

UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO

(Bacharelado)

**MULTI-AGENTES NO AUXÍLIO À UTILIZAÇÃO DE
CONHECIMENTO DISTRIBUÍDO**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA
COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA COMPUTAÇÃO –
BACHARELADO

CARLOS ALBERTO BARTH

BLUMENAU, JUNHO/1999

1999/1-08

MULTI-AGENTES NO AUXÍLIO À UTILIZAÇÃO DE CONHECIMENTO DISTRIBUÍDO

CARLOS ALBERTO BARTH

ESTE TRABALHO DE CONCLUSÃO DE CURSO FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Paulo de Tarso Mendes Luna – Orientador

Prof. José Roque Voltolini da Silva – Coordenador do TCC

BANCA EXAMINADORA

Prof. Paulo de Tarso Mendes Luna

Prof. Marcel Hugo

Prof. Sergio Stringari

SUMÁRIO

SUMÁRIO.....	iii
LISTAS DE FIGURAS.....	vi
LISTAS DE ABREVIATURAS	vii
RESUMO	vii
ABSTRACT	ix
1 INTRODUÇÃO.....	1
1.1 A DEFINIÇÃO DA PROPOSTA DE TRABALHO.....	1
1.2 O OBJETIVO DESTE TRABALHO	4
1.3 A ORGANIZAÇÃO DESTE TRABALHO.....	4
2 AGENTES AUTÔNOMOS	6
2.1 BREVE DEFINIÇÃO DE AGENTE.....	8
2.2 PROPRIEDADES DOS AGENTES.....	10
2.2.1 COORDENAÇÃO.....	10
2.2.2 COOPERATIVIDADE/HABILIDADE SOCIAL	11
2.2.3 NEGOCIAÇÃO.....	12
2.2.4 COERÊNCIA DE COMPORTAMENTO	12
2.2.5 PLANEJAMENTO	12
2.2.6 ADAPTABILIDADE/REATIVIDADE/APRENDIZADO	13
2.2.7 MOBILIDADE	14
2.2.8 PERSISTÊNCIA DE INTENÇÃO/PRO-ATIVIDADE.....	14
2.2.9 AUTONOMIA.....	14
2.2.10 COMUNICABILIDADE	16
2.2.11 DISCURSO	17
2.2.12 RACIOCÍNIO/INTELIGÊNCIA.....	18
2.3 ARQUITETURAS DE AGENTES	19
2.3.1 ARQUITETURAS REATIVAS - PEQUENO APROFUNDAMENTO.....	21
2.3.2 APLICAÇÕES.....	22
2.4 AGENTES INTELIGENTES NA INTERNET	23
2.4.1 BREVE INTRODUÇÃO	23
2.4.2 CARACTERÍSTICAS DOS AGENTES DE INERNET.....	24

2.5	METODOLOGIA DE DESENVOLVIMENTO DE AGENTES DE [WOO96]....	26
3	O PROTÓTIPO	30
3.1	A DEFINIÇÃO DO PROBLEMA	30
3.2	ESTRATÉGIA PARA SOLUÇÃO DO PROBLEMA APRESENTADO	30
3.3	A CONTRIBUIÇÃO DO MODELO DIAGNÓSTICO CLÍNICO E RECEITA AOS AGENTES QUE COMPORÃO O SISTEMA	30
3.3.1	A A INTERFACE MAPEAMENTO DO CONHECIMENTO	32
3.3.2	A INTERFACE DIAGNÓSTICO CLÍNICO E RECEITA	35
3.4	APLICANDO AGENTES NA BUSCA DO CONHECIMENTO	38
3.4.1	A IDENTIFICAÇÃO DAS TAREFAS A SEREM EXECUTADAS PELO SISTEMA DURANTE O PROCESSO DE BUSCA DE CONHECIMENTO.....	39
3.4.2	A IDENTIFICAÇÃO DAS ENTIDADES DE AGENTES A SEREM CRIADAS....	40
3.4.3	A ASSOCIAÇÃO ENTRE AS ENTIDADES DE AGENTES E SUAS TAREFAS..	40
3.4.3.1	REGISTRO DE ATIVAÇÃO DA BASE DE CONHECIMENTO	41
3.4.3.2	REQUISIÇÃO DAS PERMISSÕES DE ACESSO	41
3.4.3.3	A BUSCA DO CONHECIMENTO	42
3.4.3.4	A TRANSMISSÃO DOS RESULTADOS OBTIDOS	42
3.4.3.5	A DESATIVAÇÃO DA BASE DE CONHECIMENTO	42
3.4.4	A ESPECIFICAÇÃO DAS ENTIDADES DE AGENTES - A APLICAÇÃO DA METODOLOGIA DE [WOO96]	43
3.4.4.1	O MODELO DE RELACIONAMENTOS	43
3.4.4.2	O MODELO FUNCIONAL	44
3.4.4.3	O MODELO DINÂMICO.....	44
3.4.4.4	OS ESTADOS DE CADA AGENTE.....	47
3.4.4.4.1	O AGENTE CLIENTE	47
3.4.4.4.2	O AGENTE PESQUISADOR.....	47
3.4.4.4.3	O AGENTE ROTEADOR	48
3.4.4.4.4	O AGENTE REGISTRADOR DE ATIVAÇÃO	48
3.4.4.4.5	O AGENTE TRANSMISSOR DE PERMISSÕES	49
3.4.4.4.6	O AGENTE REGISTRADOR DE DESATIVAÇÃO	49
3.4.4.4.7	O AGENTE SERVIDOR LOCAL	50

3.4.4.4.8	O AGENTE TRANSMISSOR DE DADOS	50
3.4.5	A IMPLEMENTAÇÃO	50
3.4.5.1	A IMPLEMENTAÇÃO DOS AGENTES	51
3.4.5.2	JDBC PARA ACESSO AO BANCO DE DADOS.....	53
3.4.5.3	INTERNET COMO MEIO FÍSICO DE TRANSMISSÃO	54
3.4.5.4	HTML NA FORMATAÇÃO DOS RESULTADOS	54
3.4.6	TESTES E VALIDAÇÕES.....	55
4	CONCLUSÕES	63
5	ANEXOS.....	65
6	REFERÊNCIAS BIBLIOGRÁFICAS.....	68

LISTA DE FIGURAS

3.1	Interface Mapeamento do Conhecimento.....	32
3.2	Entidades do subsistema de dados.....	35
3.3	Regras aplicadas à representação do conhecimento.....	36
3.4	Interface Diagnóstico Clínico e Receita.....	37
3.5	Modelo de Relacionamentos.....	43
3.6	Modelo Funcional.....	44
3.7	Modelo Dinâmico.....	45
3.8	Comportamento do Sistema.....	46
3.9	Estados do agente Cliente.....	47
3.10	Estados do agente Pesquisador.....	48
3.11	Estados do agente Roteador.....	48
3.12	Estados do agente Registrador de Ativação.....	49
3.13	Estados do agente Transmissor de Permissões.....	49
3.14	Estados do agente Registrador de Desativação.....	49
3.15	Estados do agente Servidor Local.....	50
3.16	Estados do agente Transmissor de Dados.....	50
3.17	Streams de Dados.....	52
3.18	Uso da Classe serverSocket.....	53
3.19	Ponte ODBC-JDBC.....	53
3.20	SQL e JDBC.....	54
3.21	AgR localizado em RP aguardando conexão do Agc de BcV1.....	56
3.22	AgC de BcV1 informando ativação ao AgR de RP.....	57
3.23	AgC de BcV1 requisitando permissões de acesso ao AgR de RP.....	58
3.24	AgS de BcV1 aguardando conexão do AgP de BcV2.....	59
3.25	AgC de BcV1 recebendo resultados da pesquisa do AgT de BcV2.....	60
3.26	Arquivo de resultados gerado pelo processo de pesquisa em BcV2.....	61
3.27	AgC de BcV1 solicitando sua desativação ao AgR de RP.....	62

LISTA DE ABREVIATURAS

API	Application Program Interface
DPS	Distributed Problem Solving ou Sistemas de Processamento Distribuído
HTML	HiperText Markup Language
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
JDBC	Java Data Base Connection
MAS	Multi Agent Systems ou Sistemas Multi Agentes
ODBC	Padrão aberto de acesso a bancos de dados desenvolvido pela Microsoft
SAD	Sistema de Apoio à Decisão
SQL	Sequencial Query Language

RESUMO

Este trabalho propõe o uso dos agentes autônomos se para construir um protótipo de uma ferramenta que auxilie os médicos veterinários no compartilhamento de conhecimento médico para construção de diagnósticos.

ABSTRACT

This work proposes a multi-agents system prototype, which make possible the interchanging of medical knowledge to the construction of a veterinarian diagnosis.

1 INTRODUÇÃO

Este capítulo visa esclarecer a origem deste trabalho, bem como seu principal objetivo e sua estrutura.

1.1 A DEFINIÇÃO DA PROPOSTA DE TRABALHO

Este trabalho oferece continuidade à proposta de [HOF98], que se caracteriza pela aplicação de um SAD (Sistema de Apoio à Decisão) ao ambiente de decisão de clínicas veterinárias.

Em [HOF98] foram caracterizados e especificados todos os aspectos relevantes quanto à aplicação desta tecnologia a este ambiente de decisão, resultando na implementação de um protótipo de SAD composto por vários modelos. Dentre os modelos propostos por [HOF98], está o modelo Diagnóstico Clínico e Receita, o qual oferece ao decisor auxílio na construção de diagnósticos médicos.

O caráter de continuidade deste trabalho se dá pelo fato de ser aqui implementada a sugestão de [HOF98], o qual propõe a agregação de um sistema de agentes ao modelo Diagnóstico Clínico e Receita que possibilite o compartilhamento de conhecimentos entre médicos geograficamente separados.

Tem-se aqui um retrospecto do contexto que gerou o modelo Diagnóstico Clínico e Receita proposto por [HOF98] como forma de ilustrar a contribuição a ser dada pelos agentes em sua agregação a este modelo: para a construção de diagnósticos, médicos veterinários necessitam diariamente de conhecimentos relacionados às mais diversas áreas específicas da Medicina Veterinária. Muitas vezes o acesso a conhecimentos tão diversificados não é instantâneo, seja por falta de literatura especializada, falta de experiência médica ou não disponibilidade imediata de consultoria.

Ao longo dos dias de atividade do médico veterinário, não são raras as ocasiões em que são enfrentadas situações inusitadas e até mesmo confusas no que diz respeito aos sintomas apurados no paciente durante o exame médico. Isto torna a construção do diagnóstico um processo por vezes difícil e até mesmo demorado, pois são muitas as

variáveis envolvidas para sua definição e estas variáveis implicam em conhecimento diversificado.

Nos casos nos quais um diagnóstico médico se torna difícil, os médicos veterinários fazem pesquisas diversas (livros, periódicos, Internet) ou trocam pontos de vista com outros médicos para resolução do diagnóstico. Isto traz aos médicos veterinários um problema em comum que necessita ser resolvido: como ter acesso eficiente e eficaz ao conhecimento necessário ao diagnóstico? Tal pergunta se faz porque o intercâmbio de conhecimento médico entre clínicos geograficamente separados e a pesquisa são processos eficientes na maioria das vezes, mas também por vezes demorados e esta demora prejudica o desempenho do médico em termos de precisão e rapidez do diagnóstico.

São três os principais efeitos quando o processo de diagnóstico se torna demorado ou impreciso:

- 1) No que diz respeito ao cliente, a postergação do tratamento definitivo pela impossibilidade de definição de um diagnóstico correto e preciso na primeira consulta gera custos financeiros com medicamentos e consultas intermediárias que poderiam ser evitadas;
- 2) Um diagnóstico demorado ou impreciso gera situação de desconforto entre cliente e médico, pois gera insatisfações por parte do cliente;
- 3) Demora ou imprecisão na construção do diagnóstico põem em risco integridade física do paciente, podendo acarretar agravamentos no estado de saúde deste ou até mesmo morte. Embora não se tenha conhecimento de estatísticas que demonstrem o percentual de agravamentos ou até mesmo mortes em virtude de possíveis demoras na construção de diagnósticos, o

¹Dr. Luís Carlos Kriewal informa que tais casos de fato ocorrem.

Conclui-se então que, quando o processo de diagnóstico se torna difícil, o conhecimento para sua construção deve ser buscado o mais rápido possível, para que se possa principalmente preservar a integridade física do paciente. Assim, abreviando-se o

¹ Médico veterinário formado pela UDESC, atuando no estado desde 1988.

tempo que é gasto com pesquisas e consultas a outros médicos, certamente pode-se agilizar a construção dos diagnósticos considerados difíceis minimizando os efeitos de sua demora, havendo ganhos para todas as partes envolvidas - médico, cliente e paciente.

Entretanto, o conhecimento médico existente e aplicável a diagnósticos está disponível em fontes muito diversificadas, configurando-se o seguinte panorama:

- 1) Há uma grande diversidade de especialidades dentro da medicina veterinária. É então coerente afirmar que um profissional da área médico veterinária não possa ser um especialista em todas as áreas de especialização que lhe são oferecidas. Tal fato gera uma carência de conhecimento especializado que necessita ser suprido de alguma forma.
- 2) Há um grande volume de informações gerado por diversos meios de comunicação que não pode ser assimilado totalmente pelo médico. Conforme levantamento prévio tem-se à disposição uma vasta gama de periódicos, sites e até mesmo editoras especializadas os quais publicam resultados de novas pesquisas, novos tratamentos, novas técnicas, novos medicamentos entre outros assuntos relevantes aos profissionais da área. Tal fato torna improvável que um clínico esteja sempre atualizado a respeito de todas as novas técnicas e conhecimentos recentes publicadas nestes periódicos;
- 3) Também é importante mencionar que é grande o número de médicos veterinários atuantes nas diversas especialidades da medicina veterinária. Logo, é válido afirmar que existe no mercado conhecimento profissional que abrange as mais diversas áreas da medicina veterinária. Este conhecimento pode ser compartilhado entre vários profissionais que o necessitem.

Assim sendo, a construção de um módulo para o SAD proposto por [HOF98] que incorpore características de agentes e contribua afim de possibilitar a distribuição de conhecimento , foco do presente trabalho, torna-se importante.

Os agentes a serem implementados no protótipo possibilitarão o compartilhamento rápido de conhecimentos médicos entre profissionais geograficamente separados, para que os diagnósticos considerados difíceis sejam construídos de forma mais ágil e precisa, possibilitando maior ganho de tempo e evitando custos excedentes com tratamentos intermediários e não definitivos. Tais ferramentas constituirão um sistema informatizado que utilizará a Internet como suporte à comunicação e intercâmbio de conhecimento possibilitando incremento na eficiência do médico.

1.2 O OBJETIVO DESTE TRABALHO

O objetivo principal deste trabalho é a especificação e implementação de um protótipo de sistema multi-agentes que funcionará acoplado ao modelo Diagnóstico Clínico e Receita proposto por [HOF98] e será uma ferramenta que possibilitará o compartilhamento de conhecimento entre os médicos veterinários geograficamente separados.

1.3 A ORGANIZAÇÃO DESTE TRABALHO

O trabalho será desenvolvido observando as várias etapas. No capítulo 1, é demonstrada a origem deste trabalho bem como seu principal objetivo e organização aqui apresentada.

No capítulo 2, apresenta-se um estudo a respeito dos agentes. Este capítulo está dividido em vários subitens, apresentando informações relativas às propriedades dos agentes, suas arquiteturas e também conceitos específicos sobre agentes de Internet. É também apresentada neste capítulo uma metodologia que permite especificar os agentes a serem implementados no protótipo.

O capítulo 3 contempla vários aspectos relevantes ao trabalho pois tem-se aqui a especificação das entidades de agentes, sua implementação e também testes e validações que comprovam a aplicabilidade prática do protótipo.

Tem-se a seguir o capítulo 4, onde são demonstradas as conclusões a respeito da viabilidade tecnológica deste trabalho e sua aplicabilidade prática. Sugestões para

trabalhos futuros que visem aperfeiçoamento da solução proposta são apresentadas no capítulo 5, o capítulo final deste trabalho.

2 AGENTES AUTÔNOMOS

De acordo com [GAS88], a Inteligência Artificial (IA) é uma área da computação que se concentra no estudo de como um modelo computacional pode exibir comportamento inteligente para resolução de problemas. A IA propõe o uso de heurística e do conhecimento baseado em casos, como forma de gerar a percepção e aprendizagem necessários a essas entidades computacionais.

Em [GAS88] afirma-se que o desenvolvimento de novas tecnologias, como computadores poderosos, a proliferação do uso de redes de comunicação e a vasta gama de problemas humanos a serem resolvidos, influenciaram o surgimento da IAD (Inteligência Artificial Distribuída). “Esta é uma sub-área da Inteligência Artificial que se concentra na distribuição e concorrência da IA, visando a cooperação mútua entre entidades computacionais inteligentes para solução de problemas” [GAS88]. Dentro da IAD, existem dois ramos principais, os quais são alvo de estudos e pesquisas : *Multi-Agent Systems* ou Sistemas Multi-Agentes (MAS) e *Distributed Problem Solving* (DPS) ou Sistemas Distribuídos.

De acordo com [JEN96], os sistemas DPS são compostos por unidades computacionais, as quais podem ser divididas em um determinado número de módulos que distribuem conhecimento sobre o problema e sobre as soluções que o envolvem. Em sistemas DPS puros, toda estratégia de interação é incorporada como uma parte integral do sistema. Uma característica fundamental que diferencia um sistema DPS de um sistema MAS, segundo [JEN96], é que enquanto o primeiro é concebido para solução de um problema específico, o segundo é construído para soluções genéricas. Este trabalho, no entanto, não tem como objetivo o estudo da arquitetura DPS, mas sim um estudo mais profundo sobre a arquitetura MAS.

Os MAS são sistemas que se compõem de uma coleção de entidades autônomas denominadas Agentes Inteligentes. De acordo com [GAS88], essas entidades têm o poder de coordenar seu conhecimento, metas e planos, agindo cooperativamente na busca de um objetivo comum pré-definido. Segundo [WOO94], em sistemas multi-agentes, os agentes

que o compõe devem não só ter conhecimento sobre problemas e soluções, mas também ter capacidade de raciocínio para que esse conhecimento possa ser coordenado na busca da solução do problema.

De acordo com [WOO96], o paradigma dos agentes autônomos é baseado em noções de como prescrever a essas entidades reatividade, autonomia e motivação. Ele diz que essas entidades, em seu mundo virtual, são envolvidas em incertezas a respeito desse mundo. Elas devem então ter a capacidade de perceber e agir dentro dele. Um apanhado geral das propriedades que regem os agentes autônomos são vistas no item 2.2 deste capítulo.

Em [GAS88], encontra-se a definição de benefícios quanto à aplicação de agentes para solução de problemas distribuídos :

- a) *adaptabilidade*: sistemas multi-agentes são mais apropriados para lidar com problemas distribuídos em termos espaciais, lógicos, temporais ou semânticos;
- b) *custo*: o grande número de pequenas entidades computacionais pode ser mais interessante quando os custos em comunicação não são relevantes;
- c) *desenvolvimento e gerenciamento*: a inerente modularidade dos sistemas multi-agentes permite o desenvolvimento das partes de forma independente e paralela;
- d) *eficiência e velocidade*: a concorrência e a distribuição das entidades computacionais em diferentes máquinas pode aumentar a velocidade de processamento haja visto o paralelismo na execução de tarefas;
- e) *integração*: a integração de recursos distribuídos e até mesmo heterogêneos tais como hardware e software de diferentes plataformas;
- f) *segurança*: o controle dos processos locais pode ser encarado como uma maneira de proteção ou de aumento de segurança do sistema;
- g) *naturalidade*: alguns problemas são mais naturalmente resolvidos com a ajuda de uma configuração distribuída;

- h) *confiabilidade*: os sistemas distribuídos podem exibir um grau maior de confiabilidade e de segurança quando comparados a um sistema tradicional de processamento centralizado, pois podem prever redundância de dados e prover múltiplas verificações;
- i) *bom aproveitamento de recursos*: os agentes computacionais individuais, mesmo que ligados a recursos escassos, podem superar e resolver problemas complexos;
- j) *especializações*: pode-se especializar cada agente de acordo com seu domínio de conhecimento e aplicação.

Os MAS são uma inovação dentro da IA e se destinam a várias aplicações. Dentre estas aplicações, segundo [WOO96], pode-se citar controle aéreo, sistemas de simulação de um modo geral, controle de negócios, controle de comunicações, etc. De acordo com [WOO94], é um novo paradigma para o desenvolvimento de aplicações e, segundo ele próprio, muito além disso, sistemas baseados em agentes têm sido aclamados como sendo o futuro em termos de desenvolvimento e a nova revolução em software.

2.1 BREVE DEFINIÇÃO DE AGENTE

Em [WOO96], é encontrada a definição de agente como sendo entidades computacionais aptas a habitar um ambiente virtual complexo e dinâmico, raciocinando e agindo neste ambiente de modo a realizar as tarefas e objetivos para os quais foram criados.

Segundo [SOU96], agentes autônomos possuem propósitos que são gerados por eles próprios em vez de adotados de outros agentes. Estes propósitos originam-se de motivações as quais são definidas como desejos ou preferências que podem levar à geração e adaptação de objetivos. Assim, agente motivado é um agente que possui sua própria agenda para solução e comportamento, de acordo com sua motivação interna.

O comportamento do agente é determinado pelo resultado da percepção, conhecimento e interação com outros agentes. É o grau de interação entre os agentes que os posiciona em categorias. Em outras palavras dos princípios e idealizações adotados por

pesquisadores para que possa ser considerada uma entidade autônoma, um agente deve possuir diversas habilidades. Essas habilidades figuram em [JEN96] e são elas:

- percepção e interpretação quando do recebimento de mensagens;
- raciocínio com base em suas crenças;
- capacidade de decisão (seleção de metas, raciocínio a partir de uma intenção);
- planejamento de ações (seleção ou construção de um plano de ações, solução de conflitos, alocação de recursos);
- capacidade de execução de planos incluindo também capacidade de comunicação.

Entretanto, [JEN96] diz que, em um MAS, os agentes que o compõe podem ser caracterizados em vários níveis. Estes níveis são definidos de acordo com a capacidade de resolução de problemas do agente.

Para [WOO94] a inexistência de consenso universal na definição de agentes determinou dois enfoques na conceituação destes, conforme segue:

- a) fraca noção sobre agentes:* este enfoque apresenta definições relativamente insatisfatórias onde a palavra "agente" é usada para especificar um *hardware* ou um *software* que possui as propriedades de autonomia, habilidade social, reatividade e pró-atividade;
- b) robusta noção sobre agentes:* no âmbito das pesquisas realizadas em IA, este enfoque apresenta uma definição mais abrangente e satisfatória para agentes. Geralmente, identifica um sistema computacional que além de apresentar as propriedades de autonomia, habilidade social, reatividade e pró-atividade, é conceituado ou implementado utilizando-se propriedades aplicáveis a pessoas, como por exemplo, conhecimento, crenças e intenções.

2.2 PROPRIEDADES DOS AGENTES

Tem-se a seguir a descrição das propriedades que podem ser implementadas nos agentes.

2.2.1 COORDENAÇÃO

A coordenação é o método que gerencia as trocas de informações numa arquitetura MAS. De acordo com [JEN96], em um sistema multi-agentes, a coordenação é uma prioridade pois, sem coordenação, um grupo de agentes pode rapidamente degradar-se e se transformar numa coleção de entidades individualistas. O resultado seria a degradação também do sistema como um todo.

Em [GAS88] a coordenação é definida como a propriedade que permite aos agentes uma interação mútua, criando modelos de atividade e interação entre as agentes, evitando “comportamentos estranhos” que prejudicariam as metas a serem atingidas pela coletividade dos agentes.

São três os principais aspectos da coordenação, segundo [JEN96] :

1. *ajustamento mútuo*: é a mais simples forma de coordenação. Ela ocorre quando dois ou mais agentes aceitam compartilhar recursos em prol de um objetivo comum. Um agente pode usualmente modificar o destino de informações e realizar vários ajustes em seu próprio comportamento, dependendo do comportamento dos outros agentes pertencentes à sociedade. Na coordenação por ajustamento mútuo, nenhum agente tem prioridade de controle sobre outros agentes e as decisões são tomadas num processo conjunto;
2. *supervisão direta*: este tipo de coordenação ocorre quando dois ou mais agentes mantém um relacionamento onde um agente tem prioridades de controle sobre os demais agentes da comunidade. Neste tipo de relacionamento, a entidade supervisora controla o compartilhamento de recursos entre as entidades subordinadas e pode ainda definir aspectos de

comportamento dessas entidades. Esta forma de coordenação é definida em [FAR98] como Coordenação Mestre-Escravo;

3. *hierarquização*: o método de coordenação por hierarquização vem suprir limitações contidas nos modos de Ajustamento Mútuo e Supervisão. Esses dois modos possuem limitações quanto à quantidade de entidades que compõe a comunidade. De forma mais clara, quando o número de entidades é pequeno, a comunicação entre essas entidades é de fácil coordenação. À medida que o número de agentes cresce, a quantidade de informações a ser trocada entre esses agentes também cresce em quantidade e velocidade. A coordenação por hierarquização, porém, orienta a criação de sub-grupos de agentes, combinando também as formas de coordenação 1 e 2. Isto porque se a maior parte da comunicação ocorre dentro de cada grupo, então cada grupo pode coordenar a sua comunicação interna da forma que melhor lhe convier, seja por ajustamento ou supervisão, sempre dependendo do domínio da aplicação e das características das tarefas a serem executadas. Já comunicação entre os diversos grupos existentes pode ser coordenada pelo método de supervisão.

2.2.2 COOPERATIVIDADE / HABILIDADE SOCIAL

Segundo [JEN96], implementar formas de cooperação entre agentes é uma das principais questões relacionadas à arquitetura MAS. Em [JEN96] encontra-se a proposta de se caracterizar a cooperação em termos dos objetivos adotados pela comunidade de agentes. Ele diz que em sistemas multi-agentes, os agentes são independentes e compartilham o mesmo ambiente competindo pelos mesmos recursos limitados como tempo e espaço. Para alcançar a eficiência, os agentes devem cooperar entre si, obtendo benefícios ou apenas ajudando outros agentes e, conseqüentemente, evitando conflitos. Essa cooperação é necessária porque um único agente não tem conhecimentos e recursos suficientes para resolver sozinho um dado problema, mas vários agentes podem agregar as habilidades necessárias para resolve-lo em pequenas partes.

A cooperação é definida em [SOU96] como habilidade social. A habilidade social, segundo [SOU96] se traduz pela aptidão do agente em se comunicar com os demais agentes da comunidade como recurso para conclusão de suas tarefas, ou apenas em auxílio a outros agentes.

2.2.3 NEGOCIAÇÃO

A capacidade de negociação se torna necessária para que os agentes possam resolver os conflitos que interferem no comportamento cooperativo. Em [JEN96] é encontrada uma definição para negociação : “Negociação é o processo de melhoramento do entendimento de pontos de vista comuns, planos e informações relevantes”. Similarmente a [JEN96], vê-se em [WOO96] a definição de negociação conforme segue: “Agentes interagem por meio de negociação. Durante uma negociação, agentes trocam mensagens, mudam seu estado e executam ações locais”. Esta negociação permite, de acordo com a abordagem de [WOO96], que agentes tomem ações em conjunto, direcionando desta forma, suas ações em prol dos objetivos do grupo.

2.2.4 COERÊNCIA DE COMPORTAMENTO

Em [JEN96], esta propriedade é definida como sendo necessária para que se possa definir como os agentes de uma comunidade, tendo eles comportamento cooperativo, podem agir de forma que suas ações façam sentido tendo em vista o objetivo comum do grupo de agentes.

2.2.5 PLANEJAMENTO

Em [GAS88], a importância do planejamento é expressa como uma forma de raciocínio sobre como as ações de diferentes agentes podem interferir em outros agentes.

De acordo com [JEN96], o qual fornece um ponto de vista esclarecedor, o planejamento das ações a serem executadas pelos agentes de uma comunidade é importante, pois é o planejamento que previne conflitos entre os planos de ação a serem executados pelos agentes dessa comunidade. Existem duas abordagens que tratam da forma

de planejamento das ações dos agentes: planejamento centralizado e planejamento distribuído.

Segundo [JEN96], no planejamento centralizado, apenas um agente é responsável por construir o plano de ação de toda a coletividade de agentes da comunidade, enquanto que no planejamento distribuído, o planejamento das atividades é distribuído entre o grupo de agentes. Ele diz também que todo planejamento de atividades necessita também de um plano de sincronização entre as atividades planejadas. Isto porque, tendo em vista que podem ocorrer conflitos por incompatibilidades de estados, atividades ou compartilhamento de recursos, um agente coordenador é necessário, para que esses conflitos possam ser então resolvidos.

2.2.6 ADAPTABILIDADE / REATIVIDADE / APRENDIZADO

Segundo [GAS88], um agente apresenta adaptabilidade quando possui a habilidade de modificar sua própria configuração, de modo a atender da melhor forma possível os requisitos do sistema. Consequentemente, a adaptabilidade proporciona coerência e coordenação. A idéia da adaptabilidade é reforçada por [MAE91], o qual diz que em um dado ambiente os objetivos presentes podem mudar em tempo de execução. A adaptabilidade é importante, segundo [GAS88], porque as contingências que envolvem a resolução do problema podem mudar. Neste caso, uma boa estratégia é o replanejamento. Replanejamento, por consequência, exige adaptabilidade. A propriedade de adaptabilidade, quando presente, permite melhoras em termos de performance do sistema, pois respostas previamente programadas podem não ser apropriadas quando o grau de incerteza é grande.

Em [SOU96] é feito o relacionamento entre as propriedades de aprendizado e comportamento adaptativo. Estas são definidas como a habilidade apresentada pelo agente de acumular conhecimento baseado em experiências anteriores, e consequentemente, modificar seu comportamento em resposta a novas situações. Em [SOU96] é visto que aprendizagem é a capacidade que um agente deve possuir para executar uma tarefa com maior eficiência do que em execuções anteriores. Sem a capacidade de aprendizagem o agente reagirá sempre da mesma maneira para um mesmo ambiente e uma mesma situação.

A propriedade do agente em se adaptar às mudanças do ambiente é definida por [WOO96] como reatividade. A definição de reatividade de [WOO96] é dada conforme segue: “Um agente deve ser apto a modificar seu comportamento apropriadamente de acordo com as mudanças do ambiente. Um agente não deve simplesmente executar planos, mas deve fazê-lo levando em consideração o estado do ambiente em que se encontra inserido”.

2.2.7 MOBILIDADE

De acordo com [JEN98], a propriedade de mobilidade permite a um agente ser transmitido de um lugar para outro, executar suas tarefas dentro de um ambiente virtual seguro em uma locação remota qualquer, voltando em seguida ao seu endereço de origem.

2.2.8 PERSISTÊNCIA DE INTENÇÃO / PRO-ATIVIDADE

Um agente, segundo [WOO94], pode apresentar uma propriedade caracterizada como persistência de intenção. Pode-se definir esta propriedade como sendo a capacidade de não abandonar um objetivo antes que ele seja realizado. Esta propriedade garante ao agente poder manter um estado interno conciso, sem modificá-lo ao acaso.

Esta definição se assemelha à definição de proatividade de [SOU96], que diz que um agente deve exibir um comportamento direcionado a objetivos. Para tanto, ele não reage simplesmente em resposta ao ambiente, mas sim de acordo com um propósito pré-definido. Em [WOO94] a idéia de pró-atividade é reforçada, pois ele nos diz que um agente deve apresentar um comportamento oportunístico, voltado para a realização de seus objetivos.

2.2.9 AUTONOMIA

Em [SOU96], encontra-se uma definição sobre autonomia: “Autonomia é a capacidade do agente de executar o controle sobre suas próprias ações. As propriedades que permitem ao agente operar isoladamente, adicionadas às propriedades que o permitem decidir sobre o que fazer enquanto opera neste estado, juntas constituem a autonomia. Um

agente inteligente deve possuir a habilidade de praticar ações para desenvolver tarefas ou alcançar objetivos, sem necessitar da interferência do usuário final”.

Em acréscimo a esta definição, [WOO94] comenta que o agente, além de possuir controle sobre seu comportamento, deve também possuir controle sobre seu estado interno, visando autonomia.

De acordo com [SOU96], os agentes detentores de alto grau de autonomia podem manter suas agendas independentes daquelas de seus usuários. Para tanto, os agentes apresentam três aspectos básicos: ações periódicas; execução espontânea e iniciativa. Estes requisitos habilitam o agente autônomo a efetivar ações preemptivas e independentes que poderão eventualmente beneficiar o usuário.

De acordo com [WOO94], são cinco os princípios básicos que definem a autonomia de um agente:

1. possui seus próprios objetivos. Eles são endógenos, ou seja, não são derivados de outros agentes;
2. é hábil para tomar decisões;
3. ele adota objetivos de outros agentes, ou seja, é predisposto à influência de outros agentes;
4. ele adota objetivos de outros agentes como consequência de uma escolha entre seus próprios e os de outros agentes. Esta adoção não é automática nem governada por regras;
5. ele adota objetivos de outros agentes se esses objetivos podem ser uma forma de atingir seus próprios objetivos.

Segundo [WOO94], a regra 3 a qual fala de influência de um agente sobre outros, é um paradoxo frente ao termo autonomia. Para esclarecer, ele nos explica da seguinte forma : “A autonomia se torna sem sentido quando levada ao extremo. Se um agente é concebido e em momento algum tem a capacidade de adotar objetivos de outros agentes, ele vive sozinho em um comportamento anti-social. Então, um agente, para ser social, deve ter sua autonomia limitada”.

2.2.10 COMUNICABILIDADE

Segundo [SOU96], a propriedade da comunicabilidade é importante, pois possibilita o diálogo entre agentes. É coerente afirmar então que, sem a propriedade da comunicabilidade torna-se impraticável a negociação, a coordenação ou qualquer ação que exija diálogo entre os agentes de uma determinada comunidade. Consequentemente, sem diálogo, o nível de cooperatividade dentro da comunidade de agentes torna-se extremamente prejudicado, pois configura-se um ambiente de inabilidade social.

Segundo [DEM89], um agente quando implementado deve ter a capacidade de perceber o ambiente, adquirindo conhecimento sobre este ambiente. O conhecimento sobre o ambiente pode ser adquirido através da comunicação com outros agentes eventualmente inseridos naquele ambiente.

Um agente, comunicando-se com outros agentes do ambiente, adquire conhecimento sobre a personalidade destes agentes, e pode ser capaz de executar escolhas que favoreçam o comportamento cooperativo dentro do grupo, pois terá a capacidade de reconhecer os planos executados por estes agentes e suas ações.

Esta troca de informações, segundo [DEM89], é uma forma de percepção que pode ser implementada e pode ser de três tipos :

- a) *troca de conhecimento*: dentro de um ambiente, dois ou mais agentes podem ter descrições diferentes a respeito de uma mesma situação. Estas descrições podem ser complementares ou conflitantes. Dando-se aos agentes a capacidade de percepção pela troca de conhecimento, pode ser realizada uma negociação entre estes agentes, de modo a resolver conflitos ou complementar idéias, favorecendo o surgimento de um ponto de vista comum entre eles;
- b) *troca de possíveis soluções*: a troca de possíveis soluções deve ocorrer sempre que dois ou mais agentes necessitem de um ponto de vista comum de planos de ação. Uma das maneiras de se conseguir este ponto de vista comum é promover-se a intercessão entre as possíveis soluções

apresentadas por cada agente. Entretanto, devido às diferentes capacidades de raciocínio de cada agente, a técnica da intercessão pode ser inútil. Seria então adequado o raciocínio sobre a capacidade de raciocínio de cada agente como forma de deduzir-se novas possíveis soluções;

- c) *troca de possíveis escolhas*: em virtude de ser necessário o comportamento cooperativo entre agentes, estes podem aceitar sugestões de escolha quanto à execução de ações. A comunicação para troca de escolhas pode ocorrer quando um agente solicita a outro que o faça.

Esta diversidade de possibilidades de comunicação gera três níveis de interação entre os agentes conforme segue:

- a) *forte interação entre agentes*: neste tipo de integração, os agentes envolvidos no processo de comunicação negociam com base em pontos de vista relacionados a conhecimentos, ações e possíveis soluções;
- b) *média interação entre agentes*: neste tipo de integração, os agentes negociam com base em pontos de vista relacionados a conhecimento e possíveis soluções;
- c) *fraca interação entre agentes*: este é o menor grau de integração possível entre agentes, segundo [DEM89]. Neste tipo de integração, os agentes negociam com base em pontos de vista relacionados apenas a conhecimento.

2.2.11 DISCURSO

A propriedade do discurso é, segundo [SOU96], uma propriedade que permite a integração entre os agentes inseridos no ambiente do sistema e o usuário deste sistema. Através desta propriedade, [SOU96] nos diz que se cria um *feedback* entre agente e usuário, fazendo com que o usuário tome maior conhecimento das habilidades e intenções dos agentes. Como exemplo de discurso entre agente e usuário, [SOU96] exemplifica da seguinte forma: “Durante um discurso, o usuário especifica que ações poderiam ser

tomadas pelos agentes e que lhe trariam benefícios. Cabe ao agente selecionar as ações que pode executar, bem como os possíveis resultados”.

2.2.12 RACIOCÍNIO / INTELIGÊNCIA

Em função do agente atuar em um ambiente dinâmico, conforme dito por [WOO96], configura-se a necessidade de raciocínio. Esta propriedade é definida por [SOU96] conforme segue: “Inteligência é a propriedade do agente que o habilita a negociar efetivamente com ambigüidades. Durante o processo de identificação da ação mais adequada, o agente defronta-se com ambigüidades nos mais diversos níveis. Neste contexto, a inteligência pode ser considerada como um conjunto de recursos, atributos e características que habilitam o agente a decidir que ações executar”.

Conforme pode-se notar, a propriedade da inteligência tem estreita relação com a propriedade da coerência de comportamento, tendo em vista que se presume que raciocínio implique em coerência de ações. Entretanto, [SOU96] diz que a coerência tem um sentido mais amplo no que diz respeito aos objetivos gerais do grupo, isto porque a ação mais apropriada a um agente em particular pode não ser a ação mais apropriada para que se atinja do objetivo geral naquele momento.

2.3 ARQUITETURAS DE AGENTES

Segundo [FAR98], pode-se definir como arquitetura de agentes a forma como os agentes foram implementados. Esta forma diz respeito às suas propriedades, sua estrutura e como é realizada a interação entre estes agentes. A arquitetura é, em suma, de acordo com [FAR98], a especificação da estrutura e do funcionamento dos agentes.

De acordo com [SOU96], a arquitetura dos agentes pode ser assim definida: “A arquitetura é uma metodologia particular para definir agentes. Ela especifica como o agente pode ser decomposto na construção de um ambiente de módulos componentes e como estes módulos podem interagir. O conjunto de módulos e suas interações devem prover uma resposta para a questão de como os sensores de dados e o estado interno corrente do agente determinam suas ações e futuro estado interno. Uma arquitetura deve prever as técnicas e algoritmos para suportar esta metodologia”.

Os pesquisadores definem várias arquiteturas, cada qual baseada em características específicas de estrutura e funcionamento dos agentes. Daí encontra-se em [FAR98] a menção de cinco tipos de arquiteturas conforme segue nos itens numerados a seguir:

1. *arquiteturas cognitivas ou deliberativas*: “é considerada uma arquitetura clássica para agentes. Ela contém um modelo simbólico do mundo representado explicitamente e as decisões a respeito das ações a tomar são feitas via raciocínio lógico, baseado em reconhecimento de padrões e manipulação simbólica. Essa arquitetura se caracteriza pela complexidade dos agentes, pois estes apresentam mecanismos de inferência robustos, interações sofisticadas e alto grau de intenção em seu comportamento”;
2. *arquiteturas funcionais*: “nesta arquitetura, que também pode ser classificada como deliberativa, os agentes são compostos por módulos funcionais que são necessários para a operação adequada do sistema. Os agentes são dotados de percepção, comunicação, decisão e raciocínio”;

3. *arquiteturas baseadas em estados mentais*: “este tipo de arquitetura adota uma perspectiva psicológica para a definição da estrutura dos agentes. Os agentes baseados neste tipo de arquitetura possuem estados mentais a respeito do domínio em que se encontram, crenças a respeito deste domínio e capacidade de escolha de compromissos”;
4. *arquiteturas reativas*: “neste tipo de arquitetura não se inclui nenhum modelo do mundo simbólico e também não é utilizado raciocínio simbólico complexo, somente reação às ações que ocorrem no ambiente”;
5. *arquiteturas de suposição*: “este é um exemplo de arquitetura reativa onde existe uma hierarquia de níveis de comportamento tarefa-ação, onde cada comportamento compete com outros para ter controle sobre um robô. Esta arquitetura surgiu da necessidade em um mecanismo para controle de robôs móveis e autônomos. Neste tipo de arquitetura somente há reação às ações que ocorrem no ambiente”;
6. *arquiteturas híbridas*: “mistura componentes de arquiteturas deliberativas e reativas com o objetivo de se tornar mais apurada funcionalmente a determinados requisitos impostos aos agentes”.

As diferentes arquiteturas propostas implicam na definição de vários tipos de agentes, cada qual agregando características e propriedades em função da arquitetura em que está inserido. Desta forma, [JEN96] nomeia três tipos de agentes e são eles:

1. *agente reativo*: um agente reativo apenas reage a mudanças no ambiente onde está inserido ou a mensagens enviadas por outros agentes. Ele não tem habilidade de raciocínio sobre suas intenções (manipulação de metas). Este tipo de agente apenas executa ações baseadas em regras ou modelos pré-definidos, como, por exemplo, escolha e envio de mensagens relevantes a outros agentes de acordo com a atual situação do ambiente. A primeira geração de Sistemas Especialistas, compostos de uma base de conhecimento, regras e uma máquina de inferência são exemplos típicos de agentes reativos;

2. *agente intencional*: um agente intencional é capaz de raciocinar sobre suas metas e crenças, de forma a criar planos de ação e executar esses planos. Sistemas baseados em agentes intencionais podem ser chamados de sistemas planejadores. Tal definição, segundo [JEN96], advém do fato dessas entidades possuírem a capacidade de selecionar suas metas, sempre de acordo com sua motivação, e raciocinar sobre elas, detectando e resolvendo conflitos de objetivos e coincidências. Elas podem também selecionar ou criar planos, detectar conflitos entre planos e executar e revisar estes planos, se necessário. Em suma, em um sistema composto por agentes intencionais, essas entidades coordenam a si mesmas, trocando informações sobre suas crenças, metas e ações;
3. *agente social*: um agente social possui todas as habilidades de um agente intencional. Além disso, um agente social deve possuir a habilidade de processar modelos construídos por outros agentes da sociedade. Conseqüentemente, um agente social deve ter a capacidade de manter estes modelos, atualizando suas crenças, metas e eventualmente, planos. Um agente social deve ter a capacidade de raciocinar sobre o conhecimento incorporado a esses modelos, para que possa criar seus planos sempre com base na cooperatividade com as demais entidades da sociedade. Este tipo de agente é definido por [NWA96] como agente cooperativo.

Em [NWA96] encontra-se a definição de um quarto tipo de agente, o Agente de Informação ou Agente de Internet. Este será visto no próximo capítulo com maiores detalhes. Por hora, cabe um pequeno aprofundamento na arquitetura reativa, tendo em vista que este tipo de arquitetura será utilizada na implementação do protótipo.

2.3.1 ARQUITETURAS REATIVAS – UM PEQUENO APROFUNDAMENTO

Como anteriormente mencionado, [FAR98] explica que neste tipo de arquitetura não se inclui nenhum modelo do mundo simbólico e também não é utilizado raciocínio simbólico complexo, somente reação às ações que ocorrem no ambiente. Em [HEU97], encontra-se concordância com a definição de [FAR98]. Pode-se averiguar em

[NWA96] que as arquiteturas reativas são estudadas desde meados da década de oitenta, e desde então muitas linguagens de desenvolvimento e teorias têm sido desenvolvidas.

De acordo com [NWA96], um dos pontos realmente importantes a se observar a respeito das arquiteturas reativas, é que ela é composta por agentes relativamente simples e que eles interagem entre si de forma também simples. Dessas interações podem derivar comportamentos mais complexos, caso se considere o comportamento do sistema de forma global. Em [NWA96] também são descritas três características básicas que, segundo ele, definem bem uma arquitetura reativa e são elas:

1. *funcionalidade emergente*: a chamada funcionalidade emergente diz respeito ao crescimento da complexidade comportamental global do sistema, em virtude das interações simples entre agentes também simples. Sob este ponto de vista, [NWA96] comenta que sistemas “vivazes” podem ser construídos a partir de agentes simples, e que esta vivacidade deriva das interações entre estes agentes;
2. *decomposição de tarefas*: uma arquitetura reativa deve ser vista como um conjunto de módulos que operam de forma autônoma e são responsáveis em executar uma determinada tarefa, que pode ser controle de um motor elétrico, cálculos matemáticos, envio de dados, etc. Vale lembrar que nenhum modelo global de comportamento rege estes módulos, e por este motivo tem-se a funcionalidade emergente anteriormente mencionada;
3. *processamento ação/reação*: as arquiteturas reativas agem em um ambiente onde a representação do mundo real se reduz a sensores de dados, ao contrário das arquiteturas deliberativas, onde representações simbólicas de alto nível de complexidade modelam o comportamento global do sistema.

2.3.2 APLICAÇÕES

Em [NWA96] vê-se que atualmente as aplicações para agentes reativos são limitadas, restringindo-se geralmente a simulações e jogos. Ele diz que agentes reativos podem ser utilizados para simular muitos tipos de ambientes do mundo real, bem como os

fenômenos que ocorrem naturalmente nesses ambientes. Por exemplo, uma sociedade de agentes reativos pode ser utilizada para simular um mundo virtual qualquer, onde cada agente representa um elemento deste mundo (peixe, cardume e pescador). Como forma de melhor explicar, [NWA96] diz que agentes reativos tem potencial de transformar o computador em um laboratório virtual, onde o pesquisador pode modificar qualquer parâmetro experimental e validar seu modelo modificando esses parâmetros seja de forma quantitativa ou qualitativa.

Segundo [NWA96], é conveniente comentar que é pequena a quantidade de arquiteturas reativas implementadas e em funcionamento. Duas destas arquiteturas são mencionadas abaixo:

- a arquitetura de Brooks, utilizada para implementar robôs autônomos, definida como uma arquitetura puramente voltada a agentes do tipo reativo;
- a arquitetura de Suchmann, que é o modelo mais simplificado já construído e se baseia em regras de situação/ação. Os agentes implementados nesta arquitetura executam a ação mais apropriada à situação apresentada em um determinado momento. Esta situação é determinada de acordo com a combinação de um conjunto de eventos e estados internos e externos. Ela foi utilizada em um software chamado PENGI, que é um jogo.

2.4 AGENTES INTELIGENTES NA INTERNET

O item 2.3 e seus sub-itens tem como finalidade traçar um pequeno perfil da aplicabilidade dos agentes autônomos na recuperação de informações na Internet.

2.4.1 BREVE INTRODUÇÃO

Segundo [DAM96], em 1995 a Internet alcançou proporções globais, atingindo 106 países e 50 milhões de usuários. A rede estabeleceu um novo paradigma nas comunicações e tornou-se um dos assuntos prediletos da mídia, sendo referência constante nos principais meios de comunicação.

Em [NWA96] encontra-se a descrição do que seriam os agentes de Internet. Em [NWA96] tem-se que agentes de Internet executam suas buscas de forma metódica e planejada, manipulando ou buscando informações de várias fontes remotamente distribuídas. Ambos, [RUS95] e [HEU97] afirmam que o surgimento deste tipo de agente em particular se deu em função da grande demanda por ferramentas que ajudem as pessoas a manusear explosivo crescimento do volume de informações que estão experimentando com o advento da Internet, volume este que continuará crescendo. Em [HAR98], também considerando que a Internet é o lar de um enorme volume de informações, complementa [NWA96] frisando que agentes têm o potencial de filtrar este enorme fluxo de informações e passar ao usuário somente aquilo que lhe interessa.

“Nós estamos nos afundando em informações mas carentes de conhecimento” [NWA96]. De acordo com [NWA96], se for considerado o atual perfil da Internet no que diz respeito a volume de informações, esta máxima torna-se verdadeira. Por este motivo, [NWA96] afirma que os agentes para Internet podem ser melhorados, mas jamais eliminados, pois tem-se hoje um problema específico de sobrecarga de informações, bem como do manuseio de sua demanda. O problema da sobrecarga de informações é evidente e vem se agravando. Para que se possa vislumbrar tal afirmativa e perceber a utilidade e a necessidade dos agentes de Internet, basta observar os argumentos apresentados por [NWA96]: “em pesquisa feita em meados de 1985, chegou-se à seguinte projeção: o conhecimento científico, técnico e corporativo presente na Internet dobraria a cada cinco anos. Três anos depois essas projeções foram refeitas e atualizadas com base nos parâmetros atuais da época. Chegou-se à conclusão que este conhecimento dobraria a cada 2,2 anos. Em 1992, numa nova atualização, o período diminuiu para 1,6 anos”. Estas projeções, segundo [NWA96], sugerem que atualmente este volume de conhecimento dobra a cada ano. Estas projeções são a motivação para o desenvolvimento de agentes de Internet.

2.4.2 CARCTERÍSTICAS DOS AGENTES DE INTERNET

De acordo com [NWA96], agentes de Internet podem possuir várias características: podem ser estáticos ou móveis. Podem ser reativos, compondo uma arquitetura simples e pouco sofisticada, ou podem ser do tipo deliberativo (intencionais ou

sociais), possuindo ou não a capacidade de aprender e raciocinar e de traçar e executar planos e metas. As propriedades variam sempre de acordo com os requisitos do sistema. Em [COC98] encontra-se concordância com a definição descrita por [NWA96].

Entretanto, há características implícitas aos agentes, definidas pelo ambiente de implementação. Por este motivo, torna-se imprescindível traçar um pequeno perfil deste ambiente, de modo a poder-se compreender de forma mais completa o potencial dos agentes de Internet.

Os agentes tem estreita ligação com a rede mundial, pois são desenvolvidos na linguagem Java, que é tida por [SOU97] como a linguagem que fornece os alicerces para o desenvolvimento de aplicativos para Internet. Segundo [ANT97], o ambiente Java, além da transparência a plataformas e das fortes capacidades de rede, oferece os benefícios da orientação a objetos e *multithreading*. De acordo com [SOU97], são estas as vantagens oferecidas pelo ambiente:

1. *portabilidade*: a predominância de diversos sistemas operacionais e plataformas gerou uma demanda intensa por softwares portáveis. Ou seja, os programas precisam ser escritos de tal maneira que não apenas funcionem em várias plataformas diferentes, mas também se adaptem à “aparência” natural do ambiente. Além disso, o software portátil precisa ter alto desempenho, semelhante ao software tipicamente escrito para a plataforma destino;
2. *desempenho*: o ambiente Java apesar de ser uma plataforma interpretada, funciona apenas com pequenos arquivos de classes binários compilados em código de máquina virtual Java, obtendo assim alto desempenho. Outro recurso fundamental que esse ambiente oferece para aumento de desempenho é o *multithreading*;
3. *distribuição*: o rápido crescimento das redes de computadores coincidiu com um interesse cada vez maior pela computação distribuída. Esta disciplina cuida dos problemas relacionados com um software distribuído por vários computadores. As questões da computação distribuída incluem a

comunicação interprocessos, o processamento concorrente, o compartilhamento e a replicação de dados e ainda a segurança. Uma característica essencial dos aplicativos Cliente/Servidor, segundo [NEW97] deve ser a capacidade de compartilhar informações e carga de trabalho de processamento de dados. Segundo ele, o termo “distribuído” descreve o relacionamento entre objetos do sistema, estejam esses objetos em sistemas remotos ou locais. Uma das grandes vantagens dos aplicativos desenvolvidos em Java é que eles podem abrir e acessar objetos dentro da Web através de URL com a mesma facilidade com que podem acessar um sistema de arquivos local;

4. *segurança*: embora sistemas distribuídos sejam extremamente capazes, eles abrem as portas para uma série de problemas de segurança, que são particularmente importantes para uma vasta rede aberta como a Internet. No caso da transmissão normal de dados, você tem que se preocupar com quem possa estar espionando seus dados à medida que eles passam pela rede. Se tiver um servidor na Internet, você terá que se preocupar com a possibilidade de alguém entrar e danificar sua rede interna. Além disso, se estiver transferindo aplicativos da rede, você deve ficar atento à possibilidade destes aplicativos provocarem danos no seu sistema *host*. O ambiente Java foi desenvolvido com um sistema de segurança em várias camadas, cuja presença é percebida em todo o ambiente Java.

2.5 METODOLOGIA DE DESENVOLVIMENTO DE AGENTES DE [WOO96]

Em [WOO96] encontra-se a descrição de uma técnica de modelagem para sistemas multi-agentes denominada por ele mesmo como Técnica Orientada a Agentes. Esta técnica de modelagem de agentes emprega classes de objetos e instâncias destes para descrever os diferentes tipos de agentes que compõem o sistema multi-agentes nos diferentes níveis de abstração. Esta metodologia descreve o sistema pela identificação das classes-chave e o seu domínio na aplicação, bem como especificando seu comportamento e

seu relacionamento com as demais classes que compõe o sistema. Os aspectos essenciais do sistema são expressos por três diferentes tipos de modelos e são eles:

1. um *Modelo Objeto* o qual contém informações sobre os agentes que compõe o sistema, descrevendo sua estrutura de dados, relacionamentos e as operações que suportam;
2. um *Modelo Dinâmico* o qual descreve estados, transições, eventos, ações, atividades e interações os quais caracterizam o comportamento do sistema;
3. um *Modelo Funcional* o qual descreve o fluxo de dados durante a atividade do sistema dentro e entre os seus componentes.

Os modelos dinâmico e funcional servem para prover um refinamento do modelo objeto, em particular o refinamento das operações que cada componente do sistema irá prover. O refinamento das entidades de agentes que compõe o sistema determina a especificação do modelo baseado em objetos. O conceito de objeto é aplicado a todos os níveis de abstração, como uma forma de bem representar a adoção de conjuntos de especialização dos agentes. A especialização implica em tornar explícitos dois aspectos básicos que um sistema orientado a agentes deve atender.

O primeiro aspecto se relaciona ao ponto de vista externo ao sistema. Este deve ser visto como uma coleção de agentes modelados como objetos complexos, caracterizados pelas suas proposições, responsabilidades, serviços, performance, conhecimentos e interações. De acordo com o ponto de vista externo, são estas as características que um agente deve apresentar :

1. um *Modelo Agente* para descrever os relacionamentos hierárquicos entre as diferentes classes sejam concretas ou abstratas e identificar as instâncias que podem coexistir dentro do sistema, bem como sua multiplicidade;
2. um *Modelo de Interação* para descrever as responsabilidades de cada classe de agentes, os serviços que cada classe oferece e o controle dos relacionamentos entre essas classes. Este controle inclui a sintaxe e a

semântica das mensagens usadas para intercomunicação entre os agentes e outros componentes do sistema, como a interface com o usuário.

O segundo aspecto diz respeito ao ponto de vista interno ao sistema. Os elementos requeridos para se definir a arquitetura do sistema devem ser modelados para cada agente, ou seja, cada agente deve ser moldado de acordo com suas crenças, objetivos e planos. A arquitetura do agente deve então obedecer a três princípios básicos os quais descrevem seus estados informacional e motivacional. São eles os seguintes : modelo de crenças, modelo de objetivos e o modelo de planejamento do agente.

1. *modelo de crenças*: descreve o conhecimento do agente sobre o ambiente e o conhecimento sobre o estado interno que um agente pode apresentar, bem como as ações que ele pode executar. As possíveis crenças de um agente e suas propriedades, seja aquelas que o agente pode ou não modificar ao longo do tempo são descritas pelo modelo de crenças, que é composto pelo conjunto de crenças;
2. *modelo de objetivos*: descreve os objetivos que um agente pode adotar e os eventos que ele pode responder. Este modelo é composto do conjunto de objetivos o qual especifica o domínio do objetivo e o dos eventos que o agente pode responder dentro da aplicação, bem como um ou mais estados de objetivo, - que são o conjunto de objetivos básicos do agente;
3. *modelo de planejamento*: descreve os planos que um agente pode empregar para atingir os objetivos descritos no modelo de objetivos ou responder aos eventos que ele pode perceber. O modelo de planejamento é composto pelo conjunto de planos o qual descreve as propriedades e estruturas de controle de cada plano individualmente.

Implícita nesta caracterização (itens 1,2 e 3), estão as propriedades de execução da arquitetura, a qual determina como, exatamente, eventos e objetivos podem suscitar intenção, e intenções podem induzir a ações e revisão de crenças e objetivos. Estas propriedades são responsáveis por garantir que as crenças, objetivos e intenções impliquem em comportamento racional. Esta arquitetura assegura, por exemplo, que os eventos sejam

respondidos em tempo hábil, crenças se mantenham consistentes e a seleção de planos e sua execução proceda de maneira a refletir certo grau de racionalidade.

Outros aspectos são englobados por esta metodologia, como herança de crenças, planos e objetivos, agregação e noção de falha. Cada classe é caracterizada por suas crenças, objetivos e planos, descritas nos respectivos modelos os quais descrevem seu estado interno e comportamento. A herança permite que uma classe de agente seja definida como uma extensão ou uma restrição de outra classe. As crenças, objetivos e planos de uma super-classe podem ser estendidos e especializados na sua sub-classe.

Entretanto, o protótipo a ser aqui especificado implementa uma arquitetura reativa. Desta forma os modelos de crença, objetivo e planejamento bem como agregação e noção de falha não são necessários para atender aos requisitos do sistema aqui proposto. Serão ilustrados então apenas os modelos objeto, dinâmico e funcional, os quais possibilitarão a representação do sistema.

3 O PROTOTIPO

Este capítulo tem como finalidade apresentar o protótipo em sua especificação e implementação, bem como seus testes e validações.

3.1 A DEFINIÇÃO DO PROBLEMA

O contexto que permite a definição do problema a ser abordado neste trabalho está descrito no item 1.1 – A Definição da Proposta de Trabalho. Conforme mencionado neste item, o problema se define da seguinte forma: como ter acesso eficiente e eficaz ao conhecimento médico necessário ao diagnóstico?

A proposta de solução aqui apresentada sugere a busca de um conjunto de ferramentas que possibilite o compartilhamento eficiente e eficaz de conhecimentos médicos entre clínicos geograficamente separados, para que os diagnósticos considerados difíceis sejam construídos de forma mais ágil e precisa, possibilitando maior ganho de tempo e evitando custos excedentes com tratamentos intermediários e não definitivos. Tais ferramentas constituirão um sistema informatizado que utilizará a Internet como suporte à comunicação para intercâmbio de conhecimento.

3.2 A ESTRATÉGIA PARA SOLUÇÃO DO PROBLEMA APRESENTADO

É implementada a sugestão de [HOF98] que prevê a solução do problema com o uso de Agentes Autônomos. Os agentes que comporão o sistema possibilitarão que médicos geograficamente separados utilizem-se da Internet como meio de comunicação para compartilhamento de conhecimentos. Este conhecimento será organizado e representado através do modelo Diagnóstico Clínico e Receita implementado no trabalho de [HOF98].

3.3 A CONTRIBUIÇÃO DO MODELO DIAGNÓSTICO CLÍNICO E RECEITA AOS AGENTES QUE COMPORÃO O SISTEMA

São apresentadas as considerações que tornam o modelo Diagnóstico Clínico e Receita implementado por [HOF98] relevante para o processo de compartilhamento de

conhecimentos enquanto forma de organização e representação do conhecimento a ser compartilhado com o auxílio dos agentes.

Conforme mencionado por [HOF98], para que se possa falar em SAD é necessário conhecer o ambiente da decisão, ou seja, o contexto onde as decisões são tomadas e os problemas enfrentados. Desta forma, um ponto fundamental que determinará a eficiência e a eficácia no intercâmbio de conhecimentos a ser realizado pelos agentes será a forma de organização do conhecimento a ser compartilhado.

Deve-se portanto procurar a forma de melhor organizar este conhecimento, de modo que o tráfego de informações gerado a partir da atividade dos agentes seja o mais efetivo possível. Como forma de ilustrar tais considerações cabem alguns esclarecimentos de caráter técnico relevantes na implementação da solução. São eles:

1. o tratamento para uma mesma doença não é idêntico para diferentes espécies de animais. Em função da constituição física diferenciada de cada espécie, existem variações na condução do tratamento médico a ser executado;
2. o tratamento difere de acordo com a idade e peso do paciente, pois o grau de resistência física para diferentes pesos e idades é diferenciado. Estas duas características determinam como o tratamento deverá ser conduzido no que diz respeito a fatores como tipos de medicamento a serem utilizados, dosagens, duração e outros;
3. o tratamento difere de acordo com alguma sensibilidade natural do animal, sensibilidade esta determinada muitas vezes pela raça. Esta sensibilidade é fator determinante na condução do diagnóstico e do tratamento;
4. o tratamento difere em função de sensibilidades provocadas por traumatismos anteriores, por este motivo o tratamento, muitas vezes, é influenciado pelo histórico médico do animal;
5. o tratamento difere em função da apresentação do animal, como por exemplo fêmeas em estado de gestação.

O modelo Diagnóstico Clínico e Receita implementado por [HOF98] contempla estes aspectos e permite a organização sistemática do conhecimento médico do clínico.

É esta organização sistemática do conhecimento a contribuição deste modelo aos agentes que realizarão o compartilhamento, pois esta organização possibilita otimizar o acesso ao conhecimento por parte dos agentes. Tem-se a seguir a especificação das interfaces propostas por [HOF98] para o modelo Diagnóstico Clínico e Receita.

3.3.1 A INTERFACE MAPEAMENTO DO CONHECIMENTO

Observa-se na figura 3.1 a associação dos sintomas que podem ser detectados em um paciente acometido de varíola. Nesta interface são estabelecidos os relacionamentos entre a enfermidade e seus elementos, que são causas, sintomas, tratamentos e seus sub-ítem. Observa-se que é possível relacionar ao tratamento para varíola os medicamentos indicados para sua cura bem como exames de acompanhamento recomendados e dietas aconselhadas. Os agentes, conforme anteriormente mencionado utilizarão o conhecimento organizado através desta interface.

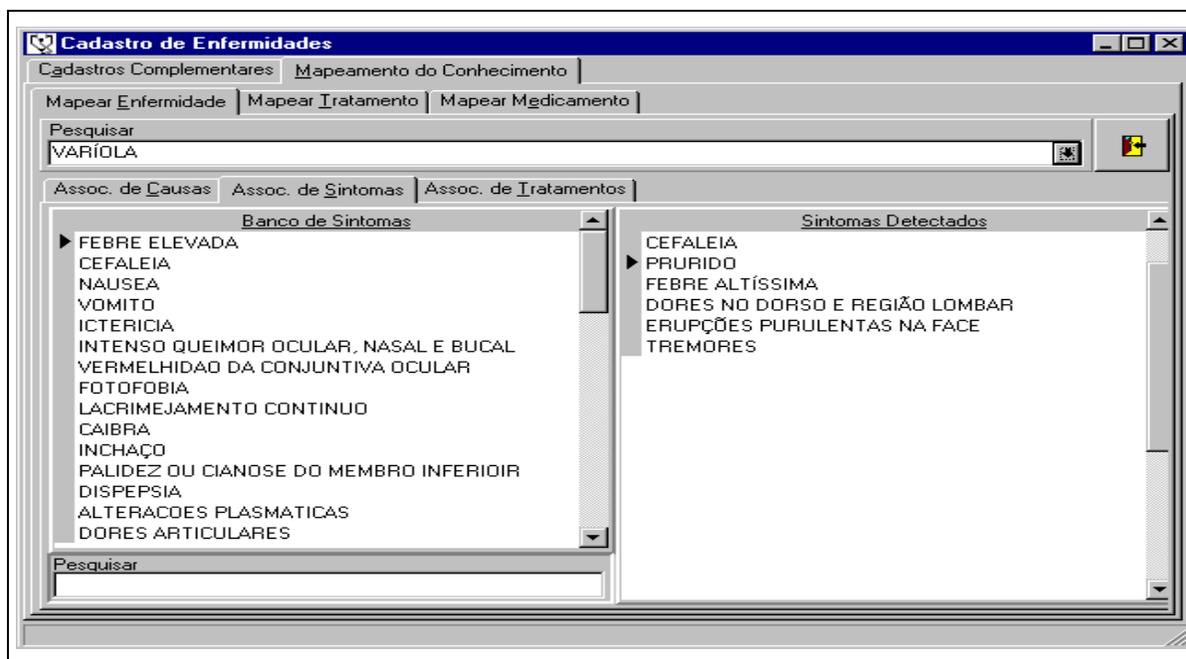


Figura 3.1 Interface Mapeamento do Conhecimento

São descritas a seguir as entidades do subsistema de dados do modelo Diagnóstico Clínico e Receita implementado por [HOF98]. Estas entidades contêm o

conhecimento organizado a partir da interface Mapeamento do Conhecimento contida neste modelo. Estas entidades de dados serão utilizadas pelos agentes em seu processo de pesquisa para compartilhamento de conhecimentos. Por este motivo torna-se relevante sua especificação:

- a) *doenças*: contém os nomes das doenças;
- b) *sintomas*: contém os nomes dos sintomas;
- c) *causas*: contém a descrição de todas as causas de doenças;
- d) *procedimentos*: contém os nomes de todos os procedimentos médicos;
- e) *tratamentos*: contém os nomes de todos os tratamentos médicos;
- f) *dietas*: contém a descrição de todas as dietas;
- g) *medicamentos*: contém os os nomes de todos os medicamentos;
- h) *indicações*: contém a descrição das indicações para os medicamentos;
- i) *restrições*: contém a descrição das contra indicações para os medicamentos;
- j) *doenças - sintomas*: esta é a entidade normalizadora que relaciona no modo [1->n] todos os sintomas para cada doença;
- k) *doenças - tratamentos*: esta é a entidade normalizadora que relaciona na forma [1->n] todos os tratamentos indicados para cada doença;
- l) *doenças - causas*: esta é a entidade normalizadora que relaciona no formato [1->n] todas as causas para cada doença;
- m) *tratamentos - medicamentos*: esta é a entidade normalizadora que relaciona na forma [1->n] todos os medicamentos indicados para cada tratamento;
- n) *tratamentos - procedimentos*: esta é a entidade normalizadora que relaciona no formato [1->n] todas os procedimentos clínicos aplicados para cada tratamento médico;

- o) *tratamentos - dietas*: esta é a entidade normalizadora que relaciona no formato [1->n] todas as dietas indicadas para cada tratamento médico;
- p) *medicamentos - indicações*: esta é a entidade normalizadora que relaciona no formato [1->n] todas as indicações para cada medicamento;
- q) *medicamentos - restrições*: esta é a entidade normalizadora que relaciona no formato [1->n] todas as contra indicações para cada medicamento;
- r) *perfis*: esta entidade relaciona cada tratamento especificado ao perfil específico a que ele se destina. Este perfil determina características do paciente como espécie, peso e idade.

Na figura 3.2, encontra-se o diagrama representando as entidades do subsistema de dados do modelo Diagnóstico Clínico e Receita. Para cada entidade são apresentados os seus respectivos relacionamentos com as demais entidades do modelo. A descrição dos dados para cada entidade pode ser encontrada nos anexos 1 e 2.

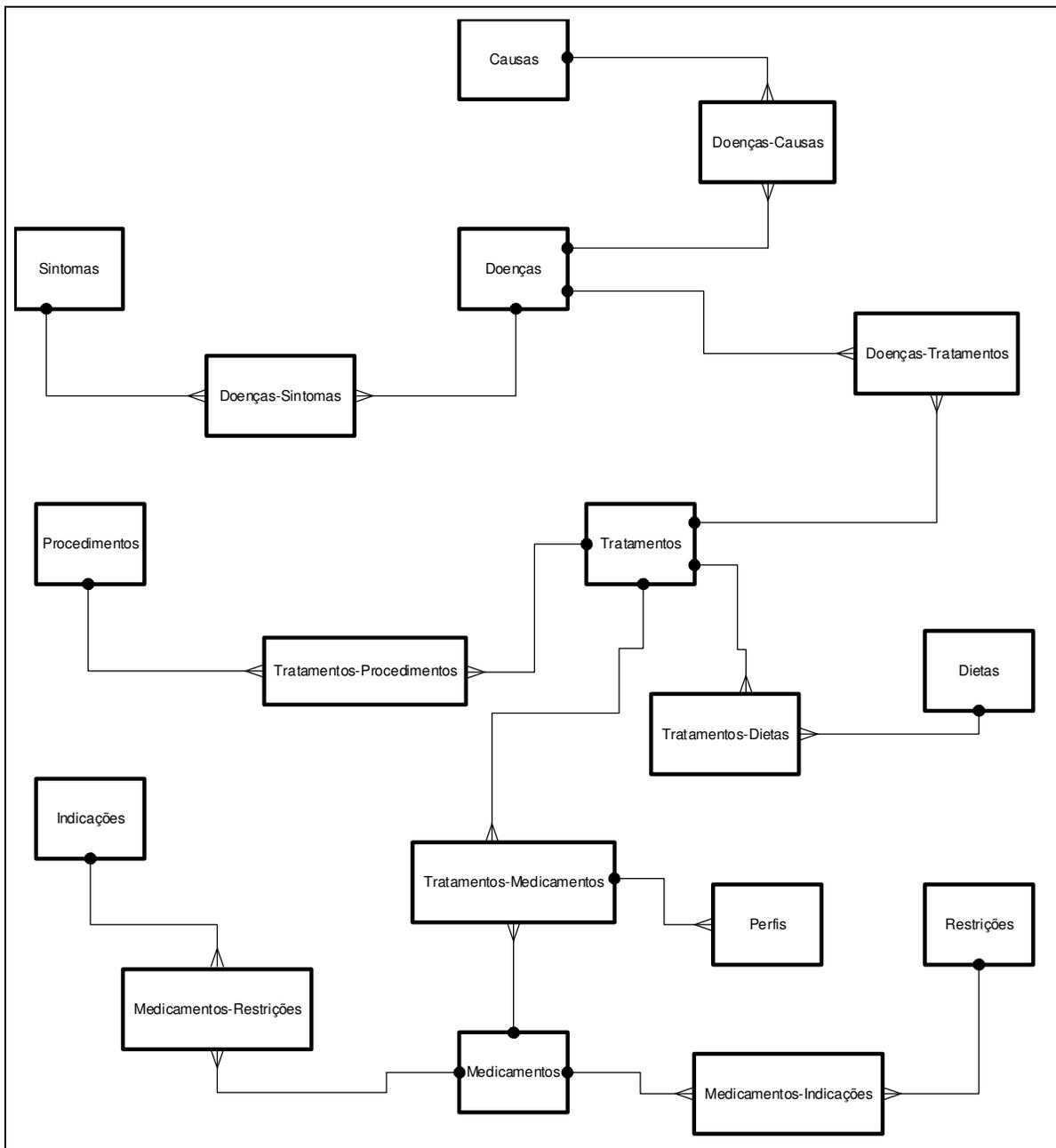


Figura 3.2 Entidades do subsistema de dados

3.3.2 A INTERFACE DIAGNÓSTICO CLÍNICO E RECEITA

Torna-se importante que os agentes que executarão ações efetivas sobre a base de conhecimento (pesquisa e transmissão dos resultados) conheçam esta estrutura, de modo a otimizar o acesso determinando assim a eficácia e a eficiência do sistema.

O conhecimento desta estrutura por parte dos agentes é especificado por um conjunto de regras que são implementadas na interface Diagnóstico Clínico e Receita, que também dá nome ao modelo. Estas regras são especificadas com base nos relacionamentos das entidades do subsistema de dados deste modelo e descrevem a representação do conhecimento a ser compartilhado pelos agentes.

As regras descritas na figura 3.3 especificam um ciclo de busca completo, ou seja, as regras aplicáveis para que, informando-se os sintomas, possam ser recuperadas informações que determinem as possíveis enfermidades relacionadas aos sintomas informados, seus tratamentos e demais informações acessórias.

Se (Febre e Anorexia e Prurido) então Varíola
Se (Meteorismo e Dispepsia) então Febre do Feno.
Se Febre do Feno então Início Pega Tratamento Febre do Feno; Pega Dieta Febre do Feno; Pega Exames Febre do Feno Fim
Se Varíola então Início Pega Tratamento Varíola; Pega Dieta Varíola; Pega Exames Varíola. Fim.

Figura 3.3 Regras aplicadas à representação do conhecimento

A interface Diagnóstico Clínico e Receita implementada por [HOF98] disponibiliza o Banco de Sintomas. O Banco de Sintomas é constituído pela entidade Sintomas que compõe o subsistema de dados do modelo e como anteriormente mencionado, contém um cadastro genérico de sintomas possíveis de serem detectados durante o exame ao paciente.

Nesta interface cabe ao profissional informar os sintomas detectados durante o exame ao paciente. Feito isto, é realizada uma pesquisa ao subsistema de dados com base na sintomatologia apresentada. As prováveis enfermidades são apresentadas ao usuário. Selecionando as enfermidades apresentadas obtêm-se os tratamentos associados a ela, os quais podem ser indefinidos em número, de acordo com o conhecimento especificado pelo profissional. Selecionando o(s) tratamento(s) apresentado(s), obtêm-se a descrição deste tratamento. Esta descrição se compõe dos medicamentos utilizados, dosagens, dietas e demais informações complementares que tenham sido acrescentadas durante a construção da base de conhecimento. A figura 3.4 representa a interface Diagnóstico Clínico e Receita.

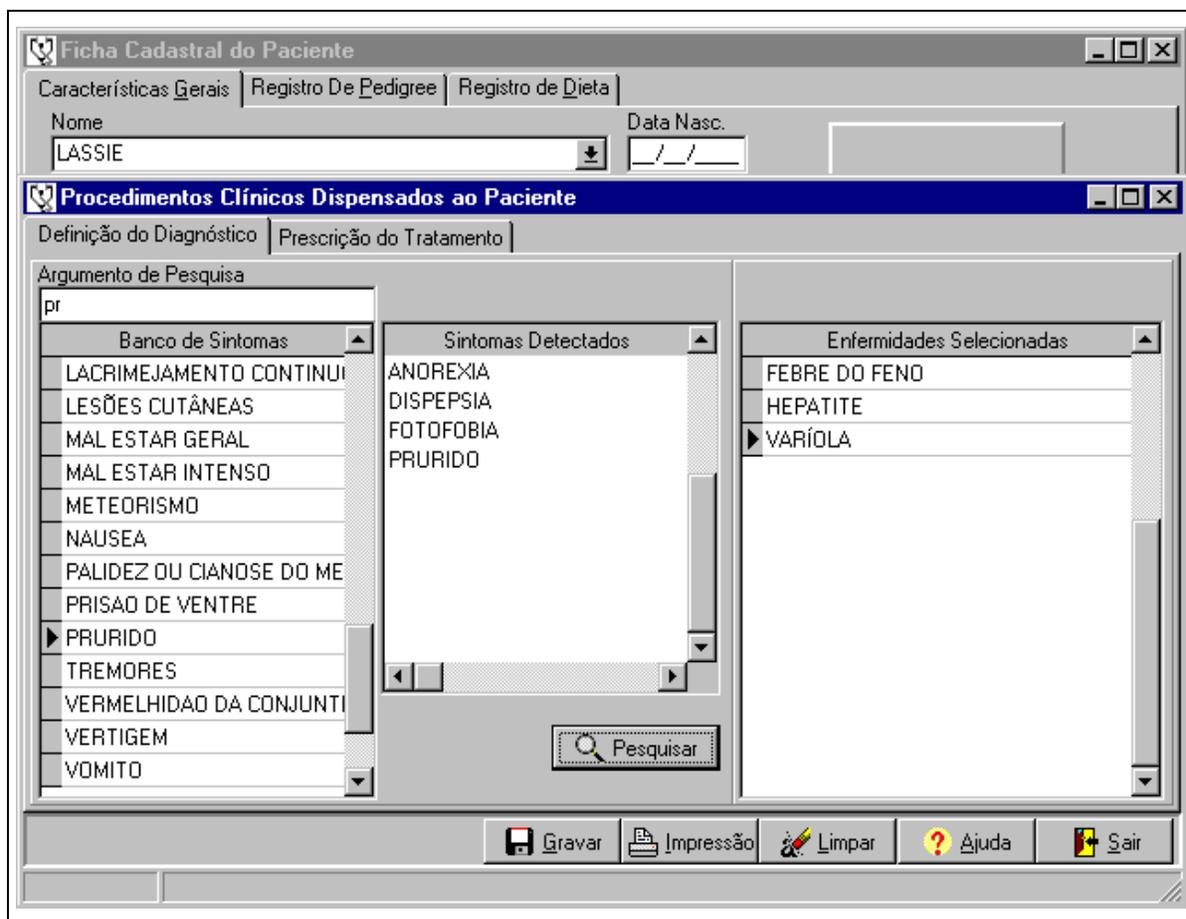


Figura 3.4 Interface Diagnóstico Clínico e Receita

Observa-se na figura 3.4 as prováveis enfermidades levantadas a partir da informação da sintomatologia apresentada pelo paciente. Observa-se então que os sintomas anorexia, dispepsia, fotofobia e prurido podem ser indicativos de três tipos diferentes de

enfermidades que, conforme mostradas na figura acima são febre do feno, hepatite e varíola. Esta interface representa a pesquisa à base de conhecimentos local, alimentada pelo próprio usuário.

3.4 APLICANDO AGENTES NA BUSCA DO CONHECIMENTO

A arquitetura adotada como já mencionado será a arquitetura reativa, e como tal obedecerá a dois princípios básicos expostos por [FAR98] e que caracterizam essa arquitetura. Estes princípios já foram mencionados no cap. 2 e são novamente descritos abaixo:

1. uma arquitetura reativa se caracteriza por não incluir nenhum modelo do mundo simbólico e também não é utilizado raciocínio simbólico complexo, somente reação às ações que ocorrem no ambiente;
2. uma arquitetura reativa deve ser vista como um conjunto de módulos que operam de forma autônoma e são responsáveis em executar uma determinada tarefa.

Por este motivo, o protótipo a ser implementado se constituirá de agentes com tarefas específicas. Estes agentes não utilizarão qualquer método de processamento simbólico que traduza raciocínio lógico, nem construção ou realização de metas e objetivos, isto porque conforme visto, tais propriedades são aplicáveis às arquiteturas deliberativas.

Os agentes que comporão o protótipo apenas responderão a estímulos vindos de outros agentes, determinando um conjunto de ações. Estas ações desencadearão um fluxo de dados entre os agentes que tem como finalidade a busca do conhecimento médico especificado pelo usuário, determinando assim o compartilhamento deste conhecimento entre clínicos geograficamente separados.

3.4.1 A IDENTIFICAÇÃO DAS TAREFAS A SEREM EXECUTADAS PELO SISTEMA DURANTE O PROCESSO DE BUSCA DE CONHECIMENTO

A princípio, um problema que possua algum grau de complexidade pode ser dividido em partes mais simples. Nestes casos, o somatório das partes compõem o problema em sua totalidade, ou o Problema Global a ser resolvido.

Sob esta ótica, feita uma análise da complexidade do problema que este trabalho se propõe a resolver (item 3.1 – A Definição do Problema), pode-se identificar várias partes distintas que o compõem. Algumas destas partes podem ser descritas conforme segue e podem ser denominadas como as tarefas a serem executadas pelo sistema durante o processo de busca e compartilhamento de conhecimentos. São elas:

- a) definir a interface com o usuário que requisita conhecimento para apresentação dos resultados obtidos com a pesquisa;
- b) monitoramento das bases de conhecimento da rede, evitando que sejam direcionadas pesquisas a bases que tenham se tornado inativas;
- c) atualização dos endereços das bases de conhecimento que se tornam ativas ou se tornam inativas na rede para que o processo de pesquisa possa ser eficiente;
- d) controle da segurança nos acessos às bases de conhecimento, para que não haja acessos não permitidos;
- e) distribuição do conhecimento relativo às permissões de acesso;
- f) pesquisa efetiva com base na requisição vigente, endereços a pesquisar e permissões de acesso;
- g) monitoramento do processo de pesquisa para fins de garantia de performance do sistema;
- h) transmissão dos resultados obtidos.

Os itens acima expressam apenas os pontos essenciais a serem considerados na implementação da solução. As demais questões a serem consideradas não foram aqui especificadas pelo fato de estarem relacionadas de forma mais direta ao aperfeiçoamento do modelo.

3.4.2 A IDENTIFICAÇÃO DAS ENTIDADES DE AGENTES A SEREM CRIADAS

Como forma de possibilitar a execução das tarefas levantadas no item anterior, sugere-se a criação de várias entidades de agentes conforme segue:

1. *agente Cliente (AgC);*
2. *agente de Pesquisa (AgP);*
3. *agente Roteador (AgR);*
4. *agente RegistradorAtividade (AgA);*
5. *agente RegistradorDesativação (AgE);*
6. *agente TransmissorPermissões (AgT);*
7. *agente ServidorLocal (AgS);*
8. *agente TransmissorDados (AgD).*

Estas entidades de agentes comporão o sistema. A divisão de tarefas para cada entidade de agente será definida e descrita a seguir.

3.4.3 A ASSOCIAÇÃO ENTRE AS ENTIDADES DE AGENTES E SUAS TAREFAS

Identificadas as várias partes que compõem o problema global, é necessário delegar tarefas aos agentes envolvidos no processo, para que cada entidade de agente realize suas ações de acordo com um objetivo comum.

3.4.3.1 REGISTRO DE ATIVAÇÃO DA BASE DE CONHECIMENTO

Esta tarefa envolve três entidades de agentes: *AgC*, *AgR* e *AgA*. Ela tem como finalidade realizar o registro de ativação de uma nova base de conhecimento. O registro de ativação implica em informar-se ao *AgR* o endereço desta base e sua identificação nominal ou nome do *host*. Este registro significa que a base de conhecimento registrada está disponível para futuras pesquisas. As ações se desenvolvem da seguinte maneira:

- a) o *AgC*, quando é ativado pelo usuário se comunica com o *AgR*, informando sua identificação e seu endereço, para que o *AgR* saiba que há uma nova base de conhecimento ativa na rede;
- b) o *AgR* então ativa o *AgA*, para que este forneça o serviço de registro de ativação para esta base de conhecimento. Esta base de conhecimento torna-se então disponível para eventuais consultas.

3.4.3.2 REQUISIÇÃO DAS PERMISSÕES DE ACESSO

Esta ação envolve novamente três tipos de entidades de agentes: *AgC*, *AgR* e *AgT*. A finalidade é fornecer ao *AgC* uma lista com as atuais bases de conhecimento ativas bem como seus endereços. Cabe esclarecer que todas as bases de conhecimento referentes aos endereços recebidos pelo *AgC* requisitante estão remotamente localizadas. A requisição das permissões de acesso é executada sempre que o *AgC* recebe parâmetros do usuário para realizar a busca de conhecimento. As ações que determinam o recebimento das permissões de acesso pelo *AgC* são as seguintes:

- a) mediante uma requisição de conhecimento feita pelo usuário, o *AgC*, que é a interface com este usuário, solicita informações de acesso ao *AgR*;
- b) o *AgR* possui uma lista das bases de conhecimento ativas na rede, bem como as permissões de acesso a essas bases de conhecimento. O *AgR* ativa um segundo agente, o *AgT*, que fornecerá o serviço requisitado pelo *AgC* que é a transmissão de todas as permissões de acesso vigentes naquele momento.

3.4.3.3 A BUSCA DO CONHECIMENTO

Estando consciente dos endereços das bases de conhecimento ativas na rede, bem como das permissões de acesso que lhe são dadas, o *AgC* ativa o *AgP*. Este agente realizará a comunicação efetiva com todas as bases de conhecimento ativas naquele momento, requisitando o conhecimento desejado e monitorando o retorno das informações. Todas as bases de conhecimento consultadas pelo *AgP* têm agregado um *AgS*, o qual é responsável por possibilitar a comunicação e interpretar a requisição recebida executando-a. O desenvolvimento das ações é dados conforme segue:

- a) o *AgS*, quando do recebimento de uma requisição de pesquisa, ativa um segundo agente, o *AgT*, que tem como tarefas a busca efetiva de informações e a transmissão física dos resultados da pesquisa ao seu destino, que é o *AgP* requisitante;
- b) o *AgP* aguarda o retorno das informações requisitadas, monitorando constantemente todos os *AgT* para poder se certificar que estão ativos e que não superaram o tempo máximo estipulado como expectativa de retorno.

3.4.3.4 TRANSMISSÃO DOS RESULTADOS OBTIDOS

Cabe ao *AgP* requisitante monitorar os *AgT* designados para a coleta das informações requisitadas por ele, questionando-os se tem ou não informações prontas para serem enviadas. À medida que os *AgT* concluem a sua tarefa, os dados são enviados ao seu endereço de destino. Os resultados são recebidos pelo *AgP* requisitante. Tendo recebido todos os dados enviados pelos *AgT* designados, o *AgP* agora é eliminado pelo *AgC* que o criou e este fica novamente à disposição para outra requisição de conhecimento.

3.4.3.5 A DESATIVAÇÃO DA BASE DE CONHECIMENTO

Quando ocorrer a desativação do *AgC*, este informará ao *AgR* sua intenção. O *AgR* por sua vez, ativa o *AgE* que fornecerá este serviço. O *AgE* eliminará o registro de ativação desta base de conhecimento eliminando seu endereço da lista de endereços de bases de conhecimento ativas.

3.4.4 A ESPECIFICAÇÃO DAS ENTIDADES DE AGENTES – A APLICAÇÃO DA METODOLOGIA DE [WOO96]

3.4.4.1 O MODELO DE RELACIONAMENTOS

Conforme citado na metodologia de [WOO96], é necessário especificar as entidades de agentes bem como sua estrutura de dados e as operações que elas suportam e também seus relacionamentos com as demais entidades que compõem o sistema. Este diagrama é inspirado na metodologia de [WOO96]. Aqui são descritas as entidades chave que vão compor o sistema, bem como as operações que essas entidades suportam e seus relacionamentos. Nota-se na figura 3.5 que os relacionamentos existentes entre as entidades de agentes são feitas apenas na forma de associações.

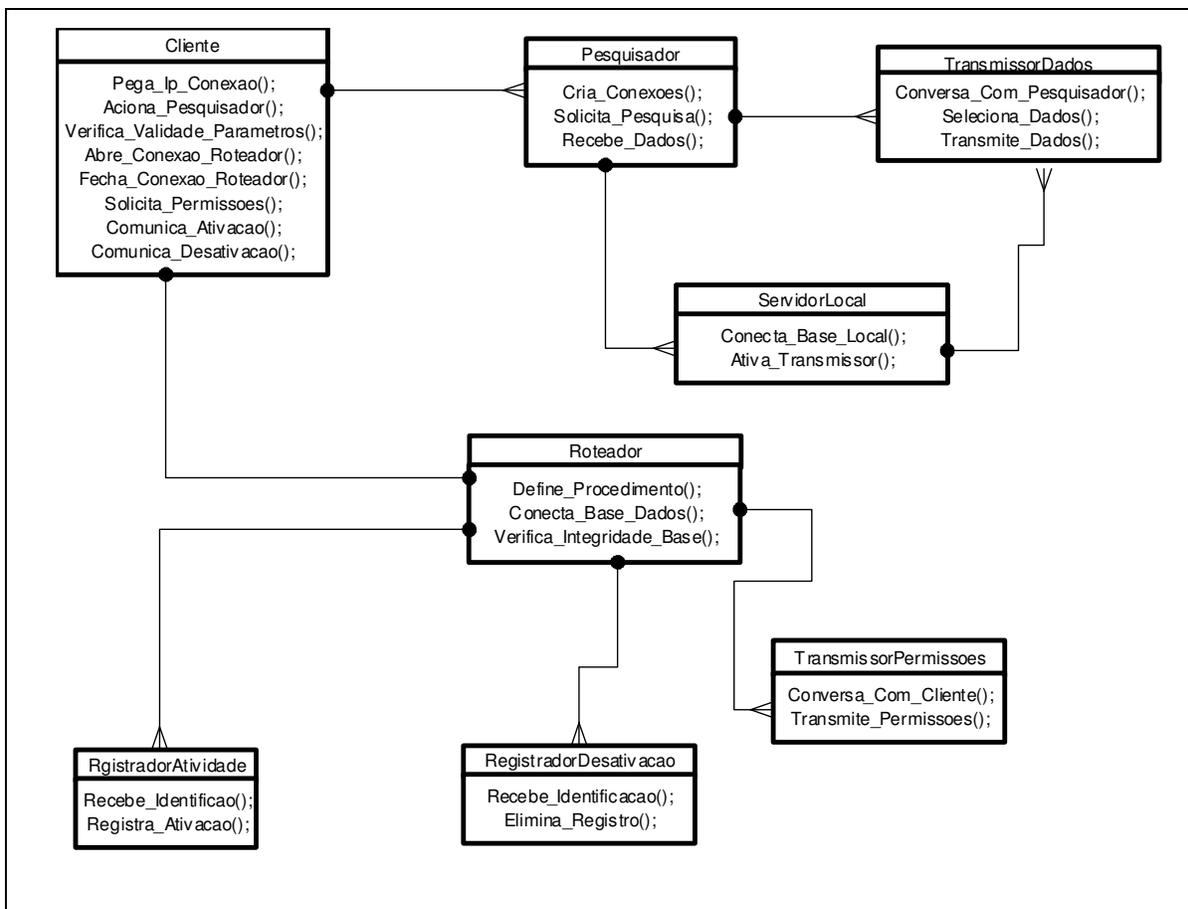


Figura 3.5 Modelo de Relacionamentos

3.4.4.2 O MODELO FUNCIONAL

Como descrito na metodologia, tem-se a figura 3.6 que caracteriza o fluxo de dados entre as classes de agentes.

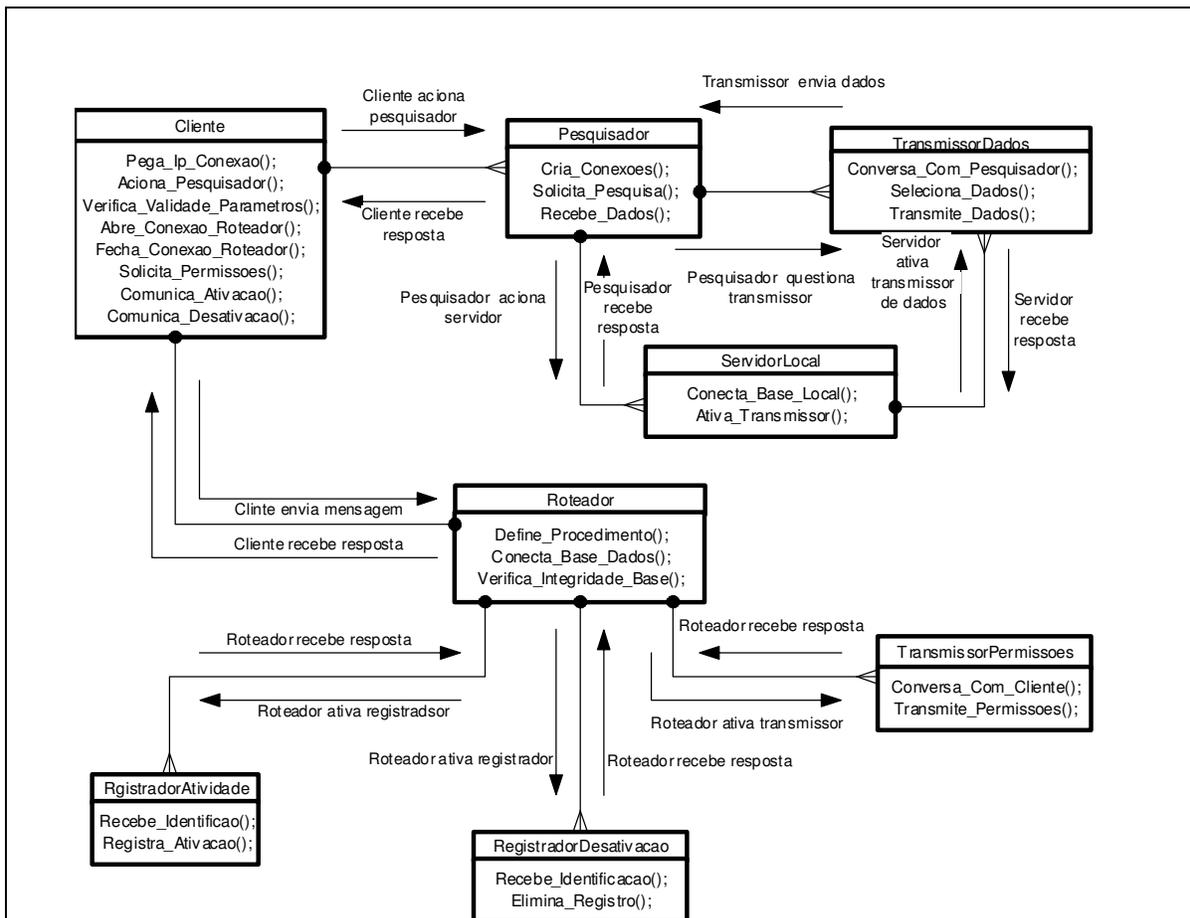


Figura 3.6 Modelo Funcional

3.4.4.3 O MODELO DINÂMICO

De acordo com a metodologia de [WOO96], aqui estão representadas as atividades que caracterizam o comportamento do sistema. Nota-se na figura 3.7 que as interações que ocorrem entre as entidades de agentes caracterizam a arquitetura reativa, pois

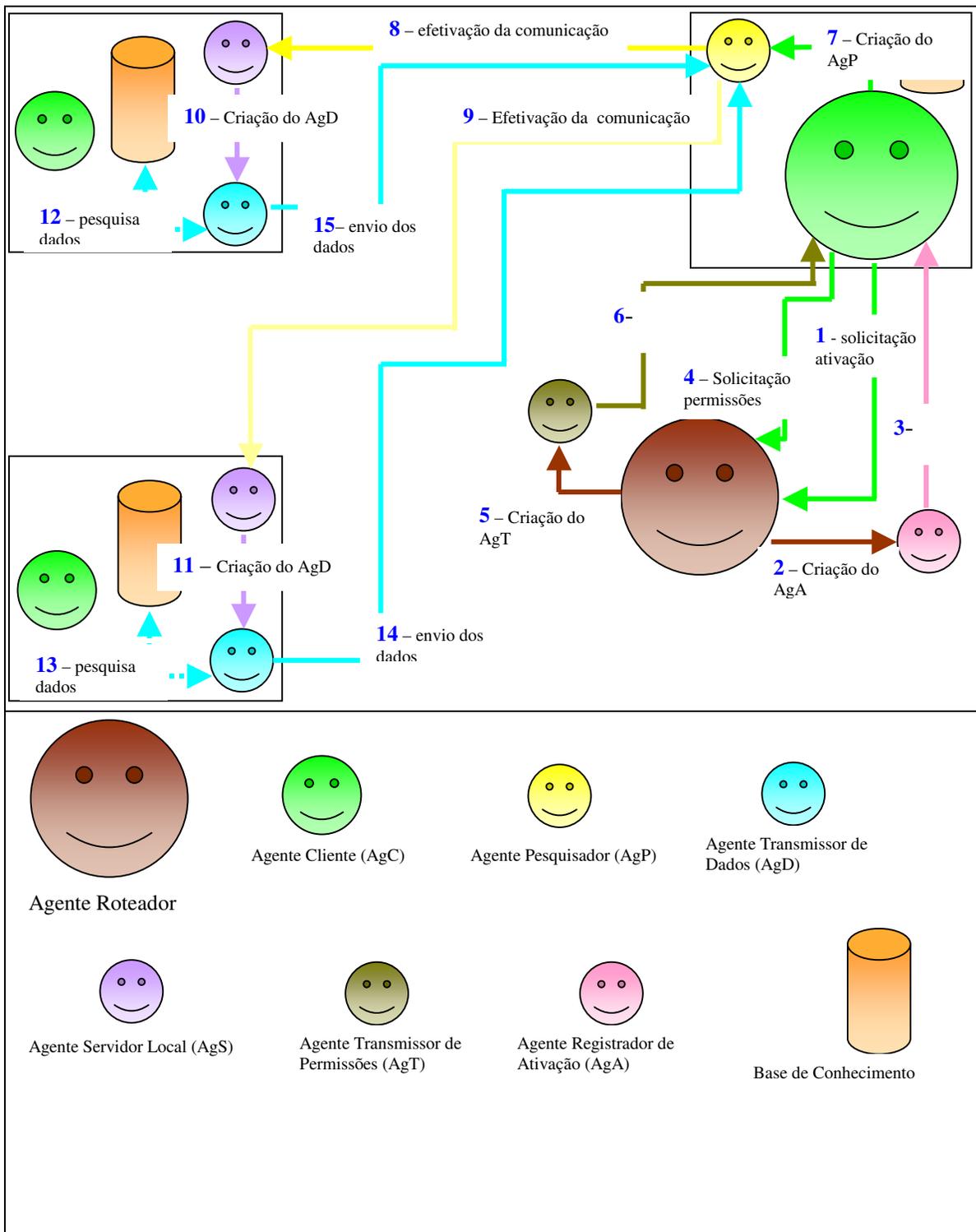


Figura 3.8 O comportamento do sistema

3.4.4.4 OS ESTADOS DE CADA AGENTE

São demonstrados a seguir os possíveis estados de cada entidade de agente durante a atividade do sistema.

3.4.4.4.1 O AGENTE CLIENTE

Este item demonstra os possíveis estados do agente Cliente. Conforme pode ser visto na figura 3.9, este agente pode assumir dois estados que são disponível ou ocupado. Este agente está presente em tempo integral pois entre outras tarefas anteriormente citadas ele representa também a interface com o usuário.

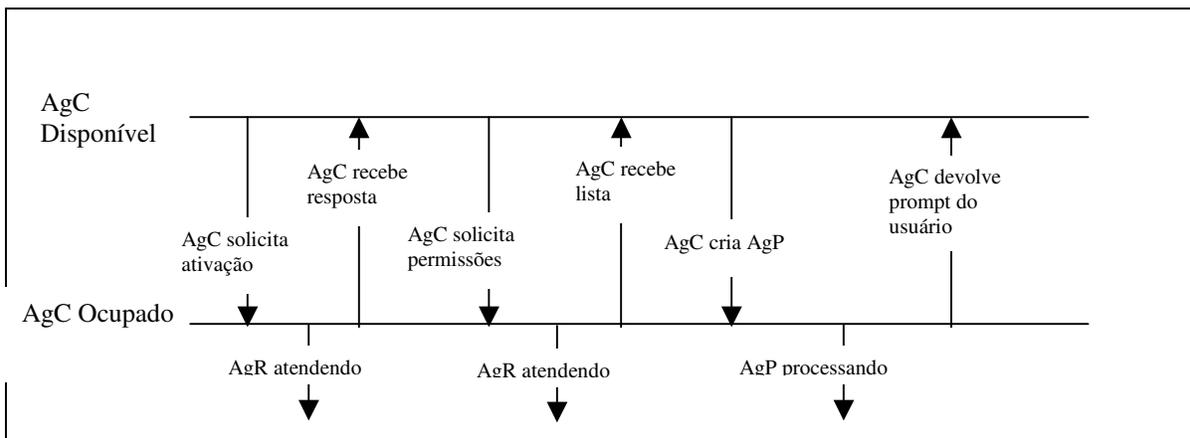


Figura 3.9 Estados do agente Cliente

3.4.4.4.2 O AGENTE PESQUISADOR

Este agente apresenta um único estado durante a execução de suas tarefas que são, conforme anteriormente mencionado, estabelecer a comunicação com as bases de conhecimento remotas e monitorar o retorno de resultados da pesquisa. A figura 3.10 representa o agente Pesquisador. Tendo sido completada sua tarefa esta entidade de agente é então destruída.

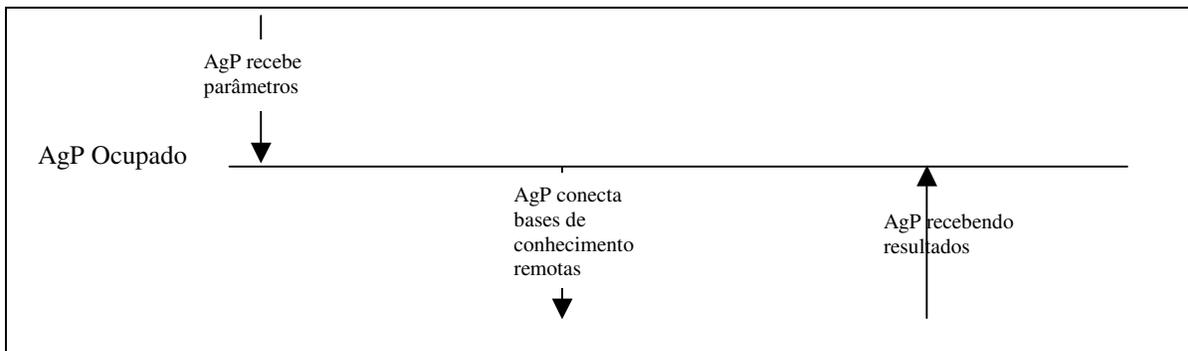


Figura 3.10 Estados do agente Pesquisador

3.4.4.4.3 O AGENTE ROTEADOR

A figura 3.11 representa o agente Roteador. Este agente apresenta sempre o estado disponível, de modo que várias requisições de prestação de serviço possam ser recebidas e atendidas de forma simultânea. Esta entidade tem sua localização remota ou seu endereço na rede conhecido por todos os AgC, de forma que possa ser sempre localizado por estes a qualquer momento.

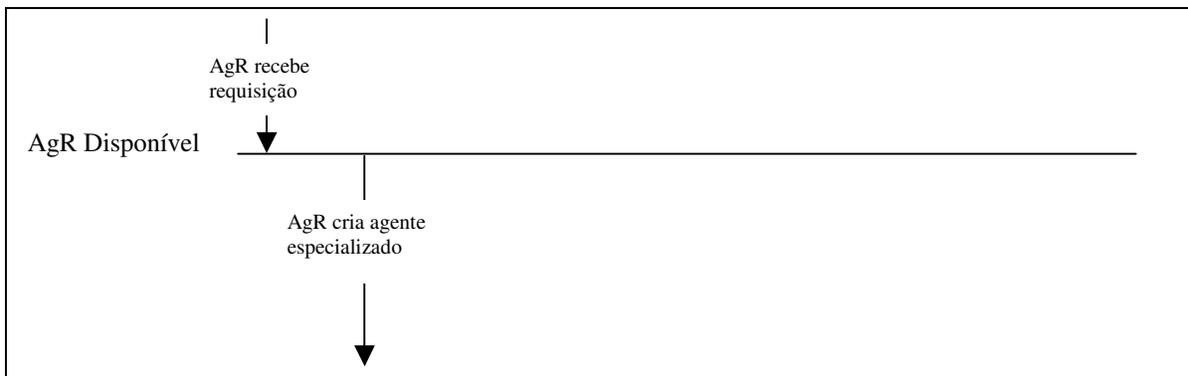


Figura 3.11 Estados do agente Roteador

3.4.4.4.4 O AGENTE REGISTRADOR DE ATIVAÇÃO

Este agente especializado apresenta apenas o estado ocupado durante a execução de sua tarefa que é registrar a conexão de uma nova base de conhecimento à rede. Esta convenção pode ser observada na figura 3.12. Tendo sido completada sua tarefa esta entidade de agente é então destruída.

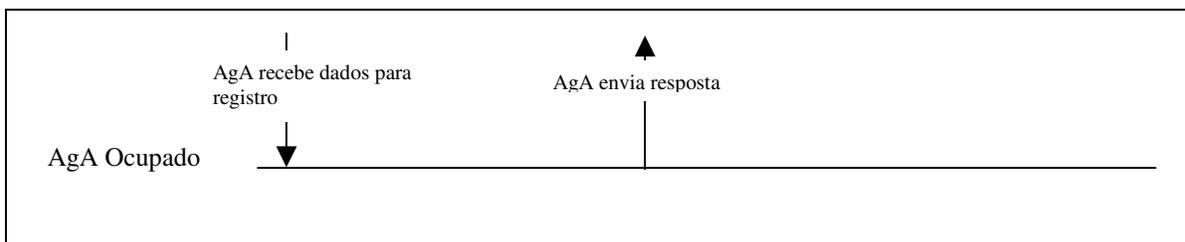


Figura 3.12 Estados do agente Registrador de Ativação

3.4.4.4.5 O AGENTE TRANSMISSOR DE PERMISSÕES

Este agente apresenta apenas um único estado durante a execução de sua tarefa que é a transmissão das permissões de acesso ao AgC requisitante. Esta convenção está representada na figura 3.13. Tendo sido completada sua tarefa esta entidade de agente é então destruída.

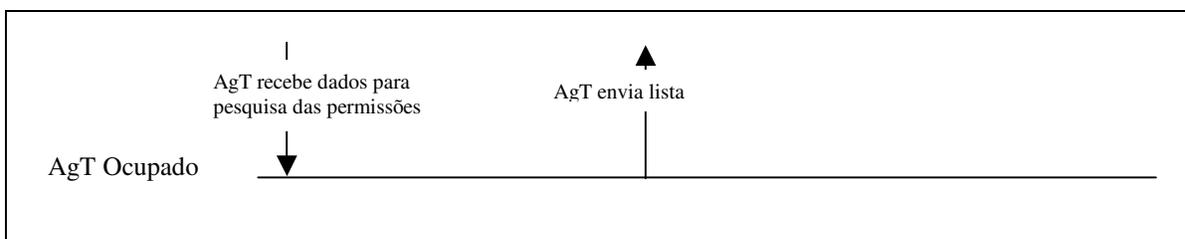


Figura 3.13 Estados do agente Transmissor de Permissões

3.4.4.4.6 O AGENTE REGISTRADOR DE DESATIVAÇÃO

Este agente permanece ocupado durante a execução de sua tarefa que é retirar do arquivo de ativos a base de dados requisitante da desativação, conforme pode ser visto na figura 3.14. Tendo sido completada sua tarefa esta entidade de agente é então destruída.

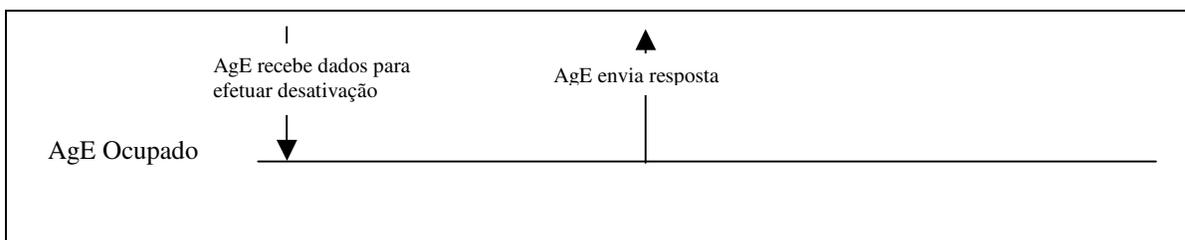


Figura 3.14 Estados do agente Registrador de Desativação

3.4.4.4.7 O AGENTE SERVIDOR LOCAL

Esta entidade de agente, conforme pode ser visto na figura 3.15, atua similarmente ao agente Roteador, ou seja, está sempre disponível para atendimento a várias requisições simultaneamente. Este agente está ativo em tempo integral junto à base de conhecimento por ele representada e cria tantos *AgD* quantas forem as requisições de conhecimento recebidas.

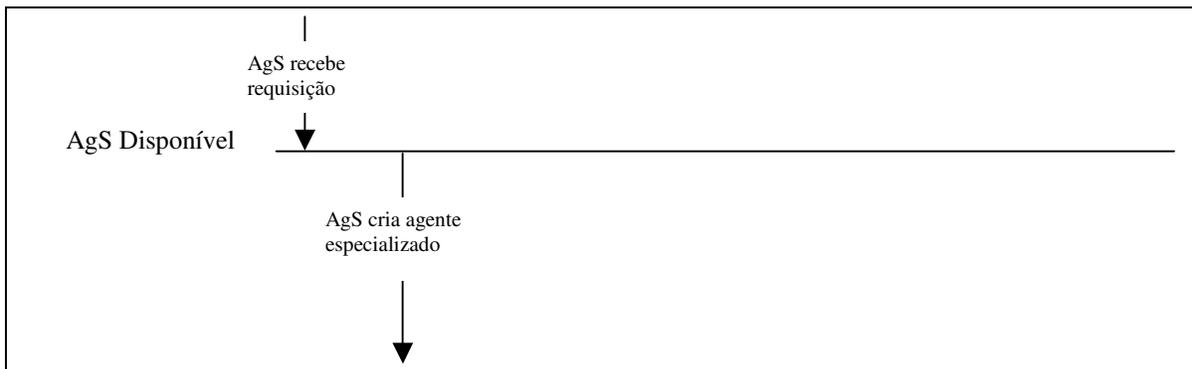


Figura 3.15 Estados do agente Servidor Local

3.4.4.4.8 O AGENTE TRANSMISSOR DE DADOS

Este agente especializado apresenta apenas um único estado durante a execução de sua tarefa que é pesquisar e transmitir os dados requisitados. Esta convenção pode ser observada na figura 3.16. Tendo sido completada sua tarefa esta entidade de agente é então destruída.

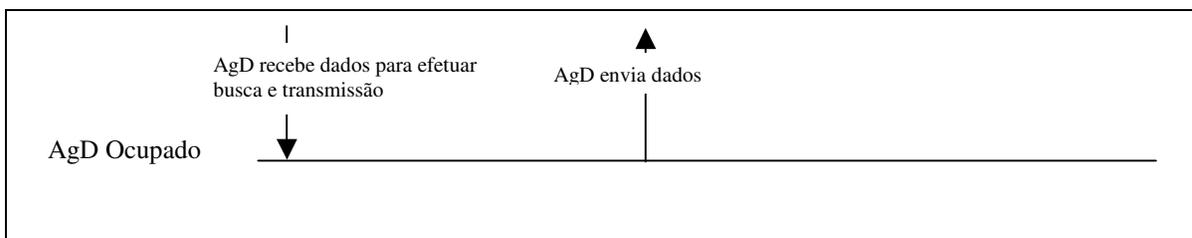


Figura 3.16 Estados do agente Transmissor de Dados

3.4.5 A IMPLEMENTAÇÃO

As tecnologias envolvidas no processo de implementação são abaixo relacionadas:

- linguagem de programação Java na implementação dos agentes;
- JDBC para acesso ao banco de dados;
- HTML na formatação dos resultados da pesquisa;
- Internet como meio físico de transmissão para os resultados.

3.4.5.1 A IMPLEMENTAÇÃO DOS AGENTES

A implementação dos agentes foi realizada utilizando-se a linguagem de programação Java. A linguagem Java ofereceu todos os recursos que possibilitaram a comunicação e troca de informações entre os agentes envolvidos no processo de compartilhamento de conhecimentos.

Os métodos `DataInputStream` e `PrintStream`, chamados *streams* de dados, possibilitaram estabelecer a troca de informações entre as entidades de agente remotamente localizadas. Isto foi possível porque estes métodos implementam o protocolo de comunicação TCP-IP utilizado na Internet. Desta forma, quando é enviada uma informação através da rede utilizando-se o método `PrintStream`, está se enviando um pacote de dados cujo formato atende aos requisitos do protocolo de comunicação de rede TCP-IP implementado na Internet. A figura 3.17 demonstra a utilização destes método na implementação do protótipo.

Um dos pontos fundamentais que possibilitou a operacionalização do sistema foi a definição de métodos que possibilitam a cada base de conhecimento capturar e informar ao agente Roteador o endereço IP provisório adquirido a cada conexão à Internet. Desta forma, quando uma base de conhecimento torna-se ativa, ou seja, estabelece sua conexão com a Internet, o agente Cliente localizado nesta base de conhecimento tem a capacidade de tomar conhecimento do endereço IP provisório obtido na conexão e informar este endereço ao agente Roteador. Estando consciente do endereço IP das bases de conhecimento ativas na rede, o agente Roteador tem condições de distribuir este conhecimento a quem requisitar, de forma que as próprias bases de conhecimento possam agora localizar umas às outras na rede.

Apesar da transparência ao protocolo de comunicação da rede implementada nos *streams* de dados, foi necessário desenvolver um protocolo de comunicação entre os agentes, de forma a organizar a comunicação entre as entidades para obter-se os resultados desejados. O protocolo de comunicação entre os agentes pode ser também observado na figura 3.17. Como forma de exemplificar a necessidade de organização na comunicação entre os agentes pode-se considerar o agente Pesquisador. Este agente tem entre outras tarefas já citadas a responsabilidade de monitorar o retorno dos resultados da pesquisa. Sendo que o retorno desses resultados pode significar receber transmissões simultâneas de várias bases de conhecimento pesquisadas, este protocolo previne que resultados de diferentes bases de conhecimento sejam misturadas em um mesmo arquivo de resultados. Cabe lembrar que a convenção adotada no sistema é que cada base de conhecimento pesquisada em uma determinada ocasião gere seu respectivo arquivo de resultados e os dados obtidos em cada base pesquisada permaneçam neste arquivo sem serem misturados com os demais resultados obtidos das demais bases pesquisadas. O protocolo de comunicação entre agentes é aplicado a todos os agentes que realizem trocas de informações com outros agentes do sistema permitindo a organização da comunicação entre as entidades.

```
Public void Comunica_Ativacao(DataInputStream ImpStream, PrintStream OutStream)
throws IOException
    DataInputStream ImpStream = null;
    PrintStream OutStream = null;
    {
        try
        {
            String Retorno = new String();
            OutStream.println("Informando ativação...");
            System.out.println("Enviando-> Informando ativação...");
            If ((Retorno = ImpStream.readLine()).equals("Registrador pronto..."))
            {
                System.out.println("Recebido-> " + Retorno);
                OutStream.println(Ip_Cliente);
                System.out.println("Enviando-> " + Ip_Cliente + "...");
                OutStream.println(Id_Cliente);
                System.out.println("Enviando-> " + Id_Cliente + "...");
                OutStream.println("Fim");
                System.out.println("Enviando-> Fim");
            }
            if ((Retorno = ImpStream.readLine()).equals("Sucesso"));
            {
                System.out.println("Registro de ativacao realizado com sucesso...");
            }
        }
    }
```

Figura 3.17 Streams de Dados

A Classe `ServerSocket` oferece métodos que permitem as múltiplas conexões entre as entidades de agentes. Tem-se como exemplo o agente Roteador, o qual pode receber várias solicitações de prestação de serviço e ainda assim estar disponível para novas conexões. Segue na figura 3.18 um pequeno trecho de código utilizado na implementação e que permite o caráter servidor do agente Roteador.

```
Public void run()
{
    while (true)
    {
        Sock = ServerSock.accept();
        Define_Procedimento(Sock, Connect);
    }
}
```

Figura 3.18 Uso da Classe `ServerSocket`

3.4.5.2 O JDBC PARA ACESSO AO BANCO DE DADOS

O JDBC (Java Data Base Connection) é uma API (Application Program Interface) de nível de chamada para Java, cuja função é dar independência de banco de dados aos desenvolvedores. Na implementação do protótipo foi utilizada uma ponte ODBC-JDBC. Esta ponte é um driver que fornece uma interface entre o padrão ODBC do banco Microsoft Access utilizado na implementação de [HOF98] e o JDBC utilizado pelo Java. É através desta ponte ODBC-JDBC que os agentes têm acesso aos dados das bases de conhecimento a serem consultadas. Na figura 3.19 está o código que permite a conexão entre o JDBC e o padrão ODBC presente no banco de dados a ser utilizado.

```
Public void Conecta_Com_a_Base_De_Dados() throws SQLException,
ClassNotFoundException
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    System.out.println("Iniciando Roteador...");
    System.out.println("Conectando-se ao banco de dados.Aguarde...");
    Try
    {
        Connect = DriverManager.getConnection("jdbc:odbc:BdRoteador", "barth",
"barth");
    }
    catch (Exception e)
    {
        System.out.println("Erro!! Conexão banco de dados : " + e);
        Connect.close();
    }
}
```

Figura 3.19 Ponte ODBC-JDBC

A função do JDBC é executar instruções SQL (Sequencial Query Language) e enviar os resultados ao usuário. Este processo é realizado através de classes abstratas e seus métodos. Estes métodos executam uma consulta SQL, recuperam uma única linha de um conjunto de resultados e retornam um objeto String, respectivamente. A figura 3.20 mostra a utilização dos métodos oferecidos pelo JDBC e utilizados na implementação

```
Public void Verifica_Integridade_Da_Base() throws SQLException,
ClassNotFoundException
{
    Statement Stment = Connect.createStatement();
    ResultSet Rset = null;
    Try
    {
        System.out.println("Verificando integridade da base de dados do
roteador.Aguarde..." );
        Rset = Stment.executeQuery("Select * from TbPermissoes,TbClientes");
        System.out.println("Base de dados do roteador Ok..." );
    }
    catch (Exception e)
    {
        System.out.println("Erro!! Base de dados do roteador corrompida : " +
e);
    }
    Stment.close();
    Rset.close();
}
```

Figura 3.20 SQL e JDBC

3.4.5.3 INTERNET COMO MEIO FÍSICO DE TRANSMISSÃO

A Internet é o único meio físico de transmissão entre as entidades de agentes remotamente localizadas. A efetiva troca de informações foi possível conforme mostra o item 3.6.4 – Testes e Validações. Não foram necessários conhecimentos aprofundados a respeito de redes de comunicação, visto que a linguagem Java implementa os aspectos complexos desta tecnologia, deixando-os transparentes ao desenvolvedor.

3.4.5.4 HTML NA FORMATAÇÃO DOS RESULTADOS

Os resultados da pesquisa foram formatados utilizando-se HTML (HiperText Markup Language). O HTML possibilitou a transferência dos resultados na forma textual como apresentada mais adiante na figura 3.26. Cabe observar que o formato textual dos arquivos de resultados é uma convenção do sistema e serve para ilustrar as inúmeras fontes de conhecimento possíveis de serem consultadas em um único processo de pesquisa. Deve-

se reconhecer que este não é o formato ideal a ser apresentado. Sob a perspectiva de SAD, estes dados deveriam ser combinados de forma a gerarem gráficos onde seriam representados percentuais ou probabilidades de ocorrência. Entretanto, cabe aqui reforçar o formato atual como uma mera convenção do sistema para comprovar a pesquisa em várias bases de conhecimento de forma simultânea.

3.4.6 TESTES E VALIDAÇÕES

Foram realizados testes com a intenção de verificar a funcionalidade da implementação realizada. Estes testes foram realizados estabelecendo-se a comunicação entre três microcomputadores PC's localizados em pontos diferentes do bairro Vila Itoupava e também no laboratório Protem, localizado na FURB - Universidade de Blumenau.

Estes PC's serão aqui chamados Base de Conhecimento 1 Vila (*Bc1V*), Base de Conhecimento 2 Vila (*Bc2V*) e Roteador Protem (*RP*). Cabe observar que não existe qualquer meio físico de transmissão disponível entre estes três PC's, como seria em uma conexão em rede via cabo coaxial. Entretanto *RP* é um servidor de Internet enquanto *Bc1V* e *Bc2V* estão conectados a provedores de acesso. A Internet foi o único recurso utilizado como forma de comunicação entre as três plataformas, sendo que o referido teste figura ao longo do desenvolvimento do texto a seguir.

A figura 3.21 representa o *AgR*, localizado em *RP* e cujo endereço remoto é conhecido pelos *AgC* de *Bc1V* e *Bc2V*. Conforme anteriormente mencionado, o *AgR* não executa as tarefas de registro de ativação, transmissão das permissões de acesso e desativação da base de conhecimento. O *AgR* cria agentes especializados em cada uma destas tarefas para que possa estar sempre disponível para o atendimento aos *AgC* que se conectam a qualquer momento. Desta forma, o número de agentes especializados criados pelo *AgR* em um dado momento será sempre igual ao número de *AgC* que solicitam serviços. Estes agentes executam suas tarefa e são então destruídos.

```
Microsoft(R) Windows 95
(C) Copyright Microsoft Corp 1981-1996.
C:\WINDOWS>cd..
C:\>cd protot~1
C:\Prototipo>java roteador
Iniciando Roteador...
Conectando-se ao banco de dados.Aguarde...
Verificando integridade da base de dados do roteador.Aguarde...
Base de dados do roteador Ok...
Roteador escutando na porta 9802...
```

Figura 3.21 *AgR* localizado em *RP* aguardando conexão do *AgC* de *BcV1*

A figura 3.22 representa a comunicação entre o *AgC* de *BcV1* e o *AgR* de *RP*. Nesta figura o *AgC* de *BcV1* estabeleceu uma conexão com *AgR* de *RP* e com base no protocolo de comunicação pré determinado para organizar a comunicação enviou seus dados de endereço e identificação para que fosse feito seu registro de ativação. O registro de ativação foi realizado com sucesso e de acordo com a convenção adotada na especificação são dois os efeitos:

- a) *BcV1* tornou-se disponível para consultas por parte de *BcV2* e;
- b) *BcV2* pôde ser localizada por *BcV1*, sendo que a esta altura *BcV2* já havia requisitado sua ativação ao *RP*;

```
Microsoft(R) Windows 95
(C) Copyright Microsoft Corp 1981-1996.

C:\WINDOWS>cd..
C:\>cd protot*1
C:\Prototipo>java Cliente
Conexao com o Roteador estabelecida com sucesso...
Enviando-> Informando ativação...
Recebido-> Registrador pronto...
Enviando-> 192.168.1.2...
Enviando-> Carlos...
Enviando-> Fim
Registro de ativacao realizado com sucesso...

..... Sistema de Consultoria Pela Internet.....

Para termino da aplicacao digite 'Fim'
Para executar busca digite:
'Pesquisar-> <Especie>;<Peso>;<Idade>;<Sintoma1, Sintoma2,..., SintomaN>'
```

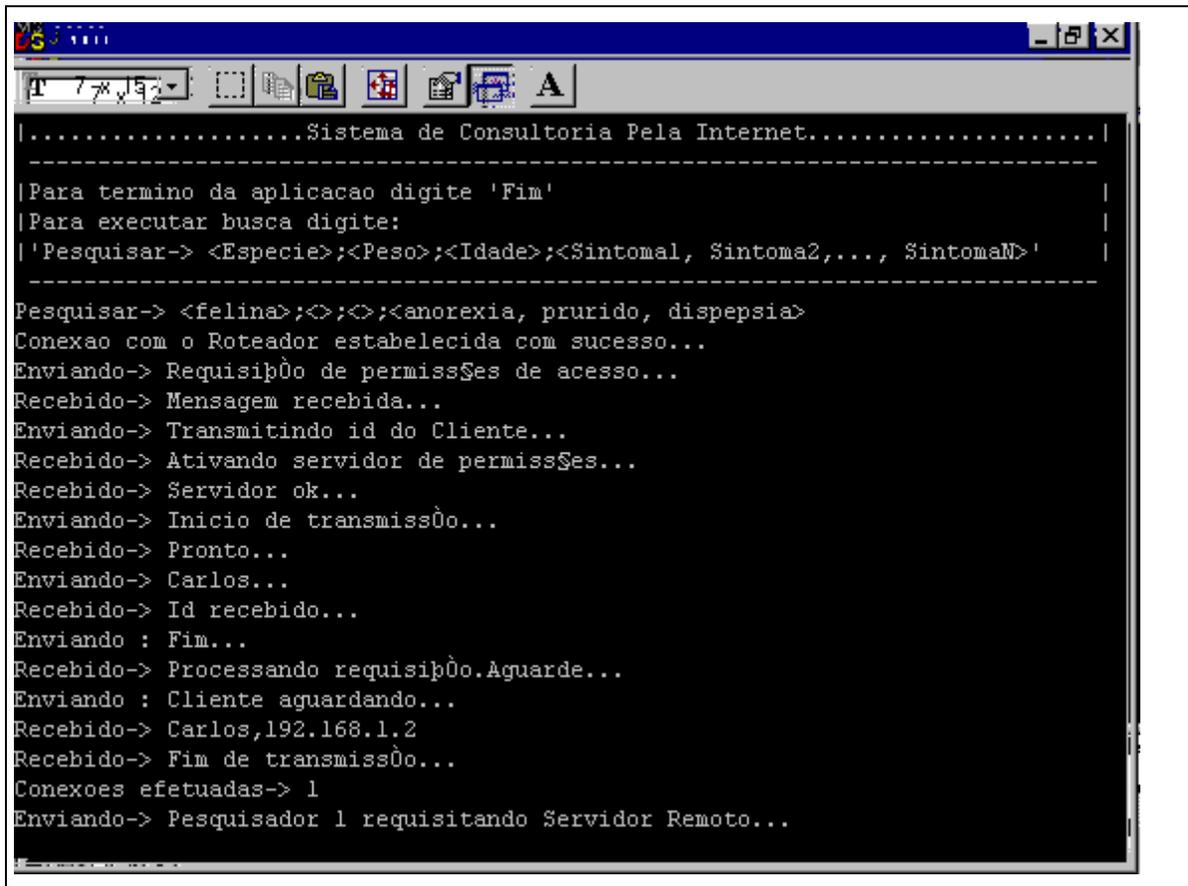
Figura 3.22 AgC de *BcVI* informando ativação ao AgR de *RP*

Durante os testes o processo de pesquisa partiu simultaneamente dos AgC localizados em *BcVI* e *BcV2*. A figura 3.23 mostra o AgC de *BcVI* requisitando as permissões de acesso disponíveis a ele neste momento. Nesta situação específica a lista de permissões de acesso relativas a *BcVI* será composta apenas do endereço de *BcV2*.

A figura 3.23 demonstra também o desenvolvimento da comunicação entre os três tipos de agentes envolvidos no processo de requisição de permissões de acesso de *BcVI*. São eles:

- AgC requisitante das permissões localizado em *BcVI*;
- o AgR localizado em *RP* e o AgT criado por ele para transmissão das permissões de acesso.

O desenvolvimento da comunicação foi executado com sucesso e o AgC de *BcVI* que requisitou as informações obteve uma lista com todos os endereços das bases de conhecimento ativas no momento da consulta. A partir desta lista foram executadas ações que efetivaram o processo de busca e compartilhamento de conhecimento.



```
.....Sistema de Consultoria Pela Internet.....
|Para termino da aplicacao digite 'Fim'
|Para executar busca digite:
|'Pesquisar-> <Especie>;<Peso>;<Idade>;<Sintoma1, Sintoma2,..., SintomaN>'
|-----
Pesquisar-> <felina>;<>;<>;<anorexia, prurido, dispepsia>
Conexao com o Roteador estabelecida com sucesso...
Enviando-> Requisição de permissões de acesso...
Recebido-> Mensagem recebida...
Enviando-> Transmitindo id do Cliente...
Recebido-> Ativando servidor de permissões...
Recebido-> Servidor ok...
Enviando-> Inicio de transmissão...
Recebido-> Pronto...
Enviando-> Carlos...
Recebido-> Id recebido...
Enviando : Fim...
Recebido-> Processando requisição.Aguarde...
Enviando : Cliente aguardando...
Recebido-> Carlos,192.168.1.2
Recebido-> Fim de transmissão...
Conexoes efetuadas-> 1
Enviando-> Pesquisador 1 requisitando Servidor Remoto...
```

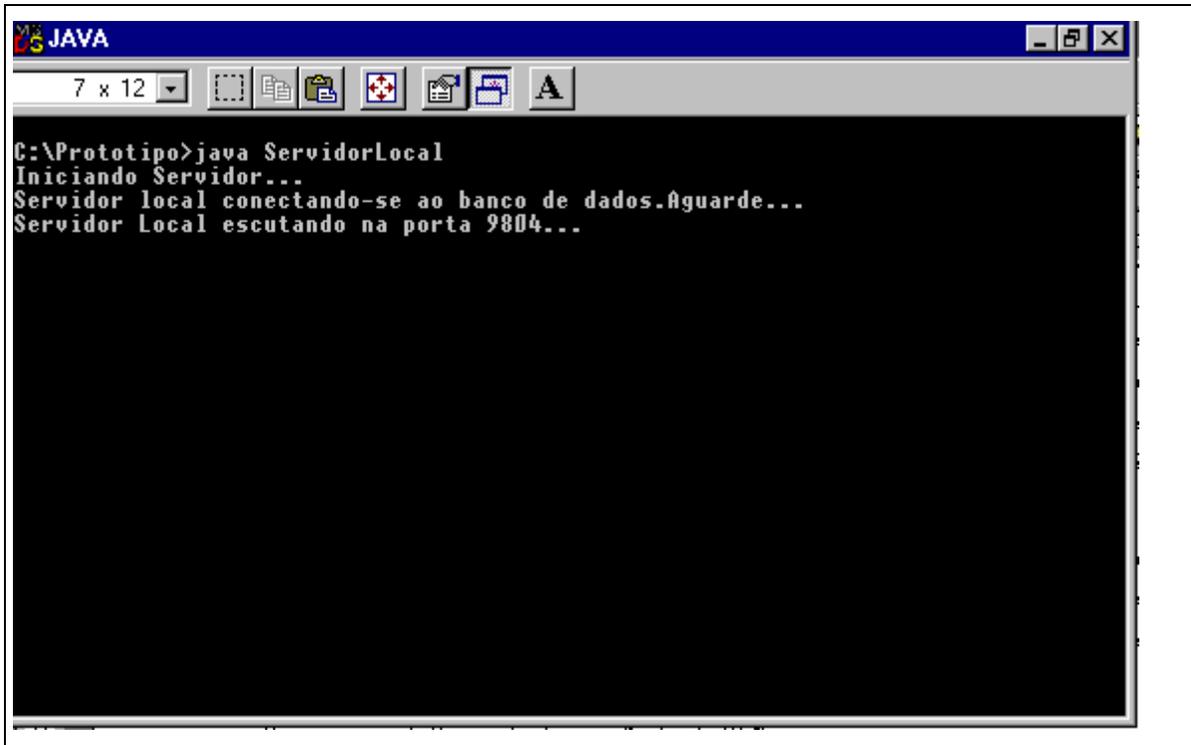
Figura 3.23 *AgC* de *BcVI* requisitando permissões de acesso ao *AgR* de *RP*

A figura 3.24 representa o *AgS* localizado em *BcVI* na ocasião dos testes. Como anteriormente mencionado, existe um *AgS* agregado a cada base de conhecimento ativa em um determinado momento.

Desta forma, tem-se um *AgS* em *BcVI* e um *AgS* em *BcV2*. Entretanto, da mesma forma que o *AgR* de *RP*, os *AgS* em *BcVI* e *BcV2* não executam a tarefa de busca e transmissão de conhecimento, mas apenas criam agentes especializados. Assim, os *AgS* estarão sempre disponíveis para atendimento a todos os *AgP* que requisitarem conhecimento.

Então, haverá tantos *AgT* ativos e agindo sobre uma mesma base de conhecimento em um determinado momento quanto forem os *AgP* requisitantes de

informações. Na ocasião específica do teste realizado havia apenas um *AgT* em *BcV1* e *BcV2*, visto que ambas pesquisaram apenas uma à outra de forma simultânea.



```
C:\Prototipo>java ServidorLocal
Iniciando Servidor...
Servidor local conectando-se ao banco de dados.Aguarde...
Servidor Local escutando na porta 9804...
```

Figura 3.24 *AgS* de *BcV1* aguardando conexão do *AgP* de *BcV2*

A figura 3.25 mostra o desenvolvimento da comunicação e que permitiu aos agentes envolvidos no processo a troca de informações necessárias à efetivação do processo. Nesta figura pode-se observar o desenvolvimento do protocolo de comunicação entre agentes criado para organizar o processo de comunicação.

```

JAVA
T 7 x 15
Enviando-> Recebido
Recebido->      REMEDIO PARA VARIOLA,
Enviando-> Recebido
Recebido-> <B><FONT SIZE="2" COLOR="#339933">INDICACOES :</FONT></B>
Enviando-> Recebido
Recebido->      IND. VARIOLA,
Enviando-> Recebido
Recebido-> <B><FONT SIZE="2" COLOR="#339933">CONTRA INDICACOES :</FONT></B>
Enviando-> Recebido
Recebido->      CONTRA IND. VARIOLA,
Enviando-> Recebido
Recebido-> </BODY>
Enviando-> Recebido
Recebido-> </HTML>
Enviando-> Recebido
Fim de arquivo...
Processo de pesquisa finalizado...
-----
|.....Sistema de Consultoria Pela Internet.....|
-----
|Para termino da aplicacao digite 'Fim'          |
|Para executar busca digite:                    |
|'Pesquisar-> <Especie>;<Peso>;<Idade>;<Sintoma1, Sintoma2,..., SintomaN>' |
-----

```

Figura 3.25 *AgC de BcVI* recebendo resultados da pesquisa do *AgT de BcV2*

Os resultados recebidos no processo de pesquisa em *BcV2* foram armazenados pelo *AgP de BcVI* em um diretório previamente especificado. Neste diretório foram criados pequenos arquivos cujo conteúdo corresponde às informações obtidas em *BcV2*. Desta forma, serão gerados tantos arquivos de resultados quantas forem as bases de conhecimento pesquisadas. A convenção adotada para formatação do arquivo foi HTML. A justificativa se baseia no fato de se poder ordenar os resultados de forma textual, expressando de forma simples e exata o conhecimento como foi originalmente organizado pelo usuário que alimentou a base de conhecimento pesquisada. A figura 3.26 representa um arquivo de resultados gerado à partir do procedimento de pesquisa realizado por *BcVI* a *BcV2* durante o teste. Nota-se que cada enfermidade traz consigo itens relevantes que possibilitam ao usuário que executou a pesquisa obter o conhecimento organizado na base de conhecimento pesquisada. Este conhecimento é apresentado, como anteriormente mencionado, da forma como foi organizado na base de conhecimento pesquisada.

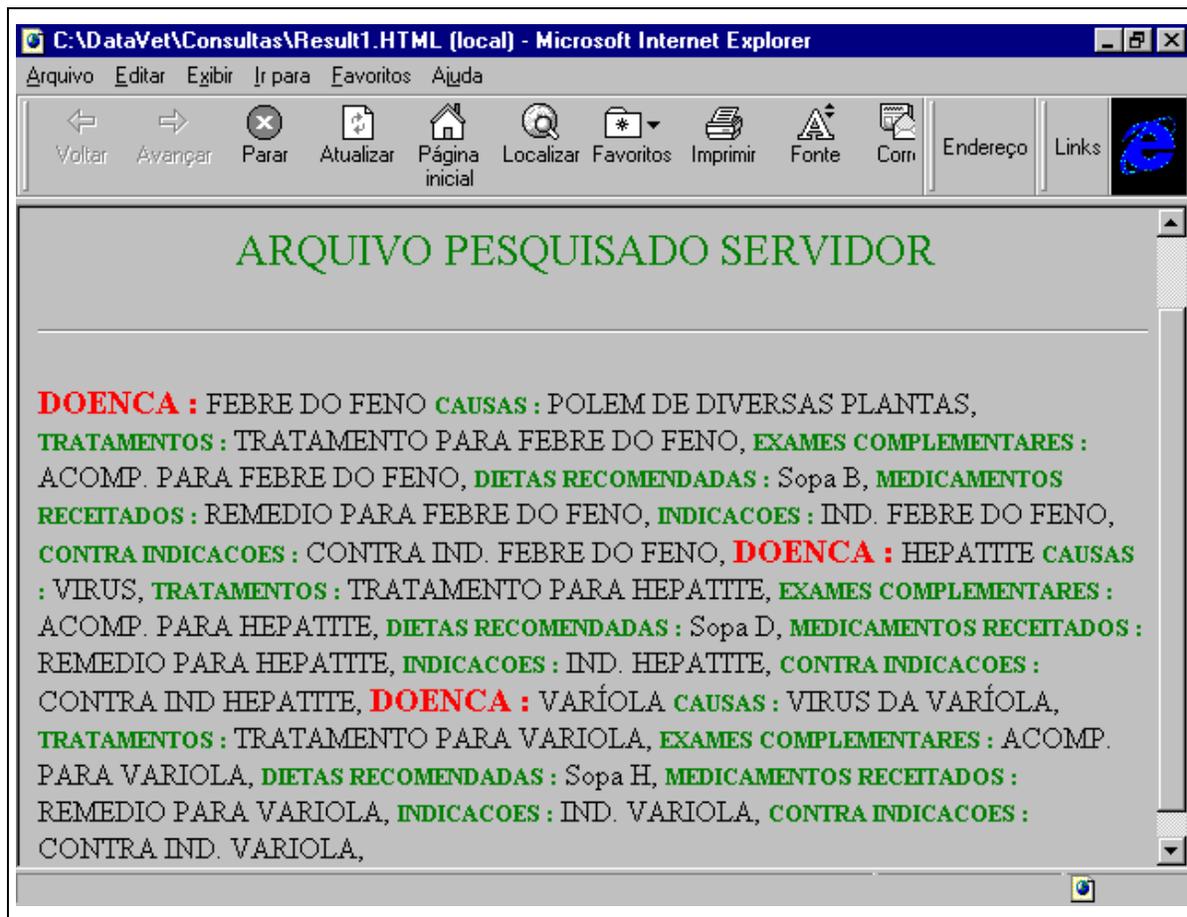


Figura 3.26 Arquivo de resultados gerado pelo processo de pesquisa em *BcV2*

A base de conhecimento *BcVI* tornou-se inativa a partir da requisição feita pelo seu *AgC* e sua solicitação de desativação aparece na figura 3.27.

Foram feitos outros testes e em todos foram obtidos resultados satisfatórios. Estes testes comprovaram:

- a) a capacidade dos agentes *AgS* e *AgR* em atender a várias requisições simultâneas ;
- b) a capacidade do agente *AgP* em efetivar o processo de busca de conhecimentos em várias bases de conhecimento de forma simultânea.

Por este motivo pode-se afirmar que o protótipo obteve funcionalidade em sua implementação.

```
Prompt do MS-DOS
7 x 15
Enviando-> Peso...
Recebido-> Parametro recebido...
Enviando-> Idade...
Recebido-> Parametro recebido...
Enviando-> Sintomas...
Enviando-> Pesquisador aguardando transmissao Servidor1...
Fim de arquivo...
Processo de pesquisa finalizado...
-----
|.....Sistema de Consultoria Pela Internet.....|
-----
|Para termino da aplicacao digite 'Fim'
|Para executar busca digite:
| 'Pesquisar-> <Especie>;<Peso>;<Idade>;<Sintoma1, Sintoma2,..., SintomaN>'
-----
Fim
Conexao com o Roteador estabelecida com sucesso...
Enviando-> Informando desativap0o...
Recebido-> Registrador pronto...
Enviando-> 192.168.1.2...
Enviando-> Carlos...
Enviando-> Fim
Registro de desativacao realizado com sucesso...
C:\Prototipo>
```

Figura 3.27 *AgC de BcVI solicitando sua desativação ao AgR de RP*

4 CONCLUSÕES

A solução implementada ilustra a viabilidade da utilização dos agentes no compartilhamento de conhecimentos. Desta forma, o presente trabalho obteve sucesso em seu principal objetivo que é demonstrar o potencial de utilização dos agentes para possibilitar o compartilhamento de conhecimentos médicos entre profissionais geograficamente separados como auxílio na construção de diagnósticos médicos.

A seguir são feitas algumas considerações a respeito das tecnologias envolvidas na implementação da solução que são a lembrar:

- agentes autônomos na execução das tarefas do sistema;
- SAD na organização do conhecimento a ser compartilhado;
- regras de produção na representação do conhecimento a ser compartilhado;
- Internet como meio de comunicação entre usuário e bases de conhecimento ativas.

No que diz respeito aos agentes, com base nas informações coletadas a partir da literatura pesquisada, foi possível mostrar que no caso da aplicação considerada os agentes tem elevado potencial na resolução de problemas, tanto de âmbito específico quanto genérico. Cabe ressaltar que a literatura a respeito dos agentes, suas propriedades e arquiteturas é volumosa. Entretanto, existe carência de literaturas que demonstrem metodologias aplicáveis à construção de sistemas multi-agentes. Esta carência está relacionada principalmente às arquiteturas deliberativas, que são arquiteturas mais complexas onde podem ser implementadas a realização de metas, planos e objetivos para os agentes, implicando assim o convívio socializado destas entidades e todos os fatores associados a este convívio como por exemplo a propriedade da cooperatividade.

Já o estudo realizado a respeito das características das várias arquiteturas existentes (reativas e deliberativas) levaram à conclusão de que o problema aqui levantado pode ser satisfatoriamente resolvido da forma como aqui foi proposto, ou seja, com base em uma arquitetura reativa. Entretanto, cabe o reconhecimento de que o sistema poderia ser

otimizado a partir da aplicação de uma arquitetura mais robusta, que implementasse algumas propriedades adicionais, como por exemplo persistência de intenção e mobilidade. Estas propriedades permitiriam aos agentes envolvidos no processo de pesquisa um grau maior de autonomia, possibilitando assim o aumento de performance do sistema.

Quanto ao SAD, este se mostrou adequado na organização do conhecimento a ser compartilhado, pois permite ao usuário organizar o seu conhecimento de forma dinâmica. Isto equivale dizer que o usuário que alimenta a base de conhecimento pode expressar através do SAD mudanças e aperfeiçoamentos em seus conceitos e idéias. Tal possibilidade representou significativa contribuição para que se pudesse alcançar o objetivo proposto.

A utilização das regras de produção na representação do conhecimento mostrou-se também adequada, oferecendo aos agentes envolvidos no processo uma especificação que permite a otimização no acesso aos dados organizados através do SAD.

A Internet como meio de comunicação para operacionalização da arquitetura proposta mostrou desempenho satisfatório, servindo como meio físico de transmissão de dados para a comunicação e o compartilhamento efetivo de informações entre várias bases de conhecimento remotamente distribuídas.

Outro aspecto a ser considerado, além da viabilidade tecnológica vista acima, é o aspecto da acessibilidade cada vez maior das pessoas às soluções de Internet. Esta acessibilidade se dá tanto por questões de custo como o barateamento progressivo dos equipamentos de hardware quanto pela atratividade em função do grande volume de informações disponíveis. Pode-se citar também o fato de que é cada vez mais comum o uso de linhas dedicadas à Internet, bem como a existência de uma perspectiva do uso da TV a cabo como meio de conexão à Internet.

Assim, pode-se afirmar que a solução aqui ilustrada e aplicada é viável e também acessível, podendo ser implementada na prática.

5 ANEXOS

Anexo1 Estrutura das entidades do subsistema de dados

NOMES	CAMPOS	INDICES
GR010003 - TbDoenças	1-Doe_ChavePri [I(*)] - chave primária 2-Doe_Descrica [A(50)] - nome da doença	DoeDescrica [2]
GR010004 - TbSintomas	1-Sin_ChavePri [I(*)] - chave primária 2-Sin_Descrica [A(50)] - nome do sintoma	SinDescrica [2]
GR010005 - NdsDoençasSintomas	1-Nds_ChavePri [I(*)] - chave primária 2-Nds_CdDoenca [I()] - chave estrangeira de doença 3-Nds_CdSintom [I()] - chave estrangeira de sintoma	NdsCdDoenca [2] NdsCdDoencaCdSintom [2,3]
GR010006 - TbTratamentos	1-Tra_ChavePri [I(*)] - chave primária 2-Tra_Descrica [A(50)] - nome do tratamento	TraDescrica [2]
GR010007 - NdmTratamentosMedicamentos	1-Ndm_ChavePri [I(*)] - chave primária 2-Ndm_CdTratam [I()] - chave estrangeira de tratamento 3-Ndm_CdMedica [I()] - chave estrangeira de medicamento	NdmCdMedica [3] NdmCdTratamCdMedica [2,3]
GR010008 - TbMedicamentos	1-Med_ChavePri [I(*)] - chave primária 2-Med_CdGrupox [I()] - chave estrangeira de grupo 3-Med_Descrica [A(50)] - nome do medicamento 4-Med_PrinAtiv [M(200)] - principio ativo 5-Med_Fabricsan [A(50)] - nome do fabricante 6-Med_ModoUsar [M(240)] - modo de usar	MedCdGrupo [2] MedDescrica [3]
GR010009 - NdtDoençasTratamentos	1-Ndt_ChavePri [I(*)] - chave primária 2-Ndt_CdDoenca [I()] - chave estrangeira de doença 3-Ndt_CdTratam [I()] - chave estrangeira de tratamento	NdtCdDoenca [2] NdtCdTratam [3] NdtCdTratamCdDoenca [3,2]
GR010010 TbPerfis	1-Per_ChavePri [I(*)] - chave primária 2-Per_CdTratam [I()] - chave estrangeira de tratamento 3-Per_Especie [A(30)] - descrição da espécie 4-Per_MinIdade [I()] - idade mínima 5-Per_MaxIdade [I()] - idade máxima 6-Per_MinPesox [N(2,2)] - peso mínimo 7-Per_MaxPesox [N(2,2)] - peso máximo	TodosCampos [2,3,4,5,6,7]
GR010011 - TbIndicações	1-Ind_ChavePri [I(*)] - chave primária 2-Ind_Descrica [A(50)] - nome da indicação	IndDescrica [2]
GR010012 - TbProcedimentos	1-Pro_ChavePri [I(*)] - chave primária 2-Pro_Descrica [A(50)] - nome do procedimento	ProDescrica [2]
GR010013	INEXISTENTE	
GR010014 - TbCausas	1-Cau_ChavePri [I(*)] - chave primária 2-Cau_Descrica [A(50)] - nome da causa	CauDescrica [2]
GR010015 - NdcDoençasCausas	1-Ndc_ChavePri [I(*)] - chave primária 2-Ndc_CdDoenca [I()] - chave estrangeira de doença 3-Ndc_CdCausas [I()] - chave estrangeira de causa	NdcCdDoenca [2] NdcCdDoencaCdCausa [2,3]

Anexo2 Estrutura das entidades do subsistema de dados (continuação)

GR010029 - TbDietas	1-Die_ChavePri [I(*)] - chave primária 2-Die_Nomexxxx [A(50)] - nome da dieta 3-Die_Descrica [M(240)] - descrição da dieta	DieNome [2]
GR010030 - NddTratamento sDietas	1-Ndd_ChavePri [I(*)] - chave primária 2-Ndd_CdTratam [I()] - chave estrangeira de tratamento 3-Ndd_CdDietax [I()] - chave estrangeira de dieta	NddCdTratamCdDieta [2,3]
GR010033 - TbGruposMedic amentos	1-Gru_ChavePri [I(*)] - chave primária 2-Gru_Descrica [A(50)] - nome do grupo	GruDescrica [2]
GR010039 - TbRestrições	1-Res_ChavePri [I(*)] - chave primária 2-Res_Descrica [A(50)] - nome da contra-indicação	ResDescrica [2]
GR010040 - NriMedicament osIndicacões	1-Nri_ChavePri [I(*)] - chave primária 2-Nri_CdMedica [I()] - chave estrangeira de medicamento 3-Nri_CdIndica [I()] - chave estrangeira de indicação	NriCdMedica [2] NriCdIndica [3] NriCdMedicaCdIndica [2,3]
GR010041 - NrrMedicament osRestricoes	1-Nrr_ChavePri [I(*)] - chave primária 2-Nrr_CdMedica [I()] - chave estrangeira de medicamento 3-Nrr_CdRestri [I()] - chave estrangeira de contra-indicação	NriCdMedica [2] NriCdRestri [3] NriCdMedicaCdRestria [2,3]
GR010025 - NdpTratamento sProcedimento s	1-Ndp_ChavePri [I(*)] - chave primária 2-Ndp_CdTratam [I()] - chave estrangeira de tratamento 3-Ndp_CdProced [I()] - chave estrangeira de procedimento	NdpCdProced [3] NdpCdProcedCdTratam [3,2]

6 REFERÊNCIAS BIBLIOGRÁFICAS

- [ANT97] ANTUNES, Alvaro Rodrigues; **Programando em Java Para Internet**. Makron Books. São Paulo : 1997.
- [COC98] COCKRAINE and ZYDA; **Agents for the Internet.**: 1996. Endereço eletrônico: www.manning.com/cockraine/chapter1.html
- [DAM96] DAMASCENO, Américo; **Java Programação na Internet**. Erica. São Paulo : 1996.
- [DEM89] DEMAIZEAU, Yves, MÜLLER, Jean Pierre. **Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World**. Elsevier Science Publishers. Amsterdam : 1989.
- [FAR98] FARACO, Rafael Avila; **Uma Arquitetura de Agentes Para Negociação Dentro do Domínio do Comércio Eletrônico**. Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia de Produção. Universidade Federal De Santa Catarina. Florianópolis : 1998.
- [GAS88] BOND, Alan H; GASSER, Les. **Readings in Distributed Artificial Intelligence**. Morgan Kaufmann Publishers. San Mateo-California : 1988.
- [HOF98] HOFFMANN, Helcio Hermes. **Um Protótipo de Sistema de Apoio à Decisão Aplicado a Clínicas Veterinárias**. Relatório de Trabalho de Conclusão de Curso Submetido à Universidade Regional de Blumenau. Blumenau: 1998. Bacharelado em Ciências da Computação, FURB.
- [HAR98] HARPER, Neal; **Intelligent Agents and the Internet.**: 1998. Endereço eletrônico: www.osiris.sunderland.ac.uk/cbowww/ai/texts/agents3/agents.html
- [HEU97] HEUNG, Kennet; **Intelligent Agents A Quick Preview**: 1997. Endereço eletrônico: www.btinternet.com/~k.heung/ia/ia.html

- [JEN96] JENNINGS, G.M.P.; O'HARE, N.R. **Foundations of Distributed Artificial Intelligence**. John Wiley & Sons. New York : 1996.
- [JEN98] JENNINGS, Nicholas R.; WOOLDRIDGE, Michael J. **Agent Technology – Foundations, Applications and Markets**. Springer. New York : 1998.
- [MAE91] MAES, Pattie. **Designing Autonomous Agents : Theory and Practice from Biology to Engineering and Back**. Bradford Book. London : 1991.
- [NEW97] NEWMAN, Alexander; **Usando Java**. Editora Campus. Rio de Janeiro : 1997.
- [NWA96] NWANA, Hyacinth S.; **Software Agents: An Overview**. Intelligent Systems Research. Advanced Applications & Technology Department BT Laboratories, Martlesham Heath Ipswich, IP5 7RE, U.K. Cambridge University : 1996. Endereço eletrônico: www.sce.carleton.ca/netmanage/docs/agentsoverview/ao.html
- [RUS95] RUSSEL, Stuart J.; NORVIG, Peter. **Artificial Intelligence – A Modern Approach**. Prentice-Hall. New Jersey : 1995.
- [SOU96] SOUZA, Eliane Moreira Sá de; **Uma Estrutura de Agentes Para Assessoria na Internet**. Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia de Produção. Universidade Federal De Santa Catarina. Florianópolis : 1996.
- [SOU97] SOUZA, Vandemberg Dantas de; **Desenvolvendo Applets com Java**. Campus. Rio de Janeiro : 1997.
- [WOO96] WOOLDRIDGE, Michael J.; JENNINGS, Nicholas R.; MÜLLER, Jörj P. **Intelligent Agents III**. Springer. New York : 1996.
- [WOO94] WOOLDRIDGE, Michael J.; JENNINGS, Nicholas R. **Intelligent Agents**. Springer-Verlag. New York : 1994.