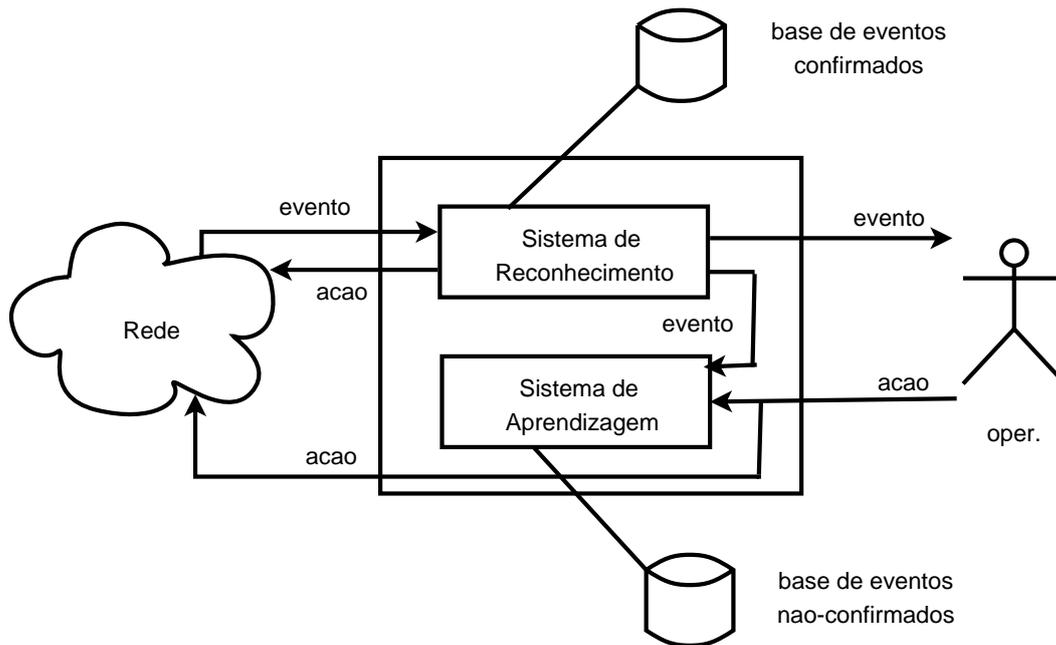


Figura 3.3: Arquitetura do agente proposto por (ESFANDIARI; DEFLANDRE; QUINQUETON, 1996)

3.3.2 Sistemas Especialistas

Jackson (1986) define sistemas especialistas como aplicações capazes de representar e raciocinar sobre algum domínio de conhecimento a fim de auxiliar na resolução de problemas ou fornecer conselhos. Estendendo essa definição, podemos caracterizar um sistema especialista como (MEDINA, 1996):

- Um programa de computador que incorpora domínio especialista abstraído de dados, frequentemente em forma de regras;
- Um sistema que emprega conhecimento humano para resolver problemas que ordinariamente requerem inteligência humana;
- Uma classe de sistemas de IA desenvolvidos para servirem como consultores na tomada de decisões que envolvam áreas restritas da ciência, normalmente apenas dominadas por especialistas humanos.

No cenário competitivo atual, construir e gerenciar conhecimento de apoio a usuários no controle de processos pode ser útil para uma organização, principalmente para processos que ocorram sob incertezas e com dados incompletos. Assim, a utilização de sistemas especialistas em um domínio de gerência de redes é muito vantajosa já que esses sistemas capturam as estratégias de resolução de problemas dos especialistas do domínio e aplicam-nas de forma a fornecer um apoio ao usuário. Esse tipo de sistemas pode ser utilizado dentro da gerência de rede a fim de auxiliar no planejamento de redes, executar diagnóstico de falhas e detectar tentativas de intrusão, por exemplo.

Em um sistema desse tipo é necessária atenção quanto à representação e aquisição do conhecimento sobre o domínio trabalhado. O conhecimento dos especialistas precisa refletir a experiência obtida pelos mesmos, ou seja, deve permitir uma concentração nas causas mais prováveis de um problema e adaptar as respostas para problemas específicos.

A representação do conhecimento, por sua vez, deve codificar o conhecimento do especialista de modo a armazenar, processar e utilizar o conhecimento especificado da melhor maneira possível.

Assim, um engenheiro de conhecimento passa a ter uma função crucial no desenvolvimento de tais sistemas. Um engenheiro de conhecimento é a pessoa que exerce o duplo papel de compreender o domínio da informação para interagir com o especialista e possuir o conhecimento necessário sobre computação, linguagens e ferramentas para selecionar o melhor ambiente e forma de implementar o sistema. Para tal tarefa, ele é responsável por realizar a interface com o especialista, dirigindo as entrevistas, acompanhando em seu trabalho (MEDINA, 1996).

Dessa forma, o Engenheiro do Conhecimento deve possuir capacidade para projetar um Sistema Especialista, dominando as diversas técnicas disponíveis para tal, além de possuir um conhecimento razoável sobre o domínio sobre o qual está desenvolvendo o sistema, sem o qual a compreensão das informações recebidas dos especialistas é prejudicada. Além disso, é imprescindível que o engenheiro de conhecimento obtenha respostas sobre as seguintes questões, que possibilitam a informação inicial que um engenheiro de conhecimento necessita para implementar um sistema especialista (SCOTT, 1991):

- Que passos um especialista realiza para resolver uma tarefa (estratégia do conhecimento)?
- Como o especialista reage desde a chegada das informações iniciais até a conclusão da solução do problema?
- Quais as características de casos o especialista usa na solução de problemas?

Sistemas Especialistas podem ser desenvolvidos utilizando-se diversos paradigmas. Dentre os paradigmas existentes, destacam-se o raciocínio baseado em regras e o raciocínio baseado em casos. Dentre alguns dos sistemas especialistas existentes na área de gerenciamento de redes (MEDINA, 1996; NUNES, 1997; TAROUCO, 1990), uma aplicação que utiliza raciocínio baseado em casos é o sistema DUMBO (MELCHORS, 1999; MELCHORS; TAROUCO, 1999), que será brevemente descrito mais adiante neste capítulo.

3.3.2.1 *Raciocínio Baseado em Regras*

O raciocínio baseado em regras consiste de uma memória de trabalho, uma base de regras e procedimentos de controle. Nesses casos, as regras podem efetuar ações como testes na rede, consultas a banco de dados e envio de avisos, entre outros. Quando uma rede se encontra num estado anormal, são selecionadas regras aplicáveis para a situação corrente e, uma dentre essas regras é de fato selecionada e executada através de estratégias de controle.

3.3.2.2 *Raciocínio Baseado em Casos*

O raciocínio baseado em casos propõe soluções para novos problemas através da recuperação de casos similares ocorridos no passado. Assim, um caso representa uma situação ocorrida anteriormente que é, por sua vez, capturada e reutilizada na solução de novos problemas, a medida que estes vão surgindo. Sistemas desenvolvidos com base em tal paradigma possuem capacidade de aprendizado, uma vez que um novo problema pode

ser armazenado após ter sido solucionado, tornando-se disponível para futuros problemas e aumentando o conhecimento presente no sistema (MELCHIORI, 1999).

A figura 3.4 representa o ciclo de processos de um sistema que se utiliza do paradigma de raciocínio baseado em casos, que é composto de 4 processos: recuperação de casos, reutilização, revisão e armazenamento.

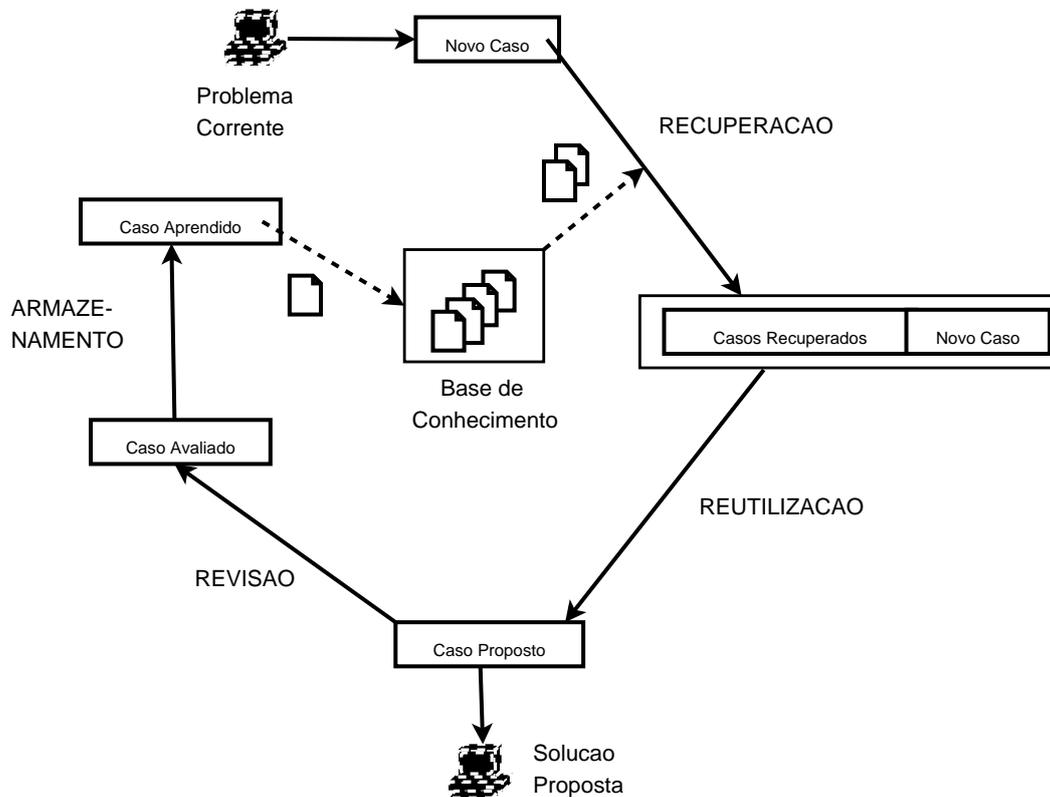


Figura 3.4: Ciclo de processos de um sistema que utiliza o paradigma CBR

3.3.2.3 Sistema DUMBO

O sistema Dumbo é um sistema orientado para o diagnóstico de redes TCP/IP que utiliza o paradigma CBR. No sistema, cada um dos casos apresentados é composto de duas partes principais: a descrição e a solução. A parte de descrição tem por objetivo apresentar os aspectos do problema. Esta parte é composta por informações de abertura de registro de problemas, além das características hierárquicas e específicas do mesmo. A parte de solução apresenta as ações que devem ser tomadas com o objetivo de minimizar ou sanar o problema previamente apresentado.

Os casos são classificados em diferentes categorias, formando do uma árvore hierárquica de tipos de problemas. O ciclo de raciocínio do sistema inicia através do módulo de definição de contexto cujo objetivo é obter as primeiras informações sobre o problema. A seguir, é acionado um módulo de procura que busca casos que casam com as informações fornecidas. Estes casos podem ser visualizados ou o usuário poderá iniciar um processo de refino através da resposta de algumas características específicas. Após este processo, o módulo de recuperação é iniciado e busca casos que contenham tais características.

Nos casos em que o sistema não é capaz de propor um conjunto de casos aproveitáveis, é então acionado um módulo de aprendizado. Este módulo tem por objetivo aprender tais

casos e incluir os mesmos na base de conhecimento. Nessa etapa, o sistema solicita, além das informações sobre encerramento de um registro de problema (tais como causas, solução, autor, data), a confirmação das características adicionais e inclusão de novas características que possam auxiliar no diagnóstico de problemas similares.

Os casos presentes no sistema DUMBO foram alimentados através de entrevistas com peritos, manuais de dispositivos de redes, referências teóricas sobre redes, entre outros.

3.4 Considerações Finais Sobre o Capítulo

Este capítulo apresentou uma breve revisão bibliográfica sobre a área de gerenciamento de redes, enfocando, especialmente, o protocolo SNMP, que é utilizado pelo sistema desenvolvido neste trabalho para prover uma interface com a rede gerenciada. Além disso, buscou enumerar exemplos de aplicações de gerenciamento de redes que se utilizam de tecnologias de IA a fim de demonstrar o potencial dessa tecnologia na área.

As redes de computadores estão se tornando maiores e mais heterogêneas, de modo que constituem um domínio em que o uso de tecnologias de IA proporcionam inúmeros benefícios. Administradores de sistemas e de redes podem utilizar tecnologias de IA para executar processos críticos na área de gerenciamento de redes, como, por exemplo, detecção e correção de falhas e monitoração de performance.

Toda essa preocupação em integrar essas duas áreas da ciência da computação pode ser justificada através do trabalho de Cheikhrouhou, Conti e Labetoulle (1998) que afirma que o número de fabricantes que estão buscando desenvolver e adaptar tecnologias de IA em dispositivos como roteadores e *switches* vêm crescendo de forma acelerada. A idéia por trás de tal crescimento é a de prover dispositivos de rede capazes de conter um certo nível de inteligência, que os permitiria ganhar certas propriedades de autonomia e beneficiaria a automatização das atividades de gerenciamento.

Esse estudo, por sua vez, é muito importante em se tratando de caracterizar a escolha da utilização de processamento de linguagem natural (também parte dos estudos de IA) para o treinamento de profissionais em uma área como a de gerenciamento de redes, que exige, através das ferramentas atuais, um certo conhecimento prévio sobre o tema.

O capítulo a seguir vem apresentar o sistema proposto neste trabalho, um chatterbot que busca treinar administradores de redes menos experientes, de forma a capacitá-los para solucionar possíveis problemas que podem ocorrer nas atuais redes de computadores. Além da motivação para a construção de tal sistema, serão apresentadas suas características e aspectos de implementação.

4 DOROTY: CARACTERÍSTICAS E IMPLEMENTAÇÃO

4.1 Motivação

Como visto no capítulo anterior, cada vez mais as tecnologias de IA vêm sendo utilizadas no domínio das redes de computadores e, principalmente, no gerenciamento de redes. Essa utilização crescente de tais tecnologias busca prover uma maneira mais eficiente de realizar o gerenciamento, automatizando alguns processos envolvidos no contexto e, dessa forma, facilitando a utilização das ferramentas de gerenciamento sob o aspecto humano.

Também verificamos que, em geral, as atividades básicas do gerenciamento de redes envolvem o controle e administração de forma racional dos recursos de hardware e software em um ambiente distribuído buscando melhor desempenho, eficiência e segurança do sistema. Assim, o gerenciamento tem por objetivo maximizar o controle organizacional das redes de computadores, de maneira mais eficiente e confiável, ou seja, planejar, supervisionar, monitorar e controlar qualquer atividade da rede (PINTO, 1995).

Porém, com o crescimento da quantidade e da heterogeneidade dos equipamentos presentes na atualidade, o gerenciamento eficaz dos recursos presentes nas redes de computadores torna-se um tanto complexo. Administradores de redes devem estar bem preparados, a fim de evitar indisponibilidades na rede e falha de seus componentes cruciais. Dessa forma, devem ser capazes de absorver informações rapidamente, além de reforçar conhecimentos teóricos e aplicá-los na utilização das mais diversas ferramentas de gerenciamento que se encontram disponíveis.

O que se observa, porém, é que nem sempre profissionais treinados e com vasta experiência atuam nas redes de grandes corporações. Muitas vezes estagiários são designados para monitorar equipamentos e redes, principalmente em turnos onde os profissionais mais qualificados não desejam permanecer trabalhando.

A motivação deste trabalho, portanto, resume-se em propor uma solução inovadora utilizando chatterbots para suprir principalmente a necessidade de treinamento e capacitação de alguns profissionais que atuam no gerenciamento de redes de computadores.

A escolha de uma tecnologia que se utiliza de linguagem natural se justifica pelo fato de que os chatterbots, além serem facilitadores no processo de interação homem-máquina, são capazes de ser adaptados facilmente a outros ambientes. Um sistema que permita usuários e computadores interagir e conversar, de forma real e natural, usando linguagem natural pode revolucionar a interação homem máquina e o papel que os computadores representam na vida cotidiana (SGANDERLA, 2002).

Além disso, sua utilização na educação tem sido bastante explorada e tem se mostrado extremamente eficaz, tanto em sistemas desenvolvidos para outras áreas de conhecimento (PRIMO; COELHO, 2000; MAGEDANZ; ROTHERMEL; KRAUSE, 1996), como na

área de redes de computadores, através de sistemas desenvolvidos para ensino de redes de computadores em níveis de graduação e pós graduação (LEONHARDT et al., 2003; LEONHARDT; CASTRO; TAROUÇO, 2003; MEDINA; TAROUÇO, 2003; TAROUÇO; DUTRA; LEONHARDT, 2004).

Por esta razão, pode se tornar um recurso bastante interessante para o treinamento de profissionais para atuarem no gerenciamento de redes de computadores. Adams e Wendt (1993) também reforçam a motivação para a construção de um sistema desse tipo quando afirmam que experiências no aprendizado de gerência de redes de computadores em níveis de graduação e pós-graduação em ciência da computação tem servido para ressaltar a necessidade de oferecer uma experiência prática e diferenciada no suporte do material teórico tradicional dos cursos de computação.

4.1.1 Problemas Encontrados no Gerenciamento de Redes

Para realizar a contento as atividades de gerenciamento de redes, os gerentes encontram várias dificuldades, entre elas (MEDINA, 1994):

- A grande diversidade de utilitários disponíveis, que atendem um ou outro aspecto de uma rede;
- Dificuldade de uso destes utilitários. Quanto mais difícil for sua utilização, menor será seu uso;
- Falta de tempo para estudo destes utilitários. Alguns utilitários são tão complexos que exigem leitura de manuais imensos e realização de muitos testes.
- Dificuldade de interpretação dos resultados apresentados. A falta de uma compreensão correta dos resultados praticamente anula o objetivo do utilitário.

Outro problema existente nas redes de computadores, em termos de gerenciamento, é que os problemas encontrados em uma rede podem apresentar o mesmo sintoma, porém, nem sempre apresentam a mesma causa. Além disso, a localização dos problemas exige que se tenha conhecimento e experiência prática na observação e manutenção da rede. Nesses casos, profissionais menos treinados muitas vezes ainda apresentam um certo receio em executar alguma rotina que danifique a rede. Também não se sentem seguros para diagnosticar efetivamente um problema por não saber como coletar todas as informações que se pode extrair de uma rede.

Assim, uma ferramenta que se utiliza do paradigma de linguagem natural para capacitar profissionais permite suprir algumas dessas características já que é capaz de fornecer um referencial teórico para seus usuários, além de fornecer links para tutoriais práticos na internet sobre as ferramentas disponíveis para o gerenciamento e quais ferramentas podem ser utilizadas para verificação de problemas em cada uma das categorias de gerenciamento definidas pela ISO e explicadas em detalhes em (TANENBAUM, 1997) e (STALLINGS, 1996). Também, é capaz de atuar como uma fonte de referência, colocando em prática conceitos que, muitas vezes, se apresentam apenas teoricamente, ou seja, sem a visualização de um exemplo prático por parte do aprendiz.

4.1.2 Linguagem Natural, Interfaces e Chatterbots

Os sistemas computacionais da atualidade ingressaram em um nível onde existe uma grande disponibilidade de hardware, fazendo com que o software seja cada vez mais complexo e especializado. Como consequência de tal crescimento, pode-se observar que os

usuários têm ao seu dispor diversas soluções de sistemas que atendem as suas mais diversas necessidades.

Porém, toda esta tecnologia requer um esforço significativo em se tratando de planejar a forma como o usuário vai interagir com um sistema computacional. O que ocorre, muitas vezes, é que os programas de computadores utilizam formas muitas vezes não intuitivas de comunicação em suas interfaces, como por exemplo, comandos específicos, botões, hyperlinks e, até mesmo, linguagens de programação. Assim, o usuário, para utilizar convenientemente algum sistema, precisa aprender seu método, o que requer tempo e dedicação.

Em sua pesquisa, Bonar e Blaise (1991) afirmam que uma interface deve apresentar o sistema computacional de forma a permitir que o usuário possa pensar exatamente da maneira como está acostumado a pensar. Cada aspecto do sistema deve ser apresentado usando termos e convenções familiares à experiência prévia do usuário quando trabalhava sem a presença de um computador. Para dar um maior escopo a tal afirmação, pode-se tomar como exemplo os programas de computador da atualidade. Suas interfaces geralmente seguem um padrão, que já se tornou conhecido de usuários comuns e técnicos. Com a reutilização deste padrão em novos sistemas computacionais, ou seja, com a utilização dos mesmos comandos utilizados anteriormente, torna-se muito mais fácil adaptar-se ao novo sistema, já que o usuário não mais precisará perder tempo aprendendo novos padrões e comandos. Por conseqüência, ele pode ser então estimulado a explorar as possibilidades que o sistema oferece ao invés de gastar tempo desnecessário aprendendo a utilizar o mesmo.

Embora existam muitas formas de manifestação externa da atividade mental como gestos, formas e posturas, a linguagem natural é onde mais se percebe a expressão do raciocínio de um ser humano. Uma justificativa para isso está no fato de que a linguagem natural apresenta um poder de expressão imenso, já que o próprio pensamento do ser humano é baseado em sua linguagem.

O processamento da linguagem natural é um conjunto de métodos formais usados para analisar textos e gerar frases escritas em um idioma humano, tendo como principal objetivo o de dar subsídios para que os programas de computadores possam ter a capacidade de entender e compor textos. Neste contexto, entender um texto engloba realizar uma análise morfológica e léxica, sintática, semântica e pragmática de seu conteúdo, além de manipulá-lo de forma a extrair informações e interpretar seu significado.

Por análise morfológica e léxica, entende-se o processo de separação de palavras ou expressões elementares de uma língua, através da leitura caracter a caracter de um determinado corpus ou texto. A morfologia trata as palavras quanto a sua estrutura, forma, flexão e classificação, no que se refere a cada um dos tipos de palavras. As palavras podem ser classificadas por seu tipo de uso (substantivo, adjetivo, artigo, numeral, verbo).

A análise sintática abrange o processo de construção de árvores de derivação para cada sentença, mostrando, assim, como as palavras estão relacionadas entre si. A língua portuguesa é muito extensa em se tratando de regras de formação e exceções. Não é intenção deste estudo detalhar as regras gramaticais da língua aos leitores, porém apenas dar uma noção superficial dos tipos de estruturas importantes que devem ser consideradas durante um processo de análise sintática. A análise sintática de uma frase, em língua portuguesa, deve levar em conta os elementos lingüísticos como: termos essenciais (sujeito e predicado), termos integrantes (complementos verbal e nominal) e termos acessórios (adjunto adverbial, adjunto adnominal e aposto). Deve ser também levada em consideração a análise de períodos como: tipo de período (simples ou composto), sua composição

(por subordinação, por coordenação) e a classificação das orações (absoluta, principal, coordenada ou subordinada).

A análise semântica envolve a verificação do sentido das estruturas das palavras que foram reagrupadas pelo analisador sintático, uma vez que o analisador morfológico permitiu identificar estas palavras individualmente. Assim, esta parte do processo é responsável pela representação do significado e do sentido da mensagem.

Por último, a análise pragmática encarrega-se de realizar as devidas associações das informações obtidas das análises anteriores sobre a frase com os resultados das análises de frases anteriores, já que em um diálogo, nem todas as informações relevantes estarão sempre em uma só sentença. Toda esta análise serve para fixar a linha de desenvolvimento de um diálogo e assim contextualizar a mensagem.

Uma aplicação de processamento de linguagem natural pode ser dividida em dois grupos: aplicações baseadas em texto (dentre os quais pode-se citar os tradutores de texto, modelos de representação de conhecimento, extração e recuperação de informações, sistemas de resumo) e aplicações baseadas em diálogo (interfaces para consulta em linguagem natural para bancos de dados e agentes conversacionais como os próprios *chatterbots*).

Por ser tão importante, a linguagem natural passa a ser, hoje em dia, foco de interesse da maioria das aplicações que envolvem uma relação com o ser humano. Entretanto, lidar com linguagem natural não é tão simples, aceitando que existem muitas formas diferentes de expressar uma mesma idéia. Além disso, uma aplicação que utilize processamento de linguagem natural torna-se muito complexa, uma vez que envolve muitas áreas de estudo, entre elas inteligência artificial, compiladores e *lingüística computacional*.

Dentro deste escopo, torna-se muito difícil a utilização do processamento de linguagem natural propriamente dita nos *chatterbots*, apesar de poder ser implementada e, até mesmo, ser realmente utilizada em alguns casos. A linguagem natural ainda sofre de uma falta de cobertura *lingüística* e conceitual. Muitas sentenças em linguagem natural são ambíguas, e os codificadores/parsers tendem a encontrar mais erros e ambigüidades do que as pessoas que porventura utilizarem esta linguagem somente na interface.

A lacuna ainda existente no processamento de linguagem natural propriamente dita pode gerar opiniões como a de Souza et. al. (1999) quando diz que uma aplicação que oferece interface em linguagem natural precisa lidar com construções vagas, ambíguas, e até gramaticalmente incorretas. Ainda não é possível desenvolver sistemas que compreendam qualquer expressão em linguagem natural, mas diversos tipos de sistemas especialistas utilizam com sucesso algum subconjunto de uma linguagem natural, nos quais o usuário deve se expressar de forma inequívoca e tendo em vista as frases que tais sistemas possam interpretar. Ainda no aspecto da linguagem natural, Saygin e Ciceklib (2002) mostram que a pragmática constitui um sério desafio para os pesquisadores de inteligência artificial. Desenvolver um programa de computador que saiba como ser relevante, como prover a quantidade correta de informação em um contexto, como fazer piadas apropriadas, como usar metáforas, alusões, figuras de linguagem, como se comportar em uma determinada situação e em uma situação mais geral e como cooperar em uma conversa não é uma tarefa fácil.

Mesmo que atualmente a capacidade de entender ou compor textos seja bastante limitada no computador, muitos resultados práticos já são possíveis e utilizados por diversos tipos de programas. Conforme Lima (1996), a linguagem concentra inúmeros desafios a serem abordados do ponto de vista da pesquisa, tanto através da ciência da computação como através da *lingüística*. Ela ainda afirma que, do ponto de vista da ciência da computação, faz-se necessário desenvolver algoritmos adequados aos processos de interpretação

e geração da linguagem, juntamente com estruturas de dados voltadas a uma representação eficiente dos modelos e das interações que ocorrem na estrutura formal, e também nas situações de uso da língua. A modelagem e a representação dos fenômenos lingüísticos, por sua vez, ainda representam desafio para a própria lingüística, de modo que o cientista da computação, para resolvê-lo, freqüentemente procura ir além da abordagem simbólica, fazendo uso de abordagens estatísticas que desvendem o comportamento 'prático' da língua, em situações de uso.

Um recurso que se utiliza, muitas vezes, nos casos em que o processamento da linguagem não é feito, é o de simular o processamento da linguagem natural através de comparação de padrões para se dar o retorno ao usuário. Isto porque a linguagem natural é expressiva o suficiente para a descrição do raciocínio artificial do programa. Apesar disso, a linguagem natural não deixa de estar presente em toda sua concepção já que aos olhos do usuário comum, seu processamento ocorre. Para ele é invisível a existência ou não de processamento detalhado de suas entradas em linguagem natural quando interage com um chatterbot.

A utilização de tal recurso pode ser justificada por Souza et. al. (1999) que afirma que a interação em linguagem natural é bastante atrativa para usuários com pouco ou nenhum conhecimento em computação. Além disso, ainda menciona que embora a interação em linguagem natural não se aplica a todos os tipos de sistemas, ambientes de consulta a informações e sistemas baseados em conhecimento são exemplos onde a utilização de interfaces em linguagem natural pode ser bastante benéfica.

Dentre as grandes contribuições do uso de linguagem natural em interfaces, destacam-se: a tornar a possibilidade de se tornar a interação mais clara e eficiente, permitindo aproximar o usuário de seu objetivo e expectativas, bem como apresentando a informação de forma mais eficiente e tornando o uso da máquina mais prazeroso. Uma boa interface, que se utiliza de linguagem natural, pode ajudar usuários a atingirem os objetivos dentro de um ambiente, entender o ambiente e a maneira como utilizá-lo e aumentar significativamente a compreensão e capacidades do usuário.

Mesmo que as limitações da linguagem natural algumas vezes possam gerar sensação de frustração nos usuários, o seu uso não se anula. Muitos pesquisadores como (SAYGIN; CICEKLIB, 2002) e (ANGELI; JOHNSON; COVENTRY, 2001) realizam estudos neste campo, procurando tornar mais qualificado o uso de tal recurso em interfaces para os programas do futuro. Os programas de conversação de hoje estão longe de ser linguisticamente competentes. Porém, não precisamos resolver todos os mistérios da lingüística prgmática antes de começar a trabalhar em melhores formas de comunicação (SAYGIN; CICEKLIB, 2002).

As interfaces entre as máquinas e seres humanos estão ficando mais complexas e planejadas, caminhando aos poucos em direção às formas mais humanas de comunicação. Com a automação computadorizada de diversos aspectos da vida cotidiana se tornando cada vez mais comum, a interação com computadores é cada vez mais necessária. À medida que os computadores se tornam mais velozes, mais poder de processamento é atribuído às interfaces. As interfaces do futuro possivelmente usarão reconhecimento de gestos e de fala, agentes inteligentes, adaptatividade, vídeo e muitas outras tecnologias que estão sendo investigadas atualmente por laboratórios corporativos e grupos de pesquisas em universidades (REDDY, 1996).

4.2 Visão Geral do Sistema: Conhecimento

Fischler e McCorduck (1987) definem conhecimento como a informação armazenada, ou os modelos usados pela pessoa ou máquina para interpretar, prever e responder apropriadamente ao mundo exterior. A definição apresentada pelos autores atenta para o fato de que a aquisição e o gerenciamento da base de conhecimento de um chatterbot são de importância central, uma vez que seu tamanho e sua qualidade têm-se mostrado fundamentais para a obtenção de bons níveis de coerência em uma conversação. Isso pode ser comprovado ao se verificar que chatterbots com técnicas simples para a identificação das sentenças dos usuários, mas que possuem bases de conhecimento robustas, são capazes de obter resultados comparáveis àqueles que utilizam técnicas mais complexas (GALVÃO, 2003).

4.2.1 Aquisição do Conhecimento

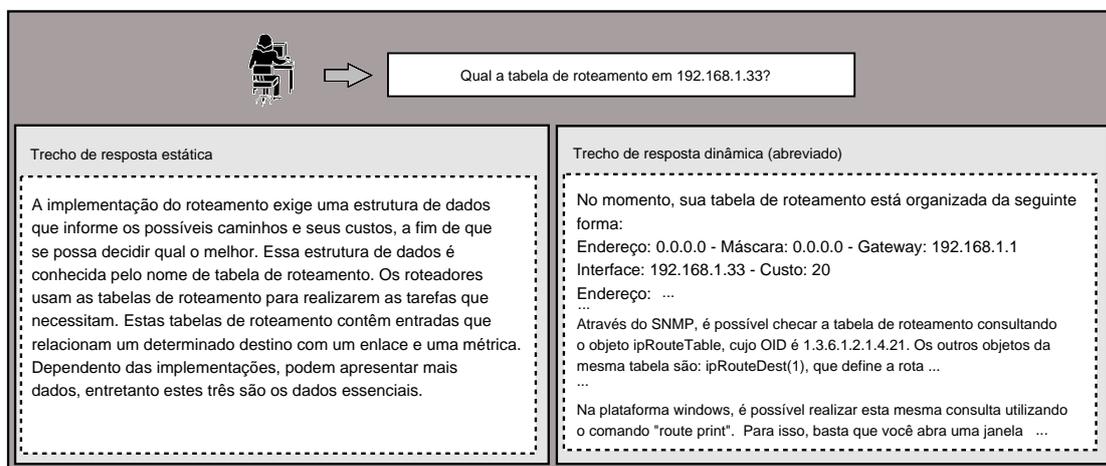


Figura 4.1: Diferenciação da parte estática e da parte dinâmica em uma resposta

O chatterbot desenvolvido neste trabalho é capaz de gerar respostas estáticas e dinâmicas (Figura 4.1). Embora essa diferença esteja sendo mencionada, as respostas são visualmente representadas exatamente da mesma maneira para o usuário. A diferença básica entre os dois tipos de resposta é que as respostas dinâmicas contêm informações reais sobre a rede gerenciada enquanto que as estáticas abrangem uma explicação conceitual de um tópico, adicionada, algumas vezes, de um exemplo hipotético direcionado para o aprendizado e a assimilação do conceito apresentado.

Para tanto, o processo de aquisição do conhecimento para criação da base de conhecimentos de Doroty foi realizado através de várias etapas. A primeira etapa envolveu uma análise bibliográfica, a fim de se construir o conteúdo teórico das respostas do chatterbot, ou seja, conforme citado anteriormente, as respostas ditas estáticas. As referências utilizadas se constituíram de livros teóricos na área de redes de computadores (SOARES, 1995; STALLINGS, 1996; TANENBAUM, 1997). Uma complementação ao conteúdo teórico envolveu uma consulta bibliográfica a trabalhos desenvolvidos na área e materiais encontrados na Internet (LIMA, 1997).

Em se tratando das respostas dinâmicas, ou seja, aquelas que envolvem a consulta a elementos da rede, foi utilizada como referência a documentação disponível sobre a MIB2, tanto em livros (STALLINGS, 1996), como em RFCs (*Request for Comments*)

que definem a mesma (MCCLOGHRIE; ROSE, 1991).

A caracterização do conhecimento de Doroty foi ampliada através da consulta a referências sobre trabalhos desenvolvidos na área, enfocando o gerenciamento de redes e alguns problemas que ocorrem durante esse processo, como o caso do sistema DUMBO (MELCHORS, 1999), já abordado no capítulo 3. Outros materiais consultados envolveram (LAMBERT, 1995).

Numa última etapa, especialistas foram convidados a avaliar as respostas teóricas e práticas do chatterbot para validar todo o processo de aquisição de conhecimento e sugerir melhorias nos conceitos apresentados.

4.2.2 Conhecimento Prático

Esta seção tem por objetivo apresentar a MIB2, que é utilizada por Doroty para realizar a interface com a rede gerenciada. Também será apresentada uma breve definição da abordagem de Aprendizado Baseado em Problemas e serão mostrados alguns problemas que se encontram especificados na base de conhecimento de Doroty. Os problemas apresentados foram retirados das referências apresentadas na seção 4.2.1 e também relatados por alguns dos especialistas consultados.

4.2.2.1 MIB2

A MIB2 é composta de objetos gerenciados, que são organizados nos grupos system, interfaces, at, ip, icmp, tcp, udp, egp, dot3 e snmp, conforme ilustra a figura 4.2.

O grupo System permite que um gerente descubra o nome de um dispositivo, quem o fabricou, o que o hardware e o software contêm, onde o dispositivo está localizado e o que ele deverá fazer. O horário da última inicialização e o nome e endereço da pessoa de contato também são fornecidos. Enfim, esse grupo fornece informações gerais sobre o dispositivo gerenciado.

O grupo Interfaces lida com os adaptadores da rede. Ele controla o número de pacotes e bytes enviados e recebidos da rede, o número de descartes, o número de difusões e o tamanho da fila de saída. Enfim, esse grupo fornece informações sobre as interfaces de rede de um dispositivo e seu tráfego.

O grupo IP trata do tráfego IP recebido e emitido por um nó. Ele é especialmente rico em contadores que controlam o número de pacotes descartados por diversas razões (por exemplo, nenhuma rota conhecida ou falta de recursos). Estão disponíveis estatísticas sobre a fragmentação e a remontagem de dados.

O grupo ICMP se refere a mensagens de erro IP. Basicamente, ele dispõe de um contador que registra quantas mensagens de erro ICMP de um determinado tipo foram encontradas.

O grupo TCP monitora o número atual e o cumulativo de conexões abertas, segmentos enviados e recebidos e diversas estatísticas de erros. Enfim, esse grupo fornece informações sobre algoritmos TCP, parâmetros e estatísticas.

Os objetos que constituem o grupo UDP servem para registrar o número de datagramas UDP enviados e recebidos e quantos dos enviados não foram entregues devido a uma porta desconhecida ou a algum outro erro, ou seja, esse grupo fornece informações sobre as estatísticas de tráfego UDP.

O grupo EGP é usado para roteadores compatíveis com o protocolo de gateway externo. Ele controla quantos pacotes de um determinado tipo foram enviados, quantos foram recebidos e encaminhados corretamente e quantos foram recebidos e descartados. Dessa forma, esse grupo é responsável por fornecer estatísticas de tráfego de protocolo de

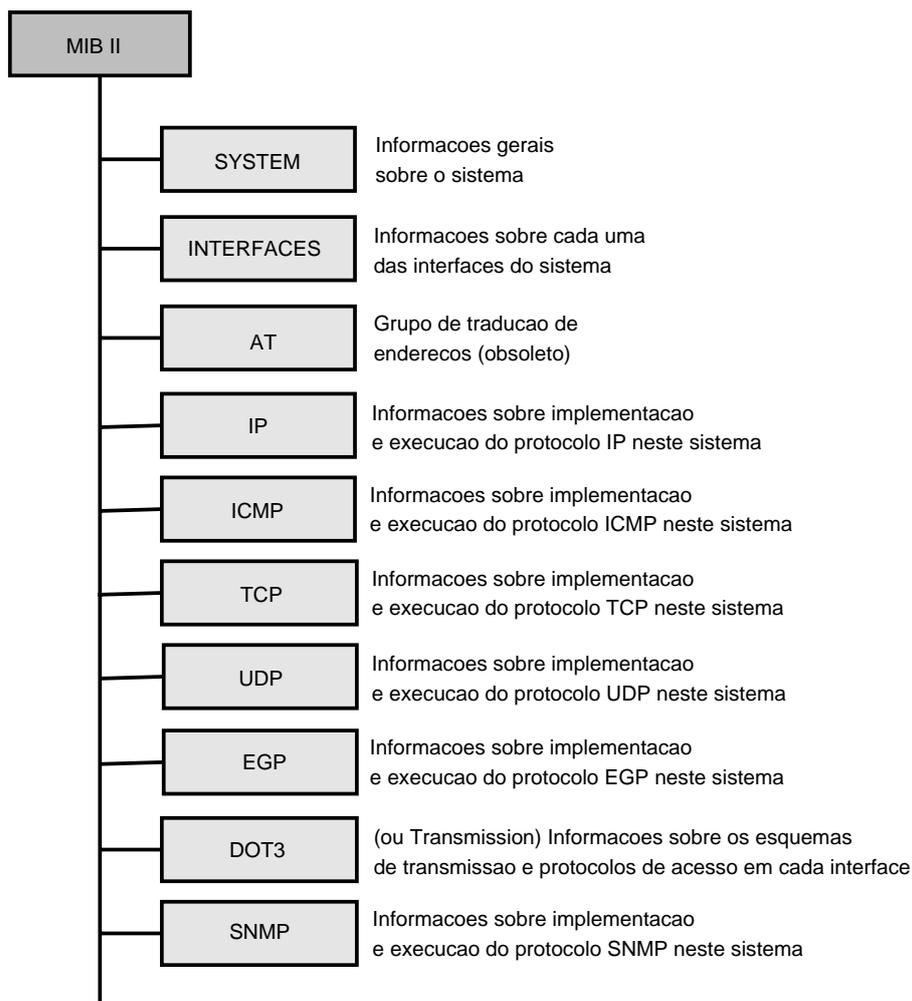


Figura 4.2: Organização da MIB II

gateway externo.

Seguindo a estrutura, o grupo Transmission é um marcador de lugar para MIBs de meios físicos específicos. Por exemplo, nesse grupo é possível manter estatísticas especificamente relacionadas à Ethernet. O objetivo da inclusão de um grupo vazio na MIB2 é reservar o identificador internet 2 1 9 para esses objetivos.

O grupo SNMP se destina ao cálculo de estatísticas sobre a operação do próprio SNMP. Quantas mensagens estão sendo enviadas, quais os tipos dessas mensagens, entre outros. Esse grupo fornece estatísticas de tráfego SNMP na rede.

Finalmente, o grupo AT tem como objetivo realizar uma tradução de endereços entre redes. Todos os objetos que compõem tais grupos podem ser vistos em detalhes em (STALLINGS, 1996) e (MCCLOGHRIE; ROSE, 1991).

4.2.2.2 Problemas

Para auxiliar no treinamento e aprendizado dos profissionais menos experientes que atuam no gerenciamento de redes, Doroty também se baseia na abordagem de Aprendizado Baseado em Problemas (ou PBL - *Problem-Based Learning*). O Aprendizado Baseado em Problemas é uma estratégia instrucional mediante a qual os alunos são expostos a problemas para achar soluções, promovendo o aprendizado.

Nessa abordagem, os problemas são apresentados aos alunos de forma mal estruturada

e em um contexto complexo, no qual eles ainda não possuem todo o conhecimento do assunto, devendo, para resolvê-los, pesquisar e buscar em diversas fontes o conteúdo de uma forma autodirecionada. Isso leva o aluno a desenvolver simultaneamente estratégias e conhecimentos para resolução de problemas e habilidades para a investigação e obtenção de novos conhecimentos (JONES, 1996). Assim, espera-se do aluno um papel mais ativo, assumindo a responsabilidade sobre o seu aprendizado e decidindo o que precisa aprender e como irá aprender.

Da mesma forma, um sistema que se baseia em tal paradigma deve tentar reaplicar o papel de um professor no modelo de aprendizagem, ou seja, no caso do PBL, o professor não deve se comportar como um agente centralizador no processo de aprendizado, mas sim, atuar como um facilitador, direcionando o aluno, sem interferir diretamente na aquisição do conhecimento. Assim, ele deve desenvolver o problema e aplicar em sala de aula, atuando como um tutor para o aprendiz durante a resolução. Por último, deve avaliar a performance do mesmo quando da resolução de tal problema.

Embora Doroty não utilize fielmente tal paradigma, algumas características do mesmo podem ser observadas em sua implementação:

- Os alunos tem a responsabilidade pelo seu próprio aprendizado. Os alunos têm que ser capazes de identificar o que eles precisam aprender e que recursos eles irão utilizar para isso. Doroty atua apenas como um facilitador neste processo, a medida que deixa o aluno livre para testar as hipóteses sobre os problemas levantados, apresentando conceitos teóricos importantes e demonstrando a forma como consulta os dados na rede pesquisada;
- Os problemas apresentados são pouco estruturados e permitem a livre investigação. Mesmo que essa investigação possa ser feita através da utilização de Doroty para interface com a rede, a atuação do aluno é imprescindível para testar, dentre as causas e soluções apontadas, quais as que melhor caracterizam o problema;
- A colaboração com Doroty permite ao aluno desenvolver a segurança necessária para se tornar responsável pelo seu próprio aprendizado;
- O aluno tem a possibilidade de receber todas as informações teóricas necessárias em um contexto, mas deve relacioná-las de forma que mais lhe parecer satisfatória;
- Por realizar interface com uma rede real, os exemplos demonstrados em Doroty fazem com que os alunos utilizem, no aprendizado, as mesmas consultas e ferramentas que os profissionais e especialistas utilizam em seu trabalho para resolver problemas reais.

Exemplos de problemas que podem ser verificados utilizando objetos da MIB2 são exemplificados na tabela 4.1, que cita alguns objetos que podem ser consultados para verificar problemas de segurança:

A tabela 4.2 apresenta um exemplo mais detalhado de problema, retirado do trabalho de Melchior (1999). Realizando consultas na rede, um usuário poderia verificar e estranhar a alta taxa de erros de saída em uma interface, por exemplo. Assim, ele poderia consultar Doroty e verificar que um possível problema seria o meio físico saturado.

Embora, em sua maioria, os problemas não possam ser diagnosticados apenas com a utilização isolada da MIB2, a utilização de Doroty pode direcionar o usuário a investigar outras causas através de outras fontes, descartando algumas hipóteses e formulando outras à medida que seu conhecimento sobre a rede aumenta.

Tabela 4.1: Exemplos de objetos da MIB2 que podem ser consultados para verificação de problemas de segurança em uma rede

Objeto	Descrição
tcpConnTable	Mostra as conexões tcp do host. Pode ser útil para descobrir conexões de programas não-autorizados (ex.: backdoors).
udpTable	Mostra as conexões tcp do host. Pode ser útil para descobrir conexões de programas não-autorizados (ex.: backdoors).
udpNoPorts	Contabiliza os datagramas UDP recebidos que tinham como destino uma porta sem aplicação associada. Útil para tentar varreduras de portas.
icmpInMsgs, icmpOutMsgs	Indica o número de mensagens ICMP recebidas e enviadas. Um atacante pode usar mensagens ICMP para realizar varredura, obter informações ou realizar um ataque de DoS.

Tabela 4.2: Características de um problema de interface de rede

Causa	Sintoma
Meio Saturado	alta taxa de erros de saída na interface, alta taxa de erros de entrada na interface, alta taxa de colisões (ethernet)

4.3 Arquitetura do Protótipo

A arquitetura de Doroty é apresentada na figura 4.3 e consiste dos seguintes elementos: a interface do chatterbot e sua base de conhecimento, o banco de dados de informações, banco de dados histórico, o módulo coletor, módulo de atualização do banco de dados histórico e o módulo central, que armazena as regras de funcionamento do bot.

O processo básico de funcionamento do chatterbot engloba algumas etapas. Um usuário interage com o chatterbot Doroty através de sua interface. A interface é constituída de um formulário para entrada da pergunta e uma área destinada para as respostas. Assim, o usuário faz uma pergunta ao chatterbot simplesmente digitando a pergunta em linguagem natural no formulário. O chatterbot processa essa entrada e procura a informação corres-

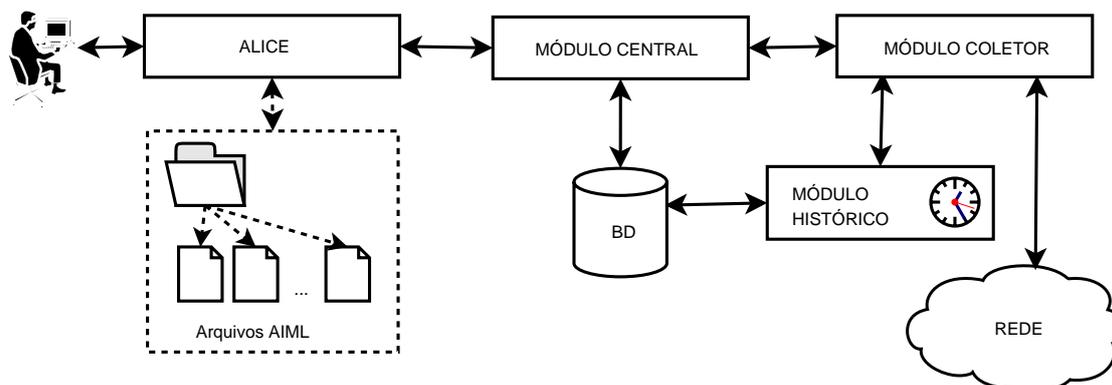


Figura 4.3: Arquitetura do chatterbot Doroty

pondente na sua base de conhecimento estática. Se a resposta não necessita explicação detalhada, o bot apenas fornece uma resposta teórica sobre o assunto. Por outro lado, nos casos em que uma explicação detalhada ou uma consulta nas condições da rede se fizer necessária, o bot ativa o módulo central e fornece, através de suas regras internas e informações em seu banco de histórico de dados, uma resposta dinâmica.

4.4 Implementação e Funcionamento

A figura 4.4 apresenta um diagrama mais detalhado da arquitetura de Doroty, demonstrando como o sistema se organiza e interage.

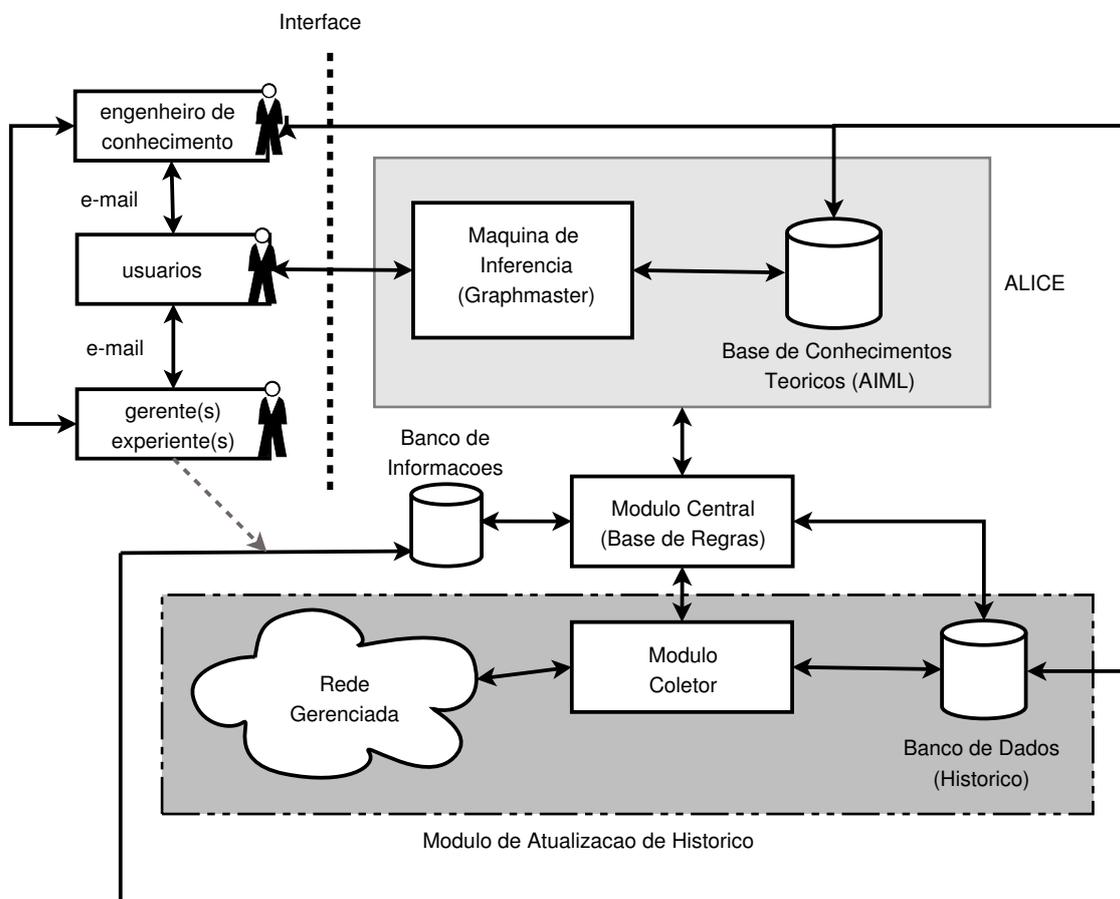


Figura 4.4: Diagrama detalhado da arquitetura de Doroty (Interação)

Neste diagrama pode-se notar a presença do engenheiro de conhecimento e do gerente mais experiente, que, neste caso, representa o papel do especialista de domínio. Quando uma resposta não satisfaz o usuário, o chatterbot então sugere que o mesmo envie um e-mail para o engenheiro de conhecimento responsável pelo conteúdo do sistema ou ainda para o especialista que, por sua vez, pode se comunicar diretamente com o engenheiro de conhecimento a fim de incluir tal conhecimento na base.

A seguir, será detalhada a implementação da arquitetura apresentada anteriormente, bem como detalhes de funcionamento interno e comportamento do chatterbot.

4.4.1 Tecnologias Utilizadas na Implementação

A escolha da dos componentes e plataforma para desenvolvimento e implementação do de Doroty foi baseada em alguns aspectos:

- Utilização de maior porcentagem possível de software livre, de forma a tentar produzir um software com maior quantidade possível de tecnologias não proprietárias;
- Execução da menor quantidade de mudanças possível em tecnologias aproveitadas como base, de forma a não comprometer a performance das mesmas;
- Utilização, sempre que possível, de tecnologias semelhantes às usadas como base, a fim de tornar possíveis trabalhos futuros sem necessidade de domínio de tecnologias desnecessárias.

Com base em tais aspectos, a linguagem de programação escolhida foi Java, por ser a mesma linguagem utilizada no desenvolvimento do chatterbot ALICE, que foi usado como base para desenvolvimento do trabalho. Além disso, essa tecnologia apresenta algumas características interessantes como:

- Portabilidade, robustez e independência sobre o sistema operacional.
- Possibilidade de adaptação de funcionalidades oferecidas em classes já existentes às necessidades da aplicação.

A interface com o usuário foi desenvolvida através de páginas HTML simples, que permitem o acesso ao o chatterbot através de navegadores gráficos comuns. Sua disponibilização, em cenário de testes, foi feita através da utilização do sistema Apache, que disponibiliza uma forma de tornar acessível o sistema desenvolvido. Porém, dependendo do ambiente em que é instalado, Doroty pode utilizar outras tecnologias para este fim.

Os dados que constituem o banco de dados de Informações e Histórico são armazenados em tabelas no sistema de banco de dados MySQL. Este sistema possibilita uma forma simples de conexão com bases de dados utilizando código Java. Para isso, deve-se utilizar um driver JDBC (*Java Database Connectivity*), que é uma API (*Application programming Interface*) que possui um conjunto de instruções específicas da linguagem Java traduzidas para a linguagem SQL (*Structured Query Language*) padrão, que permite o acesso a bases de dados externas.

4.4.2 Interface e Interação com o chatterbot

Por ser desenvolvida para visualização através de um navegador Internet, Doroty possui, ao todo, seis diferentes páginas. Ao ser acessada, sua página principal apresenta o componente de diálogo, ou seja, o formulário que permite que uma pergunta seja digitada e o campo para que o chatterbot escreva sua resposta. Além de tal componente, a interface possui uma área para apresentação das últimas notícias envolvendo o chatterbot. Essa área se destina a fornecer informações sobre a base de conhecimento do chatterbot, à medida que a mesma se expande, bem como apresentar uma análise estatística dos assuntos mais procurados e das dúvidas mais frequentes. A figura 4.5 apresenta a tela inicial do chatterbot Doroty.

As perguntas devem ser digitadas no campo Pergunta. O botão "**Limpar**" deve ser utilizado quando se deseja deixar o campo pergunta em branco, mesmo durante o processamento da resposta por parte do chatterbot. O botão "**Enviar**" envia uma pergunta para



Figura 4.5: Tela inicial de Doroty: componente de diálogo

que o chatterbot processe. Para enviar uma pergunta, a tecla **"Enter"** também pode ser utilizada.

As páginas auxiliares compreendem informações sobre as publicações sobre Doroty, informações gerais sobre o trabalho (Figura 4.6), agradecimentos, análise de dados de conversações semanais e, por fim, as últimas notícias são apresentadas de forma mais detalhada.

Além disso, para fim de testes, foi utilizada uma interface mais simples, composta apenas do componente de diálogo, de forma a concentrar o usuário na interação em linguagem natural e de explorar todos os recursos possíveis nas respostas fornecidas por Doroty (Figura 4.7).

4.4.3 Base de Conhecimento e Máquina de Inferência

Conforme mencionado no início deste capítulo, o processamento de linguagem natural em todos os seus níveis não é comum nos chatterbots. Assim, para realizar o tratamento das mensagens, optou-se por utilizar uma alternativa paralela ao processamento de linguagem natural: o tratamento da linguagem natural com a utilização de arquivos de



Figura 4.6: Tela de apresentação de informações gerais sobre Doroty

definição em XML. Esta técnica já vem sendo utilizada com sucesso no projeto ALICE, apresentado no capítulo 2, que desenvolveu sua própria linguagem de marcação, o AIML, com todas as definições na qual se baseiam as mensagens trocadas com o chatterbot. O sistema Doroty, por ser originado pelo sistema ALICE, utiliza a mesma linguagem para armazenar o seu conhecimento e a própria máquina de inferência ALICE para o processo de tratamento de mensagens.

4.4.3.1 A Linguagem AIML

O AIML é uma linguagem de fácil aprendizagem e utilização (WALLACE, 2003). Ela apresenta um conjunto de tags e comandos simples para implementação da base de conhecimento de um chatterbot e serve para analisar as mensagens enviadas pelo usuário e decidir a forma como estas mensagens devem ser respondidas.

O AIML é baseado em padrões de entrada do usuário, conhecidos como categorias. Uma frase escrita por um usuário é comparada aos padrões descritos na linguagem e com base neste processo são selecionadas ou construídas as respostas.

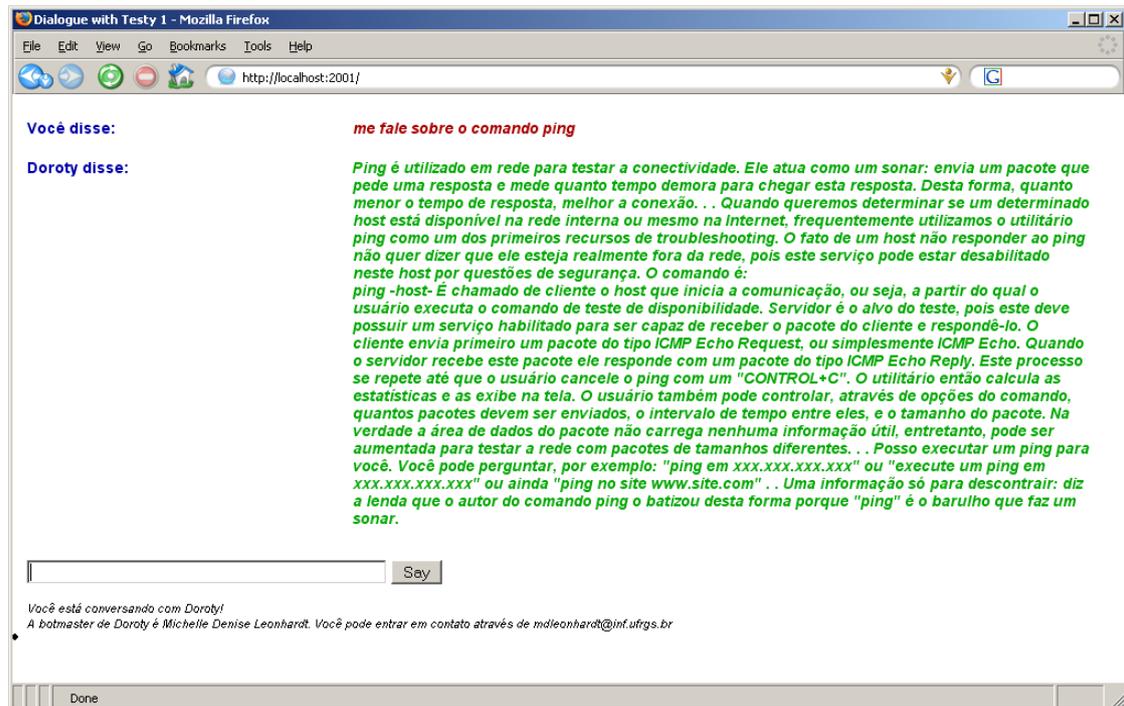


Figura 4.7: Tela simples de interação com Doroty

As principais tags do AIML são:

<aiml> inicia e termina um bloco programado em AIML

<category> identifica uma "unidade de conhecimento" na base de conhecimento

<pattern> identifica um padrão de mensagem simples

<template> contém a resposta para uma mensagem do usuário

Uma categoria em AIML é definida da seguinte forma:

```
<category>
<pattern> entrada </pattern>
<template> resposta </template>
</category>
```

Por exemplo:

```
<category>
<pattern> BOM DIA * </pattern>
<template> LINDO DIA NAO? </template>
</category>
```

Neste exemplo, o padrão de entrada "BOM DIA" seguido de qualquer informação, terá como resposta ou possível resposta a sentença "LINDO DIA NAO?". O caractere * representa um conjunto de caracteres.

Pode-se demonstrar melhor a sintaxe da linguagem AIML através da transcrição do exemplo exposto no trabalho de Russel (2002), onde um diálogo é mostrado, bem como

a codificação equivalente em AIML que poderia ser usada para implementar tal diálogo:

Usuário: *"knock knock."*

(o usuário inicia interação ao som de batidas)

Chatterbot: *"Who's there?"*

(*Quem está aí?*). Neste trecho o tópico da mensagem seria setado para "piada", já que, em inglês, muitas piadas são iniciadas com sons de batidas.

Usuário: *"ivana."*

(*ivana*).

Chatterbot: *"Ivana who?"*

(*Ivana quem?*)

Usuário: *"ivana half a billion dollars."*

(*Ivana meio bilhão de dólares*).

Chatterbot: *"that's not very funny."*

(*isso não é muito engraçado*)

Usuário: *"Why don't you tell me a joke."*

(*Porque você não me conta uma piada.*)

Chatterbot: *"do you want to hear a joke?"*

(*você quer ouvir uma piada?*). Nesse caso o sistema dispara uma resposta aleatória para o tópico piada.

A conversação acima poderia ser implementada utilizando um pequeno trecho de AIML:

```
<category>
<pattern> KNOCK KNOCK </pattern>
<template>
  Who's there?
  <think>
    <settopic> joke </settopic>
  </think>
</template>
</category>

<topic name = "joke">
<category>
<pattern> * </pattern>
<that> WHO IS THERE </that>
<template> <star/> who? </template>
</category>
```

```

<category>
<pattern> * </pattern>
<that> * WHO </that>
<template> That's not very funny </template>
</category>

<category>
<pattern> * </pattern>
<template> Do you want to hear a joke? </template>
</category>

</topic>

```

No código acima, o primeiro trecho serve para setar o tópico corrente da conversa conforme a entrada do usuário. A tag `<think>` serve para um processamento interno, invisível aos olhos do usuário. Neste diálogo, quando o tópico é setado para piada, as possíveis respostas são enumeradas em categorias dentro das possíveis interações em torno do assunto piada.

Este pequeno exemplo serve para mostrar a simplicidade do AIML e seu potencial para dar um surpreendente retorno ao usuário, mesmo com a utilização de somente recursos básicos. Com a utilização de AIML, pode-se definir mais de uma resposta para um único padrão e ainda se podem especificar critérios de escolha de cada uma das respostas. Existem, ainda, mais de 20 outras *tags* que compõem a linguagem AIML responsáveis por fornecer a necessária desenvoltura para o chatterbot propor uma solução à mensagem enviada. Algumas dessas *tags*, que foram utilizadas na criação da base de conhecimentos de Doroty, podem ser vistas em detalhes no anexo 2 desta dissertação.

A AIML também é muito extensível e customizável. Tecnologias como Javascript e comandos UNIX podem ser incorporados ao AIML, fazendo com que uma variedade de recursos externos possa ser explorada.

4.4.3.2 *Máquina de Inferência ALICE: o algoritmo Graphmaster*

Quando um usuário digita uma pergunta, ALICE aplica um processo de normalização na sentença escrita pelo usuário. Este processo é composto de três etapas distintas: substituição, separação de sentenças e ajuste de padrão. O primeiro passo consiste em substituir algumas palavras ou caracteres de forma que seja possível reter informações que podem ser posteriormente perdidas em outras etapas da normalização. Esse processo envolve a substituição de abreviaturas, caracteres como "ponto" e *emoticons* presentes na sentença. O segundo passo divide a sentença de entrada em tantas outras quanto forem necessárias, de acordo com a pontuação. O terceiro passo, de ajuste de padrão, remove a pontuação das sentenças e converte todas as letras para o maiúsculo. Quando o processo completo é aplicado, sentenças de entrada como ":-(Eu não sei! Como posso ver a tabela de roteamento da minha máquina?" ou "Eu poderia salvar essa resposta em um arquivo .txt?" são transformadas em "EU NAO SEI", "COMO POSSO VER A TABELA DE ROTEAMENTO DA MINHA MAQUINA" e "EU PODERIA SALVAR ESSA RESPOSTA EM UM ARQUIVO PONTO TXT".

Ao interpretar o AIML, ALICE procura fazer o casamento de padrões de forma perfeita, buscando palavra por palavra ao invés de categoria por categoria. Para isso, utiliza o algoritmo Graphmaster (WALLACE, 2001). A estrutura do Graphmaster é semelhante

à de um grafo e consiste de uma coleção de nós chamados Nodemappers. Os filhos são, por sua vez, palavras ou os caracteres especiais "_" e "*" (primeira palavra de todos os padrões ou caracteres especiais). Um exemplo do funcionamento do algoritmo é mostrado a seguir. Suponha que a sentença entrada pelo usuário comece com a palavra **"REDE"**. O nodo pai da estrutura do Graphmaster é um arquivo do sistema que contém todos os arquivos AIML. Quando acionado, o algoritmo ocorre da seguinte forma e, quando um casamento é feito, o processo termina. A figura 4.8 ilustra o funcionamento do Grapmaster baseado em algumas categorias.

- Se o Nodemapper contém o caracter especial "_", procure o subgrafo a partir do nó filho que corresponde a "_". Tente todas as palavras restantes para a sentença após a palavra **"REDE"** para ver se algum casa. Do contrário, volte ao Nodemapper e tente:
- Se o Nodemapper contém a palavra **"REDE"**, procure a raiz do subgrafo a partir do nó filho que corresponde a **"REDE"**, usando o restante da sentença de entrada com a palavra **"REDE"** removida. Do contrário, tente:
- Se o Nodemapper contém o caracter especial "*", procure a raiz do subgrafo a partir do nó filho que corresponde a "*". Tente todas as palavras e caracteres restantes da sentença de entrada após a palavra **"REDE"** para ver se há algum casamento. Se não houver nenhum, volte para o nó anterior e coloque a palavra **"REDE"** novamente na sentença.

No caso de Doroty, a programação dos arquivos AIML explora muitos dos recursos da linguagem. Isso porque o chatterbot deve ser preparado para responder também perguntas dinâmicas, ou seja, valores devem ser passados para o módulo central a fim de se buscar alguma informação do estado atual ou anterior da rede gerenciada. Para que esses valores sejam passados corretamente, foi definido um conjunto de parâmetros, que são atingidos pelo AIML através de comandos de recursividade e substituição de caracteres especiais de entrada. Esses parâmetros são: tópico, endereço, operação, tempo e situação. Os parâmetros são setados de acordo com cada pergunta feita pelo usuário pelo próprio programador do AIML. Depois de setados tais parâmetros, o AIML dispara a execução do módulo central, através do comando `<system >`, que permite a execução de programas externos ao ALICE.

```
<category>
<pattern> Qual a tabela de roteamento em * agora? </pattern>
<template> Tabela de roteamento e ...
  <system>
    java doroty rtable <star index = "1"\> get-table null null
  </system>
</template>
</category>
```

No exemplo acima, imagine que a pergunta digitada pelo usuário seja: "Qual a tabela de roteamento em 143.54.47.240 agora?", o AIML substitui o asterisco pelo endereço digitado pelo usuário (através de `<star index = "1">`) e passa este valor para o módulo

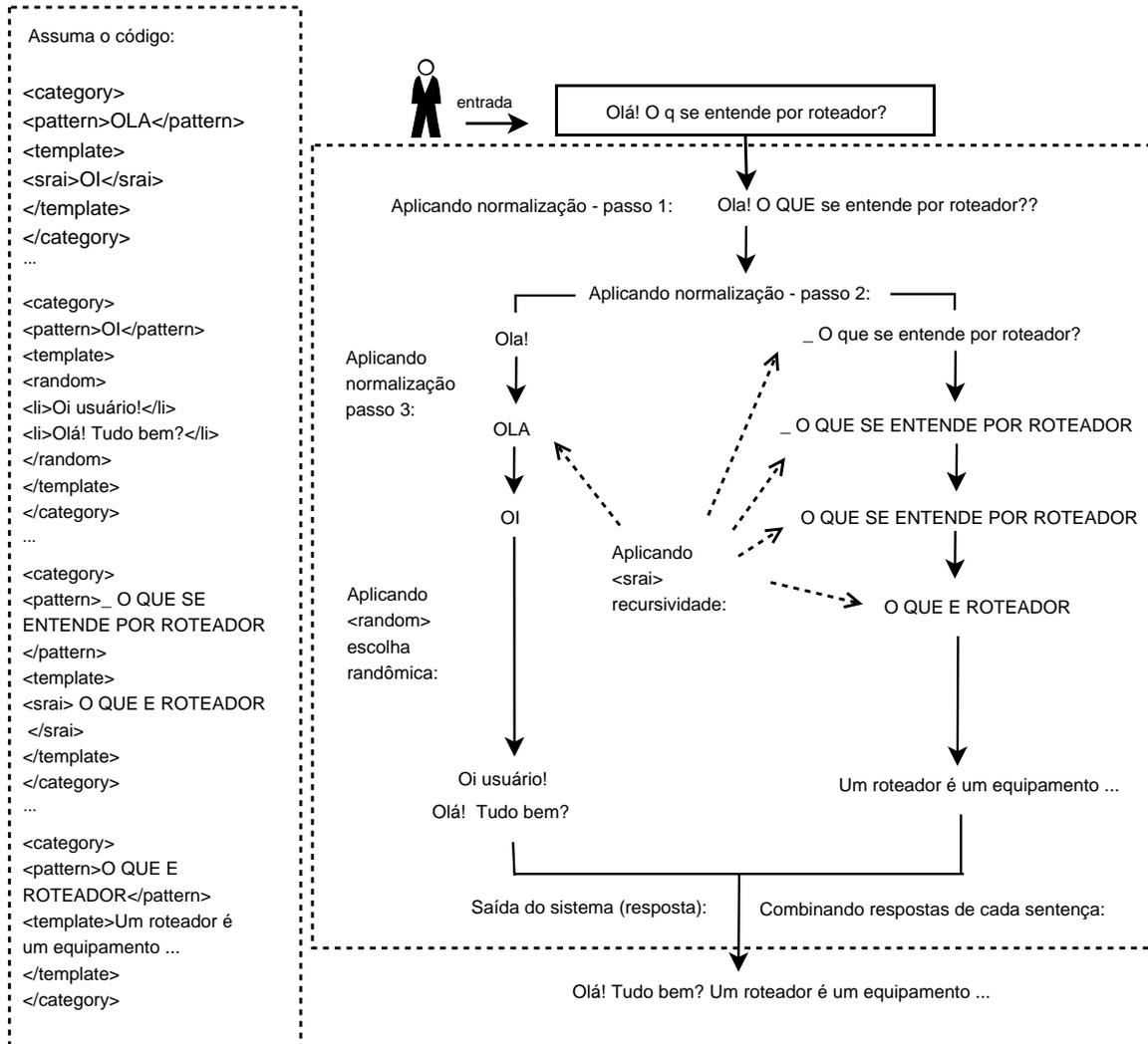


Figura 4.8: Exemplo de execução do Graphmaster

central. Dessa forma, as categorias de conhecimento de Doroty estão diretamente ligadas com o módulo central. Durante a modelagem do sistema, optou-se por essa solução para que a máquina de inferência original não precisasse ser modificada e, conseqüentemente, sua performance não fosse reduzida.

Doroty foi alimentada com mais de 2000 categorias de conhecimento, de forma a tornar possível um diálogo bastante convincente por parte do chatterbot. As categorias de conhecimento em AIML armazenam a parte estática das respostas do sistema e fornecem subsídios para a formação das respostas dinâmicas do mesmo, quando necessário. O trecho de código contido na figura 4.9 demonstra algumas categorias de conhecimento de Doroty, que realizam a passagem de parâmetros conforme mencionado.

```

...
...
<category>
  <pattern>* CONTATO NO EQUIPAMENTO *</pattern>
  <template>
    <think>
      <set name="endereco">
        <system>java ParserEndereco <star index="2"/></system>
      </set>
    </think>
    <srai>SYSCONTACT</srai>
  </template>
</category>

...
...

<category>
  <pattern>SYSCONTACT</pattern>
  <template>
    <think>
      <set name="assunto">
        objetos do grupo system da MIBII
      </set>
      <system>
        java doroty contato <get name="endereco"/> null get null
      </system>
    </think>
  </template>
</category>

...
...

```

Figura 4.9: Trecho AIML com passagem de parâmetros em Doroty

4.4.4 Banco de Informações e Histórico de Dados

O banco de dados de Doroty é constituído por 7 tabelas que permitem identificar uma rede, seus equipamentos e o tipo, os objetos que podem ser consultados e suas características, o histórico de informações da rede e outras informações que permitem associar cada tópico a um ou mais objetos. Ele está estruturado de forma a permitir que sejam cadastradas informações sobre mais de uma rede, além de conter informações detalhadas sobre cada objeto da rede. Dessa forma, é possível programar Doroty para monitorar quantas redes forem necessárias (Tabela 4.3; Figura 4.10).

A tabela histórico serve para armazenar medições da rede que vão sendo feitas em

Tabela 4.3: Tabelas pertencentes ao banco de informações e histórico de Doroty

Nome da Tabela	Descrição
REDE	Armazena informações sobre a(s) rede(s) que Doroty irá gerenciar
EQUIPAMENTOS	Armazena informações sobre os equipamentos e dispositivos presentes em cada uma das redes constantes na tabela REDE
TIPO	Armazena informações sobre tipos de equipamentos
EQUIP_TIPO	Relaciona os equipamentos da tabela EQUIPAMENTOS com os tipos da tabela TIPO
OIDS	Armazena o identificador de cada um dos objetos da MIB2 que Doroty conhece, bem como o intervalo de consulta para os objetos que integram o banco Histórico, as características e informações pertinentes
OID_TIPO	Relaciona um tópico a um ou mais identificadores
HISTORICO	Armazena o histórico de informações de uma rede

determinados intervalos. O intervalo pode variar para cada objeto e é armazenado juntamente com as informações do objeto dentro do banco. A atualização dessa tabela é constantemente feita pelo módulo de Atualização de Histórico.

4.4.5 Módulo Coletor e Módulo de Atualização de Histórico

O módulo de Atualização de Histórico é programado para percorrer, a cada minuto, a tabela de histórico de dados em busca da última atualização do valor de cada objeto. Se a última atualização de um objeto tiver ocorrido em um tempo maior que o intervalo de atualização desejado, que é consultado na tabela OIDs, o módulo coletor é chamado para que aquele objeto seja novamente consultado. Cabe salientar que o valor de cada consulta é então armazenado em um novo registro para que se possam realizar consultas do tipo maior valor, menor valor e média. O mesmo módulo se encarrega de apagar registros antigos a cada intervalo de tempo definido pelo programador (Figura 4.11). O funcionamento de tal módulo pode ser resumido através das seguintes etapas:

Para cada registro na tabela de Histórico (de cada equipamento):

1. Percorre banco para achar última data de atualização: $\max(\text{data})$;
2. Manipula as datas para comparação (ex: última atualização: hoje, 11:15:00, intervalo: 15. Então: data atual(em milisegundos) menos última atualização(em milisegundos) resulta em um valor(int). Se este valor for maior ou igual ao intervalo, então deve atualizar o banco.);
3. Compara data atual com data encontrada. Se data igual, então atualiza (colocando nova data na tabela Histórico);
4. Passa para próximo registro (até que tenha percorrido todos os registros)

O módulo Coletor é o responsável por disparar as consultas necessárias na rede. Ele pode ser acionado, sempre que necessário, pelo módulo de Atualização de Histórico e pelo módulo Central. O módulo Coletor teve seu código e funcionalidade adaptado do pacote

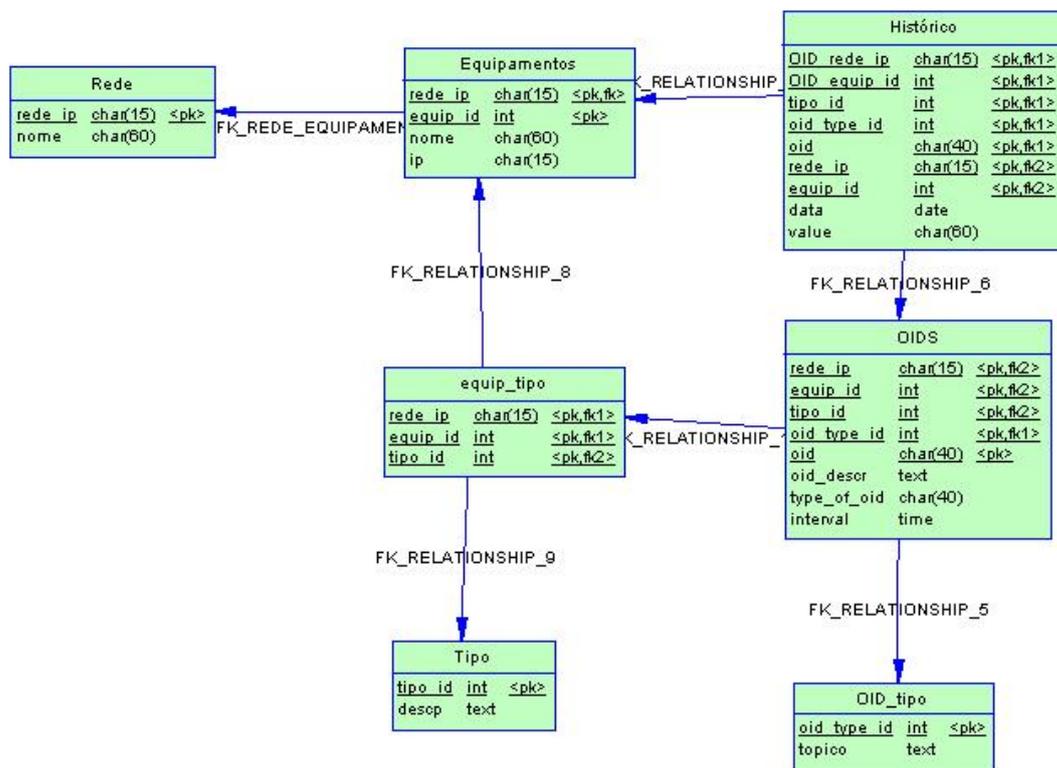


Figura 4.10: Tabelas do Banco de Dados de Doroty

SNMP, criado por Jonathan Sevy em 2002, e disponível em http://edge.mcs.drexel.edu/GICL/people/sevy/snmp/snmp_package.html. Este pacote é livre e de código aberto, permitindo sua fácil adaptação ao modelo desenvolvido. Ele define classes usadas para se realizar a comunicação com dispositivos que suportam SNMP. São definidas classes que correspondem a cada tipo de objeto SNMP, cada uma dessas classes estendendo a classe `SNMPObject`.

Este pacote provê suporte para as operações básicas SNMP conforme definidas na versão 1 e 2 do protocolo. O pacote apresenta um mecanismo para recuperar e setar valores de objetos através de uma simples interface de comunicação e representa as estruturas e tipos de dados presentes no SNMP como objetos java correspondentes. O pacote é familiar ao SNMPv1 e v2c, conforme especificado em suas respectivas RFCs.

Assim, cada classe derivada de `SNMPObject` define métodos para criar um objeto utilizando uma representação de acordo (ex: uma string como "1.2.1.1.3.2.1.1.0" para um identificador `SNMPObjectIdentifier`). Os objetos definem métodos que permitem que seus valores sejam setados.

As classes `SNMPSequence` contem vetores de outras subclasses `SNMPObjects`. Finalmente, a classe `SNMPCommunicationsInterface` define métodos para enviar consultas do tipo `GetRequest` e `SetRequest` para os dispositivos da rede. Esta classe abrange todos as necessidades, em termos de comunicação, para que sejam desenvolvidas pequenas aplicações que se utilizam do protocolo SNMP.

Muitas das funcionalidades fornecidas ao usuário são implementadas pela classe `SNMPCommunicationInterface`. Este construtor abre um socket para um host a fim de se comunicar através da porta 192. Os métodos dessa classe fornecem maneiras de se

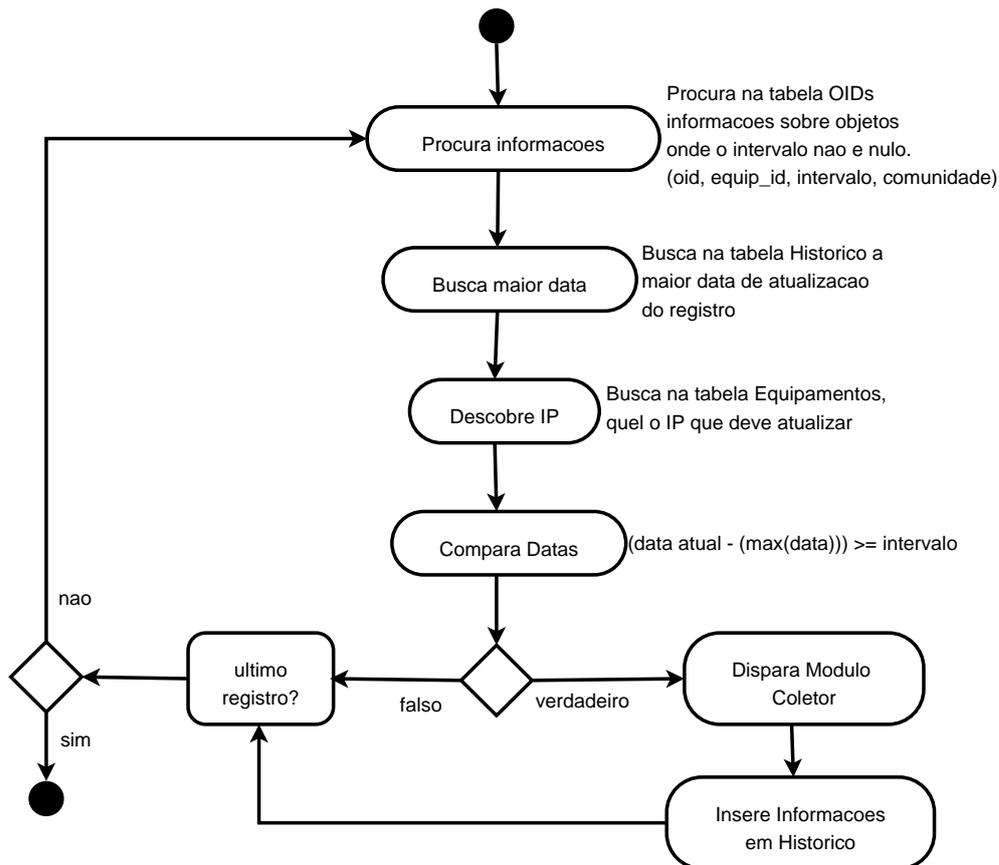


Figura 4.11: Funcionamento do módulo de atualização de histórico

recuperar ou setar valores de variáveis SNMP no dispositivo remoto, criando uma string que especifica o objeto (OID) desejado.

O construtor de `SNMPCommunicationInterface` tem três parâmetros: um inteiro que especifica a versão do protocolo a ser usada, o endereço do dispositivo remoto armazenado como um objeto `InetAddress` e uma string que armazena a comunidade a ser usada para realizar as operações.

Uma vez que uma interface de comunicação é criada para se comunicar com um dispositivo, os valores das variáveis SNMP podem ser recuperados através de chamadas ao método `getMIBEntry()`. Este método usa como argumento uma string com o OID da variável em questão. O valor de retorno é um tanto complexo devido ao fato de que o SNMP apresenta a possibilidade de consulta de valores de mais de uma variável em uma única mensagem (porém este método particular recupera apenas um valor por vez). O valor de retorno é do tipo `SNMPVarBindList`, uma subclasse de `SNMPSequence` que armazena uma sequência de pares (OID, valor). Assim, o valor de retorno é acessado da seguinte forma:

- o primeiro elemento é extraído de `SNMPVarBindList` usando o método `getSNMPObjectAt()`;
- o elemento extraído é colocado no formato da sequência - par (OID, value);
- o segundo elemento - o valor - é extraído do par usando-se o método `getSNMPObjectAt()`.

O resultado deste processo é uma subclasse `SNMPObject`. A classe do valor retornado depende do OID que foi requisitado e pode ser descoberto através do método `getClass()`. O valor interno pode ser recuperado usando-se a subclasse `SNMPObject` e o método `getValue()` que retorna uma subclasse de `java.lang.Object` apropriada para representar o valor recuperado.

O processo de setar valores em variáveis usa o método `setMIBValue()` presente em `SNMPCommunicationInterface`. Este método usa como argumento uma string, especificando o OID da variável que se deseja alterar e um objeto `SNMPObject` apropriado com o novo valor. O pacote `SNMP` define também certas exceções para indicar problemas com construtores de objetos, recuperação e alteração de valores de objetos.

Com base no pacote foi criada, para Doroty, a classe `SNMPCollect`. A classe `SNMPCollect` é responsável por disparar as ações possíveis dentro do protocolo `SNMP`. Ela combina as classes `SNMP` de forma a disparar as mensagens "GetRequest", "GetNextRequest" e "GetBulkrequest".

4.4.6 Módulo Central e Regras de Funcionamento

O módulo central é responsável por receber toda a informação gerada pelo software `ALICE` após a execução da normalização da entrada e do algoritmo `Graphmaster` e processar essa informação de modo a enriquecer as respostas de uma maneira mais prática. Além disso, também é o módulo responsável por receber os parâmetros e aplicar regras de tratamento para cada combinação de entrada possível. Entende-se por combinações de entrada o conjunto de parâmetros que são efetivamente passados, desconsiderando os que têm valor nulo. Alguns exemplos de combinações possíveis são mostrados na tabela 4.4.

Cada um desses casos apresenta um conjunto de regras de tratamento próprias e vai buscando as informações do banco de dados da forma mais adequada.

Tabela 4.4: Regras de tratamento - módulo central

Caso	Parâmetro	Tipo de pergunta
1	tópico, endereço, operação	"Qual a tabela de roteamento no noc?"
2	tópico, operação	"Qual a tabela de roteamento atual?"
3	tópico, endereço, operação, tempo	"Tráfego em 143.54.47.11 na ultima semana?"
4	tópico, operação, tempo	"Qual foi o tráfego nos últimos 7 dias?"
5	tópico, endereço, tempo, situação	"Média de tráfego em 143.54.47.11 nos últimos 3 dias?"

Como se pode ver na tabela 4.4, cada tipo de pergunta é associado a um caso. A todos os casos devem ser atribuídos tópicos para que o *chatterbot* possa se direcionar para o assunto desejado. Quando as perguntas buscam uma informação atual da rede (casos 1), sem que haja necessidade de consulta ao banco histórico, os parâmetros de tempo e situação são setados para nulos. Já os casos em que há uma consulta ao banco histórico (casos 3,4 e 5), o parâmetro tempo passa a receber o valor desejado. O caso 2 representa uma situação especial em que o usuário faz uma pergunta incompleta, sem passar a informação sobre qual equipamento deseja consultar, neste caso, o módulo central se encarrega de percorrer o banco de dados verificando quais equipamentos podem conter a informação desejada e mostrando ao usuário uma lista de endereços possíveis para a consulta, permitindo com que o usuário refaça a pergunta de forma completa, simplesmente escolhendo entre um

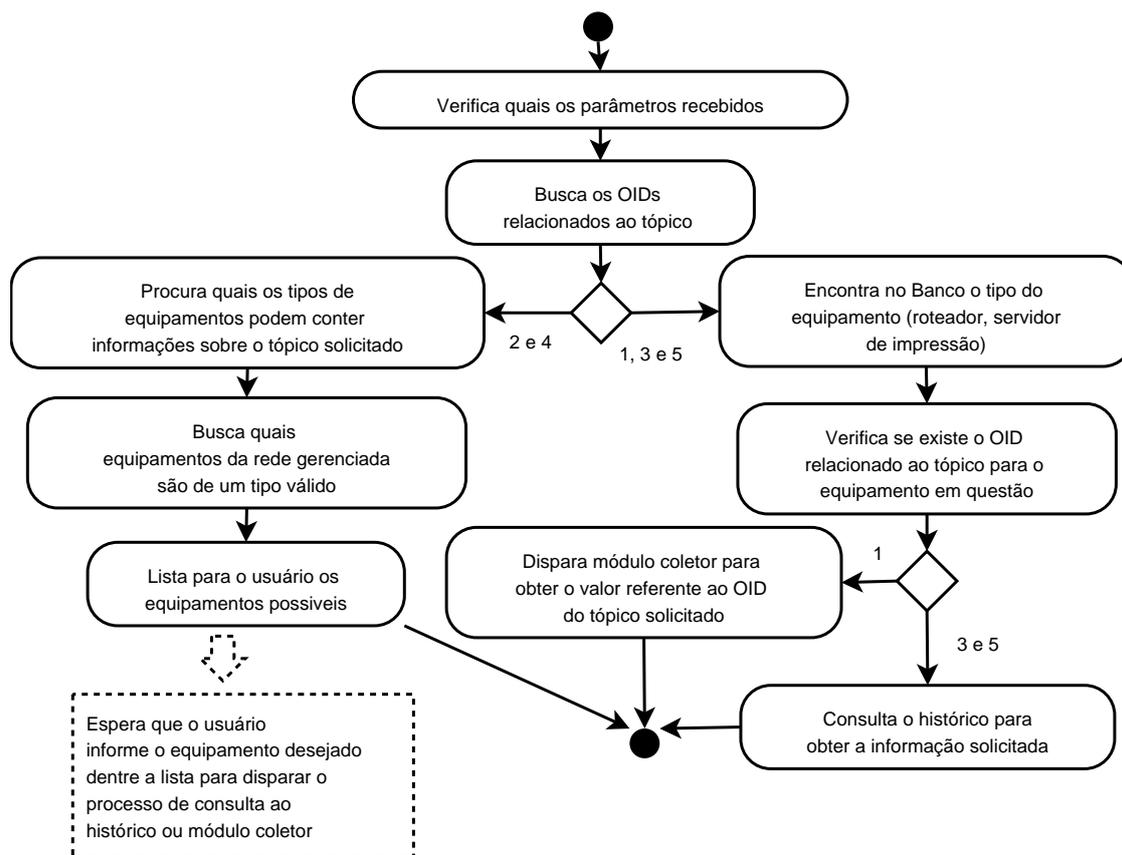


Figura 4.12: Funcionamento do módulo central para os casos citados na tabela 4.4

dos equipamentos. O caso 4 trabalha da mesma maneira, porém, quando recebe a informação completa, faz a busca no banco de histórico para que a informação seja pertinente. No caso da pergunta 5, é solicitada a média de tráfego, que é passada ao sistema através do parâmetro situação. Existem 3 tipos de situações em que Doroty é capaz de encontrar respostas: valores médios, maior valor e menor valor. Nesses casos, o sistema verifica o banco histórico e encontra a informação desejada. A figura 4.12 ilustra o funcionamento do módulo central. Os números nos nós de decisão representam os casos da tabela 4.4

Um outro tipo de pergunta possível, em Doroty, é caracterizada pela existência de um problema, por exemplo: "Porque minha rede está um pouco lenta?". Neste caso, Doroty aciona um conjunto de regras que apresenta ao usuário quais as possíveis causas do problema, explicando quais objetos da MIB2 podem ser consultados para a verificação da existência do problema na rede. Ela aponta algumas possíveis causas e quais seriam as possíveis soluções para cada caso, de forma a permitir que o usuário possa investigar o problema e resolvê-lo.

Ao final do processamento de cada tipo de pergunta, Doroty formula uma resposta, baseada no objeto consultado e nas características da pergunta do usuário. Essas respostas são programadas para, além de fornecer o estado atual da rede, ensinar ao usuário como proceder para encontrar a informação sozinho. Assim, ela adiciona em suas respostas links para tutoriais sobre o assunto, fotos, animações ou bibliografia. Muitas vezes, Doroty pode formular a resposta com dicas de outras ferramentas que podem ser utilizadas para que a informação seja obtida. Dessa forma, o sistema busca atingir seu grande foco: o de treinar usuários até que se sintam confiantes para administrar a rede.

4.4.7 Módulo Auxiliar de Diagnóstico de Problemas

Como o objetivo de Doroty é o de treinar profissionais menos experientes, optou-se por programar o módulo central de forma a não tentar diagnosticar o problema, mas sim, fornecer as possíveis causas e suas soluções, de modo que o usuário siga utilizando Doroty. Desse modo, ele pode fixar os conhecimentos particulares sobre gerenciamento quando realiza as pesquisas sugeridas pelo chatterbot para diagnosticar o problema.

Entretanto, um módulo auxiliar de diagnóstico de problemas também foi desenvolvido. Tal módulo leva em consideração o estado da rede e histórico. Para cada problema são armazenados os objetos da MIB2 que podem ser utilizados para uma tentativa de se diagnosticar o problema, bem como os valores esperados que tais objetos apresentem. Assim, quando forem disparadas consultas em tais objetos para que se verifique as condições da rede, Doroty é capaz de verificar se o valor encontrado está em desacordo com o esperado, auxiliando o usuário a fazer seu diagnóstico.

Quando ativado, o módulo auxiliar de diagnóstico funciona da seguinte forma: quando um problema é apresentado, em forma de pergunta, um tópico é atribuído ao problema, da mesma forma que o funcionamento normal do módulo central. Durante o processamento, o tópico é associado a uma ou mais causas, que, por sua vez, são associadas a um ou mais objetos da MIB2 que podem ser utilizados para verificar a existência de tal problema. Para cada causa encontrada, o módulo coletor é acionado de forma a realizar consultas na rede para verificação de alguma característica ou ainda uma consulta na base de dados de histórico pode ser feita. O resultado de cada uma das consulta é comparado ao valor esperado para o objeto de forma a verificar se a rede está em desacordo com o seu estado desejado. Quando uma condição de desacordo é encontrada, são apontadas as possíveis soluções para o problema específico, de forma a ensinar o usuário como proceder para solucionar o problema.

Assim, o módulo auxiliar leva em conta todas as causas que podem gerar o problema. Porém, no final do processamento Doroty ordena as causas de forma a apresentar ao usuário uma lista de causas possíveis que estejam em desacordo coma rede, ordenada por sua importância.

4.5 Considerações Finais Sobre o Capítulo

Este capítulo apresentou a motivação e as características da implementação e do modelo de Doroty. Foram descritos seus módulos de funcionamento e as tecnologias que foram utilizadas para que fosse possível capacitar o chatterbot para fazer interface com uma rede.

O capítulo a seguir vem apresentar os experimentos realizados com o objetivo de avaliar Doroty. Serão descritos os experimentos realizados e, posteriormente, serão levantadas as constatações verificadas em cada experimento, a fim de se validar o uso de Doroty como ferramenta de aprendizado.

5 AVALIAÇÃO E ANÁLISE DE RESULTADOS

Este capítulo apresenta alguns experimentos que foram feitos utilizando o chatterbot implementado para avaliar a influência da utilização de Doroty no treinamento de administradores de redes. Os testes realizados foram avaliados sob vários aspectos, de forma a fornecer conclusões independentes sobre os quesitos de sua utilização. Em uma primeira etapa, o chatterbot foi utilizado apenas com sua base de conhecimento teórica e o grande objetivo de tal experimento foi o de avaliar o impacto do uso de uma tecnologia inovadora na área, além de verificar a abrangência do vocabulário teórico e sua performance em termos de linguagem natural. Em uma segunda etapa, Doroty foi testada utilizando sua base de conhecimento teórica e dinâmica, através da disponibilização de funcionalidades específicas, a fim de verificar o impacto das respostas dinâmicas no processo de aprendizado. Finalmente, a última etapa de sua avaliação envolveu a análise dos registros de conversação armazenados durante todo o período em que Doroty esteve disponível para testes.

A seguir, serão descritos os experimentos realizados e, posteriormente, os resultados e as observações retiradas de cada experimento serão abordados.

5.1 Testes Realizados e Etapas de Avaliação

5.1.1 Teste de bot Estático

Entende-se por teste de bot estático, a avaliação das respostas do chatterbot Doroty, sem a interação efetiva com uma rede, ou seja, apenas avaliando seus conceitos teóricos e verificando a profundidade do seu conhecimento. Doroty foi utilizada, através da internet, por estudantes de graduação para fixação de conceitos básicos de gerenciamento de redes. Os estudantes questionados foram escolhidos pela autora dentre diversas instituições de ensino superior do estado, todos em cursos da área de computação. Embora essa validação preliminar não atinja o mesmo público exato desejado, o impacto desse experimento no processo de aprendizagem foi eficiente porque os estudantes reconheceram a importância do software enquanto assistente para resolução de dúvidas e assimilação de conceitos na área.

Quando questionados livremente sobre os pontos fortes e fracos do protótipo, os pontos destacados foram:

- Disponibilidade em tempo integral;
- Apresentação de várias alternativas de respostas para a mesma dúvida, importante quando a primeira não esclarece o conceito;
- Processo mais dinâmico, não há disputa pela atenção do professor;

O segundo item apresentado pode ser verificado na resposta de um aluno, transcrita abaixo:

"O ponto forte, para mim, foi paciência que ela teve em responder várias vezes sobre o mesmo assunto. O mais interessante foi que as respostas não eram as mesmas e então me pareceu que ela é capaz de entender que eu não entendi todo o conteúdo, sem que eu precisasse dizer uma palavra. Me senti super bem interagindo com ela pois ninguém estava avaliando minha dificuldade de entender novos conceitos, coisa que, com os colegas, na sala de aula, fico meio constrangida em ficar perguntando mais de uma vez". (e-mail)

Entretanto, mesmo que a sua utilização represente um grande atrativo para tais estudantes, o aspecto da linguagem natural deve ser cuidadosamente observado. Neste experimento, alguns estudantes se sentiram meio frustrados, já que a base de conhecimentos abrangia apenas questões sobre alguns temas dentro do gerenciamento de redes. Essa observação, que pode ser vista no diálogo transcrito abaixo atentou para a necessidade de inclusão de categorias gerais na base de conhecimentos, ou seja, aquelas que devem entrar em ação quando nenhuma resposta pode ser fornecida por parte do chatterbot. Conforme apresentado no capítulo 2, algumas estratégias de conversação foram adotadas a partir de tal constatação, como a utilização de respostas genéricas que diminuíssem a frustração do usuário, mudando o tópico da conversação ou apresentando algum conceito relacionado.

"Acho que Doroty é um bom recurso para assimilação de conceitos e é muito bom para ser usado para estudar para uma prova, por exemplo. Mas sobre a conversação, acho que poderia ser bem melhor. Muitas vezes tive dificuldade em fazer fluir a conversa. Doroty ainda precisa aprender muitos conceitos porque cada pergunta sem resposta desaponta nossa expectativa.". (e-mail)

Embora alguns alunos demonstraram se sentir intimidados ou inseguros com o aspecto da linguagem natural, sua utilização ainda é justificável. Cabe salientar que somente com uma extensa utilização de um bot por alunos e conseqüente análise dos registros e das impressões desta utilização é possível melhorar o conhecimento e a interação, tornando a conversação mais natural e abrangente. Porém, em um novo teste, com a utilização da estratégia mencionada para contornar tal problema, os alunos apresentaram opiniões mais positivas, salientando que preferem uma resposta qualquer a nenhuma. Foi observada a necessidade de um *feedback* para que o chatterbot não passasse a impressão de estar processando outra informação ou ainda a impressão de que ele simplesmente ignorasse o usuário quando não obtivesse resposta, fazendo com que o mesmo ficasse em dúvida sobre quando deveria ou poderia fazer novo questionamento.

5.1.2 Teste de bot Dinâmico

Considerando os aspectos levantados nos testes de bot estático, foram realizadas algumas melhorias na base de conhecimento antes da aplicação dos testes de bot dinâmico. A principal melhoria foi a adoção de novas estratégias de conversação, que permitiram capacitar Doroty para encontrar respostas gerais quando algum conceito não fosse encontrado em sua base, provendo o feedback necessário comentando na seção 5.1, subseção

5.1.1.

Os testes de bot dinâmico são aqueles que habilitam os módulos de Doroty de interface com a rede, permitindo que o usuário visualize exemplos que se utilizam de dados reais.

5.1.2.1 *Teste com categorias de conhecimento dinâmicas: etapa parcial*

Em uma etapa parcial, Doroty foi utilizada somente com suas respostas estáticas e uma parte das respostas dinâmicas, abrangendo somente tópicos simples (sugestões de que objetos deveriam ser consultados para cada tópico e que informações poderiam ser encontradas através da coleta de dados de cada objeto). O principal interesse do experimento em questão foi de visualizar o efeito causado pelo chatterbot atuando como administrador de uma rede, além de medir o impacto causado pelo uso de um novo paradigma para o gerenciamento de redes (usando linguagem natural). Foi solicitado que os estudantes manifestassem suas opiniões sobre o chatterbot e a idéia de aprender a gerenciar uma rede utilizando linguagem natural como fonte de aprendizado, bem como respondessem perguntas sobre o quão preparados se sentiam para aplicar os conhecimentos sobre gerenciamento para resolver possíveis problemas encontrados em uma rede.

Nesta etapa, os estudantes mostraram interesse na utilização do sistema, fazendo com que se aprofundassem na busca de detalhes sobre a área. Um exemplo de diálogo realizado entre um estudante e Doroty é transcrito abaixo:

Usuário: *"Como posso ver a tabela de roteamento em uma máquina?"*

Doroty: *"Para consultar a tabela de roteamento de uma máquina, utilizando a MIB2, é necessário verificar alguns objetos específicos. Os campos e outras formas de consultar a tabela de roteamento você pode ver nesta página:*

http://143.54.47.63/paginas/roteamento.html ." (A figura 5.1 apresenta a página que é apresentada ao usuário, para que ele verifique os objetos necessários e as maneiras de se consultar a tabela de roteamento. Dependendo da interface de diálogo utilizada, esta página pode estar diretamente linkada, sem a necessidade de se especificar o endereço na resposta.)

Usuário: *"Mas quais os campos que existem na tabela de roteamento?"*

Doroty: *"A tabela de roteamento deve guardar informações sobre que conexões estão disponíveis para se atingir uma determinada rede e alguma indicação de performance ou custo do uso de uma dada conexão."*

Usuário: *"Tu pode me dar mais detalhes sobre quais os campos que posso encontrar em uma tabela de roteamento?"*

Doroty: *"Existem alguns campos que podem compor as entradas de uma tabela de roteamento:*

Endereço de Rede: Pode ser o endereço de uma rede (por exemplo: 10.10.1.0), o endereço de um equipamento da rede, o endereço de uma sub-rede ou o endereço da rota padrão (0.0.0.0). A rota padrão significa: "a rota que será utilizada, caso não tenha sido encontrada uma rota específica para o destino". Por exemplo, se for definida que a rota padrão deve ser enviada pela interface com IP 10.10.6.2 de um determinado roteador,

sempre que chegar um pacote, para o qual não existe uma rota específica para o destino do pacote, este será enviado pela rota padrão.

Máscara: A máscara de sub-rede utilizada para a rede de destino.

Next Hop ou próximo salto (Endereço de Gateway): Endereço IP da interface para a qual o pacote deve ser enviado.

Interface: É a interface através da qual o pacote deve ser enviado.

Métrica ou Custo: A métrica é um indicativo da distância da rota, entre destino e origem, em termos de hops (saltos). Pode haver mais de um roteador entre origem e destino. Também pode haver mais de um caminho entre origem e destino. Se forem encontradas duas rotas para um mesmo destino, o roteamento será feito pela rota de menor valor no campo métrica. Um valor menor indica, normalmente, um número menor de hops (saltos) entre origem e destino."

Usuário: *"Então, por favor, consulte a tabela de roteamento dessa máquina em que estou?"*

Doroty: *"Desculpe amigo! O meu módulo de consultas ainda não está habilitado nesta etapa de testes. Mas você pode tentar realizar estas consultas sozinho, utilizando algumas das alternativas que eu mencionei anteriormente. O que você acha?"*

Usuário: *"Vou tentar depois. Antes, queria saber mais sobre o que são métricas"*

Doroty: *"As métricas são padrões de medida de uma variável da rede. Os algoritmos de roteamento utilizam diferentes métricas para determinar a melhor rota, sendo que os mais sofisticados podem basear sua escolha em múltiplas métricas, combinando-as de forma que resulte em um valor único."*

O trecho de diálogo acima mostra a busca por mais informações sobre um mesmo conceito. Nele, apesar de Doroty estar com o módulo de consultas desabilitado, o usuário continua buscando informações sobre o conteúdo. O diálogo apresentado também serve para ilustrar outras características envolvendo o uso de chatterbots. Podemos observar que, neste caso, o usuário se encontra envolvido na conversação, como se estivesse falando com outro humano já que utiliza uma forma polida de solicitar que seja feita uma consulta na rede, através do uso de *"por favor"*. Também se pode verificar a possibilidade de um chatterbot de incorporar recursos em suas respostas, quando apresenta a página com as informações sobre tabela de roteamento diretamente linkada, bastando que o usuário clique uma vez com o mouse para acessar toda a informação (Figura 5.2).

Uma outra observação importante, analisando o trecho de diálogo apresentando, envolve a influência geográfica na linguagem e no modo de se expressar dos usuários. Podemos observar que o usuário utilizou uma forma regionalista de fazer perguntas quando observamos o pronome *"Tu"* empregado na frase *"Tu pode me dar mais detalhes sobre quais os campos que posso encontrar em uma tabela de roteamento?"*.

Assim, a conclusão desta etapa de testes envolveu uma preparação mais detalhada para o teste final, onde foram analisadas novamente características já abordadas, com o

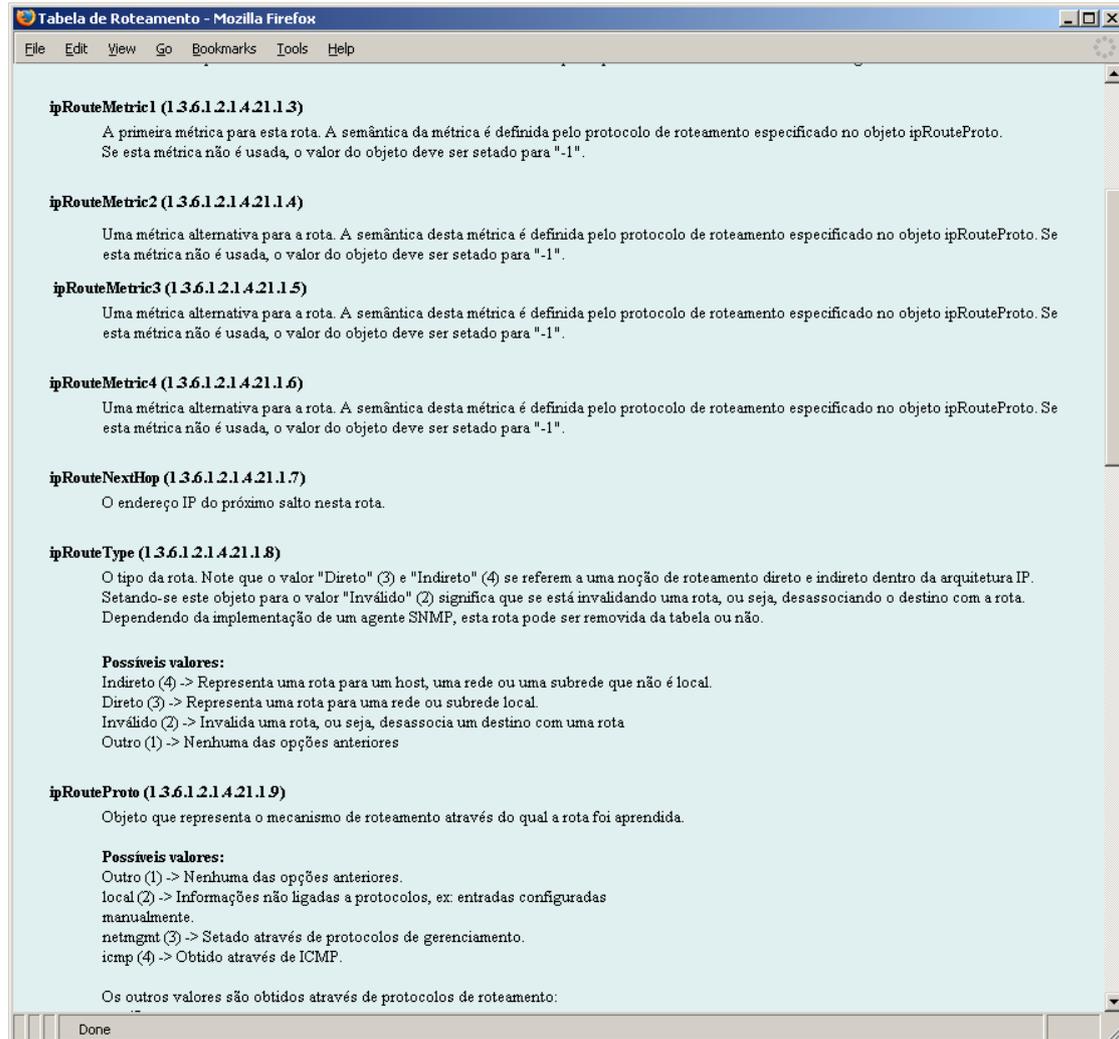


Figura 5.1: Tela que apresenta informações sobre a tabela de roteamento

objetivo de melhorar Doroty como um todo.

5.1.2.2 Teste com categorias de conhecimento dinâmicas: etapa total

O último experimento utilizando Doroty foi realizado com o módulo de consulta da rede ativo, além da base de conhecimento estática. Para aplicação deste experimento, foram selecionados 30 usuários, dos mais diversos níveis de conhecimento sobre o assunto e de diferentes faixas etárias. Buscou-se um público bem diversificado (que apresentasse tanto pessoas com conhecimento sobre a área como sem, que já tivesse trabalhado na área, que não tivesse trabalhado na área), a fim de verificar quais os aspectos mais relevantes a cada um dos avaliadores, bem como qual o impacto exercido pelo chatterbot em cada grupo específico. A escolha dos candidatos envolveu uma breve entrevista, a fim de se selecionar os avaliadores conforme seu perfil.

Um questionário foi elaborado para ser preenchido pelos avaliadores. Esse questionário pode ser visto no anexo 1 deste documento e é dividido em três grupos de perguntas. O primeiro grupo se destina a perguntas as características pessoais dos avaliadores, importantes para mensurar o grau de satisfação de Doroty segundo cada tipo de avaliador, por exemplo: nível de conhecimento, experiência na área, idade, localização geográfica, entre outros. O segundo grupo se destina a avaliar, de forma geral, o protótipo desenvolvido

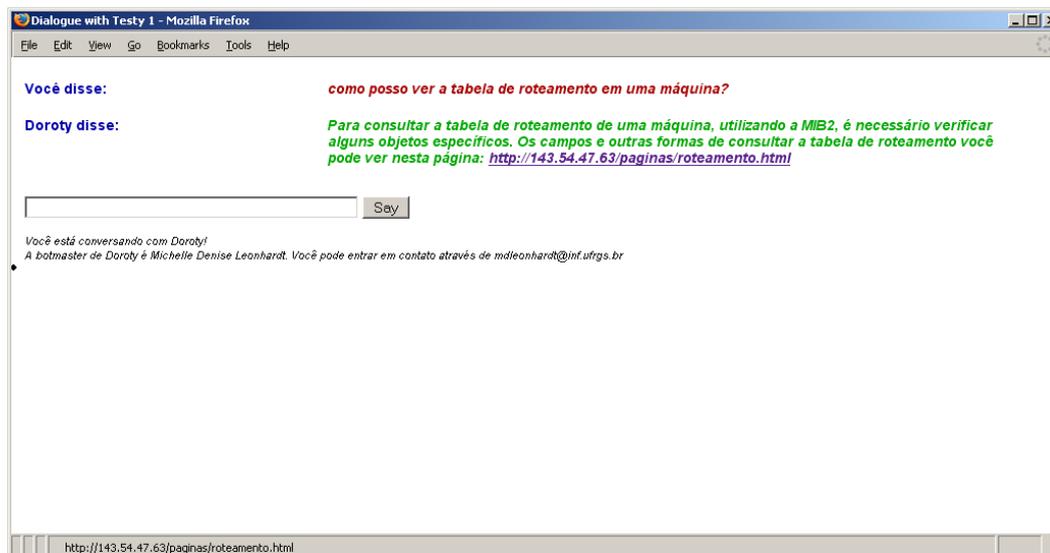


Figura 5.2: Exemplo da interface simples de Doroty apresentando link direto para uma página com informações sobre o conteúdo

utilizando algumas tarefas e instruções pré-definidas. Dentro desse grupo também é possível avaliar Doroty livremente, a fim de reforçar ou avaliar o desempenho da mesma nos aspectos que o avaliador julgar necessário. O terceiro grupo apresenta apenas um campo para sugestões e observações, deixando o avaliador livre para se expressar da forma que achar conveniente.

Antes da aplicação, os usuários foram avisados de que estavam avaliando um protótipo e para tanto, foram instruídos a:

- Considerar a hipótese de reformular a questão caso alguma resposta não fosse encontrada;
- Nem todos os objetos da MIB2 estavam disponíveis para consulta;
- Considerar que o conteúdo das respostas do bot é originado de referências bibliográficas sobre o assunto.

Dessa forma, se buscou conscientizar os avaliadores para o fato de que outros aspectos do sistema estariam sendo avaliados. A primeira observação teve como objetivo forçar a elaboração de novas perguntas por parte do usuário, a fim de se observar como o mesmo se expressa quando busca uma informação e como uma resposta negativa influencia esse processo. A segunda observação atentou para a possibilidade de não serem encontradas respostas para algumas perguntas nos testes livres e a terceira atentou para o fato de que os conceitos constantes na base de conhecimento eram retirados de referências, embora também pudessem ser julgados, já que cabe ao engenheiro de conhecimento absorver os conteúdos de forma que se aproxime do que o seu público alvo espera.

A análise das respostas do grupo 1 aponta uma diversidade no grupo de avaliadores. Em se tratando do grau de instrução dos avaliadores, 44% dos entrevistados possuem grau de ensino superior e outros 20% possuem ensino ainda mais avançado (Tabela 5.1). Porém, nem todos os usuários com grau de instrução tão alto atuam na área da computação, fazendo com que o grau de conhecimento sobre o assunto seja bem diversificado (Tabela

5.2). Dessa forma, apesar da grande quantidade de usuários com grau de instrução elevado, a maioria não possui experiência na área, ou seja, representam bom potencial como público alvo do sistema (Figura 5.3). Para os entrevistados com ensino médio ou inferior, a pergunta sobre a área de atuação foi modificada para a área que a pessoa gostaria de atuar.

Tabela 5.1: Nível de instrução dos avaliadores

Parâmetros	Pergunta 5
Ler sem grau de ensino	0
Fundamental	1
Médio	5
Médio Técnico	1
Superior	13
Especialização	4
Mestrado ou Superior	6
Total	30

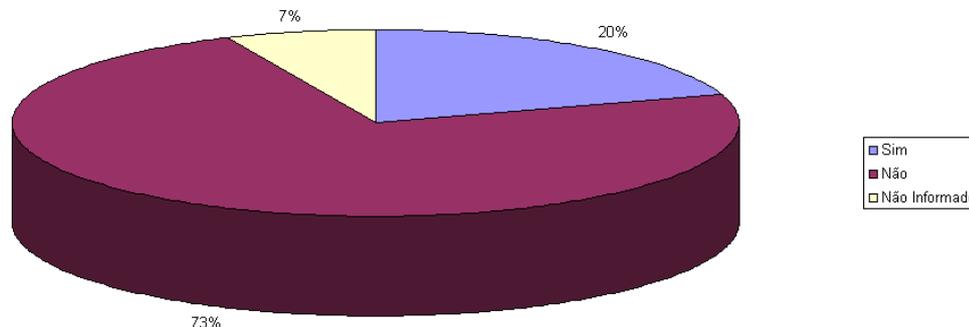


Figura 5.3: Experiência dos avaliadores

Além disso, o grupo de usuários selecionado abrange 12 pessoas nascidas em Porto Alegre, 12 no interior do estado e 6 em outras regiões do Brasil. Dessas, apenas uma ainda reside fora do estado e 6 residem no interior. A maioria apresenta idade entre 21 e 30 anos e possui acesso ao computador em casa e no trabalho ou escola.

As informações do grupo 1 foram relacionadas com as informações constantes no grupo 2 a fim de avaliar Doroty segundo cada perfil apresentado. As perguntas 1 até 4 direcionavam o usuário a fazer perguntas apenas teóricas, a fim de avaliar novamente o poder de conversação do sistema. Dentre os resultados obtidos nessas avaliações, foi constatado que usuários com maior nível de experiência consideravam as respostas do

Tabela 5.2: Área de atuação dos avaliadores

Parâmetros	Pergunta 5a
Exatas (Computação, Sistemas de Informação, Análise de Sistemas)	14
Exatas	7
Humanas	7
Saúde	2
Total	30

chatterbot muito longas, geralmente atribuindo grau "Parcialmente Insatisfeito" para tal característica. Entretanto, usuários com menor nível de experiência e com menor conhecimento na área atribuíram, em sua maioria, o grau "Plenamente Satisfeito" no quesito. Uma resposta possível apresentada pelo bot para a pergunta é transcrita abaixo:

O modelo SNMP de gerenciamento de redes apresenta alguns componentes básicos como: nós gerenciados, estações de gerenciamento, informações de gerenciamento e protocolo de gerenciamento. Os nós gerenciados podem ser hosts, roteadores, impressoras ou qualquer outro tipo de dispositivo capaz de comunicar suas informações de status para o mundo externo. Para ser diretamente gerenciado através de SNMP, um nó deve ser capaz de executar um processo de gerenciamento SNMP, que é denominado agente SNMP. Cada agente mantém um banco de dados local contendo variáveis que descrevem seu estado e histórico, além de afetar sua operação. O gerenciamento de rede é feito a partir de estações de gerenciamento que contém um ou mais processos que se comunicam com os agentes espalhados pela rede, emitindo comandos e obtendo respostas. Para permitir que uma estação de gerenciamento se comunique com todos os componentes de uma rede, é necessário que a natureza das informações de cada equipamento seja rigidamente especificada através das MIBs. Assim, uma estação de gerenciamento pode consultar características e informações sobre cada dispositivo enviando uma solicitação de informações a ele ou fazendo com que ele atualize seu estado. O agente, por sua vez, simplesmente responde com as informações solicitadas ou confirma se atualizou os seus dados da forma como foi solicitado. De outra forma, quando ocorrerem eventos significativos não esperados em uma rede, um agente pode enviar uma trap ao gerente, comunicando que algo não esperado aconteceu. O protocolo SNMP define sete mensagens que podem ser usadas: "get-request", "get-next-request", "get-bulk-request", "set-request", "Inform-request", "trap" e a mensagem de resposta.

Embora 58% dos avaliadores tenha atribuído um grau 4 ou 5 para as respostas teóricas de Doroty, optou-se por realizar um refinamento nas mesmas, de forma a diminuir o tamanho das respostas e a quantidade de informação presente nelas, expandindo a base de conhecimento com novas categorias, conforme sugestão dos avaliadores da área com maior grau de experiência. Tal procedimento não frustrou grupos com menor grau de experiência, já que as respostas foram cuidadosamente revisadas e outras categorias de conhecimento inseridas a fim de complementar as respostas quando mais informações são solicitadas. Porém, a possibilidade de múltiplas respostas permitiu que se elaborassem

respostas mais longas e mais curtas, de forma que um usuário inexperiente, recebendo uma resposta que não o agrada, pode fazer a pergunta novamente e receber respostas com maior quantidade de informações.

Nos demais quesitos, o grau de satisfação dos usuários foi bastante elevado. Houve poucos casos em que uma resposta não foi encontrada. Na maioria dos casos, problemas de grafia e abreviação impediram uma correta identificação da pergunta por parte de Doroty. Assim, a maneira encontrada para contornar tal problema foi verificar os problemas de linguagem e aumentar o número de possibilidades de substituições que Doroty realiza antes de aplicar o algoritmo de casamento de padrões. Sobre a performance de Doroty, ou seja, o tempo que os usuários esperaram para obter uma resposta, 100% dos mesmos atribuiu o grau "Plenamente Satisfeito" a esse quesito.

A pergunta 5, que solicitava que um usuário executasse um comando *ping* ou *trace-route* através de Doroty, foi colocada no questionário com o objetivo de verificar se a utilização de Doroty para realizar interface com a rede através de outras ferramentas que não a MIB2 seria vantajoso. Também se buscou verificar se o conteúdo apresentado na pergunta 4 ajudaria os usuários nessa tarefa, uma vez que Doroty foi programada a dar dicas sobre a interação em linguagem natural, como pode ser visto na resposta transcrita abaixo sobre o comando ping.

Ping é utilizado em rede para testar a conectividade. Ele atua como um sonar: envia um pacote que pede uma resposta e mede quanto tempo demora para chegar esta resposta. Desta forma, quanto menor o tempo de resposta, melhor a conexão.

Quando queremos determinar se um determinado host está disponível na rede interna ou mesmo na Internet, frequentemente utilizamos o utilitário ping como um dos primeiros recursos de troubleshooting. O fato de um host não responder ao ping não quer dizer que ele esteja realmente fora da rede, pois este serviço pode estar desabilitado neste host por questões de segurança.

O comando é: ping -host- É chamado de cliente o host que inicia a comunicação, ou seja, a partir do qual o usuário executa o comando de teste de disponibilidade. Servidor é o alvo do teste, pois este deve possuir um serviço habilitado para ser capaz de receber o pacote do cliente e respondê-lo. O cliente envia primeiro um pacote do tipo ICMP Echo Request, ou simplesmente ICMP Echo. Quando o servidor recebe este pacote ele responde com um pacote do tipo ICMP Echo Reply. Este processo se repete até que o usuário cancele o ping com um "CONTROL+C". O utilitário então calcula as estatísticas e as exibe na tela. O usuário também pode controlar, através de opções do comando, quantos pacotes devem ser enviados, o intervalo de tempo entre eles, e o tamanho do pacote. Na verdade a área de dados do pacote não carrega nenhuma informação útil, entretanto, pode ser aumentada para testar a rede com pacotes de tamanhos diferentes. .

Posso executar um ping para você. Você pode perguntar, por exemplo: "ping em xxx.xxx.xxx.xxx" ou "execute um ping em xxx.xxx.xxx.xxx".

Assim, a pergunta fácil teve uma aceitação bastante grande. Quase que a totalidade dos usuários se sentiu satisfeita com a resposta fornecida por Doroty, atribuindo grau "Plenamente Satisfeito" para todos os quesitos. Porém, 26% dos usuários atribuíram um

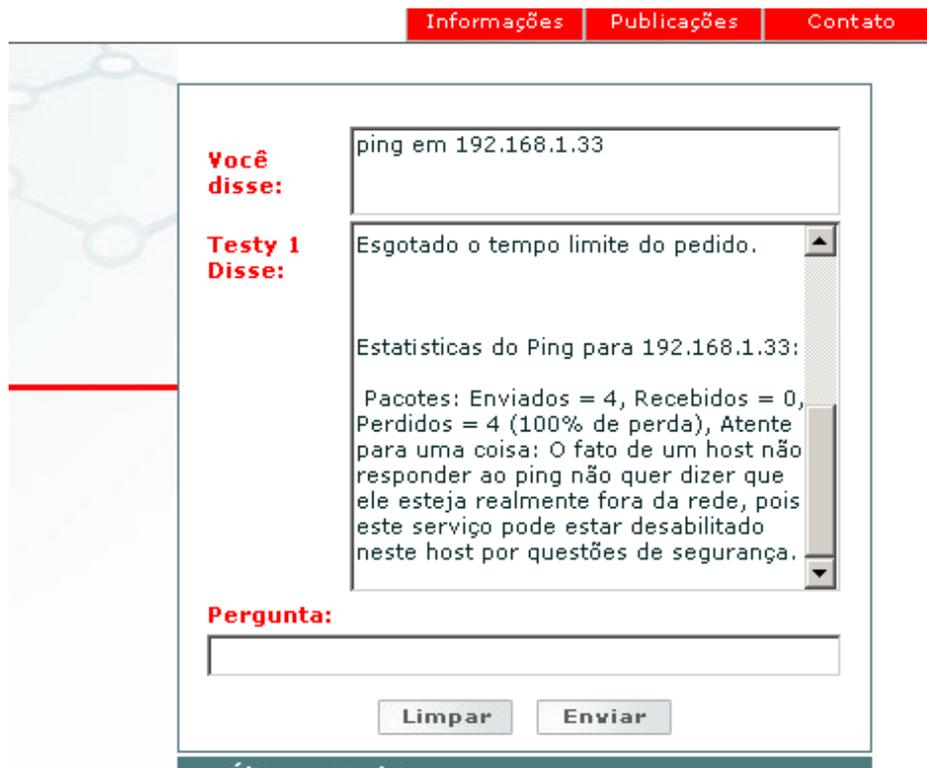


Figura 5.4: Um exemplo de execução do comando ping através de Doroty

grau "Parcialmente Satisfeito" em relação ao tempo esperado pela resposta. Essa atribuição pode ser justificada pelo fato de que a resposta só é enviada para o usuário ao final da execução do comando, o que leva um certo tempo. A resposta é apresentada exatamente como se o comando fosse digitado em uma janela comum do sistema operacional (figura 5.4). Se o usuário desejar mais informações sobre o comando, pode continuar questionando Doroty já que a mesma apresenta muitas informações sobre o comando em sua base de conhecimento (por exemplo: parâmetros do comando).

A pergunta 6 pode ser considerada uma das mais importantes do questionário e serve para verificar o comportamento de Doroty quando da realização de perguntas que envolvem tanto consultas momentâneas na rede, como perguntas que devem acessar a base de histórico de dados implementada no sistema. Em um primeiro momento, é solicitado que o usuário faça perguntas teóricas, apenas com o objetivo de obter conhecimentos básicos sobre o assunto, preparando o mesmo para avaliar as demais perguntas. A seguir, o usuário é solicitado a fazer perguntas que não fazem sentido, ou seja, que não apresentam todos os parâmetros necessários para que se verifique uma informação na rede, como é o caso das perguntas 6.2 e 6.6. As demais perguntas envolvem a consulta no banco histórico e a consulta de valores médios, máximos ou mínimos sobre um objeto armazenado no banco de histórico de dados.

Um fato importante de se observar é o de que a maioria dos usuários com conhecimento sobre o assunto não realizou as perguntas que tinham informações incompletas. Dentre esses, os que realizaram demonstraram sentir um certo receio quando solicitados a executar a tarefa, como podemos ver nos comentários apontados por uma avaliadora, com experiência, cursando mestrado na área de computação:

Não tive resposta, mas não sei se eu perguntei certo. Acho que não en-

tendi como devo proceder nesse teste.

Nesse tipo de pergunta, Doroty é programada para verificar quais equipamentos da rede que ela gerencia podem ser consultados e deve passar os endereços para o usuário. A resposta passada ao usuário pode ser verificada na figura 5.5.

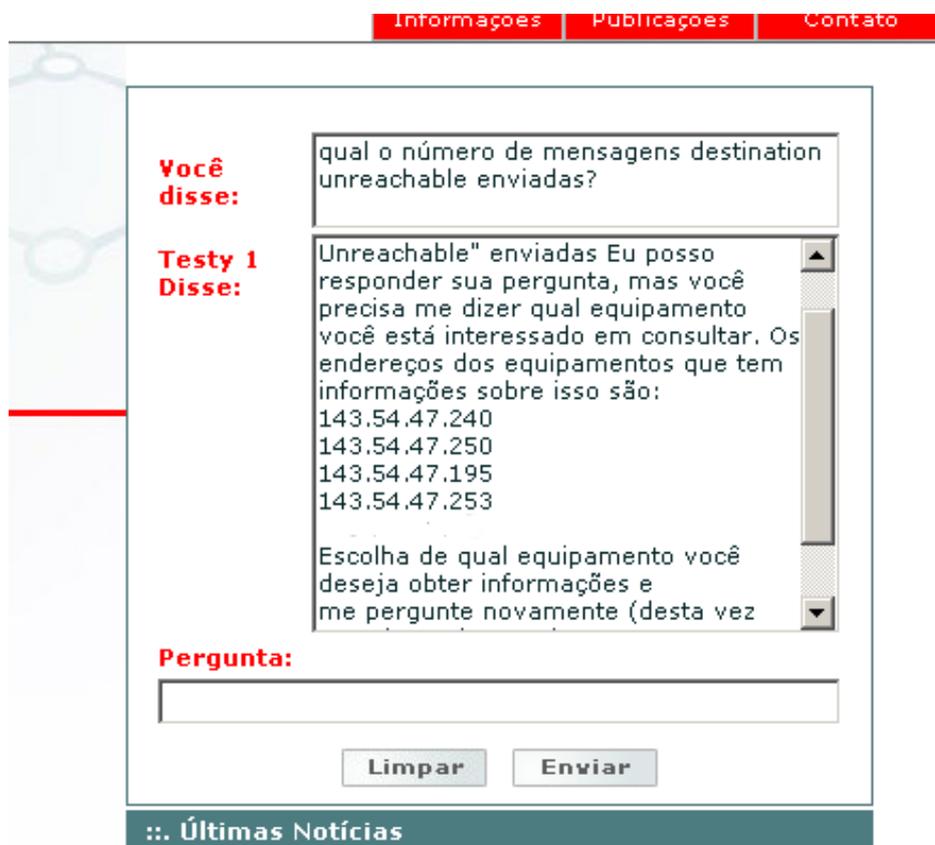


Figura 5.5: Um exemplo de resposta dada por Doroty quando uma pergunta incompleta sobre a rede é feita.

Entre os usuários sem experiência de gerenciamento, o grau atribuído quase que na totalidade dos casos foi "Plenamente satisfeito". A grande sugestão feita pelos usuários que atribuíram grau "Parcialmente satisfeito" foi o de que se implementasse um modo de escolha de endereço sem que fosse necessário realizar novamente a pergunta, como sugere Doroty. O comportamento apresentado pelos usuários mais experientes e com conhecimento pode ser justificado pelo fato de que os mesmos já possuem uma visão crítica da área e apresentam certa relutância em formular perguntas com informações incompletas. Para tais usuários, o grau atribuído variou bastante, porém a maioria marcou "Indiferente, neutro ou a situação não se aplica". Quando avisados sobre a intencionalidade de se formular uma pergunta com dados incompletos, os mesmos não apresentaram dificuldades, atribuindo grau "Plenamente Satisfeito" para a resposta recebida.

Ainda na pergunta 6, os questionamentos que envolviam consulta ao banco de dados histórico apresentaram alguns problemas em se tratando de linguagem natural. Porém, esse problema era esperado, visto que Doroty foi programada sem nenhum estudo prévio sobre como as pessoas se expressam para fazer perguntas que envolvem o passado. Assim, o grau atribuído a tais perguntas foi um tanto baixo, sendo em sua maioria "Parcialmente Insatisfeito". Em vista disso, foi iniciado um trabalho para expansão das categorias de

conhecimento de Doroty de forma a abranger o modo de expressão dos avaliadores. Ao ser novamente testada, o grau atribuído, na maioria dos casos passou a "Parcialmente Satisfeito", mas ainda com problemas de processamento de linguagem natural. Entretanto, os usuários demonstraram muito interesse em interagir com Doroty, realizando muitas tentativas para obter a resposta solicitada.

As perguntas 7 ("Descubra para que serve o objeto sysDescr?") e 8 ("Obtenha informações sobre o que é uma tabela de roteamento e qual a tabela de roteamento de um equipamento específico."), buscam demonstrar e avaliar novas maneiras de expressão e a apresentação de diferentes possibilidades de apresentação de respostas, uma vez que referenciam diretamente um objeto e, no caso da pergunta 8, apresentam links para páginas contendo mais informações e explicações. A figura 5.2 demonstrou o uso da interface com recursos html incorporados na resposta, permitindo um conforto em se tratando de interação. Na interface padrão, este conforto não está disponível, fazendo com que o usuário precise copiar o link apresentado e colar na barra de endereços de um navegador.

Para a pergunta 7, o grau atribuído por todos os avaliadores que executaram a tarefa foi "Plenamente Satisfeito". Nestes casos, Doroty apenas responde a pergunta de forma simples, sem acrescentar informações extras de nenhum tipo (figura 5.6). Esse comportamento ocorre porque a pergunta é mais técnica e, portanto, deve ter sido realizada por alguém que tem noções sobre gerenciamento de redes. Porém, Doroty questiona o usuário se o mesmo deseja consultar o valor do objeto em algum dos equipamentos gerenciados por ela, verificando os endereços e repassando-os ao usuário.

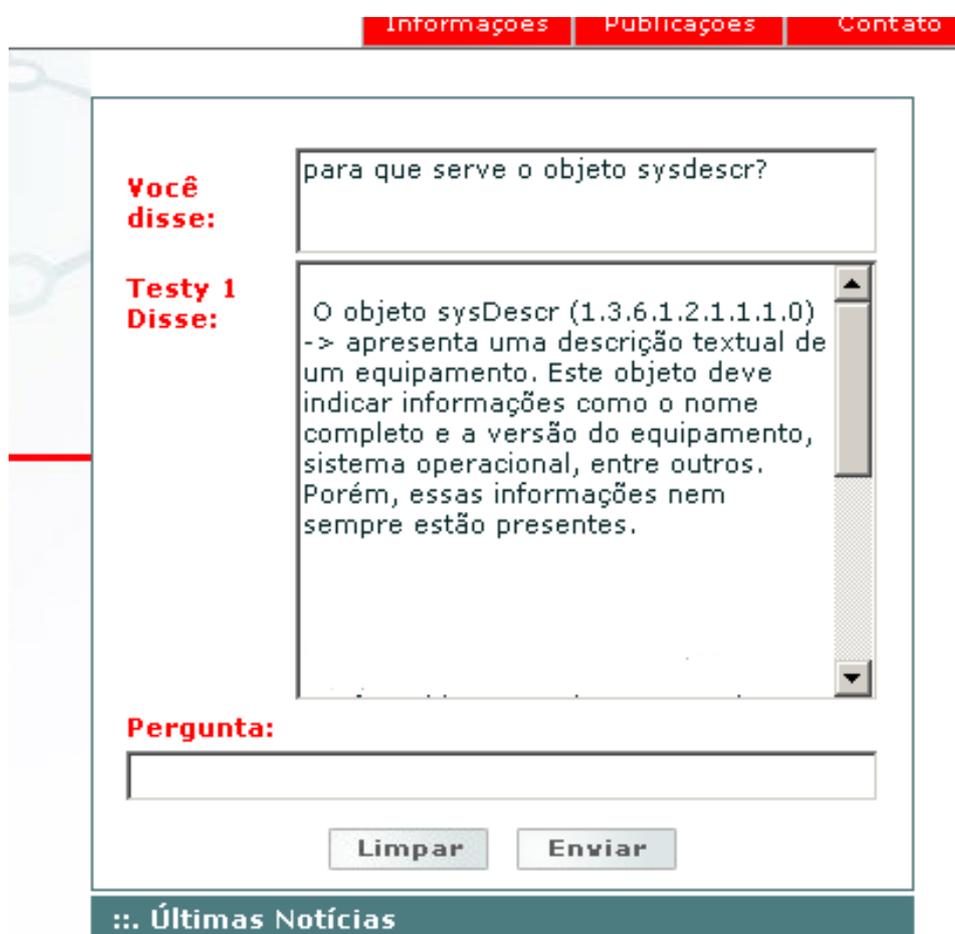


Figura 5.6: Resposta para uma pergunta sobre o objeto sysDescr

Tabela 5.3: Notas atribuídas ao protótipo

Nota	Qtd. Avaliadores	Com experiência	Com Conhecimento
10	9	2	1
9	12	1	5
8	4	1	3
7	1	0	1
6	3	2	3
5	0	0	0
4	1	0	1
3	0	0	0
2	0	0	0
1	0	0	0
Total	30	6	14

A pergunta 8 teve como avaliação geral o grau "Parcialmente Satisfeito" quando da utilização da interface padrão e "Plenamente Satisfeito" na interface com incorporação de recursos html. O grau mais alto para esta última interface aponta para o fato de que imagens e links são de extrema importância para o usuário de qualquer nível de conhecimento ou experiência. Isso porque nesse tipo de interface, o acesso aos conhecimentos ocorre de forma simples, sem a necessidade de muitas ações diferenciadas.

A pergunta 9 ("Você se sente capaz de consultar uma tabela de roteamento, executar um comando ping ou tracert sem utilizar Doroty?") teve uma aceitação total. Todos os usuários sem experiência que responderam a pergunta afirmaram se sentir capazes realizar as tarefas solicitadas. Os avaliadores que tiveram o teste monitorado foram solicitados a executar alguma das tarefas. Entre tais usuários, a maioria voltou a consultar Doroty através das perguntas feitas anteriormente e executou os comandos sem enfrentar problemas.

Em se tratando de performance geral e satisfação global, foi possível constatar que a avaliação do protótipo tende a ser mais alta por usuários com menos conhecimento e menos experiência. Esses usuários atribuíram uma nota alta para Doroty, enquanto que os usuários com mais conhecimento e experiência atribuíram notas mais baixas. As notas atribuídas ao protótipo podem ser vistas na tabela 5.3. Calculando a média final, o protótipo obteve como nota final 8,4. A única avaliação negativa obtida no processo, porém, acredita que Doroty pode ainda atingir um nível bastante positivo.

Vale salientar que os questionários de avaliação foram distribuídos por grupos de avaliadores em períodos distintos, assim, as notas mais baixas foram recebidas logo no início do processo, enquanto que as mais altas foram atribuídas após constantes constatações e modificações no protótipo, elevando seu nível de aceitação a cada teste realizado. Os aspectos que mais influenciaram a atribuição das notas por parte dos avaliadores foram: conteúdo teórico, o formato das respostas e o detalhamento das explicações. A dificuldade com linguagem natural e a interface obtiveram segundo lugar em influência.

O quesito impressões gerais (parte D) apontou a preferência pela interface padrão, porém quase que a totalidade dos usuários sugeriram a modificação da mesma a fim de aceitar recursos html nas respostas produzidas por Doroty. Da mesma forma, a conversa com o chatterbot foi considerada agradável e produtiva e as premissas apresentadas na pergunta 4 também tiveram um resultado bastante positivo.

5.2 Considerações Finais sobre os Resultados Alcançados

Em se tratando da aceitação de uma tecnologia cujo principal atrativo é o uso da linguagem natural e a conseqüente facilidade de interação com o usuário, foi observado (teste estático) que o uso da solução no ambiente acadêmico representa um enorme potencial no sentido de aumentar o interesse e curiosidade dos estudantes. Por imitar a realidade e ser de fácil uso, o bot permite que uma pessoa com menos experiência não se sinta constrangida em interagir com ele, como muitas vezes acontece em ambientes profissionais e acadêmicos com os chefes e professores, respectivamente. Esse resultado foi ao encontro de resultados de outros experimentos, nas mais diversas áreas, que utilizaram chatterbots para interação com usuários em seus campos de domínio (BICKMORE, 1999; RICKENBERG; REEVES, 2000). Os estudantes se sentiram confortáveis com Doroty e também motivados o suficiente para executar alguns experimentos sugeridos por ela.

O experimento descrito na subseção 5.1.2.1 gerou resultados favoráveis ao uso da ferramenta para a atividade fim que se destina. Os estudantes salientaram que o uso do chatterbot foi atrativo e intuitivo, colocando a visão de gerenciamento de redes em outra perspectiva. Dentre as principais características citadas pelos estudantes destacam-se: a possibilidade de consulta ao recurso em qualquer hora do dia e a possibilidade de sanar dúvidas de maneira rápida, sem a necessidade de uma consulta detalhada em bibliografia e referências muito técnicas. Ambas características podem ser consideradas muito importantes para os profissionais aos quais o trabalho é dirigido.

Os resultados do teste apresentados na seção 5.1.2.2 confirmam os resultados comentados e demonstram o amplo potencial de Doroty para treinamento de profissionais com pouca experiência. Embora existam características que devem ser constantemente refinadas, o protótipo apresenta uma grande variedade de recursos ao usuário, apresentados de uma maneira fácil. Além disso, permite fácil adaptação a outras tecnologias que podem agregar conhecimentos relevantes para a área. Uma das características que ainda deve ser trabalhada é a questão da linguagem natural. No início do processo de avaliação da etapa total do bot, este fator foi decisivo. Porém, a constante verificação dos registros de conversação pode aumentar a aceitação do sistema e a gama de possibilidade em termos de conversação. Além disso, a constante análise de tais registros também pode ajudar o engenheiro de conhecimento a verificar quais as dúvidas mais frequentes do usuário e isso pode ser de grande valia para outros em uma mesma situação, o que contribui para a melhoria do protótipo como um todo.

6 CONSIDERAÇÕES FINAIS

O gerenciamento de redes é uma tarefa complicada. Nos últimos anos muitas organizações fizeram tentativas no sentido de criar e definir novas ferramentas e paradigmas para simplificar a tarefa de gerenciamento (DERI, 1999). Com o avanço dos estudos de processamento de linguagem natural e de Inteligência Artificial, os chatterbots cada dia desenvolvem novas habilidades e são utilizados para os mais diversos fins. O presente trabalho salienta que a Inteligência Artificial tem buscado desenvolver diferentes meios de fazer com que um computador possa realizar tarefas de forma racional, não mais se prendendo a um conjunto pré-programado de instruções.

O presente trabalho apresentou Doroty, um chatterbot projetado e adaptado para ajudar profissionais menos treinados a adquirir conhecimentos de gerenciamento de redes utilizando o protocolo SNMP. A grande vantagem da utilização desse recurso se encontra no fato de que o usuário não necessita conhecimento prévio para usar a ferramenta e interagir com a rede na qual deve trabalhar.

Para isso, foi realizado um estudo sobre chatterbots, a fim de apresentar o leitor a tal tecnologia. Também foi realizado um estudo sobre o gerenciamento de redes e algumas ferramentas que utilizam IA e foram desenvolvidas para facilitar esse processo. Todo esse estudo serviu como base para a elaboração de um protótipo que fosse capaz de utilizar linguagem natural como forma de comunicação com o usuário, traduzindo os questionamentos levantados para ações efetivas em uma rede, buscando contextualizar suas explicações e fornecer ao usuário uma fonte rica em exemplos e informações teóricas sobre os conceitos e ações envolvidos no processo de gerenciamento.

É importante salientar que o objetivo principal do bot não é o de encontrar e resolver problemas em uma rede. O chatterbot apresentado deve ser usado como ponto de partida para ensinar um usuário quais os possíveis problemas que podem ocorrer em uma rede, bem como encorajá-lo a tomar atitudes que ajudem a resolver tal problema. A formação de profissionais no domínio de Redes de Computadores tem tradicionalmente utilizado abordagens centradas no professor induzindo os alunos a uma condição mais passiva em lugar de prepará-los para a pesquisa e aplicação de novos conhecimentos. As abordagens centradas nos alunos que estimulam habilidades de auto-aprendizado e pensamento crítico têm se mostrado mais apropriadas principalmente na área de tecnologia que está em constante aperfeiçoamento.

Conforme os estudos realizados, pode-se afirmar que o sistema contribui para seu público alvo, sendo capaz de levar informações sobre gerenciamento de redes a usuários sem nenhum conhecimento na área. Entretanto, a realização desse trabalho apresentou algumas dificuldades. A primeira delas se refere aos experimentos realizados para fins de avaliação. A avaliação de Doroty exigiu bastante tempo por parte dos avaliadores, além de interações constantes, à medida que o protótipo era incrementado. Esse fato causou um

certo abandono por parte de alguns avaliadores, que preencheram apenas alguns aspectos da avaliação, desconsiderando outros.

Outra dificuldade encontrada está no fato de que a construção da base de conhecimentos do chatterbot precisa ser bem planejada, uma vez que os conceitos apresentados devem apenas guiar o usuário sem influenciar no seu comportamento geral. Em se tratando de problemas, por exemplo, foi bastante difícil encontrar a maneira mais adequada de programar o protótipo para ajudar o usuário, uma vez que intenção era de não decidir por ele. Da mesma forma, o sistema DUMBO, utilizado como referência para a alimentação de problemas, apresenta uma interface que fornece dados bastante específicos sobre o problema. Muitos desses dados podem ser incorporados ao conhecimento de Doroty, uma vez que para isso, a sua capacidade de diálogo deveria ser expandida, através da adição de novas *tags* no chatterbot original.

As constantes avaliações realizadas durante o desenvolvimento do trabalho e a interação com outros desenvolvedores de sistemas chatterbots e também com usuários serviram como ponto de partida para uma reestruturação de todos os aspectos envolvidos no protótipo. Essa reestruturação permitiu que fossem integradas características de sistemas especialistas, através da participação dos especialistas e do engenheiro de conhecimento para a construção do protótipo. também foram incorporadas características de sistemas baseados no paradigma PBL, possibilitando um processo de aprendizado mais colaborativo.

6.1 Trabalhos Futuros e Possíveis Expansões

Alguns trabalhos que podem agregar valor ao chatterbot desenvolvido englobam:

- Transformação do sistema em uma ferramenta completa de auxílio ao gerenciamento de redes. A integração de um módulo responsável por realizar a verificação e correlação de alarmes transformaria a ferramenta de treinamento em um assistente virtual completo para operadores de rede, sem a necessidade de conhecimento teórico prévio. Porém, salienta-se que o objetivo da ferramenta criada não é o de substituir as ferramentas existentes, mas sim, o de prever uma ferramenta gratuita para o aprendizado de gerenciamento de redes, encorajando usuários a utilizar tantos recursos quanto tiverem disponíveis para esta tarefa, até que estejam preparados o suficiente para decidir qual a melhor forma de gerenciar uma rede;
- Refinamento das funcionalidades básicas de Doroty, através da criação de um módulo automatizado para a alimentação e expansão da base de conhecimentos da mesma, sem que o usuário precise efetivamente trabalhar com as tabelas do banco de dados. Atualmente, o chatterbot Doroty pode ser estendido através da adição de arquivos AIML à sua base de conhecimento, bem como de objetos ao banco de dados. A adição de arquivos AIML pode ser feita a qualquer momento, já que o software realiza verificação constante de novos arquivos AIML, mesmo quando o bot está ativo;
- Desenvolvimento de um módulo para que o usuário possa avaliar a qualidade de cada resposta e sugerir novos recursos, a fim de tornar o conteúdo existente na base de conhecimentos e no banco de informações mais adaptado para cada perfil de usuário;

- Desenvolvimento de um módulo para criação automática de AIML, baseado em extração de conhecimento de textos, livros e artigos da área;
- Adaptação de outras tecnologias de Inteligência artificial apresentadas no capítulo 3, como, por exemplo, a utilização de agentes móveis para descoberta da rede quando da instalação do sistema e utilização de agentes de interface para observação do usuário a fim de tornar mais atrativo e completo o processo de aprendizado.

Além disso, outra futura revisão do sistema poderia incorporar trabalhos já desenvolvidos na área e ferramentas de geração de gráfico de forma automática. Um sistema como DUMBO poderia ter todos os casos incorporados à base de conhecimento através da adaptação de sua interface de entrada de características de casos. Uma forma de adaptação envolveria o refinamento das possibilidades de conversação do chatterbot, expandindo sua memória de forma a armazenar as entradas e, posteriormente acionar o sistema DUMBO para processar as informações armazenadas.

REFERÊNCIAS

- ADAMS, T.; WENDT, C. Defining and training administrators for distributed networking. In: ACM SIGUCCS - CONFERENCE ON USER SERVICES, 21., 1993, New York, USA. **Proceedings...** New York: ACM Press, 1993. p.199–203.
- AGENTLAND. **Cybelle**. Disponível em: <<http://www.agentland.com/>>. Acesso em: setembro 2003.
- ANGELI, A. D.; JOHNSON, G.; COVENTRY, L. The unfriendly user: exploring social reactions to chatterbots. In: INTERNATIONAL CONFERENCE ON AFFECTIVE HUMAN FACTORS DESIGN, 2001, London. **Proceedings...** [S.l.]: Asean Academic Press, 2001. p.467–474.
- ANGELI, A. D.; LYNCH, P.; JOHNSON, G. Personifying the e-market: a framework for social agents. In: IFIP TC.13 CONFERENCE ON HUMAN-COMPUTER INTERACTION, 8., 2001, Tokyo, Japan. **Proceedings...** [S.l.: s.n.], 2001.
- APPLEBY, S.; STEWARD, S. Software Agents for Control. In: COCHRANE, P.; MEATLEY, D. (Ed.). **Modelling Future Telecommunication Systems**. [S.l.]: Chapman & Hall, 1994.
- ARNOLD, W. R.; BOWIE, J. S. **Artificial Intelligence: a personal commonsense journey**. Englewood Cliffs, New Jersey: Prentice-Hall, 1986.
- BICKMORE, T. W. **Social Intelligence in Conversational Computer Agents**. Cambridge, USA: Massachusetts Institute of Technology, 1999. ProSeminar Conceptual Analysis of Thesis Area.
- BIESZCAD, A.; PAGUREK, B.; WHITE, T. Mobile Agents for Network Management. **IEEE Communications Surveys**, [S.l.], v.1, n.1, Sept. 1998.
- BONAR, J.; BLAISE, L. Communicating with High Level Plans. In: WORKSHOP ON ARCHITECTURES FOR INTELLIGENT INTERFACES: ELEMENTS AND PROTOTYPES, 1991, New York, USA. **Proceedings...** New York: ACM Press, 1991. p.129–156. (Frontier Series).
- CASE, J.; FEDOR, M.; SCHOFFSTALL, M.; DAVIN, J. **A Simple Network Management Protocol (SNMP): RFC 1157**. [S.l.]: IETF, 1990.
- CASE, J.; ROSE, M.; WALDBUSSER, M. **Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2): RFC 1442**. [S.l.]: IETF, 1993.

CASSEL, J. Genderizing HCI. In: JACKO, J.; SEARS, A. (Ed.). **The Handbook of Human-Computer Interaction**. Mahwah, NJ: Lawrence Erlbaum Associates, 2002. p.402–411.

CHEIKHROUHOU, M. M.; CONTI, P.; LABETOULLE, J. Intelligent agents in network management a state-of-the-art. **Networking and Information Systems**, [S.l.], v.1, n.1, p.1–29, 1998.

COLBY, K. Artificial Paranoia. **Artificial Intelligence**, [S.l.], v.2, 1971.

DERI, L. H. Desktop versus web-based network management. **International Journal of Network Management**, New York, NY, USA, v.9, n.6, p.371–378, Nov./Dec. 1999.

ESFANDIARI, B.; DEFLANDRE, G.; QUINQUETON, D. An interface agent for network supervision. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1996, Budapest, Hungary. **Proceedings...** [S.l.: s.n.], 1996.

FISCHLER, M. A.; MCCORDUCK, P. **The Eye, the Brain, and the Computer**. [S.l.]: Addison-Wesley, 1987.

FRANKLIN, S.; GRAESSER, A. **Is it an Agent, or Just a Program?: a taxonomy for autonomous agents**. Disponível em: <<http://www.msci.memphis.edu/franklin/AgentProg.html>>. Acesso em: jul. 2003.

FULLER, W. Network management using expert diagnostics. **International Journal of Network Management**, New York, USA, v.9, n.4, p.199–208, July/Aug. 1999.

GALVÃO, A. M. **Persona-AIML: uma arquitetura para desenvolver chatterbots com personalidade**. 2003. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Pernambuco, Recife, Pernambuco.

GALVÃO, A. M.; BARROS, F. A.; NEVES, A. M. M.; RAMALHO, G. L. Persona-AIML: an architecture for developing chatterbots with personality. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 3., 2004, New York, USA. **Proceedings...** New York: ACM Press, 2004. p.1266–1267.

GUZELDERE, G.; FRANCHI, S. **Dialogues with Colorful Personalities of Early AI**. Disponível em: <<http://www.stanford.edu/group/SHR/4-2/text/dialogues.html>>. Acesso em: jan. 2004.

JACKSON, P. **Introduction to Expert Systems**. Wokingham: Addison-Wesley, 1986.

JENKINS, R. D. Why web-based network monitoring? Leveraging the platform. **International Journal of Network Management**, [S.l.], v.9, n.3, p.175–183, May/June 1999.

JONES, D. **The Learning Tree: problem based learning**. Disponível em: <<http://edweb.sdsu.edu/clrit/learningtree/Ltree.html>>. Acesso em: jun. 2003.

L'ABBATE, M.; THIEL, U. Helping conversational agents to find informative responses: query expansion methods for chatterbots. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 1., 2002, Bologna, Itália. **Proceedings...** New York: ACM Press, 2002. p.796–797.

LAMBERT, M. **A Model for Common Operational Statistics: RFC 1857**. [S.l.]: IETF, 1995.

LEONHARDT, M. D.; CASTRO, D. D.; DUTRA, R. L. S.; TAROUCO, L. M. R. Elektra: um chatterbot para uso em ambiente educacional. **RENOTE: Revista Novas Tecnologias na Educação**, Porto Alegre, v.1, n.2, set. 2003. Disponível em: <<http://www.cinted.ufrgs.br/renote/set2003/artigos/elektra-chatterbot.pdf>> Acesso em: abr. 2004.

LEONHARDT, M. D.; CASTRO, D. D.; TAROUCO, L. M. R. Elektra: inteligência artificial na educação a distância de jovens e adultos. In: CONGRESSO DE EDUCAÇÃO A DISTÂNCIA MERCOSUL, 2003, Florianópolis, SC - Brasil. **Anais...** [S.l.: s.n.], 2003. p.165–170.

LEONHARDT, M. D.; NEISSE, R.; TAROUCO, L. M. R. MEARA: um chatterbot temático para uso em ambiente educacional. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 14., 2003, Rio de Janeiro, Brasil. **Anais...** [S.l.: s.n.], 2003. p.85–92.

LIMA, M. M. A. E. Introdução a Gerenciamento de Redes TCP/IP. **News Generation**, [S.l.], v.1, n.3, ago. 1997.

LIMA, V. L. S. Processamento da linguagem natural: premissas e desafios. In: ESCOLA REGIONAL DE INFORMÁTICA, 3., 1996, Canoas. **Anais...** [S.l.: s.n.], 1996.

LOEBNER, H. **In Response**. Disponível em: <<http://loebner.net/Prizef/In-response.html>>. Acesso em: jun. 2004.

MAES, P.; KOZIEROK, R. Learning Interface Agents. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AAAI, 11., 1993. **Proceedings...** [S.l.]: MIT Press: AAAI Press, 1993.

MAGEDANZ, T.; ROTHERMEL, K.; KRAUSE, S. Intelligent Agents: an emerging technology for next generation telecommunications? In: INFOCOM, 1996, USA. **Proceedings...** [S.l.]: IEEE, 1996. p.464–472.

MAULDIN, M. Chatterbots, tinymuds, and the turing test: entering the loebner prize competition. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 12., 1994. **Proceedings...** [S.l.: s.n.], 1994.

MCCLOGHRIE, K.; ROSE, M. **Management Information Base for Network Management of TCP/IP-based internets (MIB-II)**: RFC 1213. [S.l.]: IETF, 1991.

MEDINA, R. D. **SAFO Sistema Agregador de Ferramentas de Operação de Rede**. 1994. Trabalho Individual (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

MEDINA, R. D. **SAFO Sistema Agregador de Ferramentas de Operação de Rede**. 1996. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

MEDINA, R. D.; TAROUCO, L. M. R. Tecnologias Aplicadas no Ensino de Redes de Computadores: um protótipo de laboratório virtual para facilitar a aprendizagem significativa. In: CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACION - CACIC, 2003, La Plata, Argentina: UNLP/Facultad de Informática. **Anais...** [S.l.: s.n.], 2003. p.256–264.

MELCHIORS, C. **Raciocínio Baseado em Casos Aplicado ao Gerenciamento de Falhas em Redes de Computadores**. 1999. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

MELCHIORS, C.; TAROUCO, L. M. R. Fault management in computer networks using case-based reasoning: dumbos system. In: INTERNATIONAL CONFERENCE ON CASE-BASED REASONING, 1999, Seon Monastery, Germany. **Proceedings...** Berlin: Springer, 1999. p.510–524.

MULLER, J. P. **The Design of Intelligent Agents: a layered approach**. Secaucus, NJ, USA: New York: Springer-Verlag, 1996. (Lecture Notes in Artificial Intelligence, v.1177).

MULLER, N. J. Improving network operations with intelligent agents. **International Journal of Network Management**, [S.l.], v.7, n.3, p.116–126, 1997.

NIKOLOPOULOS, C. **Expert systems**. New York, USA: Marcel Dekker, Inc., 1997.

NUNES, C. **Um Discriminador Inteligente de Eventos de Rede para o Ambiente CINEMA**. 1997. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

OLIVEIRA, R.; LABETOULLE, J. Intelligent Agents: a way to reduce the gap between applications and networks. In: INTERNATIONAL WORKSHOP ON FACTORY COMMUNICATIONS SYSTEMS, 1., 1995, Leysin, Switzerland. **Proceedings...** [S.l.: s.n.], 1995. p.81–90.

PINTO, A. C. **SEGREDE: mecanismos para gerência de segurança em redes**. 1995. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

PRAS, A. **Network Management Architectures**. 1995. Tese (Doutorado em Ciência da Computação) — University of Twente, Netherlands.

PRIMO, A. F. T.; COELHO, L. R. A chatterbot Cybelle: experiência pioneira no Brasil. **Mídia, Textos e Contextos**, Porto Alegre, p.259–276, 2001.

PRIMO, A. T.; COELHO, L. R. Júnior, um chatterbot para educação a distância. In: CONGRESSO IBERO-AMERICANO DE INFORMÁTICA NA EDUCAÇÃO - RIBIE, 2000. **Anais...** [S.l.: s.n.], 2000.

REDDY, R. To Dream the Possible Dream (Turing Award Lecture). **Communications of the ACM**, [S.l.], v.39, n.5, p.105–112, 1996.

RICKENBERG, R.; REEVES, B. The Effects of Animated Characters on Anxiety, Task performance, and Evaluations of User Interfaces. **CHI Letters**, [S.l.], v.2, n.1, p.49–56, 2000.

- RINGATE, T. **The AIML Reference Manual**. Disponível em: <<http://www.alicebot.org/documentation/aiml-reference.html>>. Acesso em: ago. 2003.
- RUSSEL, R. S. **Honors Project Report Artificial Intelligence and Computer Science**. [S.l.]: University of Edinburg, 2002.
- RUSSEL, S.; NORVIG, P. **Artificial Intelligence: a modern approach**. Englewood Cliffs: Prentice-Hall, 1995.
- SAYGIN, A. P.; CICEKLIB, I. Pragmatics in human-computer conversations. **Journal of Pragmatics**, [S.l.], n.34, p.227–258, 2002.
- SCOTT, A. C. **A Practical Guide to Knowledge Acquisition**. Massachusetts: Addison-Wesley, 1991.
- SGANDERLA, R. B. **BonoBOT: um chatterbot para interação como usuários em um sistema tutor inteligente**. 2002. Trabalho de Conclusão de Curso — La Salle, Canoas.
- SOARES, L. F. G. **Redes de Computadores: das lans, mans e wans as redes atm**. 7.ed. Rio de Janeiro, Brasil: Campus, 1995.
- SOUZA, C. S. de; LEITE, J. C.; PRATES, R. O.; BARBOSA, S. D. J. Projeto de Interfaces de Usuário - Perspectivas Cognitivas e Semióticas. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 19., 1999, Rio de Janeiro, RJ. **Anais...** [S.l.: s.n.], 1999. v.2, p.425–476.
- STALLINGS, W. **SNMP, SNMPv2 and RMON: practical network management**. 2nd ed. Massachussets, USA: Addison Wasley Longman Inc., 1996.
- TANENBAUM, A. S. **Redes de Computadores**. 3.ed. Rio de Janeiro, Brasil: Campus, 1997.
- TAROUCO, L. M. R. **inteligência Artificial Aplicada ao Gerenciamento de Redes de Computadores**. 1990. Tese (Doutorado em Ciência da Computação) — Escola Politécnica da USP, São Paulo, Brasil.
- TAROUCO, L. M. R.; DUTRA, R. L. S.; LEONHARDT, M. D. Ambiente de Aprendizagem Colaborativo Orientado a Problemas para Gerência de Redes. In: CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACION, 2004, Buenos Aires, Argentina. **Proceedings...** [S.l.: s.n.], 2004. p.1156–1165.
- TAYLOR, A. **The AIML Mini Reference and Primer**. Disponível em: <<http://hippie.alicebot.com/ataylor/index.html>>. Acesso em: jul. 2003.
- TEIXEIRA, S.; MENEZES, C. S. de. Facilitando o uso de Ambientes Virtuais através de Agentes de Conversação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCACÃO, 14., 2003, Rio de Janeiro, Brasil. **Anais...** [S.l.: s.n.], 2003. p.483–492.
- TERPLAN, K. **Communication Networks Management**. New Jersey, USA: Prentice-Hall, 1992.
- TSAI, C.-W.; CHANG, R.-S. SNMP through WWW. **International Journal of Network Management**, New York, NY, USA, v.8, n.2, p.104–119, Mar./April 1998.

TURING, A. Computer machinery and Intelligence. **Mind**, [S.l.], v.59, p.433–460, 1950.

WALLACE, R. **ALICE - Artificial Linguistic Internet Computer Entity - The A.L.I.C.E A.I. Foundation**. 1995. Disponível em: <<http://alicebot.org>>. Acesso em: jul. 2003.

WALLACE, R. **AIML Overview**. Disponível em <<http://www.pandorabots.com/pandora/pics/wallaceaimltutorial.html>> Acesso em: jul. 2003.

WALLACE, R. S. **AIML Pattern Matching Simplified**. 2001. Disponível em <<http://www.alicebot.org/documentation/>> Acesso em: jul. 2003.

WEIZEMBAUM, J. Eliza - A Computer Program For the Study of Natural Communication Between Man and Machine. **Communications of the ACM**, [S.l.], v.9, n.1, p.36–45, January 1966.

WILKS, Y.; CATIZONE, R. **Human Computer Conversations**. New York, USA: University of Sheffield Dekker, 2000.

WOOLDRIDGE, M. Intelligent Agents: theory and practice. **The knowledge engineering review**, [S.l.], v.2, n.10, p.115–152, 1995.

WOOLDRIDGE, M. Agent-Based software engineering. **IEEE Proceedings on Software Engineering**, [S.l.], v.1, n.144, p.26–37, Feb. 1997.

APÊNDICE A QUESTIONÁRIO DE AVALIAÇÃO

Prezado(a) avaliador(a),

Através deste questionário, você estará avaliando Doroty, um protótipo desenvolvido durante o meu curso de mestrado em Ciência da Computação na Universidade federal do Rio Grande do Sul. Sua tarefa consiste em utilizar o sistema e avaliar os aspectos positivos e negativos do mesmo. Por isso estamos lhe disponibilizando o questionário e solicitamos que nos responda, pois contamos com suas informações para avaliar se a ferramenta desenvolvida está condizente com os objetivos desejados para a mesma.

Os dados coletados neste questionário destinam-se apenas a avaliar as expectativas dos usuários da ferramenta, bem como a melhorar a mesma. Lembre-se que suas conversas com Doroty serão posteriormente lidas em sua integridade, mas os dados coletados não serão repassados a terceiros, bem como suas informações pessoais.

Instruções e procedimentos: O procedimento de avaliação, nesta etapa, é constituído de duas partes: No primeiro momento, você deverá seguir as instruções constantes neste documento, buscando avaliar cada um dos aspectos solicitados. Em um segundo momento, você poderá utilizar Doroty livremente. Esta segunda parte da avaliação é importante pois permitirá que a base de conhecimentos de Doroty seja melhorada, através da análise dos registros de conversação para avaliação da maneira como diferentes usuários (com diferentes níveis de conhecimento sobre o assunto) interagem com o chatterbot. Além disso, esse segundo momento servirá para que você possa reforçar suas impressões iniciais e sugerir melhorias baseadas na sua observação pessoal.

Observações: Antes de qualquer coisa, tenha em mente que Doroty é apenas um protótipo. Assim:

1. Durante sua avaliação pessoal considere a hipótese de reformular sua questão quando alguma resposta não for encontrada. Isso será de grande valia para a próxima etapa de avaliação do protótipo, que abrangerá, entre outros aspectos, refinamento da capacidade do bot de "entender" linguagem natural;
2. Doroty utiliza o protocolo SNMP para monitorar a rede em que atua (através da MIB2). Dessa forma, em sua fase atual, ela é capaz de consultar informações sobre aproximadamente 15% dos objetos presentes nessa MIB. Os objetos selecionados para sua fase de protótipo foram escolhidos com o objetivo de demonstrar que sua atuação pode abranger todos os grupos da MIB, além de demonstrar que outros objetos podem ser incluídos, bem como as consultas e base de conhecimento adap-

tadas para abranger os mais diversos problemas que um administrador de rede pode enfrentar;

3. O conteúdo das respostas de Doroty foi retirado de livros e trabalhos acadêmicos dentro de sua área de abrangência.

Breve resumo e contextualização do trabalho: O gerenciamento de redes exige dos administradores a disponibilidade de uma grande quantidade de informações sobre os seus equipamentos, as tecnologias envolvidas e os problemas associados a elas. Nesse cenário, administradores de redes devem, cada vez mais, aprofundar o seu conhecimento através de constante treinamento, até que estejam aptos a administrar uma rede de maneira mais eficiente e confiável.

Alguns estudos têm sido feitos buscando integrar tecnologias de Inteligência Artificial na área de gerenciamento de redes. Abordagens utilizando sistemas multiagentes, agentes de interface e raciocínio baseado em casos já foram utilizadas com o objetivo de facilitar a tarefa de gerenciamento de rede aos olhos do usuário e de empregar novos paradigmas de programação para a tarefa, como no caso dos sistemas multiagentes. Os chatterbots representam um grande potencial para a tarefa de treinamento e gerenciamento de redes já que utilizam linguagem natural e são capazes de ser facilmente integrados em ambientes mais complexos.

O principal objetivo deste trabalho é investigar o uso de chatterbots como uma ferramenta de gerenciamento utilizada por administradores menos treinados. O trabalho envolveu a adaptação do chatterbot ALICE (<http://www.alicebot.org>) para permitir o treinamento e a gerência de redes através da inclusão de módulos que permitem a monitoração de equipamentos de uma rede (através do protocolo SNMP) e módulos que permitam consultar e armazenar histórico de informações da mesma. Desta forma, a grande contribuição da arquitetura proposta é a de prover uma comunicação mais efetiva entre o administrador menos experiente e a rede, através do papel do chatterbot assistente, que interpreta os dados coletados e expõe os conceitos envolvidos no processo de gerenciamento.

Desde já, gostaria de agradecer por sua colaboração e disponibilidade.

Michelle Denise Leonhardt
Mestranda do Instituto de Informática
PPGC - UFRGS
E-mail: mdleonhardt@inf.ufrgs.br

Lembre-se: As perguntas devem ser digitadas no campo Pergunta. Utilize o botão Limpar para deixar o campo pergunta em branco e o botão Enviar para enviar uma pergunta. Para enviar uma pergunta, você também pode digitar a tecla enter.

(Grupo 1) - Características do entrevistado:

Este grupo se destina a perguntas sobre suas características pessoais, importantes para avaliação estatística do grau de satisfação de Doroty segundo características pessoais de cada avaliador (por exemplo: nível de conhecimento e experiência na área, grau de interesse pelo assunto, nível de escolaridade, idade, localização geográfica, entre outros).

1. Nome:

2. Idade:

3. Cidade e estado onde reside *:

4. Cidade e estado de nascimento *:

5. Grau de escolaridade (finalizado ou em andamento):

Sabe Ler e escrever sem possuir grau de ensino

Ensino Fundamental

Ensino Médio

Ensino Médio Técnico

Ensino Superior

Especialização

Mestrado, doutorado ou pós-doutorado

Área ou curso (para quarta opção em diante):

6. Possui experiência prática na área de gerência de redes de computadores? (nesta pergunta deve responder sim quem já trabalhou ou trabalha profissionalmente na área ou que realiza estudos práticos sobre o assunto, ou seja, realiza experimentos práticos)

Sim

Não

7. Tem acesso à Internet? Onde?

casa

trabalho

em qualquer lugar pois possuo computador portátil

outros locais

8. Costuma recorrer à Internet para consulta de informação/pesquisa sobre assuntos que despertem sua curiosidade?

Sim

Não

(Grupo 2) - Avaliação geral.

Este grupo se destina a avaliar, de forma geral, o protótipo Doroty, através de instruções pré-definidas.

Parte A - Utilização de Doroty:

Serão enumerados alguns assuntos e tópicos. Para cada um, utilize Doroty para descobrir informações sobre o assunto. (Em alguns casos, mais de uma pergunta pode ser necessária.) Em seguida, avalie os aspectos solicitados utilizando a escala abaixo: (5) Plenamente Satisfeito

- (4) Parcialmente Satisfeito
- (3) Parcialmente Insatisfeito
- (2) Plenamente Insatisfeito
- (1) Indiferente, Neutro ou a situação não se aplica.

Se desejar, utilize o espaço de comentários para expressar alguma opinião.

1. Tente descobrir como funciona o protocolo SNMP e o que é uma MIB.

- Encontrei a(s) informação(ões) solicitada(s)?
- Esperei muito tempo pela resposta?
- Achei que a resposta estava muito longa?
- O conteúdo da resposta foi suficiente para que eu absorvesse a informação?
- Comentários:

2. Descubra como está organizada uma MIB.

- Encontrei a(s) informação(ões) solicitada(s)?
- Esperei muito tempo pela resposta?
- Achei que a resposta estava muito longa?
- O conteúdo da resposta foi suficiente para que eu absorvesse a informação?
- Comentários:

3. Descubra o que é roteamento e o que são protocolos de roteamento.

- Encontrei a(s) informação(ões) solicitada(s)?
- Esperei muito tempo pela resposta?
- Achei que a resposta estava muito longa?
- O conteúdo da resposta foi suficiente para que eu absorvesse a informação?
- Comentários:

4. Descubra o que é um comando ping ou um comando tracert.

- Encontrei a(s) informação(ões) solicitada(s)?
- Esperei muito tempo pela resposta?
- Achei que a resposta estava muito longa?
- O conteúdo da resposta foi suficiente para que eu absorvesse a informação?
- Comentários:

5. Execute um comando ping ou tracert através de Doroty.

- Encontrei a(s) informação(ões) solicitada(s)?
- Esperei muito tempo pela resposta?
- O comando foi executado?
- Doroty apresentou alguma informação extra para ajudar na absorção do conteúdo?
- Comentários:

6. Descubra o máximo possível de informações sobre o protocolo ICMP. Então, peça para que Doroty consulte:

Observação: você pode e deve explorar o conhecimento de Doroty, fazendo perguntas sobre assuntos que não estão claros sobre o conteúdo. Isso será de grande auxílio para enriquecimento de sua base de conhecimento, mesmo que Doroty não saiba responder todas as suas dúvidas.

6.1 Que informações você conseguiu descobrir sobre o protocolo?

6.2 Quantas mensagens ICMP foram recebidas no total ou qual o número total de mensagens do tipo "Destination Unreachable" recebidas ou enviadas. (ou mensagens do tipo "Time Exceeded", por exemplo)

- Encontrei a(s) informação(ões) solicitada(s)
- Esperei muito tempo pela resposta?
- O comando foi executado?
- Doroty apresentou alguma informação extra para ajudar na absorção do conteúdo?
- Comentários (Qual foi o comportamento apresentado por Doroty?):

6.3 Descubra quantas mensagens ICMP foram recebidas no total ou qual o número total de mensagens do tipo "Destination Unreachable" recebidas ou enviadas por um equipamento específico? ** (Note que você deve digitar o endereço IP do equipamento que deseja consultar)

- Encontrei a(s) informação(ões) solicitada(s)
- Esperei muito tempo pela resposta?
- O comando foi executado?
- Doroty apresentou alguma informação extra para ajudar na absorção do conteúdo?
- Comentários:

Observação: Para as próximas atividades, se você não conseguir descobrir a informação desejada, escreva qual foi(ram) as perguntas feitas para que possamos ensinar Doroty a entender o seu modo de se expressar. (Você pode pensar que faltam informações em algumas perguntas para que as mesmas façam sentido. Não se preocupe, este é o objetivo!)

6.4 Descubra quantas mensagens ICMP foram recebidas no total ou qual o número total de mensagens do tipo "Destination Unreachable" recebidas ou enviadas por um equipamento específico num intervalo de tempo definido (últimos 4 minutos, últimas 3 horas, últimos 2 dias, por exemplo).

- Encontrei a(s) informação(ões) solicitada(s)
- Esperei muito tempo pela resposta?
- O comando foi executado?
- Doroty apresentou alguma informação extra para ajudar na absorção do conteúdo?
- Comentários:

6.5 Descubra qual o número médio (ou número máximo e mínimo) de mensagens ICMP que foram recebidas ou enviadas por um equipamento específico num intervalo de tempo definido (últimos 4 minutos, últimas 3 horas, últimos 2 dias, por exemplo).

- Encontrei a(s) informação(ões) solicitada(s)
- Esperei muito tempo pela resposta?
- O comando foi executado?
- Doroty apresentou alguma informação extra para ajudar na absorção do conteúdo?
- Comentários:

6.6 Tente descobrir qual o número total de mensagens ICMP recebidas num intervalo de tempo definido. (últimos 25 minutos, últimas 25 horas, últimos 5 dias, por exemplo).

- Encontrei a(s) informação(ões) solicitada(s)
- Esperei muito tempo pela resposta?
- O comando foi executado?

- Doroty apresentou alguma informação extra para ajudar na absorção do conteúdo?
- Comentários:

7. Descubra para que serve o objeto sysDescr?

- Encontrei a(s) informação(ões) solicitada(s)
- Esperei muito tempo pela resposta?
- Achei que a resposta estava muito longa?
- O conteúdo da resposta foi suficiente para que eu absorvesse a informação?
- Comentários:

8. Obtenha informações sobre o que é uma tabela de roteamento e qual a tabela de roteamento de um equipamento específico.

- Encontrei a(s) informação(ões) solicitada(s)
- Esperei muito tempo pela resposta?
- Achei que a resposta estava muito longa?
- O conteúdo da resposta foi suficiente para que eu absorvesse a informação?
- Comentários:

9. (Apenas para pessoas sem experiência na área) Você se sente capaz de consultar uma tabela de roteamento, executar um comando ping ou tracert sem utilizar Doroty?

- Sim
- Não

10. Se você fosse um administrador sem experiência, consideraria utilizar Doroty para tirar dúvidas e aprender sobre conceitos e ferramentas sobre gerenciamento de redes?

- Sim

Não

Parte B - Utilização Pessoal

Esta parte está aberta para que você utilize Doroty livremente, questionando-a a fim de avaliar seu desempenho nos aspectos que ainda julgar necessário.

Observação: Alguns assuntos que já estão na base de conhecimento de Doroty: Informações sobre um equipamento: localização, contato, tempo ativo do mesmo. Protocolo ICMP, tipos de mensagens, RFC que define o protocolo, entre outros. Número máximo de conexões TCP em uma máquina. Número de solicitações ICMP feitas em uma máquina, maior datagrama MTU que pode trafegar em uma rede, estado desejado das interfaces de uma rede, lista de interfaces de uma rede, entre outros.

Comentários:

Parte C - Performance Geral e Satisfação Global:

1. Considerando toda a sua experiência com Doroty, qual é a nota geral que você atribui ao protótipo? Use a escala de 10 pontos onde 1 significa "muito insatisfeito" e 10 "muito satisfeito". (Note que as respostas de 1 a 5 correspondem a uma avaliação negativa e de 6 a 10 a uma avaliação positiva)

Em caso de avaliação negativa:

A. Você acredita que Doroty, quando terminada sua fase de protótipo e testes, pode atingir um nível bastante positivo em relação ao avaliado agora? (Note que essa pergunta se refere as suas expectativas de melhora através da inclusão de novos módulos de conversação e/ou avaliação de registros de conversação semanais com o objetivo de tornar Doroty mais completa.)

Sim

Sim, mas não estou muito confiante quanto a isso.

Não

2. Enumere de 1 a 4, de acordo com o grau de importância, os aspectos que influenciaram na sua decisão sobre a nota geral atribuída a Doroty.

(1). Nada importante

(2). Alguma importância

- (3). Importante
- (4). Muita importância

- Interface
- Conteúdo teórico
- Tempo para resposta
- Dificuldade com linguagem natural
- Formato de Resposta
- Detalhamento das explicações
- Outro. Qual?

Parte D - Impressões Gerais

1. Qual a interface que você acha que possibilita um melhor entendimento do conteúdo?***.
 - Interface padrão (que apresenta caixa de texto com scroll)
 - Interface secundária (que não apresenta caixa de texto com scroll)

2. Você considerou a conversa com um chatterbot agradável?
 - Sim
 - Não

3. Sentiu-se confortável para utilizar o Doroty?
 - Sim
 - Não

4. Marque as opções que julga estarem de acordo com sua opinião. (ainda que não totalmente)
 - Acredito que a utilização de Doroty pode facilitar o processo de aprendizado de um administrador de rede sem experiência.
 - Acredito que a utilização de linguagem natural pode ser muito útil em ferramentas na área de redes de computadores.
 - Acredito que Doroty possa ser adaptada para se tornar uma ferramenta completa de gerenciamento, através do desenvolvimento de novos módulos ou integração com ferramentas já existentes na área de gerenciamento de redes.

(Grupo 3) - Sugestões:

Você tem algum comentário final? Alguma sugestão? Este espaço é dedicado a isso.

Legenda:

* Perguntas sobre cidade e estado de residência e nascimento são importantes para avaliação da influência de individualidades no modo de se expressar de pessoas de diferentes localizações geográficas.

** Caso a lista dos equipamentos monitorados por Doroty não tenha sido fornecida na pergunta anterior, utilize o endereço 143.54.47.240 ou 143.54.47.250 para realizar esta pergunta.

*** O responsável pela aplicação do seu teste pode lhe fornecer informações sobre a utilização da interface não padrão. Caso não esteja disponível, entre em contato por e-mail em mdleonhardt@inf.ufrgs.br

APÊNDICE B CONHECIMENTO - EXEMPLO DE CATEGORIAS

Este anexo apresenta alguns trechos de categorias de conhecimento presentes no "cérebro" de Doroty. O objetivo é ilustrar alguns dos recursos da linguagem que foram utilizados, explicando sua funcionalidade (algumas vezes através de diálogos retirados dos registros de conversação da mesma).

B2.1 Substituições

Com o objetivo de tornar mais flexível o diálogo, Doroty realiza substituições antes de aplicar o algoritmo de busca de padrões na base de conhecimento. Essas substituições são definidas em um arquivo e podem ser modificadas a qualquer hora.

```
...
<substitute find=" tb " replace=" tambem "/>
<substitute find=" tbem " replace=" tambem "/>
<substitute find=" vc " replace=" voce "/>
<substitute find=" pq " replace=" porque "/>
<substitute find=" roteametno " replace=" roteamento "/>
<substitute find=" q " replace=" que "/>
<substitute find=" tab " replace=" tabela "/>
...
```

B2.2 Memória

Através de predicados, que podem ser definidos pelo programador, Doroty é capaz de armazenar algumas variáveis em sua memória. Os predicados são definidos através de um nome, um valor padrão e um valor de retorno.

```
<predicates>
  <predicate name="name" default="colega" set-return="value"/>
  <predicate name="endereco" default="127.0.0.1" set-return="value"/>
  <predicate name="assunto" default="gerenciamento" set-return="value"/>
  <predicate name="tempo" default="0" set-return="value"/>
  <predicate name="situacao" default="medio" set-return="value"/>
</predicates>
```

No diálogo transcrito abaixo, o exemplo do uso de memória é ilustrado. O usuário começa a fazer perguntas sobre topologias de rede. Radicalmente, ele muda de assunto, apenas dizendo oi para Doroty. Quando ele pergunta sobre o que estavam falando, Doroty

usa os predicados de memória para responder que estavam falando sobre a topologia de barramento.

Usuário: *"me fale sobre topologias de rede"*

Doroty: *"A topologia de uma rede de comunicação, refere-se a forma como os enlaces físicos existentes e os nós de uma comutação estão organizados, determinando caminhos físicos existentes e utilizáveis entre quaisquer pares de estações conectadas a essa rede. A topologia de uma rede muitas vezes caracteriza o seu tipo, eficiência e velocidade."*

Usuário: *"quais são as possíveis topologias de uma rede?"*

Doroty: *"Algumas das possíveis topologias de uma rede são: Mesh, estrela, barramento, anel, árvore e topologias mistas. Eu ficaria feliz em falar mais sobre cada uma das topologias possíveis em uma rede, basta que você me diga sobre qual delas deseja saber mais, certo colega?"*

Usuário: *"desejo saber mais sobre a topologia em anel"*

Doroty: *"Na topologia de Anel, o barramento toma a forma de um anel, com ligações unidirecionais ponto a ponto. A mensagem é repetida de estação para estação até retornar a estação de origem, sendo então retirada do anel. Como o sinal é recebido por um circuito e reproduzido por outro há a regeneração do sinal no meio de comunicação; entretanto há também a inserção de um atraso mínimo de 1 bit por estação. O tráfego passa por todas as estações do anel, sendo que somente a estação destino interpreta a mensagem. É de fácil expansão, obtida através da ligação de módulos que implementam anéis independentes e que tornam-se um grande anel quando conectados. Pode ter sua confiabilidade incrementada pela adoção de dispositivos que realizam o bypass da estação no anel em caso de falha nos circuitos de conexão da mesma."*

Usuário: *"e a topologia barramento?"*

Doroty: *"Na topologia de Barramento, as estações são conectadas através de um cabo de cobre (coaxial ou par trançado), com difusão da informação para todos os nós. É necessária a adoção de um método de acesso para as estações em rede compartilharem o meio de comunicação, evitando colisões. É de fácil expansão mas de baixa confiabilidade, pois qualquer problema no barramento impossibilita a comunicação em toda a rede."*

Usuário: *"oi"*

Doroty: *"Bom te ver"*

Usuário: *"do que estávamos falando?"*

Doroty: *"Estávamos conversando sobre a topologia de rede barramento"*

B2.3 Armazenando Predicados na Memória

Para armazenar predicados na memória, a sintaxe é mostrada abaixo:

```
<set name="assunto">a topologia de rede barramento</set>
```

Esse comando faz com que o predicado assunto receba o valor "a topologia de rede barramento". Observe uma categoria de Doroty que ilustra tal comando:

```
<category>
  <pattern>BARRAMENTO</pattern>
  <template>
    <think><set name="assunto">a topologia de rede barramento</set></think>
    Na topologia de Barramento, as estações são conectadas
    através de um cabo de cobre (coaxial ou par trançado),
    com difusão da informação para todos os nós.
    É necessária a adoção de um método de acesso para as estações em rede
    compartilharem o meio de comunicação, evitando colisões. É de fácil
    expansão mas de baixa confiabilidade, pois qualquer problema no
    barramento impossibilita a comunicação em toda a rede.
  </template>
</category>
```

B2.4 Recuperando Predicados na Memória

Para recuperar predicados na memória, a sintaxe é mostrada abaixo:

```
<get name="assunto"/>
```

Esse comando faz com que o predicado seja recuperado e enviado para o usuário. Observe uma categoria de Doroty que ilustra tal comando:

```
<category>
  <pattern> * ESTAVAMOS FALANDO</pattern>
  <template><srai>ASSUNTO</srai></template>
</category>

<category>
  <pattern>ME PERDI</pattern>
  <template><srai>ASSUNTO</srai></template>
</category>

<category>
  <pattern>DO QUE ESTAVAMOS FALANDO *</pattern>
  <template><srai>ASSUNTO</srai></template>
</category>

<category>
  <pattern>ASSUNTO</pattern>
  <template>Estávamos conversando sobre <get name="assunto"/></template>.
</category>
```

B2.5 Pensando

Quando se quer executar uma ação, sem que isso seja visível ao usuário, utiliza-se a tag `<think>`. Tudo o que for colocado dentro dessa tag não é mostrado ao usuário.

```
<think>O que estiver escrito aqui dentro, o usuário não vai ler</think>
```

O código apresentado em B2.3 ilustra o funcionamento. Se o comando de armazenamento de memória não estivesse dentro da tag, o conteúdo armazenado seria impresso na tela para o usuário. Exemplo:

Doroty: *"a topologia de rede barramento Na topologia de Barramento, as estações são conectadas através de um cabo de cobre (coaxial ou par trançado), com difusão da informação para todos os nós. É necessária a adoção de um método de acesso para as estações em rede compartilharem o meio de comunicação, evitando colisões. É de fácil expansão mas de baixa confiabilidade, pois qualquer problema no barramento impossibilita a comunicação em toda a rede."*

B2.6 Reduzindo Simbolicamente

A Redução Simbólica permite que sejam criadas diversas categorias que se direcionam a outras. A sintaxe é mostrada abaixo:

```
<srai>categoria desejada</srai>
```

O código apresentado em B2.4 ilustra o funcionamento. As primeiras três categorias direcionam a resposta do chatterbot para a categoria assunto, permitindo que sejam feitas associações sem repetição de código.

B2.7 Executando Comandos ou Programas Externos

Através da tag `<system>` é possível disparar a execução de comandos ou programas externos. A sintaxe é mostrada abaixo:

```
<system>categoria desejada</system>
```

O código abaixo ilustra o funcionamento. Na categoria apresentada a de um comando ping é disparada.

```
<category>
  <pattern>EXEPING</pattern>
  <template>
    <system> ping <get name="endereco"/> </system>
  </template>
</category>
```

B2.8 Escolhendo uma Resposta Randômica

Através da *tag* `<random>` é possível realizar a escolha de uma resposta randômica para ser mostrada ao usuário:

```
<random>
  <li>resposta 1</li>
  <li>resposta 2</li>
  ...
</random>
```

O código abaixo ilustra o funcionamento.

```
<category>
  <pattern>ROTEADOR</pattern>
  <template>
    <random>
      <li>O roteador é o responsável pelo encaminhamento
        de pacotes usando as melhores rotas disponíveis
        consultando tabelas de roteamento que são
        dinamicamente atualizadas a partir da configuração
        inicial.
      </li>
      <li>Um roteador é um dispositivo que tem como
        objetivo determinar o próximo ponto da rede para
        onde um pacote deve ser transmitido até que chegue
        em seu destino. Um roteador deve ser conectado com pelo
        menos duas redes e decide por qual caminho um pacote
        deve ser enviado baseado no conhecimento que possui
        das redes ao qual está conectado. Um roteador usualmente
        mantém uma tabela das rotas disponíveis e suas condições e
        deve usar tais informações, juntamente com a distância e
        o custo dos algoritmos de roteamento, para decidir a melhor
        rota para um pacote.
      </li>
      ...
    </random>
  </template>
</category>
```

B2.9 Utilizando Valores Presentes nas Perguntas

Através da *tag* `<star index>` é possível utilizar valores presentes nas perguntas feitas pelo usuário:

```
<star index="n"/>
```

O código abaixo ilustra o funcionamento:

```

<category>
  <pattern>QUAL O NUMERO TOTAL DE DATAGRAMAS IP RECEBIDOS EM *</pattern>
  <template>
    <think>
      <set name="endereco">
        <system>java ParserEndereco <star index="1"/></system>
      </set>
    </think>
    <srail>IPINRECEIVES3</srail>
  </template>
</category>

```

O valor *n* é substituído pelo número do asterisco presente na frase. Nesse caso, se a pergunta for "*Qual o número de datagramas ip recebidos em 192.168.1.33?*", o valor a ser substituído será 192.168.1.33. E a chamada externa executada pela *tag* <system> será:

```
java ParserEndereco 192.168.1.33
```

B2.10 Recuperando Sentenças Anteriores

A *tag* <that> armazena a declaração anterior do chatterbot. Especificamente se este responde com um parágrafo de múltiplas sentenças, o valor é setado para a última sentença na seqüência. A seguir um exemplo de como isto pode ocorrer.

```

<category>
  <pattern>TOPOLOGIA</pattern>
  <template>
    A topologia de uma rede de comunicação, refere-se a
    forma como os enlaces físicos existentes e os nós de uma comutação
    estão organizados, determinando caminhos físicos existentes e utilizáveis
    entre quaisquer pares de estações conectadas a essa rede. A topologia de
    uma rede muitas vezes caracteriza o seu tipo, eficiência e velocidade. Sobre
    qual topologia você gostaria de obter informações?
  </template>
</category>

<category>
  <pattern>BARRAMENTO</pattern>
  <that>SOBRE QUAL TOPOLOGIA VOCE GOSTARIA DE OBTER INFORMAÇÕES</that>
  <template>
    <srail>TOPOLOGIA BARRAMENTO</srail>
  </template>
</category>

```

Neste caso a categoria "BARRAMENTO" é ativada somente quando o aluno digita esta frase respondendo a pergunta do chatterbot: "*Sobre qual topologia você gostaria de obter informações?*". Este recurso enriquece a conversação, simulando uma memória da conversa apesar de ainda limitada a última sentença. A sintaxe individual do comando é apresentada abaixo:

```

<category>
  <pattern>padrão de entrada</pattern>

```

```
<template>pergunta 1</template>
</category>

<category>
  <pattern>resposta possível para pergunta 1</pattern>
  <that>pergunta 1</that>
  <template>resposta</template>
</category>
```

Outras *tags* presentes especificadas na linguagem podem ser estudadas em detalhes em (RINGATE, 2003) e (TAYLOR, 2003). Foram exemplificadas aqui aquelas que são utilizadas por Doroty, a fim de ilustrar o potencial de conversação que se pode atingir através do uso de tais *tags*.