

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**Centro Tecnológico**  
**Curso de Pós-Graduação em Ciência da Computação**

**SISTEMA ESPECIALISTA PROBABILÍSTICO:  
BASE DE CONHECIMENTO DINÂMICA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, para obtenção do grau de Mestre em Ciência da Computação.

**Carlos Efrain Stein**

Florianópolis, Agosto de 2000.

# DEDICATÓRIA

À minha esposa, minha filha,  
com carinho e apoio em todos os momentos,  
suportando minha ausência em  
momentos importantes.  
Agradeço pela compreensão.

## AGRADECIMENTOS

Agradeço, inicialmente, a **Deus** por me dar forças e me deixar terminar o mestrado com paz e harmonia.

À professora **Sílvia Modesto Nassar** pela maravilhosa orientação, carinho, estímulo, apoio, incentivo e paciência que teve durante o desenvolvimento desta pesquisa.

À minha querida esposa, **Eleonora Lopes Stein** e estimada filha **Eleonora Cristine Lopes Stein**, que sempre estiveram ao meu lado durante a realização do mestrado.

Aos professores e colegas da FURB, **Geraldo Moretto**, **Cláudio Loesch**, que me incentivaram a fazer o curso.

Aos meus **colegas** e **amigos**, do curso, pelo companheirismo.

À **Verinha** e à **Valdete** pela atenção e apoio.

A todos que, diretamente ou indiretamente, colaboraram comigo nessa caminhada.

# SISTEMA ESPECIALISTA PROBABILÍSTICO: BASE DE CONHECIMENTO DINÂMICA

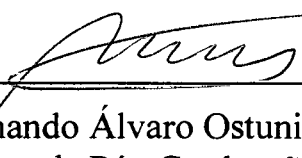
**Carlos Efrain Stein**

Esta Dissertação foi julgada adequada para a obtenção do título de

## MESTRE EM CIÊNCIA DA COMPUTAÇÃO

na área de concentração *Sistemas de Conhecimento*, sub-área *Inteligência Computacional* e, aprovada em sua forma final pelo Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina.

  
\_\_\_\_\_  
Prof.ª Sílvia Modesto Nassar, Dra. Eng.  
Orientadora

  
\_\_\_\_\_  
Prof.º Fernando Álvaro Ostuni Gauthier, Dr. Eng.  
Coordenador do Curso de Pós-Graduação em Ciência da Computação

Banca Examinadora:

  
\_\_\_\_\_  
Prof.ª Sílvia Modesto Nassar, Dra. Eng.

  
\_\_\_\_\_  
Prof.º Bruno H. Kopittke, Dr. Eng.

  
\_\_\_\_\_  
Prof.ª Maria Marlene de Souza Pires, Dra. Med.

## SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>IV</b>
<b>LISTA DE TABELAS.....</b>	<b>V</b>
<b>LISTA DE SIGLAS.....</b>	<b>VI</b>
<b>RESUMO.....</b>	<b>VII</b>
<b>ABSTRACT.....</b>	<b>VIII</b>
<b>1 – INTRODUÇÃO.....</b>	<b>1</b>
1.1 – APRESENTAÇÃO.....	1
1.2 – OBJETIVOS.....	2
1.2.1 – <i>Objetivo Geral</i> .....	2
1.2.2 – <i>Objetivos Específicos</i> .....	2
1.3 – ESTRUTURA DO TRABALHO.....	2
<b>2 – O ESTADO DA ARTE.....</b>	<b>4</b>
2.1 – SISTEMAS ESPECIALISTAS PROBABILÍSTICOS.....	4
2.2 – SHELLS DISPONÍVEIS.....	8
2.3 – CONSIDERAÇÕES FINAIS.....	14
<b>3 – FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>15</b>
3.1 – SISTEMAS ESPECIALISTAS.....	15

3.1.1 – Componentes de um Sistema Especialista.....	16
3.1.2 – Conceitos Básicos .....	17
3.1.3 – Sistema Especialista Bayesiano .....	18
3.2 – REDES BAYESIANAS (RB).....	19
3.3 – PROBABILIDADE BAYESIANA.....	21
3.3.1 – Biografia de Thomas Bayes.....	22
3.3.1 – Espaço de Probabilidade .....	23
3.3.2 – Eventos Compostos .....	24
3.3.3 – Terminologia para os conetivos E e OU.....	24
3.3.4 – Eventos Disjuntos.....	25
3.3.5 – Axiomas e Propriedades de Probabilidade.....	25
3.3.6 – Probabilidade Condicional.....	26
3.3.7 – Propriedades da Probabilidade Condicional .....	26
3.3.8 – Teorema de Bayes .....	27
3.3.9 – Independência de Eventos.....	28
3.3.10 – Probabilidade Condicional e Implicação Lógica .....	28
3.3.11 – Razões de Probabilidade.....	29
3.3.12 – Atualização Bayesiana para uma Nova Evidência .....	30
3.3.13 – Hipóteses Multivaloradas .....	31
3.4 – AGENTES INTELIGENTES.....	33
3.4.1 – Inteligência Artificial Distribuída (IAD).....	33
3.4.2 – Definição de agente.....	34
3.4.3 – Propriedades.....	35
3.4.4 – Classificação de Agentes.....	38
3.4.5 – Funções dos Agentes.....	42
3.5 – APRENDIZAGEM.....	43
3.5.1 – Paradigmas de Aprendizagem de Máquina .....	44
3.5.2 – Métodos de Aprendizagem de Máquina .....	48
3.6 – CONSIDERAÇÕES FINAIS .....	51
<b>4 – O SISTEMA SEDIN .....</b>	<b>53</b>

4.1 – CONCEPÇÃO DO SISTEMA SEDIN .....	53
4.1.1 – <i>Arquitetura Geral do Sistema SEDIN</i> .....	53
4.1.2 – <i>Base de Conhecimento do Sistema SISPAN</i> .....	54
4.1.3 – <i>Shell para SEP</i> .....	60
4.1.4 – <i>Interface Gráfica do SEDIN</i> .....	60
4.1.5 – <i>Módulo de Ajuda do Sistema SEDIN</i> .....	60
4.1.6 – <i>Base de Conhecimento do Sistema SEDIN</i> .....	60
4.1.7 – <i>Considerações Finais</i> .....	61
4.2 – O SISTEMA SEDIN .....	61
4.2.1 – <i>Softwares Utilizados</i> .....	61
4.2.2 – <i>Implementação</i> .....	63
4.2.3 – <i>Avaliação</i> .....	68
<b>5 – CONCLUSÃO</b> .....	<b>69</b>
5.1 – <b>TRABALHOS FUTUROS</b> .....	<b>70</b>
<b>BIBLIOGRAFIA</b> .....	<b>71</b>
<b>ANEXO 1 – CÓDIGO FONTE DO SEDIN</b> .....	<b>76</b>
<b>ANEXO 2 – MANUAL DO USUÁRIO</b> .....	<b>86</b>
<b>ANEXO 3 – INSTALAÇÃO</b> .....	<b>91</b>

## LISTA DE FIGURAS

Figura 1 – Componentes de um Sistema Especialista.....	17
Figura 2 – Formato das Regras de Produção no Raciocínio Bayesiano.....	18
Figura 3 – Exemplo de RB, parte qualitativa, [NETICA 99].....	20
Figura 4 - Exemplo de RB, parte quantitativa, [NETICA 99].....	21
Figura 5 – Retrato de Thomas Bayes .....	22
Figura 6 – Dimensionamento de um agente.....	39
Figura 7 – Arquitetura do Sistema .....	54
Figura 8 – Representação gráfica da base de conhecimento (RB) para Avaliação do Estado Nutricional .	55
Figura 9 – Tabela no Netica que representa o vetor das probabilidades <i>a priori</i> da hipótese.....	56
Figura 10 – Representação gráfica das probabilidades condicionais <i>a priori</i> da variável Edema .....	57
Figura 11 – Probabilidades <i>a posteriori</i> após consulta .....	59
Figura 12 – Visualização da tela principal do SEDIN .....	63
Figura 13 – Visualização da tela Variáveis.....	64
Figura 14 – Visualização da tela Adicionar Nova Variável – 1ª página .....	65
Figura 15 – Visualização da tela Adicionar Nova Variável – 2ª página .....	66
Figura 16 – Visualização da tela Variável Sexo.....	66
Figura 17 – Visualização da tela Probabilidades a Priori.....	67



## **LISTA DE TABELAS**

Tabela 1 – Exemplos de Sistemas Especialistas Probabilísticos.....	5
Tabela 2 – Relação de shells para desenvolvimento de SEP.....	8
Tabela 3 – Métodos de Aprendizagem.....	48

## **LISTA DE SIGLAS**

API – Application Program Interface

DLL – Dynamic Link Library

DPS – Resolução Distribuída de Problemas

IA – Inteligência Artificial

IAD – Inteligência Artificial Distribuída

IAS – Inteligência Artificial Simbólica

MAS – Sistemas Multi-Agentes

RB – Rede Bayesiana

SE – Sistema Especialista

SEP – Sistema Especialista Probabilístico

SEDIN – Sistema Especialista Probabilístico de Base de Conhecimento Dinâmica

SISPAN – Sistema Pediátrico Para Avaliação Nutricional

## RESUMO

Um dos pontos em evidência em Inteligência Artificial é o tratamento da incerteza nos sistemas especialistas. Mais especificamente para os bayesianos a concepção de Sistemas Especialistas Probabilísticos capazes de oferecer um diagnóstico sobre uma determinada situação apresentada, onde a incerteza presente no domínio de aplicação é tratada utilizando a teoria da probabilidade.

Muitos sistemas especialistas probabilísticos apresentam no momento de sua utilização inicial um desempenho que se manterá o mesmo durante seu ciclo de vida. Esses sistemas podem ser considerados como sistemas especialistas probabilísticos com base de conhecimento estática. Isto é, uma vez construída a base de conhecimento com as variáveis especificadas pelo especialista, o sistema não se altera mais.

Entendendo que se poderia modificar este quadro, acrescentar ou excluir novas informações (variáveis), desenvolveu-se um sistema especialista probabilístico com base de conhecimento dinâmica, isto é, permitindo modificação em sua base de conhecimento original, dessa forma proporcionando a aprendizagem.

Este sistema de base de conhecimento dinâmica (SEDIN) foi concebido a partir do sistema especialista probabilístico (SISPAN) cuja base de conhecimento é estática. Ambos os sistemas oferecem um diagnóstico para avaliar o estado nutricional em crianças com até 2 anos de idade. Para a construção do SEDIN utilizou-se a linguagem de programação C++Builder, versão 3.0, da Borland e a shell Netica 1.06. Mais especificamente o ambiente de trabalho do Netica API (Application Program Interface) com sua DLL (Dynamic Link Library) que contém uma biblioteca com todas as funções definidas na linguagem C para representar o conhecimento do especialista em uma rede bayesiana, composta de seus nós (variáveis), seus respectivos arcos e probabilidades condicionais a priori.

O sistema SEDIN pode ser usado para verificar o grau de importância das variáveis introduzidas no sistema bem como oportunizar a aquisição continuada de conhecimento. Por outro lado, um sistema especialista probabilístico tal como este exposto aqui pode servir como material didático para apoiar o professor na transmissão do conhecimento de Sistemas Especialistas Bayesianos.

## Palavras-chave

Inteligência Artificial, Sistemas Especialistas, Redes Bayesianas, Base de Conhecimento Dinâmica.

## ABSTRACT

One of the points in evidence in Artificial Intelligence is the treatment of the uncertainty in the experts systems. More specifically for the Bayesians the conception of Probabilistic Experts Systems capable to offer a diagnosis about a certain presented situation, where the present uncertainty in the application domain is treated using the theory of the probability.

Many probabilistic experts systems present in the moment of your initial use a performance that will stay the same during your life cycle. Those systems can be considered as probabilistic experts systems with static knowledge base. That is, once built the knowledge base with the variables specified by the specialist, the system he doesn't lose temper more.

Understanding that she could modify this picture, to increase or to exclude new information (variables), he grew a probabilistic experts systems with dynamic knowledge base, that is, allowing modification in your base of original knowledge, in that way providing the learning.

This system of dynamic knowledge base (SEDIN) it was conceived starting from the probabilistic experts systems (SISPAN) whose knowledge base is static. Both systems offer a diagnosis to evaluate the state nutritional in children with up to 2 years of age.

For the construction of SEDIN the programming language C++ Builder was used, version 3.0, of Borland and the shell Netica 1.06. More specifically the ambient of work of Netica API (Application Program Interface) with your DLL (Dynamic Link Library) that contains a library with all the defined functions in the language C to represent the specialist's knowledge in a Bayesian network, composed of your knots (variables), your respective arches and conditional probabilities a priori.

The SEDIN system can be used to verify the degree of importance of the variables introduced in the system as well as to make possible the continuous acquisition of knowledge. On the other hand, a probabilistic expert system just as this exposed here it can serve as didactic material to support the teacher in the transmission of the knowledge of Bayesians Experts Systems.

## Key – Words

Artificial Intelligence, Expert Systems, Bayesian Networks, Dynamic Knowledge Base.

# 1 – INTRODUÇÃO

## 1.1 – Apresentação

Este trabalho procura reunir em um conjunto, os conceitos de Rede Bayesiana, Sistemas Multi-agentes e Aprendizagem de Máquina entendendo que num primeiro ponto está representada a base de conhecimento inicial formada por um especialista num determinado domínio de aplicação. Em um segundo ponto, entende-se que esta base de conhecimento pode ser modificada (pode ser dinâmica) produzindo assim um novo conhecimento, isto é, o sistema pode aprender. E em um terceiro ponto, representa o fato de que cada componente do sistema representa um agente que tem níveis diferentes de inteligência.

O método Bayesiano passou a fazer parte dos Sistemas Especialistas por se tratar de uma teoria consistente e que permite a representação de novos conhecimentos no tratamento da incerteza.

Entende-se que uma base de conhecimento é **estática** em um SEP quando esta base uma vez construída com variáveis especificadas pelo especialista não se altera mais. No caso de uma base de conhecimento ser **dinâmica**, afirma-se de que as variáveis especificadas pelo especialista podem alterar o sistema com a aquisição de novas informações e assim podendo realizar um novo aprendizado.

## **1.2 – Objetivos**

### **1.2.1 – Objetivo Geral**

O objetivo geral desta pesquisa é desenvolver um protótipo de um Sistema Especialista Probabilístico que tenha capacidade de aprendizagem.

### **1.2.2 – Objetivos Específicos**

Têm-se como objetivos específicos:

- a) Realizar um levantamento envolvendo Sistemas Especialistas Probabilísticos e por conseguinte eleger um Sistema Especialista Probabilístico com base de conhecimento estática.
- b) Capacitar a aprendizagem no Sistema Especialista Probabilístico tornando-o um sistema com base de conhecimento dinâmica.

## **1.3 – Estrutura do Trabalho**

O presente documento está estruturado em cinco capítulos. No primeiro capítulo é apresentado a introdução e os objetivos da pesquisa.

No segundo capítulo, o Estado da Arte. Os SEP desenvolvidos nos últimos anos e uma relação das shells disponíveis para o desenvolvimento do SEP.

No terceiro capítulo é apresentada a fundamentação teórica desta pesquisa, dando ênfase aos conceitos de Probabilidade Bayesiana, Sistemas Especialistas, Redes Bayesianas, Agentes Inteligentes, Aprendizagem de Máquina.

O quarto capítulo apresenta o protótipo do sistema desenvolvido SEDIN - Sistema Especialista Probabilístico de Base de Conhecimento Dinâmica, bem como a metodologia empregada e implementação do sistema. na

As conclusões obtidas, possibilidades de ampliações para trabalhos futuros são apresentados no quinto capítulo.

E, posteriormente, está a bibliografia e os anexos.

## **2 – O ESTADO DA ARTE**

Neste capítulo são apresentados alguns dos Sistemas Especialistas Probabilísticos desenvolvidos utilizando uma abordagem bayesiana bem como algumas ferramentas de desenvolvimento (*shells*) de Sistemas Especialistas probabilísticos.

### **2.1 – Sistemas Especialistas probabilísticos**

Muitos problemas na vida real são repletos de incerteza. Especialistas humanos são capazes de formar julgamentos e tomar decisões baseados em informação incerta, incompleta e algumas vezes, até mesmo contraditória. Para que um SE seja confiável, o mesmo deve lidar com este tipo de informação com a mesma facilidade que o ser humano [KOEHLER 98].

Depois da metade da década de 80, a pesquisa sobre raciocínio probabilístico em SE resultou na introdução de Redes Bayesianas, também chamadas de Redes Causais. Estas redes têm sua origem na Teoria da Probabilidade e são caracterizadas por um poderoso formalismo que representa o conhecimento no domínio e pelas incertezas associadas a este domínio.

Desde sua introdução a estrutura de RB vem rapidamente ganhando popularidade e agora está começando a mostrar o seu valor em domínios complexos. Aplicações práticas estão sendo desenvolvidas por exemplo, para diagnóstico e prognóstico médico e para recuperação de informação probabilística [GAAG 96].

Segue abaixo, na Tabela 1, os principais SEP desenvolvidos nas áreas de aplicação de medicina e engenharia.



**Tabela 1 – Exemplos de Sistemas Especialistas Probabilísticos**

NOME	LOCAL	DATA
DOMBAL		1972
Dxplain	Laboratório de Ciência da Computação do Hospital Geral de Massachusetts.	1987
Lógica Autônoma de Controle – Lockheed Martin	Lockheed Martin Systems e Artificial Intelligence Center – Sunnyvale, California	1988
Microsoft Pregnancy and Child Care	Microsoft e Knowledge Industries (KI)	1989
General Electric Aircraft Engines Services	Knowledge Industries, ENTER Software	1989
DIAVAL	Departamento de Informática e Automática – U.N.E.D. – Madrid – Espanha	1994
MammoNet	Laboratório de IA, Departamento de Engenharia Elétrica e Ciência da Computação da Universidade de Wisconsin – Milwaukee	1996
SISPAN	Centro Tecnológico – Pós-Graduação em Ciência da Computação – UFSC – Florianópolis	1998

1 – **DOMBAL**: Um dos primeiros, e talvez um dos melhores, sistemas desenvolvidos na área bayesiana foi o de DOMBAL et al. para diagnóstico de dor abdominal. Este sistema utiliza os dados de sensibilidade, especificidade e prevalência para vários sinais, sintomas e testes laboratoriais para calcular a probabilidade de sete possíveis diagnósticos de dor abdominal (apendicite, diverticulite, úlcera perforada, colecistite, obstrução intestinal, pancreatite e dor não específica). O sistema, implementado em computador, passou a funcionar como um tipo de exame subsidiário, onde o médico interessado preenchia uma requisição padronizada informando os achados clínicos e a enviava, em seguida, ao serviço de computação. Após alguns minutos o médico recebia um impresso, dando as probabilidades para os diagnósticos diferenciais [DOMBAL 72].

2 – **Dxplain**: É um sistema de apoio a decisão o qual utiliza um conjunto de dados clínicos (sinais, sintomas e dados laboratoriais) para produzir uma lista de diagnósticos, os quais

podem expor (ou serem associados com) as manifestações. Este software possui uma grande base de dados de probabilidades sobre 5000 manifestações clínicas associadas com mais de 2000 diferentes doenças. O sistema utiliza um formato interativo para coletar informações clínicas e faz uso de uma forma modificada do raciocínio Bayesiano para oferecer interpretações clínicas e tem sido utilizado por milhares de médicos e estudantes de medicina. O banco de dados e o sistema estão sendo continuamente aperfeiçoados e atualizados. DXplain tem característica de livro texto eletrônico e um manual de referência médico [DXPLAIN 97].

**3 – Lógica Autônoma de Controle – Lockheed Martin:** A Lockheed Martin Marine Systems, em Sunnyvale Califórnia, em conjunto com o Artificial Intelligence Center em Palo Alto, vem desenvolvendo um sistema de controle para o UUV (Unmanned Underwater Vehicle). Este veículo submarino poderá permanecer, semanas submerso, sem necessidade de intervenção humana. O objetivo da pesquisa é desenvolver software que seja capaz de controlar eventos planejados e eventos não-antecipados, de modo a diminuir o risco de perda do veículo e maximizar a probabilidade de sucesso na missão. O sistema possui maior capacidade de manusear a grande gama de anomalias e alarmes falsos que ocorrem em missões de longa duração, o que o torna bastante superior aos sistemas da época. Quando completamente desenvolvida, a Lógica Autônoma de Controle permitirá ao UUV monitorar o progresso de sua missão, analisar o estado de seu equipamento, detectar e analisar eventos que causam impacto nos objetivos da missão, tomar decisões e efetuar ações para compensar os eventos, além de modificar o plano da missão quando o atual estiver completamente fora de alcance.

**4 – Microsoft Pregnancy and Child Care:** A *Microsoft* e a *Knowledge Industries* (KI) uniram-se para criar um serviço on-line de informação sobre saúde, com capacidade de multimídia, que emprega raciocínio bayesiano. O produto é chamado de *Microsoft Pregnancy and Child Care*. Uma RB é um componente central da opção *Find by Symptom* (Busca por Sintoma). Sintomas que ocorrem freqüentemente são conectados a módulos especialistas que guiam os pais da criança através de uma série de questões. O sistema repetidamente determina qual a melhor pergunta a se fazer aos pais, mostrando as apresentações em multimídia que correspondem às exatas condições de saúde da criança. Os pais podem escolher entre artigos ilustrados para obter informação detalhada sobre os sintomas e a doença da criança. Os artigos também contêm *links* para outras fontes de informação relacionadas, incluindo outros *Web* sites. O *Microsoft Pregnancy and Child Care* foi criado pela Divisão de Consumo da Microsoft. A KI desenvolveu e testou as bases de conhecimento em RB,

trabalhando com os especialistas Dr. Richard Behrman, Professor de Pediatria na *Stanford University* e na *University of California, San Francisco*, e Dr. David Kerns, Chefe de Pediatria no Hospital *Santa Clara Valley Medical Center*.

5 – **General Electric Aircraft Engines Services**: O diagnóstico de falhas em turbinas de aviões em grandes empresas aéreas implica em grandes custos, pois os dados retirados da análise possuem muitos ruídos e não podem atuar para um defeito específico. As solicitações mecânicas para efetuar os testes e inspeções, assim como a excessiva verificação do motor pode duplicar os custos de manutenção. Para um avião 757 típico, este custo pode chegar a 500000 dólares. Este sistema possui modelos de RB que guiam um mecânico de motores através de uma seqüência otimizada de testes e passos de inspeção. O sistema de recomendação de testes utiliza os resultados de um modelo de desempenho termodinâmico do motor da aeronave, que processa os dados vindos de sensores montados sobre o motor no estande de teste [KIK 97].

6 – **DIAVAL**: É um SE destinado a ajudar médicos no diagnóstico de enfermidades cardíacas a partir, principalmente, de dados ecocardiográficos. Este software faz a introdução dos dados, apresentação dos resultados do diagnóstico e explicações sobre o raciocínio, destacando as características que diferem o DIAVAL de outros SE e de outras RB. Este software é o resultado da tese de doutorado do Professor Fernandez J. Díez. Este sistema consiste em calcular a probabilidade *a posteriori*, selecionar a doença mais provável e a mais relevante, gerando um relatório [DÍEZ 97].

7 – **MammoNet**: É um sistema de suporte à decisão médica para auxiliar no diagnóstico do câncer de mama dado fatores de risco demográficos, informações radiológicas e sintomas físicos. As probabilidades condicionais foram obtidas através de livros e especialistas médicos. MammoNet foi implementado como uma base de conhecimento de regras. A rede que especifica o problema foi construída utilizando um algoritmo de construção de RB. A performance do sistema foi avaliada com 77 casos de um livro texto [MAMMO 98].

8 – **SISPAN** : É um sistema tendo como domínio à área médica. Foi desenvolvida para auxiliar na avaliação do estado nutricional em crianças de até 2 anos de idade com base nos sinais e sintomas e dados antropométricos. Foi testada a sensibilidade do sistema às mudanças nos valores de probabilidades e foi analisado o comportamento do sistema em função do tamanho da base de conhecimento [KOEHLER 98].

A seguir veremos as principais shells disponíveis para o desenvolvimento de Sistemas Especialistas probabilísticos.

## 2.2 – Shells Disponíveis

**Tabela 2 – Relação de shells para desenvolvimento de SEP**

NOME	LOCAL	DATA
SPIRIT	Universidade de Hagen (Fern Universität Hagen)	1990
Netica	Norsys Software Corp. – Vancouver, BC, Canada	1990
IDEAL	Rockwell Palo Alto Laboratory	1990
Bayes On-Line	Knowledge Based Engineering (KBE) – África do Sul	1993
Strategist	Prevision Incorporatec – Davis – Canadá	1994
DxPress	Rockwell Palo Alto Laboratory – Universidade de Stanford	1994
Hugin	Hugin Expert A/S e Universidade de Aalborg – Dinamarca	1995
Graphical-Belief	Departamento de Estatística – Universidade de Washington	1995
BNG	Departamento de Engenharia Elétrica e Ciência da Computação – Universidade de Wisconsin – Milwaukee	1995
Microsoft Belief Networks	Microsoft	1996
BANTER	Laboratório de IA, Departamento de Engenharia Elétrica e Ciência da Computação – Universidade de Wisconsin – Milwaukee	1996

1 – **SPIRIT**: É uma shell para SE criada na Universidade de Hagen (Fern Universität Hagen), na Alemanha, desenvolvida pela equipe do Professor Willhem Rödden. É uma ferramenta que possui interface gráfica de desenvolvimento. Permite a criação de variáveis de diversos tipos, tais como: booleana, ordinal, cardinal e nominal. A parte qualitativa da RB é implementada a partir da inserção de regras de produção. Em seguida, valores de probabilidades

são associados a estas regras e às variáveis, caracterizando a parte quantitativa da RB. Após a caracterização da RB, é necessário inicializá-la, preparando-a para a compilação, ou seja, aprendizagem das regras.

Possui **vantagens** como: facilidade de acesso e trabalha em várias plataformas, inclusive Windows.

Versões de demonstração estão disponíveis pelo endereço eletrônico: <http://pawpaw.fernuni-hagen.de/BWLOR/forsch.html>.

2 – **Netica**: Este software utiliza RB para realizar vários tipos de inferência usando algoritmos modernos e rápidos. Dado um novo caso, pelo qual o usuário tem conhecimento limitado, Netica encontrará os valores ou probabilidades apropriadas para todas as variáveis desconhecidas. O caso pode ser, convenientemente, salvo em um arquivo e depois incluído na rede (ou em uma rede diferente) para incrementar a consulta, ou para trazer uma nova informação sobre o caso. Utiliza diagramas de influência para encontrar as decisões ótimas, as quais maximizam os valores esperados das variáveis especificadas. Variáveis que não são de interesse por muito tempo podem ser removidos sem uma mudança geral entre os relacionamentos entre as variáveis remanescentes.

#### **Vantagens:**

- gera uma apresentação gráfica de qualidade, a qual pode ser incorporada dentro de outros documentos,
- pode encontrar decisões ótimas para problemas de decisão seqüencial,
- soluciona diagramas de influência eficientemente pelo uso de clique na árvore,
- pode aprender relações probabilísticas através de dados,
- permite atualização fácil da RB e dos diagramas de influência, incluindo: excluir, colar e duplicar nós da RB e dos diagramas de influência,
- mantém diagramas complexos ordenados,
- permite comentários,
- permite desfazer comando ilimitadamente,
- permite a entrada de relações probabilísticas através de equações, com uma extensa biblioteca de funções probabilísticas, e funções matemáticas,
- tem facilidade para realizar a discretização de variáveis contínuas.

A Shell Netica é um software desenvolvido pela Norsys Software Corp. em Vancouver, BC, Canadá e está disponível em versões para Windows 95, 98, Windows NT 4.0, Macintosh e Unix. Maiores informações, bem como versões de avaliação da shell, podem ser obtidas pelo endereço: <http://www.norsys.com>.

3 – **IDEAL**: (Influence Diagram Evaluation and Analysis in Lisp): É uma base de testes para trabalhar com RB e diagramas de influência. Contém diversos algoritmos de influência para RB e algoritmos de avaliação para diagramas de influência e RB [SRINIVAS 90]. IDEAL foi escrito em Common Lisp. Este programa possui uma clareza do código e oferece uma programação de alto nível. Desta forma, IDEAL é adequado para implementações experimentais que necessitem tecnologia de RB. Em aplicações mais avançadas, 'IDEAL' pode ser usado como uma biblioteca de sub-rotinas as quais oferecem influência em RB e avaliação de diagramas de influência. IDEAL vem com uma interface gráfica opcional que depende da versão do Common Lisp que o usuário utiliza. Uma versão para pesquisa do IDEAL pode ser obtida através de *download* na forma de um arquivo .TAR para Unix. O manual vem na forma de um arquivo *Postscript* [IDEAL 97].

4 – **Bayes On-Line (BOL)**: É uma ferramenta que ajuda a determinar as causas do problema e prognostica eventos futuros em muitas aplicações, principalmente industriais. Ele utiliza uma rede para modelar uma aproximação baseada na regra de Bayes da Teoria da Probabilidade. Automatiza a identificação de causas prováveis de um problema detectado e facilita a resolução do problema. BOL é valioso, para sensores de validação, diagnósticos, prognósticos, análise de decisão, análise de risco, análise probabilística de segurança e monitoramento de alarmes. É aplicável a problemas que requerem respostas em tempo real, conectividade, ou ferramentas sofisticadas para decisões estratégicas. Trabalha, preferencialmente, com distribuições de probabilidades complexas, elaboração de árvores de decisão. Para o desenvolvimento de aplicações que possam trabalhar com o nível de componentes utilizando o BOL, é necessário especificar somente as causas imediatas e os efeitos de um evento. Estes relacionamentos são usados para desenvolver gráficos, que podem ser compreendidos por usuários não-especialistas. BOL permite uma fácil representação de fenômenos dinâmicos, tais como: persistência, efeitos atrasados e *feedback* [BOL 97].

5 – **Strategist**: Oferece uma ferramenta para construir e analisar modelos graficamente. O usuário poderá analisar graficamente o seu modelo de decisão. Os modelos incluem diagramas de influência para mostrar a estrutura dos relacionamentos entre as variáveis.

*Strategist* é baseado na probabilidade bayesiana e na teoria de decisão estatística. As tecnologias existentes para operacionalização da análise de decisão, tais como: árvores de decisão; diagramas de influência, ainda deixa muito a desejar por parte do usuário. As principais características do *Strategist* são [STRATEGIST 97]:

- trabalha com RB,
- trabalha com diagramas de influência hierárquico,
- uma rica linguagem para representar o relacionamento entre as variáveis (contingências, relacionamentos noisy-and, divisões),
- trabalha com variáveis contínuas,
- realiza testes com os modelos utilizando situações do mundo real,
- possui análise de sensibilidade.

6 – **Dxpress**: É uma ferramenta para aquisição de conhecimento para desenvolvimento rápido de modelos probabilísticos para diagnósticos. Escrito em C++, este programa utiliza várias técnicas para acelerar a aquisição do conhecimento, para reduzir a quantidade de tempo necessário para desenvolver um SE. Por exemplo, foi desenvolvido um protótipo de sistema de diagnóstico para uma turbina à jato usando energia elétrica e pneumática no DC-10. Este sistema diagnostica 60 diferentes falhas utilizando aproximadamente 50 testes ou observações diferentes [KIK 97].

#### **Vantagens:**

- avaliação rápida das distribuições de probabilidades utilizando funções de independência causal,
- localiza o progresso durante as diferentes fases da aquisição do conhecimento,
- permite ao usuário dar entrada em notas e definições pertinentes para o processo de aquisição de conhecimento.

7 – **Hugin**: É uma shell de origem dinamarquesa para construção de RB. Foi desenvolvida por Frank Jensen e Lars Bo Nielsen. É uma ferramenta capaz de construir modelos baseados em SE em domínios caracterizados pela incerteza. Os modelos suportados são RB e diagramas de influência. Permite ao usuário definir nós discretos e alguns nós contínuos em seus modelos. A origem desta shell deu-se durante o projeto do programa ESPRIT sobre diagnóstico de doenças neuromuscular. Um grupo de pesquisa da Universidade de Aalborg, na Dinamarca, trabalhou no desenvolvimento correto e eficiente dos métodos de cálculo para o problema do

diagnóstico. Naquela época, os pesquisadores obtiveram alguns resultados, mas ainda restava um problema que impedia a utilização de RB na construção de SE. O problema foi chamado de *rumour problem*: “Posso ouvir a mesma estória através de diversos canais: mas a estória pode se originar da mesma fonte. Sem saber se meus canais são, ou não independente, não posso combinar as declarações corretamente”.

Em RB o *rumour problem* aparece quando uma causa pode influenciar o mesmo evento através de diferentes caminhos na rede. O problema foi resolvido e métodos gerais foram disponibilizados para serem utilizados em vários domínios, os quais podem ser modelados por uma RB. Os métodos têm sido programados dentro de um sistema de desenvolvimento geral, de fácil operação por qualquer usuário que deseja construir um SE baseado em uma RB. Este sistema foi chamado HUGIN. Desde então o sistema tem sido aprimorado com a facilidade dos diagramas de influência.

HUGIN apresenta parte qualitativa da rede de forma clara, incluindo a direção dos grafos e a parte quantitativa é apresentada em forma de tabelas. A primeira fase na implementação de uma base de conhecimento no HUGIN é a construção da parte qualitativa. Com estas informações é possível definir todas as variáveis e seus respectivos atributos. A parte quantitativa consiste em preencher as tabelas que são apresentadas quando cada variável é selecionada. O tamanho da tabela varia conforme o número de *links* que o nó observado possui com *patern nodes* e outros nós. *Patern Nodes* são os nós onde chegam grafos de dependências. Finalizando este processo, está terminada a etapa de edição da RB. A etapa seguinte é a compilação e o programa passa para o modo de execução.

O usuário tem a oportunidade de utilizar o HUGIN através HUGIN Runtime – um ambiente gráfico de fácil acesso para o usuário. O usuário pode também utilizar o HUGIN API, que é uma biblioteca de funções para C (ou C++) [HUGIN 97].

**8 – Graphical-Belief:** É um programa que oferece um ambiente interativo para explorar modelos baseados em RB. Uma RB é um modelo de rede utilizado para descrever os relacionamentos entre variáveis. Este software é dividido em dois módulos principais, a saber: Análise do Modelo e Construção do Modelo. É uma ferramenta para explorar os aspectos preditivos dos modelos. É baseado na tecnologia de modelos gráficos (também conhecidos como diagramas de influência, redes de crença ou RB) as quais já têm sido muito utilizadas na análise de decisão e IA. Estes modelos têm sido utilizados com sucesso em diversas áreas, tais como:



segurança de sistemas, tomada de decisões médicas, planejamento financeiro e gerenciamento operacional. Este programa é um ambiente completo para construir e explorar modelos de risco. Este software várias ferramentas para modelar tarefas, tais como: a) **Organização**: O modelo somente especifica a interação direta entre as variáveis no modelo; calcula e mostra todas as dependências envolvidas entre variáveis distantes. b) **Engenharia do Conhecimento**: Uma grande quantidade de conhecimento fica armazenada dentro de um modelo gráfico tradicional. *Graphical-Belief* oferece ferramentas para manter este conhecimento. O modelo pode ser projetado através de uma biblioteca anteriormente construída e de estruturas de conhecimento genérico. Uma outra vantagem em se utilizar o *Graphical-Belief* é que ele utiliza um esquema orientado a objetos para armazenar o conhecimento; uma alteração de uma simples regra do protótipo ou de uma variável é prontamente propagada para todo o modelo. c) **Flexibilidade**: o usuário pode selecionar a forma de representação dos relacionamentos entre as variáveis: probabilidade (para relacionamentos com incerteza) e lógica (para relacionamentos com certeza). *Graphical-Belief* completa esta flexibilidade com uma máquina de inferência genérica a qual pode ser simplesmente expandida para incluir outras representações para os relacionamentos. d) **Visualização Dinâmica**: o usuário manipula o modelo, explorando as implicações entre os relacionamentos [GRAPHICAL 97].

9 – **Bauesian Network Generator (BNG)**: É um sistema para construção de base de conhecimento com RB. Uma classe de RB é especificada com uma base de conhecimento de regras. Regras podem conter variáveis temporais e não-temporais, bem como restrições do contexto. BNG constrói uma RB estruturalmente mínima para calcular a probabilidade *a posteriori* de uma consulta. Este sistema oferece duas vantagens sobre o uso das RB tradicionais: representacional e computacional. A incorporação das variáveis quantificadas na base de conhecimento permite a representação da informação não expressível em RB tradicionais. Pois somente uma pequena porção de um grande modelo probabilístico pode ser relevante para a computação de uma particular probabilidade posterior, usando BNG pode resultar em economias computacionais significantes. Estas economias podem ser particularmente grandes com RB temporais. BNG está escrito em CommonLisp e possui interface para a ferramenta IDEAL [BNG 97].

10 – **Microsoft Belief Network Tools**: Este software permite a criação e avaliação de RB. É composto pelo executável MSBN32.EXE que utiliza uma versão proposta pelo *Bayes Net Interchange Format*, isto é, uma representação de um arquivo texto de RB. A aplicação está

escrita em Visual Basic 4.0 e possui uma .DLL para Windows (MSBN32.DLL). Este módulo é o componente ativo do conjunto, e suporta as seguintes operações: a) procura e armazena RB em texto para criação e modificação da rede através da adição de nós e arcos da avaliação das probabilidades discretas das RB utilizando um método chamado “*exact clique-tree propagation methods decision-theoretic troubleshooting*”; b) diagramas assimétricos de influência de simples decisão. É uma aplicação padrão com interface multi-documentos para Windows. Possui uma barra de ferramentas e um help sensível ao contexto.

11 – **BANTER**: É uma ferramenta educacional para trazer a informação contida em uma RB para o usuário final de uma forma inteligível. A shell BANTER foi projetada para usuários tutores na avaliação de hipóteses e seleção dos melhores processos diagnósticos. BANTER pode ser usado com qualquer RB que contém nós que podem ser classificados dentro de hipóteses, observações e processos diagnósticos. Oferece capacidade para calcular a probabilidade *a posteriori* de hipóteses, determinar o melhor processo diagnóstico para afirmar (“*rulle in*”) ou excluir (“*rulle out*”) uma hipótese, interroga o usuário na seleção do processo diagnóstico ótimo e explica o raciocínio do sistema. Oferece também, uma interface para modelar RB e cria problemas para uso instrucional. Foi escrito em C e necessita da shell HUGIN, descrita anteriormente, para executar [BANTER 97].

### 2.3 – Considerações Finais

Este capítulo teve como objetivo, realizar uma pesquisa e posteriormente apresentar um relato os principais SEP desenvolvidos nos últimos anos. Entre os SEP existentes optou-se pelo sistema SISPAN com sua RB original e assim desenvolver o sistema SEDIN.

De modo semelhante, descreveu-se as principais shells existentes e optou-se pela utilização da shell Netica 1.06. Justifica-se o emprego da shell Netica 1.06 no presente trabalho pelo fato de que a RB do sistema SISPAN ser também construída com a mesma shell.

No próximo capítulo será apresentada a fundamentação teórica necessária ao desenvolvimento do sistema SEDIN.

## **3 – FUNDAMENTAÇÃO TEÓRICA**

Neste capítulo será feita uma apresentação sobre os tópicos: Sistemas Especialistas, Redes Bayesianas, Probabilidades Bayesianas, Agentes Inteligentes e Aprendizagem. No final do capítulo serão feitas algumas considerações finais.

### **3.1 – Sistemas Especialistas**

O principal objetivo dos SE (Sistemas Especialistas) é a partir do conhecimento capturado junto a um especialista, em uma área particular do conhecimento humano e representado em uma estrutura modular e expansível, transferi-lo para outros usuários deste domínio [PACHECO 91].

Assim, a proliferação de SE (Sistemas Especialistas) pode trazer benefícios nas mais variadas áreas de conhecimento. Por exemplo, um médico pode vir a ter uma segunda “opinião” para seus diagnósticos clínicos, ou seja, qualquer outro profissional pode potencialmente dispor de um Sistema Especialista para fundamentar suas decisões.

Um SE é uma aplicação da área de IA (Inteligência Artificial) que toma as decisões ou soluciona problemas em um domínio de aplicação, pelo uso do conhecimento e regras definidas por especialistas neste domínio. Os SE solucionam problemas que normalmente são solucionados por “especialistas” reais [KOEHLER 98].

O primeiro sistema a ser considerado foi o DENDRAL (SE para detectar a estrutura molecular de um componente orgânico), em 1965, no entanto, sua aplicabilidade ficou restrita ao meio acadêmico. Em 1976 desenvolveu-se o SE mais conhecido, o MYCIN (SE para diagnóstico

e terapia de doenças infecciosas) que ficou restrito aos meios acadêmicos. Somente em 1982, com o desenvolvimento do XCON (SE para configurar sistemas operacionais para computadores a partir das solicitações dos usuários), os SE saíram das Universidades e ganharam o interesse das indústrias. E a partir dessa época, então, o setor manufatureiro e empresarial faz uso de SE [GAAG 96].

### 3.1.1 – Componentes de um Sistema Especialista

Os principais componentes de um SE são:

- ***Máquina de Inferência***: indica a operação a ser realizada sobre o conhecimento contido no SE; busca as regras necessárias a serem avaliadas, ordena-as de uma maneira lógica e direciona o processo de inferência; compara *Strings* com padrões, e se a comparação for satisfatória, uma atitude será tomada, buscando determinado objetivo.
- ***Base de Conhecimento***: fornece as características de funcionamento do sistema, armazena os fatos e regras a serem usados.
- ***Subsistema de Aquisição do Conhecimento***: é usado para alimentar a base de conhecimento. Nele pode-se introduzir, alterar ou excluir regras de acordo com a necessidade; construído por um editor de texto construído especialmente para ser usado pelo especialista e/ou engenheiro de conhecimento na transferência do conhecimento para o sistema.
- ***Subsistema de Explicação***: tem por função, elucidar a linha de raciocínio que o SE usa para chegar a uma conclusão. Interage com o usuário esclarecendo-o de como o sistema chegou a conclusão, se isso for solicitado.
- ***Interface***: exhibe toda a transação de informações que ocorrem durante a consulta. Pode ser em forma de menus, perguntas ou até mesmo, ícones; exhibe todas as respostas, perguntas e resultados das consultas e imprime relatórios.

Na Figura 1, abaixo, estão ilustrados os componentes básicos de um SE, conforme Koehler [KOEHLER 98].

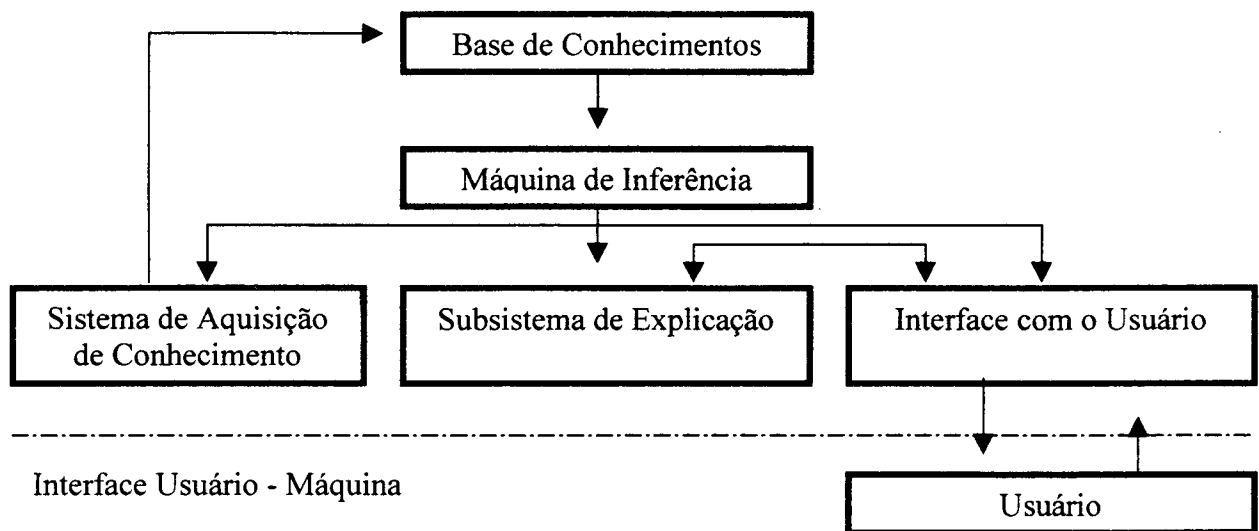


Figura 1 – Componentes de um Sistema Especialista

### 3.1.2 – Conceitos Básicos

Seguem abaixo, alguns conceitos pertinentes ao desenvolvimento e funcionamento dos SE.

- **Backward Chaining** (Encadeamento regressivo): é o procedimento para solução de um problema, que começa com uma proposição e um conjunto de regras que conduzem a proposição, e então, trabalha para trás, unificando as regras com informações da base de fatos até a proposição ser verificada ou considerada errada.
- **Forward Chaining** (Encadeamento progressivo): é o procedimento para solução de um problema, que começa com um conjunto de regras e um banco de fatos e trabalho para alcançar uma conclusão baseada em fatos que unifique todo o conjunto de premissas contido nas regras.
- **Shell para SE**: é um software que facilita a construção de SE pelo fornecimento de esquemas de representação do conhecimento e de máquinas de inferência. O Engenheiro de Conhecimento apenas adiciona conhecimento.
- **Aquisição de Conhecimento**: é o processo de coletar conhecimento especializado do especialista real para disponibilizar em um SE.



Na Abordagem Bayesiana, a *premissa* é referida como *evidência* e a conclusão como hipótese. Note-se que as hipóteses em uma regra podem tornar-se evidência em outra [PACHECO 91].

Para se utilizar Métodos Probabilísticos no Tratamento de Incerteza em SE, é necessário obter um conjunto de *probabilidades a priori*, a fim de descrever a população (conjunto de hipóteses ou evidências) que formam o conjunto universo. Quando surgem evidências relacionadas a determinadas hipóteses, há uma atualização nas probabilidades a priori das primeiras. O resultado é um conjunto de *probabilidades a posteriori* [PACHECO 91].

### 3.2 – Redes Bayesianas (RB)

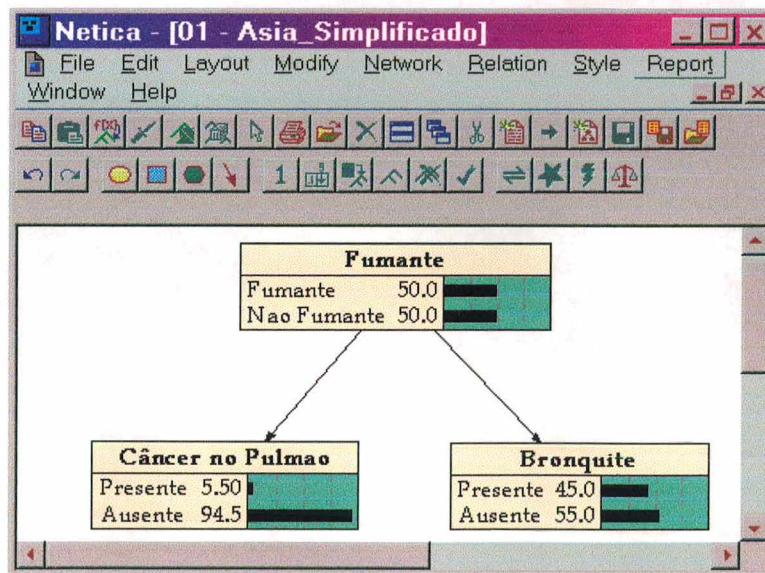
As redes bayesianas são esquemas de representação de conhecimento, utilizadas para desenvolver a base de conhecimentos de um Sistema Especialista probabilístico. Possuem uma parte estrutural refletindo relações causais e valores de probabilidade refletindo a força da relação. A RB é extraída de um especialista que transfere seus conhecimentos do domínio de aplicação para o engenheiro de conhecimentos do sistema. Na construção de uma RB, nem sempre se pode identificar numericamente as dependências entre as variáveis. Na maioria das vezes esta tarefa é feita através do julgamento humano. Para se designar que uma variável B será descendente direto (filho) de uma variável A, deve-se constatar que A é causa direta ou então influencia a variável B diretamente [PEARL 88].

Segundo Gaag [GAAG 96], as redes bayesianas são compostas de duas partes complementares:

A parte **qualitativa** é um modelo gráfico (grafo direcionado acíclico) onde as variáveis são os nós e as regras, que são as relações de dependência condicional entre essas variáveis, são os arcos direcionados. Assim um arco ligando as variáveis A e B, na seguinte forma  $A \rightarrow B$ , indica que a variável B é a consequência e a variável A é a causa, e estas apresentam uma relação de dependência. Por outro lado, se não houver um arco ligando duas variáveis então se supõe que essas variáveis são independentes.

A parte **quantitativa** de uma RB é o conjunto de probabilidades condicionais associadas aos arcos existentes no modelo gráfico acima descrito e as probabilidades estimadas *a priori* das hipóteses diagnósticas. Associada à parte qualitativa de uma RB está um grupo de funções representando valores numéricos da distribuição em questão. A cada vértice do grafo é associada uma função de atribuição de probabilidades que é basicamente um conjunto de probabilidades condicionais que descreve a influência dos valores dos vértices predecessores nas probabilidades dos valores deste vértice. Estas funções de atribuição de probabilidade em conjunto constituem a parte quantitativa da RB.

Para ilustrar as duas partes, **qualitativa** e **quantitativa** de uma RB, será demonstrado um exemplo - didático - simplificado de uma RB extraída do diretório "exemplos" que acompanha a shell Netica [NETICA 99]. A RB inicialmente intitulada: Asia, foi simplificada apenas para ilustração. A RB é composta por três variáveis binárias: a variável *Fumante* representa a presença ou a ausência do evento em uma pessoa, isto é, se a pessoa é fumante ou não fumante, a variável *Câncer* representa a presença ou ausência de câncer, a variável *Bronquite* representa a presença ou ausência de bronquite, conforme a Figura 3.



**Figura 3 – Exemplo de RB, parte qualitativa, [NETICA 99]**

Na Figura 3, as variáveis estão representadas pelos nós. Os nós estão ligados através de setas, também chamados de arcos, (  $\longrightarrow$  ) que liga um nó ao outro. As setas ligando a variável *Fumante* às variáveis *Câncer no Pulmão* e *Bronquite*, significam que as variáveis *Câncer no Pulmão* e *Bronquite* são consequência da variável *Fumante*. Assim pode-se dizer que:



“dado que uma pessoa é Fumante ou não (causa), a probabilidade dessa pessoa ter Bronquite (conseqüência) será de 45%. Portanto a parte qualitativa é composta de nós (variáveis) e as setas (regras e/ou fatos)”.

A parte quantitativa é compreendida pelas probabilidades condicionais associadas aos arcos e as probabilidades *a priori* das hipóteses diagnósticas.

A seguir, a Figura 4 ilustra as probabilidades condicionais do nó (variável) Cancer\_no\_Pulmão em forma de uma tabela organizada.

Smoking	Presente	Ausente
Fumante	10.000	90.000
Nao_Fu...	1.000	99.000

Figura 4 - Exemplo de RB, parte quantitativa, [NETICA 99]

### 3.3 – Probabilidade Bayesiana

A probabilidade Bayesiana é uma das principais medidas de incerteza em Estatística, também chamada de chance ou frequência, associa a probabilidade com a frequência de um evento obtendo o grau de certeza de ocorrência de um certo evento.

Os métodos Bayesianos possibilitam representar numericamente o grau de certeza sobre condições de incerteza, e manipulá-lo de acordo com as regras definidas na Teoria da Probabilidade [HECKERMAN 95].

A estatística Bayesiana teve origem com Thomas Bayes, Teólogo e Matemático inglês, realizando vários trabalhos na área da probabilidade. Mais tarde, o grande Matemático francês, Piere Simon de Laplace, desenvolveu o teorema de Bayes [HECKERMAN 95].

O teorema de Bayes é de grande importância para o cálculo de probabilidades. Quando melhor compreendido, o teorema se mostra como a lei fundamental que governa o

processo de inferência lógica. O teorema é a base para analisar um conjunto de informações disponíveis e chegar a uma conclusão, expressa numericamente [KOEHLER 98].

O teorema de Bayes é um método quantitativo para a revisão de probabilidades conhecidas, com base em nova informação amostral. No processo de tomada de decisão, significa calcular uma probabilidade pela aplicação de um teste diagnóstico (*probabilidade a posteriori*), considerando uma probabilidade já disponível (*probabilidade a priori*) [KOEHLER 98].

Pela enorme importância que o Teorema de Bayes representa na atual pesquisa, é conveniente apresentar, a seguir, uma biografia de Thomas Bayes.

### 3.3.1 – Biografia de Thomas Bayes



**Figura 5 – Retrato de Thomas Bayes**

Reverendo Thomas Bayes, nasceu em Londres em 1702 e faleceu em 17 de abril de 1761, em Poços de Tunbridge, 35 milhas do sudeste de Londres. Filho de Joshua e da Ann Ditos Bayes.

Teologista e Matemático, o primeiro a usar a probabilidade indutiva e que estabeleceu uma base matemática para a inferência da probabilidade.

Em 1763, Bayes teve sua obra póstuma publicada por Richard Price com o título: “Ensaio para Resolver um Problema na Doutrina das Probabilidades”. Este trabalho se transformou na base de uma técnica estatística, chamada agora de estimação Bayesiana, para calcular a validade da probabilidade de uma proporção na base de uma estimativa prévia de sua probabilidade e nova evidência relevante. As desvantagens do método indicado por estatísticos mais conservadores incluem as maneiras diferentes de atribuir distribuições prévias dos parâmetros e possíveis sensibilidade das conclusões à escolha das distribuições.

Os únicos trabalhos conhecidos que Bayes publicou em vida foram:

- “Divina Providência e o Governo São a Felicidade de Suas Criaturas (1731)”.
- “Uma Introdução a Doutrina dos Fluxos e Defesa do Analista (1736)” que opôs os ataques pelo Bispo Berkeley nas fundações lógicas do cálculo de Newton.

Um dos teoremas mais famosos na teoria da probabilidade é o Teorema de Bayes:  $P(H_i / e) = \frac{P(e / H_i) \cdot P(H_i)}{P(e)}$ . Mais tarde, este teorema generalizado por Laplace, foi o ponto de partida básico para resolver problemas de inferência usando a teoria da probabilidade como lógica.

Portanto, pode-se observar que desde o século XVII, com Thomas Bayes, entre outros estudiosos, a Teoria da Probabilidade tem sido desenvolvida e aplicada nos mais variados campos científicos. A partir dos anos 70, quando os pesquisadores da Inteligência Artificial (IA) defrontaram-se com a questão do Tratamento de Incerteza em Sistemas Especialistas, a Inteligência Artificial passou a integrar estes domínios de aplicação.

A seguir, será feita uma apresentação formal dos conceitos de probabilidade [NASSAR 98].

### 3.3.1 – Espaço de Probabilidade

Seja  $\varepsilon$  um conjunto finito e  $P$  uma função de  $\varepsilon$  para os números não negativos, tal que:

$$\sum P(e) = 1 \quad \forall e: e \in \varepsilon$$

O par  $(\mathcal{E}, P)$  é chamado de espaço de probabilidade. Os elementos de  $\mathcal{E}$  são chamados de eventos simples ou elementares.  $P$  é chamado de distribuição de probabilidade ou função de probabilidade. Desta definição resulta que a probabilidade de qualquer evento  $e$  denotada por  $P(e)$ , é medida por um número no intervalo  $[0, 1]$ .

Intuitivamente,  $\mathcal{E}$  é a coleção de resultados que se pode esperar em um domínio de aplicação. O valor  $P(e)$  é uma estimativa da crença de que o resultado  $e$  ocorra.

### 3.3.2 – Eventos Compostos

Seja um espaço de probabilidade  $(\mathcal{E}, P)$ .

Um subconjunto  $A$  do conjunto  $\mathcal{E}$  é denominado de evento composto.

Para um evento composto  $A$ ,  $A \subseteq \mathcal{E}$  define-se a probabilidade do evento  $A$  da seguinte forma:

$$P(A) = \sum_{e \in A} P(e)$$

$$e \in A$$

Se  $e$  ocorre e  $e \in A$  então diz-se que  $A$  ocorre.

Ao evento composto consistindo do conjunto de elementos de  $\mathcal{E}$  que não estão em  $A$  é chamado de negação ou complemento de  $A$  e é denotado por  $\neg A$  ou  $\bar{A}$ .

### 3.3.3 – Terminologia para os conectivos E e OU

Considerando dois eventos  $A$  e  $B$  e adotando a terminologia  $\wedge$  e  $\vee$  respectivamente, então:

$P(A \wedge B)$  significa a probabilidade de que ambos os eventos ocorram; isto é a probabilidade do evento  $A$  ocorrer e do evento  $B$  ocorrer.

$P(A \vee B)$  significa a probabilidade de que pelo menos os eventos ocorra; isto é a probabilidade do evento **A** ocorrer **ou** do evento **B** ocorrer.

### 3.3.4 – Eventos Disjuntos

Denotando por  $\cup$  e  $\cap$  a união e a interseção de conjuntos então:

$$A \cup B = \{e: e \in A \text{ OU } e \in B\}$$

$$A \cap B = \{e: e \in A \text{ E } e \in B\}$$

Se  $A \cap B = \emptyset$  então **A** e **B** são eventos disjuntos ou mutuamente excludentes, e  $P(A \wedge B) = 0$ .

### 3.3.5 – Axiomas e Propriedades de Probabilidade

a)  $0 \leq P(A) \leq 1$

b)  $P(\varepsilon) = 1$

c)  $P(A \vee B) = P(A) + P(B)$  se **A** e **B** são eventos disjuntos.

d) Se  $A \subseteq B \subseteq \varepsilon$  então  $0 \leq P(A) \leq P(B) \leq 1$

e) Se  $A \subseteq B \subseteq \varepsilon$  então  $P(A \wedge B) + P(A \wedge \bar{B}) = P(A)$  e  $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

f)  $P(A) + P(\bar{A}) = 1$

g) Se  $A_i \subseteq \varepsilon$  para  $1 \leq i \leq n$  e  $A_i \cap A_j = \emptyset$  então para todo  $i \neq j$

$$P(A_1 \vee A_2 \vee \dots \vee A_n) = P(A_1) + P(A_2) + \dots + P(A_n)$$

### 3.3.6 – Probabilidade Condicional

Considerando agora que duas moedas de um real são lançadas. Adicionalmente, sabe-se que pelo menos uma cara ocorreu. Como construir um novo espaço de probabilidade  $(\epsilon, P')$  que seja consistente como o espaço de probabilidade  $(\epsilon, P)$  anterior e com a nova informação? A construção de um novo espaço de probabilidade  $(\epsilon, P')$  onde o anterior  $(\epsilon, P)$  é alterado devido a uma nova informação, é o problema central no desenvolvimento de SEP. O conceito de probabilidade condicional permite considerar as novas informações de forma a obter as novas probabilidades.

Sejam **A** e **B** eventos compostos de um espaço de probabilidades  $(\epsilon, P)$ . Suponha que um evento simples **e** ocorra. A probabilidade  $P(B)$  é a probabilidade de que  $e \in B$  dado nosso conhecimento inicial refletido por  $P$ . Intuitivamente,  $P(B/A)$  é a probabilidade que  $e \in B$  quando se tem informação adicional de que  $e \in A$ .

Seja  $(\epsilon, P)$  um espaço de probabilidade e seja  $A \subseteq \epsilon$  tal que  $P(A) \neq 0$ . Definindo um novo espaço de probabilidade  $(\epsilon, f)$  da seguinte forma:

$$P(e) / P(A) \quad \text{se } e \in A$$

$$f(e) = 0 \quad \text{se } e \notin A$$

para qualquer  $B \subseteq \epsilon$  a probabilidade condicional de  $B$  dado a ocorrência de  $A$  é igual a  $f(B)$ .

Observe que neste caso  $A$  é o novo espaço de probabilidade, onde  $B$  deve ser analisado.

$$\text{Se } A = \epsilon \text{ então } P(B/A) = P(B)$$

### 3.3.7 – Propriedades da Probabilidade Condicional

Seja um espaço de probabilidade  $(\epsilon, P)$ . Se  $C \subseteq \epsilon$  então  $P(C) \neq 0$ . Segue-se que:

1.  $P(A/C) = P(A \cap C) / P(C)$

2. Se  $A \subseteq B \subseteq \varepsilon$  então  $0 \leq P(A/C) \leq P(B/C) \leq 1$

3. Se  $A, B \subseteq \varepsilon$  então  $P(A/C) = P(A \wedge B / C) + P(A \wedge \bar{B} / C)$  e

4. Se  $A_i \subseteq \varepsilon$  para  $1 \leq i \leq n$  e  $A_i \cap A_j = \emptyset$  então para todo  $i \neq j$

$$P(A_1 \vee A_2 \vee \dots \vee A_n / C) = P(A_1 / C) + P(A_2 / C) + \dots + P(A_n / C)$$

5. Se  $A \subseteq \varepsilon$ ,  $B_1 \vee B_2 \vee \dots \vee B_n$  para  $1 \leq i \leq n$  e  $B_i \cap B_j = \emptyset$  para todo  $i \neq j$  então

$$P(A) = P(A/B_1) \cdot P(B_1) + P(A/B_2) \cdot P(B_2) + \dots + P(A/B_n) \cdot P(B_n)$$

### 3.3.8 – Teorema de Bayes

Seja o espaço de probabilidade  $(\varepsilon, \mathbf{P})$  e os eventos compostos  $H_1, H_2, \dots, H_k \subseteq \varepsilon$ , desde que nenhum desses eventos tenha probabilidade nula, então:

$$P(H_i / e) = \frac{P(e/H_i) \cdot P(H_i)}{P(e)}$$

Se  $P(H_i \wedge e) \neq 0$  para todo  $i$  então

$$\frac{P(H_i / e)}{P(H_j / e)} = \frac{P(H_i)}{P(H_j)} \cdot \frac{P(e/H_i)}{P(e/H_j)}$$

Se os eventos  $H_1 \cup H_2 \cup \dots \cup H_k = \varepsilon$  e  $H_i \cap H_j = \emptyset$  para todo  $i \neq j$  então

$$P(e) = P(H_1) \cdot P(e/H_1) + P(H_2) \cdot P(e/H_2) + \dots + P(H_k) \cdot P(e/H_k) \text{ resultando}$$

$$P(H_i / e) = \frac{P(e/H_i) \cdot P(H_i)}{\sum_{j=1}^k (P(H_j) \cdot P(e/H_j))}$$

Nas aplicações dos SEP os  $H_i$ 's são as hipóteses concorrentes. O evento  $e$  pode ser pensado como uma evidência. O conhecimento da ocorrência desta evidência leva a mudanças na probabilidade *a priori*  $P(H_i)$  para a probabilidade condicional  $P(H_i/e)$ , que por sua vez considera a evidência  $e$ .

### 3.3.9 – Independência de Eventos

Seja um espaço de probabilidade  $(\mathcal{E}, \mathbf{P})$ .

E, sejam os eventos  $e_1, e_2 \subseteq \mathcal{E}$ . Segue-se que:

a) Se  $P(e_1 \wedge e_2) = P(e_1) \cdot P(e_2)$  então os eventos  $e_1$  e  $e_2$  são independentes.

b) Genericamente, para qualquer subconjunto  $E = \{e_{i1}, e_{i2}, \dots, e_{ik}\}$  de  $\{e_1, e_2, \dots, e_n\}$

se

$$P(e_{i1} \wedge e_{i2} \wedge \dots \wedge e_{ik} / H) = P(e_{i1}/H) \cdot P(e_{i2}/H) \cdot \dots \cdot P(e_{ik}/H) \text{ então}$$

pode-se dizer que os eventos  $e_i$ 's são eventos mutuamente independentes dado a hipótese H.

A idéia básica subjacente ao conceito probabilístico de independência é que o conhecimento de certa informação não faz informação adicional sobre outra coisa. Isto é, se e somente se, saber que o evento  $e_1$  ocorreu não trazer informação sobre o evento  $e_2$  e saber que o evento  $e_2$  ocorreu não trazer informação sobre o evento  $e_1$ , então se diz que ocorre a independência entre estes eventos.

Por exemplo, dois testes diagnósticos são independentes quando não há associação entre os resultados de ambos os testes.

### 3.3.10 – Probabilidade Condicional e Implicação Lógica

Na teoria da probabilidade foi possível observar que  $P(A \vee B)$ ,  $P(A \wedge B)$ ,  $P(\neg B)$  eram adequadas aos conectivos lógicos OU, E e NÃO, respectivamente. Será que o conceito de probabilidade condicional  $P(A/B)$  é adequado à implicação lógica  $(B \rightarrow A)$ , isto é “se B então A”?

Sabe-se que  $(B \rightarrow A) = (A \vee \neg B)$  então:

$$\begin{aligned} P(B \rightarrow A) &= P(A \vee \neg B) = P(\neg B) + P(A \wedge B) = P(\neg B) + P(B \wedge A) = \\ &= P(\neg B) + P(B) \cdot P(A/B) \end{aligned}$$

como os valores de probabilidade são não negativos então



$$P(B \rightarrow A) > P(A/B)$$

Dessa forma, pode-se concluir que a probabilidade condicional não é adequada à implicação lógica.

A probabilidade condicional utiliza a ocorrência de evidência(s) para chegar à uma conclusão sobre uma hipótese diagnóstica, assim pode ser considerada uma forma de raciocínio.

### 3.3.11 – Razões de Probabilidade

Razões de probabilidade são chamadas ODDS ou LIKELIHOOD ratio. No contexto de SEP as seguintes razões de probabilidade podem ser utilizadas:

a) **ODDS a priori** – é a razão entre as probabilidades de duas hipóteses diagnósticas.

$$O(H) = P(H_i)/P(H_j)$$

b) **ODDS a posteriori** – é a razão entre as probabilidades condicionais de duas hipóteses diagnósticas dado que se conhece a ensaio cognitivo e.

$$O(H/e) = P(H_i/e)/P(H_j/e)$$

c) **LIKELIHOOD ratio** – é a razão entre as probabilidades condicionais de uma evidência e considerando a ocorrência de uma outra hipótese diagnóstica.

$$L(e/H) = P(e/H_i)/P(e/H_j)$$

Observe que a ODDS a priori mede quão mais provável é a hipótese  $H_i$  em relação à hipótese  $H_j$ . A ODDS a posteriori mostra quão mais provável é a hipótese  $H_i$  em relação à hipótese  $H_j$ , porém considerando a ocorrência da evidência e. Já a LIKELIHOOD ratio mede a sensibilidade (confiabilidade) da evidência e considerando duas hipóteses diagnósticas.

Geralmente estes termos ODDS e LIKEHOOD ratio são empregados quando se têm somente duas hipóteses diagnósticas  $H_1$  e  $H_2$ . Estas hipóteses, neste caso, são complementares, isto é,  $H_1 = \bar{H}_2$ .

Pode-se demonstrar que as seguintes expressões são verdadeiras:

$$O(H/e) = O(H) \cdot L(e/H) \quad \therefore \quad P(H/e) = O(H/e)/(1 + O(H/e)) = 1/(1 + O(H/e))$$

Agora será expandido o conceito acima considerando a ocorrência de uma outra evidência  $e_2$ . Isto é, deseja-se saber qual o efeito de uma outra evidência  $e_2$  na probabilidade de uma hipótese  $H_i$ .

$$\begin{aligned} O(H/e_1 \wedge e_2) &= P(H_i / e_1 \wedge e_2) / P(H_j / e_1 \wedge e_2) = \\ &= (P(H_i / e_1) / P(H_j / e_1)) \cdot (P(e_2 / H_i \wedge e_1) / P(e_2 / H_j \wedge e_1)) = \\ &= ((P(H_i) / P(H_j)) \cdot (P(H_i / e_1) / P(H_j / e_1))) \cdot (P(e_2 / H_i \wedge e_1) / P(e_2 / H_j \wedge e_1)) = \end{aligned}$$

supondo que as evidências  $e_1$  e  $e_2$  são independentes então

$$O(H/e_1 \wedge e_2) = P(H_i) / P(H_j) \cdot (P(H_i / e_1) / P(H_j / e_1)) \cdot (P(e_2 / H_i) / P(e_2 / H_j))$$

Note que a primeira parcela da multiplicação acima é a ODDS a priori que reflete a falta de informação de evidências. A segunda parcela reflete a informação sobre a ocorrência da evidência  $e_1$ . A terceira parcela reflete a informação sobre a ocorrência da evidência  $e_2$ . Estas duas últimas parcelas são as LIKELIHOOD ratios para as evidências  $e_1$  e  $e_2$ , respectivamente.

Supondo agora a ocorrência de  $k$  hipóteses diagnósticas obtém-se a probabilidade condicional da hipótese  $H_i$  dado às evidências  $e_1$  e  $e_2$ :

$$P(H_i / e_1 \wedge e_2) = \left\{ \sum_{j=1}^{k-1} [ P(H_i) / P(H_j) \cdot (P(H_i / e_1) / P(H_j / e_1)) \cdot (P(e_2 / H_i) / P(e_2 / H_j)) ] \right\}$$

resultando que

$$P(H_i / e_1 \wedge e_2) = [O(H_j / e_1) \cdot L(e_2 / H_j \wedge e_1)]^{-1}$$

### 3.3.12 – Atualização Bayesiana para uma Nova Evidência

Uma característica interessante da regra de atualização de Bayes é seu esquema computacional recursivo e incremental.

Seja  $H$  uma hipótese e  $e^n = (e_1, e_2, \dots, e_n)$  uma seqüência de dados independentes observados no passado e seja  $e$  um novo fato. A probabilidade condicional para nova evidência é:

$$P(H/e^n \wedge e) = P(H \wedge e^n \wedge e) / P(e^n \wedge e) =$$

$$= (P(e^n) \cdot P(H/e^n)) \cdot P(e/e^n \wedge H) / ((P(e^n) \cdot P(e/e^n)))$$

resultando

$$P(H/e^n \wedge e) = P(H/e^n) \cdot (P(e/e^n \wedge H) / P(e/e^n))$$

O resultado acima mostra que uma vez calculada a probabilidade condicional da hipótese  $H$  dado o conjunto  $e^n$  de evidências, isto é, o valor  $P(H/e^n)$ , os dados passados  $P(H/e^n)$ , os dados passados  $e^n$  podem ser desprezados e assim pode ser obtido o impacto da nova evidência  $e$ . A crença velha ( $H/e^n$ ) assume o papel de crença *a priori* no cálculo do impacto da nova informação  $e$ ; a probabilidade  $P(H/e^n)$  sumariza completamente a experiência passada e para sua atualização necessita ser multiplicada pela LIKELIHOOD ratio  $P(e/e^n \wedge H)$ . Esta razão mede a probabilidade do novo dado  $e$  considerando a hipótese  $H$  e os dados passados  $e^n$ .

Geralmente, adota-se que a LIKELIHOOD ratio é independente dos dados passados e considera somente a nova evidência  $e$ . Por exemplo, é razoável supor que um paciente desenvolverá um certo sintoma  $e$  considerando que tem a patologia  $H$  independentemente de que sintoma  $e^n$  teve no passado.

A natureza incremental do processo de atualização para a nova evidência  $e$  pode ser explorado utilizando a razão ODDS:

$$O(H/e^n \wedge e) = O(H/e^n) \cdot L(e/H)$$

$$\log O(H/e^n \wedge e) = \log O(H/e^n) + \log L(e/H)$$

assim o logaritmo da LIKELIHOOD ratio da evidência  $e$  pode ser visto como um peso da própria evidência  $e$ . Caso a evidência  $e$  suporte a hipótese  $H$  então terá um peso positivo, se for oposta a  $H$  então terá um peso negativo.

Vale aqui ressaltar que atualizar recursivamente as medidas de crenças está fortemente relacionada ao conceito de independência condicional, visto anteriormente.

### 3.3.13 – Hipóteses Multivaloradas

Nos domínios de aplicação de SEP são comuns a existência de várias hipóteses concorrentes cada uma com um conjunto distinto de evidências, dizemos então que o sistema tem

um conjunto multivalorado de hipóteses. Assim, o SEP deve realizar o diagnóstico, isto é, selecionar a hipótese (*output*), mais provável, a partir de um conjunto de evidências conhecidas (*input*) e que os dados de uma nova consulta ao sistema.

Supondo que se tenha  $m$  hipótese e  $n$  evidência, a seguir, será obtido o vetor de probabilidades condicionais das várias hipóteses concorrentes.

Dado um conjunto de evidências  $e_1, e_2, e_3, e_4, \dots, e_n$  a crença na  $i$ -ésima  $H_i$  é dada por

$$P(H_i / e_1, e_2, e_3, \dots, e_k, \dots, e_n) = \frac{P(e_1, e_2, e_3, \dots, e_k, \dots, e_n / H_i) \cdot P(H_i)}{P(e_1, e_2, e_3, \dots, e_k, \dots, e_n)}$$

Chamando a probabilidade  $[P(e_1, e_2, e_3, \dots, e_k, \dots, e_n)]^{-1}$  de constante de normalização e denotando por  $\alpha$ . E, supondo a independência condicional das evidências com relação a cada hipótese  $H_i$ , então

$$P(H_i / e_1, e_2, e_3, \dots, e_k, \dots, e_n) = \alpha \cdot P(H_i) \cdot \prod_{k=1}^n P(e_k / H_i)$$

Note que:

a)  $P(H_i)$  é o vetor de probabilidade *a priori* das diferentes hipóteses em consideração:

b) A constante de normalização  $\alpha$  deve ser calculada pela restrição de soma unitária das probabilidades  $P(H_i / e_1, e_2, e_3, \dots, e_k, \dots, e_n)$ .

Assumindo ainda, que para evidência  $e_k$  define-se o vetor  $\lambda^k$  de probabilidades condicionais das  $m$  hipóteses, isto é:

$$\lambda^k = (\lambda_1, \lambda_2, \dots, \lambda_m) \text{ onde } \lambda_i = P(e_k / H_i)$$

Finalmente, pode-se rescrever a equação original na seguinte forma:

$$P(H_i / e_1, e_2, e_3, \dots, e_k, \dots, e_n) = \alpha \cdot P(H_i) \cdot \Delta$$

onde 
$$\Delta = \lambda^1 \cdot \lambda^2 \cdot \lambda^3 \cdot \dots \cdot \lambda^n$$

Assim fica computacionalmente mais fácil obter o vetor de probabilidades condicionais das hipóteses  $H_i$  dado um conjunto de evidências. Por outro lado, com a expressão acima, fica também fácil verificar que a regra de atualização Bayesiana apresenta um esquema computacional recursivo e incremental a cada evidência nova que ocorrer.

No raciocínio probabilístico são utilizadas variáveis aleatórias para representar eventos e/ou objetos do domínio de aplicação. Ao se fazer várias instanciações deste pode-se alterar o estado atual do domínio. Assim envolve a obtenção de probabilidade conjunta de variáveis aleatórias. O que se faz necessário obter informações sobre as relações entre estas variáveis. No pior caso, necessitar-se-ia das probabilidades de todas as combinações possíveis de variáveis aleatórias, isto levaria à explosão combinatória. Por outro lado, associa-se valores de probabilidades condicionais a uma cadeia de regras SE-ENTÃO. Dessa forma, ao invés de se explicitar as distribuições conjuntas de probabilidades, são explicitadas as tabelas de probabilidades condicionais que por sua vez podem ser utilizadas para obter as distribuições conjuntas [KOEHLER 98].

### **3.4 – Agentes Inteligentes**

#### **3.4.1 – Inteligência Artificial Distribuída (IAD)**

A tecnologia de agentes inteligentes faz parte de pesquisas na área de Inteligência Artificial Distribuída, que pode ser conceituada como “o estudo do comportamento computacionalmente inteligente, resultante da interação de múltiplas entidades dotadas de certo grau, possivelmente variável, de autonomia. Estas entidades são usualmente chamadas de agentes e o sistema como um todo é usualmente chamado de “sociedade” [SOUZA 97].

Na Inteligência Artificial clássica, a metáfora da inteligência é baseada no comportamento humano individual e a ênfase é na representação do conhecimento e métodos de inferência, e na Inteligência Artificial Distribuída a metáfora usada é baseada no comportamento social, e sua ênfase é nas ações e interações.

Para Demazeau [DEMAZEAU 92], a Inteligência Artificial Distribuída pode ser dividida em duas áreas principais: Resolução Distribuída de Problemas (DPS) e Sistemas Multi-Agente (MAS). Em ambos os casos, são usados o termo agente, e não sistema baseado em conhecimento ou outra expressão para descrever as entidades que fazem parte da resolução do problema.

### **Resolução Distribuída de Problemas (DPS)**

Na resolução distribuída de problemas os agentes cooperam uns com os outros, dividindo e compartilhando conhecimento sobre o problema e sobre o processo para obter uma solução. Os agentes são projetados para resolver um problema ou classe de problemas específicos, possuindo porém uma visão incompleta do objetivo global, [DEMAZEAU 92].

Isto significa que a priori os agentes não podem ser usados para resolver qualquer outro problema similar. Em outras palavras, existe um domínio específico para ação dos agentes.

### **Sistemas Multi-Agentes (MAS)**

Um ambiente multi-agente pode ser definido como um sistema no qual diversos agentes interagem. No paradigma MAS o foco principal consiste em coordenar o comportamento de um conjunto de agentes autônomos, para obter a solução de um problema apresentado.

Demazeau [DEMAZEAU 92] expõe que Sistema Multi-Agente diz respeito as atividades do agente autônomo. Por autônomo, entende-se que o agente tem sua própria existência, independente da existência de qualquer problema a resolver. Os agentes existem em um ambiente comum e cada um pode colaborar com os outros para realizar um objetivo comum.

#### **3.4.2 – Definição de agente**

Ainda não existe um acordo sobre uma definição do que é exatamente um agente. Portanto, segue algumas definições de agentes segundo alguns autores da área.

Wooldridge e Jennings [WOOLDRIDGE 95] consideram que a resposta à questão “o que é um agente?” é tão imprecisa quanto a resposta à questão “o que é inteligência?” visto que, nos dois casos, no cenário computacional, não existe uma definição para o tema aceita universalmente.

De acordo com Souza [SOUZA 97], agentes inteligentes é um software que realiza um conjunto de tarefas em benefício do usuário ou de outro programa, utilizando certo grau de independência ou de autonomia e ao fazê-lo, empregam algum conhecimento ou representação dos objetivos ou preferências do usuário.

Agente pode ser definido como um software que atua como um representante para outro partido, com o propósito expresso de desempenhar ações que são benéficas para a parte representada, conforme [FLEISCHHAUER 96].

Já Jennings [JENNINGS 96] considera agente como sendo um “programa de computador que simula um relacionamento humano fazendo alguma coisa que outra pessoa poderia fazer por você”.

### **3.4.3 – Propriedades**

#### **Agência**

Pode ser conceituada como o grau de autonomia e autoridade investida no agente e pode ser medida, ao menos qualitativamente, através da natureza da interação entre o agente e outras entidades do sistema, segundo Souza [SOUZA 97].

#### **Autonomia**

É definida como a capacidade do agente de executar o controle sobre suas próprias ações. A capacidade do agente de operar isoladamente, adicionada à capacidade de decidir sobre o que fazer enquanto opera neste estado, juntas constituem a independência do agente [AUER 95].

Autonomia é a capacidade de tomar ações conduzindo para o término de alguma tarefa ou objetivo, sem a interferência do usuário final. Os agentes tomam o interesse, necessidades, desejos, etc. como entrada e se apropria delas para realizar as tarefas esperadas, segundo Fleischhauer [FLEISCHHAUER 96].

## **Comunicabilidade**

Os agentes, no curso da realização de seus objetivos, devem acessar informações sobre o estado atual do ambiente externo. Isso requer uma habilidade de comunicar-se com os repositórios dessa informação, conforme citado Fleischhauer [FLEISCHHAUER 96].

Fleischhauer [FLEISCHHAUER 96], define comunicabilidade como a capacidade de trocar informações com outras entidades, que podem ser agentes, humanos, objetos, seu ambiente, etc.

## **Inteligência**

A habilidade de raciocinar é uma das principais características que diferenciam agentes inteligentes de agentes ditos mais robóticos [SOUZA 97].

Auer [AUER 95] afirma que durante o processo de determinação da ação mais adequada à situação, o agente defronta-se com ambigüidade nos 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.

Wolldridge [WOOLDRIDGE 95] identificou duas idéias que tem norteado sua pesquisa:

- Situação e Incorporação – a inteligência “real” está situada no mundo e não em sistemas incorpóreos como provadores de teoremas e sistemas especialistas;
- Inteligência e Emergência – o comportamento inteligente surge com o resultado das interações do agente com seu ambiente. Além disso, “inteligência está nos olhos de quem vê”, não sendo portanto uma propriedade inata isolada.



## **Mobilidade**

A característica de mobilidade é especificada por Franklin & Graesser [FRANKLIN 96] como a capacidade do agente de transportar-se de uma máquina à outra.

Para Wooldridge [WOOLDRIDGE 95], mobilidade é a habilidade para movimentar-se pela rede eletrônica.

## **Reatividade**

Conforme Wooldridge & Jennings [WOOLDRIDGE 95], a reatividade é a propriedade que permite aos agentes perceberem seus ambientes e responderem adequadamente às mudanças neles ocorridas.

O ambiente pode ser o mundo físico, um usuário via uma interface gráfica, uma coleção de agentes, a Internet, ou todos esses combinados, segundo [SOUZA 97].

## **Flexibilidade**

A flexibilidade é verificada nos agentes, em função da característica das ações executadas, quando estas não são preestabelecidas em roteiros, segundo Franklin & Graesser, [FRANKLIN 96].

Desta forma, a flexibilidade reside na habilidade dos agentes de escolher dinamicamente as ações e a sua seqüência de execução, em resposta a um estado do ambiente, [AUER 95].

## **Planejamento**

A habilidade do agente de sintetizar e escolher entre diferentes cursos de ações, com o propósito de alcançar seus objetivos, é chamado de planejamento. Segundo Auer [AUER 95], o termo “planejamento” (*planning*) é o processo pelo qual um agente deliberativo move-se do

modelo para a ação. Especificamente é o processo através do qual o agente determina qual ação é apropriada à situação.

### **3.4.4 – Classificação de Agentes**

Dependendo das características mais fortemente evidenciadas, pode-se classificar o agente dentro de 7 (sete) tipos que estão descritos abaixo, pesquisados por Fleischhauer [FLEISCHHAUER 96]:

#### **Agentes Inteligentes**

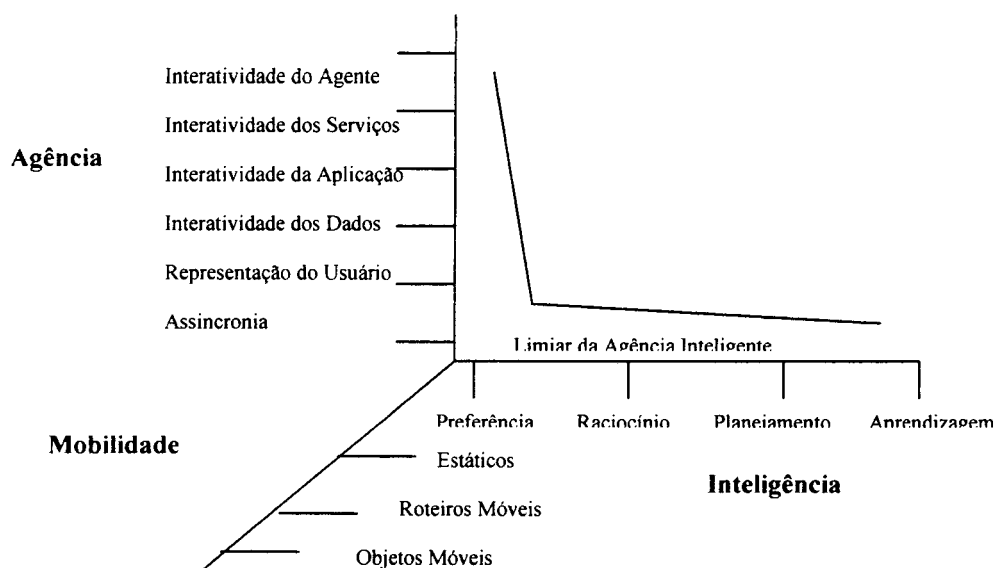
Agentes inteligentes são entidades que realizam um conjunto de operações em favor de um usuário ou outro programa com algum grau de independência ou autonomia, e assim, empregam algum conhecimento ou representação dos objetivos ou aspirações do usuário. Agentes inteligentes podem ser descritos em termos de espaço definido por três dimensões de agência, inteligência e mobilidade, conforme apresentado por O' Connor da IBM em [O'CONNOR 96].

O grau de agência é realçado, aumentado se o agente representa o usuário em algum meio. Um agente mais avançado pode interagir com outras entidades tal como, dados aplicações ou serviços. Além disso, colaboram e negociam com outros agentes.

Inteligência é o grau de raciocínio e comportamento sábio: a habilidade do agente para aceitar a declaração de objetivo do usuário e realizar a tarefa delegada a ele. No mínimo, pode partir de declarações de preferências, talvez na forma de regras com um motor de inferência ou algum outro mecanismo de raciocínio. Níveis altos de inteligência incluem um modelo de usuário ou outra forma de compreender e raciocinar sobre o que o usuário deseja fazer, e planejar o meio para atingir estes objetivos. Em um nível mais alto da escala de inteligência, estão os sistemas de aprendizagem e adaptação ao ambiente, em termos de objetivos do usuário, e em termos de recursos avaliáveis pelo agente.

A mobilidade aparece numa terceira dimensão, quando a aplicação do agente é projetada para atuar em uma rede. Mobilidade é o grau para o qual agentes por si próprios navegam pela rede. Alguns agentes podem ser estáticos, outros residem numa máquina cliente (para gerenciar a interface do usuário, por exemplo) ou são instanciados no servidor.

Na figura abaixo se pode observar a forma de medir o grau de inteligência que um agente pode atingir.



**Figura 6 – Dimensionamento de um agente**

Fonte: IBM Corporation

## Agentes Móveis

Conforme levantamento feito por Fleischhauer [FLEISCHHAUER 96], observa-se que a tecnologia de agentes estáticos tem sido utilizada nas últimas décadas em muitas áreas de aplicações, como, Interface Gráfica (Windows 95 Wizards), Aplicações (Assistente Ortográfico

do Word Microsoft), Sistemas Especialistas. Entretanto, o uso da tecnologia de agentes móveis é novo. Não há muitas aplicações desenvolvidas utilizando tecnologia de agentes móveis. Porém, há alguns candidatos, como, (Down loaded applets Isun Microsoft Java applets ou Microsoft OLE controls), Agentes de viagem (General Magic Corporation's Telescript agentes) e Agentes de colaboração comercial (Crystaliz, Inc,'s MuBot agentes). Na literatura ainda algumas linguagens para criação de agentes móveis: Java da Sun; Telescript da Genaral Magic; Safe-Tcl da Sun e Object Rexx da IBM.

### **Agentes Autônomos**

Agentes autônomos são sistemas capazes de apresentar autonomia, propondo-se a agir no mundo real.

Um agente autônomo é um sistema situado dentro e em uma parte de um ambiente que percebe o ambiente e age sobre ele, através do tempo, em perseguição de sua própria agenda e, deste modo, afetando o que percebe no futuro. Ou seja, em suas ações correntes, afeta o seu ambiente, [FRANKLIN 96].

Franklin [FRANKLIN 96] complementam que agentes autônomos operam sem a intervenção direta do usuário ou outros, e tem algum tipo de controle sobre suas ações e estado interno.

### **Agentes Coordenados**

Coordenação é o ato de gerenciar interdependências entre atividades. Uma conduta de coordenação envolve o mais básico comportamento de um agente, assim como criação de regras, criação de organização de informação e negociação, ou outros mecanismos de resolução de conflitos. A coordenação pode ser dividida em três partes: agente apresenta comportamentos de especificação (a criação de objetivos compartilhados), comportamento de planejamento (expressando conjunto de tarefas ou estratégias para acompanhamento dos objetivos) e comportamento de seqüencialidade (distribuindo tarefas para grupos e indivíduos, criando planos e seqüências compartilhadas, alocando recursos, etc.), segundo Fleischhauer [FLEISCHHAUER 96].

## **Agentes Aprendizes e Adaptativos**

Um agente aprendiz observa as ações do usuário em background, encontra padrões respectivos e automatiza-os sob aprovação. O paradigma de agente aprendiz usa a metáfora de assistente pessoal. Agentes aprendizes são, em particular, aplicáveis quando o domínio da aplicação contém comportamento repetitivo significativo e, ainda quando o comportamento repetitivo difere através do espectro do usuário [FLEISCHHAUER 96]. Pode ainda ser treinado pelo usuário com exemplos específicos [MAES 94].

## **Agentes Reativos**

Segundo Frozza [FROZZA 97], os agentes reativos são considerados entidades mais simples que os cognitivos devido às suas características de não apresentarem estados mentais (desejos, intenções, crenças e outros) e, portanto, não possuem capacidade de raciocínio e de planejamento.

Em um sistema multiagente, não é necessário que cada agente seja individualmente inteligente, pois a idéia principal é que um comportamento global inteligente seja alcançado a partir do comportamento individual do agente. Para ilustrar esta idéia Frozza [FROZZA 97] utiliza o exemplo de uma colônia de formigas: mesmo que uma única formiga apresente poucas capacidades (pareça não ser muito inteligente), o comportamento de uma colônia de formigas, como um todo, é bem estruturado e fez com que as formigas sobrevivessem por milhões de anos.

Tudo que um agente reativo sabe sobre as ações e o comportamento dos outros agentes é percebido pelas mudanças no ambiente. Fazendo-se novamente a analogia com as formigas, sabe-se que não há um modelo de comunicação direta entre elas, que na busca por comida elas seguem um rastro químico. Cada agente, individualmente, exibe comportamento de acordo com a situação na qual se encontra no ambiente de solução de algum problema. Quando o ambiente altera-se, os agentes mudam seu comportamento.

O modo de funcionamento dos agentes reativos é por estímulo e resposta: a relação do agente reativo com seu ambiente ocorre através de respostas a estímulos recebidos. O agente executa uma determinada ação quando uma certa condição for satisfeita. Uma característica importante dos agentes reativos é que eles não possuem memória das suas ações executadas no

passado nem qualquer previsão das ações a serem executadas no futuro, segundo Demazeau [DEMAZEAU 92].

## **Agentes Cognitivos**

Agentes cognitivos são baseados em modelos de organizações sociais, no senso de sociedades humanas (grupos, hierarquias, comércios) conforme Demazeau [DEMAZEAU 92]. Com os agentes, há uma representação explícita de ambiente e de membros da sociedade. Eles podem raciocinar sobre as ações tomadas no passado e planejar as ações a serem tomadas no futuro. Do ponto de vista de uma sociedade, ele é geralmente composta de um número pequeno de membros. Um exemplo de agente cognitivo pode ser observado em Demazeau [DEMAZEAU 92].

### **3.4.5 – Funções dos Agentes**

Segundo Maes [MAES 94], um grande número de usuários farão uso dos computadores e estações de trabalho, num futuro muito próximo, tornando imperiosa a necessidade de mudanças do paradigma de interação usuário - computador, gerenciamento direto, o qual exige que o usuário inicialize e monitore a totalidade dos eventos.

O paradigma emergente, denominado gerenciamento indireto, está embasado na tecnologia de agentes. Nele, o usuário é engajado em um processo onde homem e agentes computacionais inicializam a comunicação, monitoram eventos e executam tarefas.

No gerenciamento indireto o agente é visto como um assistente pessoal do usuário, sendo a ele atribuídas quatro funções:

- desempenhar tarefas em favor do usuário;
- treinar ou ensinar o usuário;
- ajudar diferentes usuários colaboradores;
- monitorar eventos e processos.

### 3.5 – Aprendizagem

Existem várias definições de aprendizagem, entre as quais, algumas estão mencionadas abaixo:

A aprendizagem é “o processo de adquirir conhecimento ou habilidade”, ou numa abordagem psicológica a aprendizagem é “a modificação do comportamento através da experiência” [PATRICK 86].

Segundo Simom [SIMOM 84], aprendizagem é “qualquer mudança em um sistema que permita que o sistema se comporte melhor a próxima vez que fizer a mesma tarefa ou várias tarefas tiradas do mesmo grupo”.

Para Saitta [SAITTA 95]: “aprendizagem é uma atividade fundamental de qualquer agente inteligente ou artificial. Para cada tarefa que o agente tem que fazer necessita adquirir as ferramentas em forma de conhecimento declarativo procedimental ou de habilidades cognitivas/motoras”.

A essência da aprendizagem é a aquisição de novo conhecimento, além de descrever e modelar sistemas físicos e seus comportamentos, e incorporar uma variedade de representações (desde simples modelos intuitivos mentais, exemplos de imagens, testes de equações matemáticas e leis físicas). Uma pessoa aprende se seu conhecimento se expande num amplo alcance de situações e está mais bem capacitada para predizer o comportamento físico do mundo. Neste contexto, aquisição de conhecimentos é definida como aprender nova informação simbólica acoplada a uma capacidade de aplicar esta informação de uma maneira mais efetiva, [PEREZ 97].

Um outro tipo de aprendizagem é o melhoramento gradual das habilidades motoras e cognitivas através da prática. Para realizar uma tarefa tal como aprender a dirigir um carro ou bicicleta, adquirir o manual de como fazer estas atividades representa só a parte inicial para desenvolver a habilidade requerida. O processo de aprendizagem em refinar a habilidade mental ou a coordenação motora aprendida, por práticas e por correção do desvio, até atingir o comportamento desejado. A pessoa que aprende sempre mistura ambas atividades [CARBONELL 84].

Pode-se considerar a aprendizagem como :

Aquisição de conhecimentos – *adquirir novas informações simbólicas.*

Refinamento de habilidades – *ocorre em um nível subconsciente em virtude de práticas repetidas.*

A Inteligência Artificial pesquisa sobre o processo de aprendizagem humana com o objetivo de desenvolver sistemas que procuram emular o comportamento inteligente humano, tal como tomar decisões, fazer inferências, e a capacidade de adquirir novos conhecimentos. Quando um sistema tem a capacidade de aprender novos conhecimentos pode-se dizer que possui a característica de Aprendizagem de Máquina (AM) (“Machine Learning”) [PEREZ 97] .

Considerando que a aprendizagem de máquina está relacionada com a aquisição de conhecimentos, é necessário o desenvolvimento de ferramentas relacionadas com a aprendizagem automática. Quando isto ocorrer poderá facilitar o desenvolvimento de sistemas especialistas pelos próprios especialistas, diminuindo assim o trabalho atual do engenheiro de conhecimentos [CARVALHO 95] .

### **3.5.1 – Paradigmas de Aprendizagem de Máquina**

Em Aprendizagem de Máquina, os quatro paradigmas que mais se evidenciam são: **aprendizagem indutiva, analítica, conexionista e evolucionária (paradigma genético)**. Com exceção do paradigma conexionista , os demais são paradigmas de aprendizagem simbólica. Os paradigmas simbólicos servem para: reconhecimento de padrões discretos, aquisição de novas descrições de conhecimentos, aquisição de regras para Sistemas Especialistas melhorarem a eficiência de sistemas baseados em regras, aprendizagem com raciocínio por instruções e arquiteturas de raciocínio integrado, para apoio em sistemas de solução de problemas em cooperação, quando o usuário e o sistema devem combinar fontes e raciocínio em conjunto ou quando cada um tenta instruir o outro com conhecimento para ser codificado de maneira explícita [PEREZ 97] .

A seguir, serão mencionados, de maneira resumida, os quatro paradigmas da Aprendizagem de Máquina.



## Aprendizagem Indutiva

A aprendizagem indutiva é aquela que induz uma descrição de um conceito geral (saída do sistema) através de uma seqüência de exemplos (exemplos positivos) e do conhecimento de contra - exemplos (exemplos negativos) do referido conceito. Desenvolve-se uma descrição do conceito do qual todos os exemplos positivos possam ser outra vez derivados quando se realiza uma instanciação, mas nenhum exemplo negativo prévio será outra vez derivado pelo mesmo processo [CARBONELL 90].

**Fatores** que determinam o poder dos sistemas indutivos, segundo [PEREZ 97]:

- **A descrição da linguagem.** É a representação da linguagem na qual exemplos de entrada e conceitos de saída são expressos. Esta representação pode ser, por exemplo, através de lógica proposicional.
- **A classificação de exemplos e o ruído.** Os sistemas que aprendem através de exemplos supõem que cada exemplo foi classificado corretamente como positivo ou negativo com respeito ao conceito desejado (saída), isto é, supõe-se que existe precisão pelo nível dos dados bem estruturados. Mas uma suposição é sempre restrita a aplicações do mundo real. Novos sistemas estão explorando uma nova classificação dos exemplos: precisos e imprecisos. A nova classificação se realiza através de exemplos especificados parcialmente, isto é, exemplos com alguns atributos desconhecidos, determinando os erros nos valores dos atributos, e as diferenças existentes entre os atributos. Se o número de exemplos é muito grande é apropriado utilizar técnicas estatísticas dentro dos métodos de aprendizagem.
- **Tipos de conceitos (saídas).** Para diferenciar os tipos de conceitos em um sistema há um conjunto de testes, os quais separam os exemplos de um conceito de exemplos pertencentes a outros conceitos conhecidos pelo sistema. Frequentemente a descrição de um conceito é codificada como um caminho dentro de uma árvore de decisão. Outros sistemas aprendem através de “conceitos característicos” os quais são compactos e bem estruturados. Tais sistemas podem se comunicar com o usuário mais facilmente e pode ter uma maior utilidade quando estes conceitos característicos devem ser interpretados por alguma outra parte do sistema, embora percam a completa exatidão. Os conceitos característicos são frequentemente codificados como molduras (“frames”) ou fórmulas

lógicas. A capacidade de um sistema aprender de forma indutiva é devido ao tipo de conceito adquirido.

- **A fonte dos exemplos.** Para o sistema de aprendizagem inicial é preciso ter um fluxo de exemplos classificados para um único conceito adquirido. Para que um sistema adquira novos conceitos estes conceitos podem ser dados por um tutor ou por uma fonte externa, se disponível. Neste caso, o sistema deverá fazer sua própria classificação de exemplos.
- **Indução incremental versus indução fixa.** Os sistemas de aprendizagem por indução fixa, consideram todos os exemplos (positivos e negativos) que podem ocorrer. Técnicas incrementais realizam um processo gradual da formação de conceitos, como um reflexo das situações do mundo real, no qual aprender é um processo contínuo.

## Aprendizagem Analítica

A aprendizagem analítica é um paradigma que vem sendo estudado recentemente e utiliza poucos exemplos (frequentemente de um único exemplo) à luz de uma teoria do domínio (um conjunto de regras que descrevem relacionamentos entre objetos e ações em um domínio). Os métodos envolvidos são mais dedutivos do que indutivos, utilizam experiências passadas de problemas já solucionados (os exemplos) para se chegar à solução de novos problemas. Os métodos analíticos melhoram a eficiência de um sistema sem sacrificar a exatidão ou a capacidade de generalização, ampliando sua biblioteca de descrições de conceitos (conceito-objetivo) [PEREZ 97].

Pode-se citar alguns **aspectos** dos métodos analíticos [PEREZ 97]:

- **A representação dos exemplos:** corresponde a uma parte familiar da solução de um problema, e para aprender utiliza esse exemplo somado a uma teoria do domínio (conhecimento inicial).
- **Aprender através de falhas ou êxitos:** os sistemas podem aprender através da repetição dos êxitos ou através das falhas já realizadas para evitar novas falhas similares.
- **Grau de generalização:** o controle dos conhecimentos adquiridos pode ser: específico (para situações do exemplo) ou generalizado (através da teoria do

domínio), onde a generalização é alcançada através da eliminação de informação irrelevante.

- **Aprendizagem de laço aberto e fechado:** **Aberto** – implica em aquisição de novo conhecimento; **Fechado** – permite avaliações futuras dos novos conhecimentos, para decidir sobre a modificação ou mesmo a eliminação tendo em vista a melhora do desempenho desejado do sistema.

## **Aprendizagem Conexionista**

É uma aprendizagem não simbólica. A aprendizagem conexionista é conhecida como aprendizagem por redes neurais. As redes neurais procuram imitar a estrutura física do cérebro humano como seu modelo base.

## **Aprendizagem Evolucionária**

O paradigma evolucionário foi projetado para desenvolver algoritmos inspirados na evolução biológica. Os algoritmos que utilizam métodos computacionais (computação evolucionária), inspirados na teoria da evolução se chamam algoritmos evolucionários e constituem um paradigma de aprendizagem de máquina [PEREZ 97].

Os dados dos algoritmos evolucionários consistem em uma população de objetos abstratos, exemplos:

- estruturas simbólicas,
- redes neurais,
- palavras de um alfabeto,
- variáveis de um problema de otimização,
- programas de computador.

Estes objetos são manipulados por operadores inspirados nas regras de seleção e outros operadores inspirados na evolução biológica, ou operadores genéticos que agem como operadores de busca na solução de um determinado problema. Os mais comuns são recombinação e mutação. Cada indivíduo da população tem um certo grau de aptidão, o qual

depende do ambiente. Desta forma a população tem a tendência de, após várias gerações, ter elementos com altos valores de aptidão [PEREZ 97].

### 3.5.2 – Métodos de Aprendizagem de Máquina

Segundo Pérez [PEREZ 97], existem alguns métodos de aprendizagem cuja classificação está baseada em “entendimento de estratégias de aprendizagem para **adquirir** e **refinar** o conhecimento de um sistema, conforme Tabela 3.

**Tabela 3 – Métodos de Aprendizagem**

MÉTODOS DE APRENDIZAGEM	CARACTERÍSTICAS
Por Memorização e Implantação Direta de Novo Conhecimento.	Inclui aprendizagem por memorização, por imitação e por ser programado.
Por Instruções	Aprender por instruções e alguns conselhos (heurísticas), que devem ser operacionalizados. Refinamento de conhecimento.
Por Analogia	Baseia-se na aprendizagem analítica a partir de poucos exemplos. Processos alternativos: - por transformação. - por derivação.
Por Dedução	Conduz inferências dedutivas.
Por Exemplos	Inferências indutivas.
Baseada em Explicação	Inferências dedutivas.
Por Observação e Descoberta	Não supervisionado e com inferências indutivas

#### **Por Memorização e Implantação Direta de Novo Conhecimento**

Em síntese este método requer pouco ou nenhuma inferência por parte do estudante que neste caso é a máquina. Inclui aprendizagem por memorização, aprendizagem por imitação, aprendizagem por ser construído ou por ser programado. É um método amplamente estudado para fornecer conhecimento a um sistema computadorizado: incorporando conhecimento no

hardware da máquina através de programação, e construindo-se a base de dados para todo tipo de aplicações [MICHALSKI 87].

### **Por Instruções**

Uma maneira trivial para que uma máquina aprenda uma tarefa é ser programada pelo programador (tutor), aceitando conselhos de alto nível. Uma máquina pode aprender um jogo por instruções próprias (regras) do jogo e alguns conselhos de como proceder para vencer. Estes conselhos são chamados de heurísticas. Por exemplo, no jogo de cartas copas poderia utilizar a heurística: “evite pegar pontos”. As heurísticas e as instruções próprias de tarefa devem ser operacionalizadas no programa. A operacionalização transforma um conselho em vários subconselhos até que o sistema reconheça que o conselho é executável [PEREZ 97].

### **Por Analogia**

O método por analogia envolve transformar ou estender conhecimento (ou habilidade) aplicável em um domínio para desempenhar uma tarefa similar em outro domínio. Por exemplo, fazer o diagnóstico de um novo paciente “paciente B” conhecendo os dados para o diagnóstico de um caso “paciente A” similar ao caso “paciente B”. A aprendizagem por analogia requer uma maior quantidade de inferência por parte da máquina do que a aprendizagem por instrução. Os conhecimentos ou habilidades pertinentes devem ser acessados na memória e apropriadamente transformados para serem aplicações em uma nova situação ou para um novo problema.

A aprendizagem por analogia é baseada na aprendizagem analítica a partir de poucos exemplos, porém com uma rica teoria subjacente do domínio. Os conhecimentos relativos aos problemas já solucionados são transferidos ao problema atual e este conhecimento é utilizado na solução do novo problema [CARBONEL 84].

## **Por Dedução**

Um sistema com aprendizagem que usa este método conduz inferências dedutivas no conhecimento que este possui e o conhecimento fornecido na sua utilização. Este é feito a fim de estruturar o conhecimento dado em formas mais úteis e efetivas, ou determinar conseqüências importantes do conhecimento. Por exemplo, dado um conjunto 1, 2, 6, 24, 120, 720, um sistema com aprendizagem poderia representá-lo em um equivalente, mas de forma mais curta como  $n!$ ,  $n=1...6$ . Para fazê-lo, o sistema deve, naturalmente, saber o conceito de um fatorial [PEREZ 97].

## **Por Exemplos**

Este método é um caso especial de aprender por indução. Para criar a capacidade de aprendizagem de um sistema segundo este método precisa-se de um conjunto de exemplos (positivos) de um conceito, e opcionalmente de contra - exemplos (negativos), para que o sistema possa induzir uma descrição geral do conceito. Este processo continuaria até que todos os conceitos fossem aprendidos pelo sistema, devendo-se chegar a uma base de conhecimento consistente, através da inferência de exemplos ou fatos disponíveis (aprendizagem indutiva) [PEREZ 97].

## **Baseado em Explicação**

O desenvolvimento da aprendizagem baseada em explicação tem sido um esforço de transição gradual através de pesquisas exploratórias para métodos mais gerais e bem definidos. A raiz da aprendizagem baseada em explicação pode ser a descoberta dos primeiros programas de aprendizagem analítica, que melhoravam seus desempenhos com base na experiência. A aprendizagem dependia de analisar por que exemplos observados tinham alguma propriedade importante. Estes projetos enfatizaram aprendizagem baseada em conhecimentos de um único exemplo, aprendizagem analítica, em contraste a muitas das pesquisas em aprendizagem indutiva que estava sendo conduzidas naquele momento. Finalmente para identificar aquelas abordagens em um só método foi sugerido o termo aprendizagem baseada em explicação [PEREZ 97].

## **Por Observação e Descoberta**

Representa uma outra forma da aprendizagem indutiva. Este método de "aprender sem professor" inclui uma variedade de processos, tais como criar classificações de observações dadas, relações de descoberta e leis para governar um dado sistema ou formar uma teoria para explicar um fenômeno [PEREZ 97].

O método é aplicado quando são dadas uma coleção de fatos (observações) e alguém deseja desenvolver uma descrição geral (uma teoria) que explique os fatos.

### **3.6 – Considerações Finais**

Pelo estudo bibliográfico realizado, pode-se constatar que os tópicos abordados neste capítulo são de grande importância em SEP que utilizam uma base de conhecimento dinâmica.

A importância das probabilidades bem como o Teorema de Bayes que compõem as redes bayesianas refletem uma teoria que possibilita o cálculo das evidências sem que haja conflitos como ocorre na lógica tradicional. Pode-se reduzir o valor da crença. Por exemplo, quando se tem uma probabilidade a priori de um paciente estar com febre, observa-se o seguinte: sabendo-se que a febre pode ser derivada de uma gripe ou de uma inflamação no ouvido, o SEP terá que decidir qual das evidências é mais provável, sem descartar totalmente a menos provável. Assim, o sistema SEDIN pode chegar ao cálculo das diversas evidências da RB, umas com maior probabilidades e outras com menor probabilidades.

Ao estudar os conceitos de Agentes Inteligentes pode-se associar os elementos que compõem o sistema SEDIN como agentes que são entidades dotadas de um certo grau de inteligência. No SEDIN os agentes são: a RB do SISSPAN, a interface gráfica do SEDIN, o módulo de ajuda e o usuário, cada um com níveis diferentes de inteligência.

Ao estudar os conceitos de Aprendizagem pode-se notar que, no momento em que o usuário utiliza o SEDIN, ele interage com os outros agentes do sistema e obtém aprendizagem, isto é, o sistema aprende. Deste modo, o SEDIN parte de uma RB estática e esta pode ser modificada pelo usuário.

O próximo capítulo apresenta o protótipo do sistema SEDIN, sua concepção e implementação.



## **4 – O SISTEMA SEDIN**

Neste capítulo será apresentado o protótipo do sistema SEDIN na fase de concepção e implementação. Na fase da concepção será feita uma descrição da arquitetura geral do SEDIN, da base de conhecimento do SISPAN, da base de conhecimento do SEDIN bem como a metodologia empregada. Na fase da implementação serão feitos comentários sobre a shell e linguagem de programação utilizada bem como uma descrição das telas que compõem a interface do SEDIN .

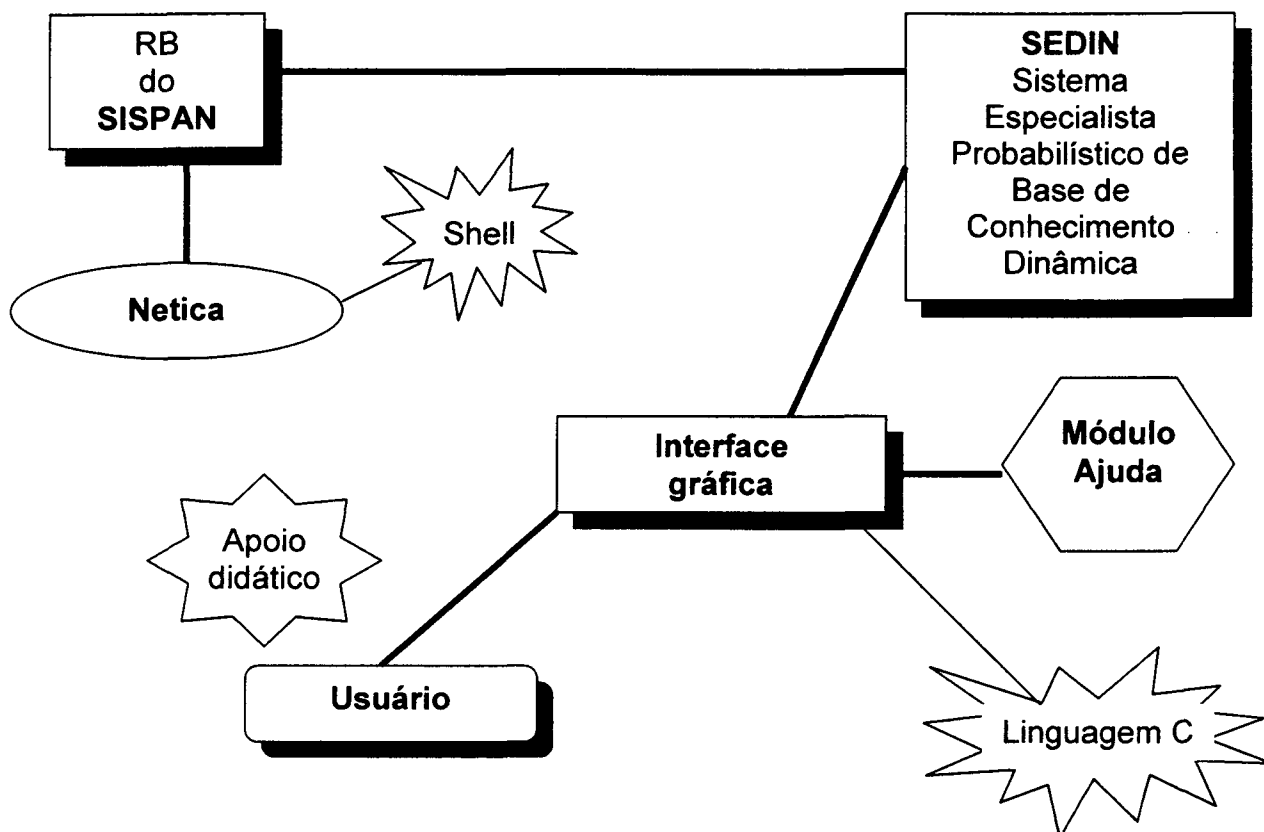
### **4.1 – CONCEPÇÃO DO SISTEMA SEDIN**

#### **4.1.1 – Arquitetura Geral do Sistema SEDIN**

A arquitetura desta aplicação proposta foi composta das seguintes partes:

- a base de conhecimento do SISPAN (Sistema Pediátrico Para Avaliação Nutricional),
- a base de conhecimento do SEDIN;
- a interface gráfica para fazer a interação com o usuário,
- o módulo de Ajuda.

Na Figura 7 pode-se ver o esquema de arquitetura geral que foi proposto nesta pesquisa:



**Figura 7 – Arquitetura do Sistema**

A seguir será dada uma descrição das partes que compõem esta arquitetura.

#### **4.1.2 – Base de Conhecimento do Sistema SISPAN**

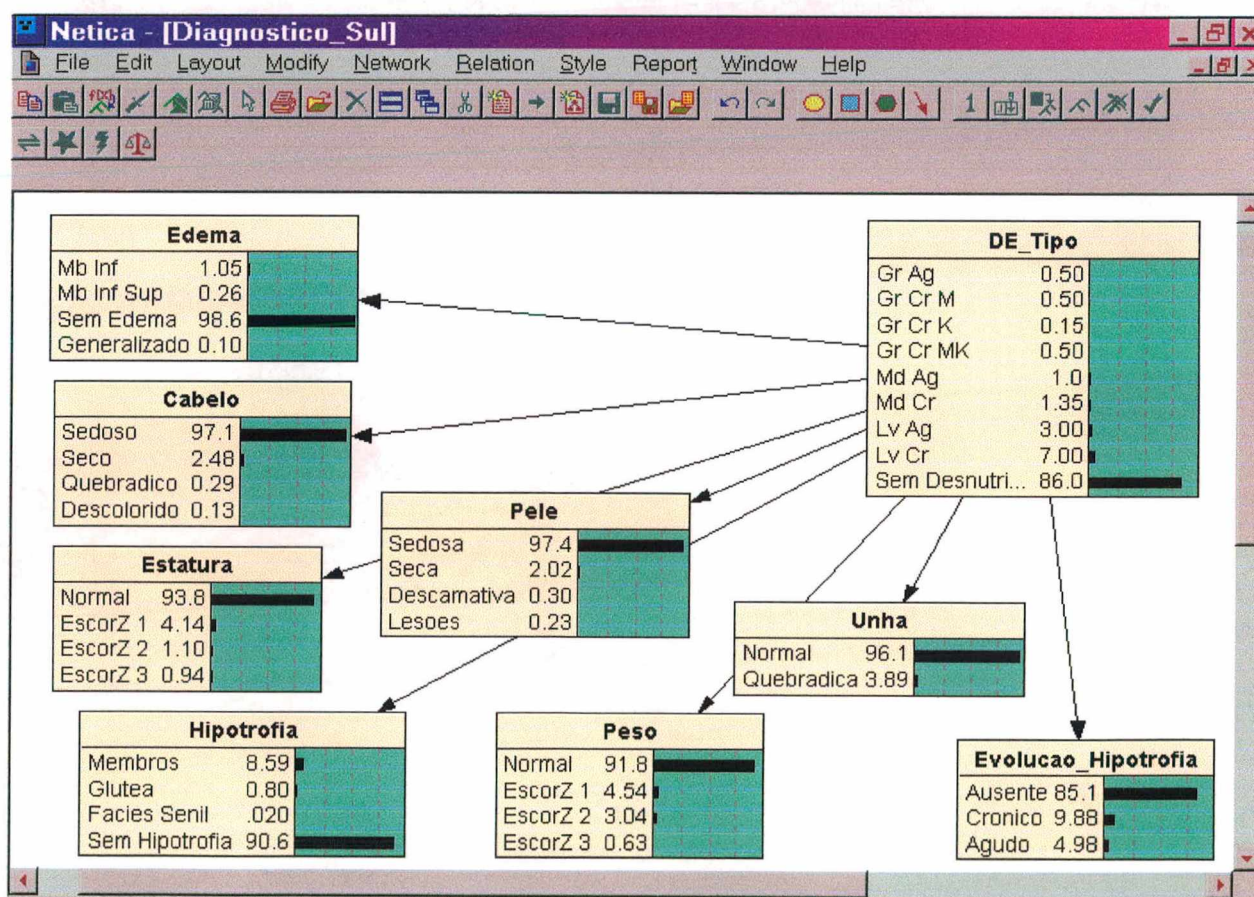
Para o desenvolvimento do aplicativo utilizou-se a Rede Bayesiana do SISPAN (Sistema Pediátrico Para Avaliação Nutricional) que serviu de base de conhecimento.

No SISPAN as variáveis de entrada são os sinais e sintomas do paciente relativos à desnutrição. A variável de saída é um vetor de probabilidades que indica quão provável é cada uma das hipóteses diagnósticas para o paciente em questão. Deste modo, a base de conhecimento foi composta por variáveis que foi estabelecida através da definição de regras e/ou fatos. Esta base de conhecimento foi representada por distribuições probabilidades.

A base de conhecimento do Sistema Especialista Probabilístico: SISPAN – Sistema Pediátrico Para Avaliação Nutricional, constitui um Sistema Especialista para avaliação do estado nutricional em crianças com até 2 anos de idade desenvolvida por Koehler [KOEHLER 98].

Neste Sistema Especialista Probabilístico as variáveis do sistema são os sinais e sintomas do paciente relativos à desnutrição. A variável de saída é um vetor de probabilidades que indica quão provável é cada uma das hipóteses diagnósticas para o paciente em questão.

A seguir será apresentado um exemplo da representação gráfica da base de conhecimento da Rede bayesiana (RB) para Avaliação do Estado Nutricional. Esta representação aparece no estado chamado “normal”, isto é, nenhum nó está clicado.



**Figura 8 – Representação gráfica da base de conhecimento (RB) para Avaliação do Estado Nutricional**

## A Rede Bayesiana a Priori

Uma vez construída a rede Bayesiana na Shell Netica, com as probabilidades das Hipóteses e as probabilidades das evidências, pode-se apresentar o cálculo, como exemplo, de algumas probabilidades da rede como seguem abaixo. Deste modo, será apresentada a rede do Sistema Especialista probabilístico da Avaliação Nutricional.

Na Figura 9, tem-se representado o conhecimento do especialista em uma (RB) rede bayesiana, onde as probabilidades condicionais *a priori* estão distribuídas nos nós. Esta RB é composta por nove nós, onde um representa o diagnóstico e os restantes representam os sinais e sintomas considerados.

Sejam apresentadas as seguintes probabilidades condicionais *a priori* que estão na Shell Netica.

Gr_Ag	Gr_Cr_M	Gr_Cr_K	Gr_Cr_MK	Md_Ag	Md_Cr	Lv_Ag	Lv_Cr	Sem_De...
0.500	0.500	0.150	0.500	1.000	1.350	3.000	7.000	86.000

**Figura 9 – Tabela no Netica que representa o vetor das probabilidades *a priori* da hipótese**

onde,

Gr_Ag	significa Desnutrição do tipo Grave Aguda
Gr_Cr_M	significa Desnutrição do tipo Grave Crônica Marasmo
Gr_Cr_K	significa Desnutrição do tipo Grave Crônica Kwashiokor
Gr_Cr_MK	significa Desnutrição do tipo Grave Crônica Marasmo-Kwashiokor
Md_Ag	significa Desnutrição do tipo Moderada Aguda

Md_Cr	significa Desnutrição do tipo Moderada Crônica
Lv_Ag	significa Desnutrição do tipo Leve Aguda
Lv_Cr	significa Desnutrição do tipo Leve Crônica
Sem_Desnutrição	significa ausência de Desnutrição.

E também as probabilidades condicionais *a priori* da variável **Edema**, conforme Figura abaixo.

The screenshot shows the Netica software window titled "Netica - [Edema Table (in net Diagnostico\_Sul)]". The interface includes a menu bar (File, Edit, Layout, Modify, Network, Relation, Style, Report, Window, Help) and a toolbar. Below the toolbar, there is a "Node:" dropdown menu set to "Edema" and a "Chance:" dropdown menu. To the right of these are buttons for "Apply", "Okay", "Load", and "Close". The main area contains a table with the following data:

DE_Tipo	Mb_Inf	Mb_Inf_...	Sem_Ed...	Generali...
Gr_Ag	1.000	0.200	98.700	0.100
Gr_Cr_M	1.000	0.200	98.700	0.100
Gr_Cr_K	90.000	7.000	0.000	3.000
Gr_Cr_MK	90.000	9.800	0.100	0.100
Md_Ag	4.000	0.900	95.000	0.100
Md_Cr	4.000	0.900	95.000	0.100
Lv_Ag	1.000	0.900	98.000	0.100
Lv_Cr	1.000	0.900	98.000	0.100
Sem_De...	0.300	0.100	99.500	0.100

**Figura 10 – Representação gráfica das probabilidades condicionais *a priori* da variável Edema**

Através do cálculo pode-se chegar a uma probabilidade de 98,6% das crianças não apresentarem Edemas.

$$\begin{aligned}
 P(\text{Edema}) = & \\
 = & P(\text{Gr\_Ag e Peso Normal}) + P(\text{Gr\_Cr\_M e Peso Normal}) + P(\text{Gr\_Cr\_K e Peso Normal}) + \\
 & + P(\text{Gr\_Cr\_MK e Peso Normal}) + P(\text{Md\_Ag e Peso Normal}) + P(\text{Md\_Cr e Peso Normal}) + \\
 & + P(\text{Lv\_Ag e Peso Normal}) + P(\text{Lv\_Cr e Peso Normal}) + P(\text{Sem\_Desnutrição e Peso Normal})
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Edema}) = & \\
 & + P(\text{Gr\_Ag}) \cdot P(\text{Peso Normal/Gr\_Ag}) + P(\text{Gr\_Cr\_M}) \cdot P(\text{Peso Normal/Gr\_Cr\_M}) + \\
 & + P(\text{Gr\_Cr\_K}) \cdot P(\text{Peso Normal/Gr\_Cr\_K}) + P(\text{Gr\_Cr\_MK}) \cdot P(\text{Peso Normal/Gr\_Cr\_MK}) + \\
 & + P(\text{Md\_Ag}) \cdot P(\text{Peso Normal/Md\_Ag}) + P(\text{Md\_Cr}) \cdot P(\text{Peso Normal/Md\_Cr}) + \\
 & + P(\text{Lv\_Ag}) \cdot P(\text{Peso Normal/Lv\_Ag}) + P(\text{Lv\_Cr}) \cdot P(\text{Peso Normal/Lv\_Cr}) + \\
 & + P(\text{Sem\_Desnutrição}) \cdot P(\text{Peso Normal/Sem\_Desnutrição})
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Edema}) = & \\
 = & 0,005 \cdot 0,987 + 0,005 \cdot 0,987 + 0,0015 \cdot 0,0 + 0,005 \cdot 0,1 + 0,01 \cdot 0,95 + 0,0135 \cdot 0,95 + \\
 & 0,03 \cdot 0,98 + 0,07 \cdot 0,98 + 0,86 \cdot 0,995 = 0,9864 \quad \text{ou} \quad 98,6\%
 \end{aligned}$$

De modo análogo a shell Netica calcula as probabilidades condicionais das outras evidências do sistema.

### A Rede Bayesiana *a posteriori*

Após a aquisição do conhecimento, isto é, tendo as probabilidades *a priori*, pode-se realizar **inferências na rede**. Informar os sinais e sintomas e obter como resposta o vetor de probabilidades *a posteriori* das hipóteses diagnósticas. É a atualização da rede bayesiana, ou seja, a RB propagará a(s) evidência(s) e atualizará as probabilidades das hipóteses diagnósticas.

Esta etapa da rede consiste na realização de **consultas** por parte do usuário ao Sistema Especialista.

A seguir, será exemplificado o cálculo que a shell Netica realiza para obter as probabilidades *a posteriori* diante de uma consulta por parte do usuário.

Considere-se que o usuário faça a seguinte pergunta ao sistema: Qual a probabilidade de uma criança estar com Desnutrição do tipo Leve Crônica, dado que esta criança esta com uma perda de Estatura no Escore\_Z1?

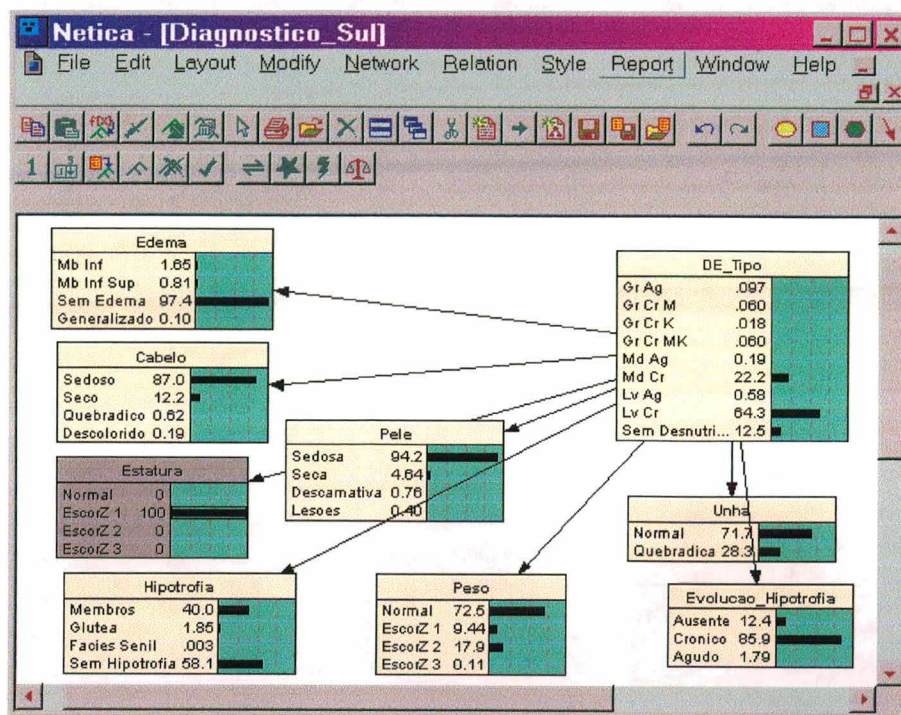


Figura 11 – Probabilidades *a posteriori* após consulta

Sejam

$$P(\text{Lv\_Cr}) = 0,07$$

$$P(\text{Estatura\_Escore\_Z1/Lv\_Cr}) = 0,38$$

$$P(\text{Estatura\_Escore\_Z1}) = 0,04136$$

Então, pelo Teorema de Bayes

$$P(\text{Lv\_Cr} / \text{Estatura\_Escore\_Z1}) = \frac{P(\text{Lv\_Cr}) \cdot P(\text{Estatura\_Escore\_Z1/Lv\_Cr})}{P(\text{Estatura\_Escore\_Z1})}$$

$$P(\text{Lv\_Cr} / \text{Estatura\_Escore\_Z1}) = \frac{0,07 \cdot 0,38}{0,0413575} = 0,6432$$

Da forma como foi demonstrada, com a utilização da rede bayesiana, pode-se obter facilmente qualquer probabilidade sobre o tipo de desnutrição de uma criança aonde o vetor de probabilidades *a posteriori* vão se alterando através da aquisição de informação das evidências.

### **4.1.3 – Shell para SEP**

Após ter feita uma pesquisa sobre SEP elegeu-se o sistema SISSPAN e utilizou-se a sua RB para desenvolver o SEDIN. Neste sentido, optou-se em utilizar a mesma shell.

Deste modo, foi utilizada a shell Netica 1.06. Mais especificamente o ambiente de trabalho do Netica API (Application Program Interface) com sua DLL (Dynamic Link Library) que contém uma biblioteca com todas as funções definidas na linguagem C para representar o conhecimento do especialista em uma rede bayesiana, composta de seus nós (variáveis), seus respectivos arcos e probabilidades condicionais a priori.

### **4.1.4 – Interface Gráfica do SEDIN**

Interface gráfica: Tem como objetivo principal fazer a comunicação entre o usuário e o Sistema Especialista Probabilístico Dinâmico (SEDIN). Será através da interface que o usuário poderá introduzir ou excluir variáveis a rede bayesiana e verificar a importância destas no SEP. A interface será projetada utilizando-se a linguagem de programação C++ Builder, versão 3.0, da Borland.

### **4.1.5 – Módulo de Ajuda do Sistema SEDIN**

Módulo ajuda: Este módulo tem como objetivo auxiliar o usuário oferecendo explicações, em forma de tópicos, sobre o domínio pesquisado. Conterá textos sobre o sistema SEDIN de um modo geral.

### **4.1.6 – Base de Conhecimento do Sistema SEDIN**

No desenvolvimento do sistema SEDIN a base de conhecimento inicial é a mesma base do SISSPAN, que desta maneira serve como parâmetro inicial (variáveis já existentes). Sendo que, as variáveis já existentes podem ser acrescentadas uma a uma pelo sistema SEDIN. Além das variáveis já existentes, o sistema SEDIN ainda tem a capacidade de introduzir novas variáveis. Isto é, se houverem novos sinais e sintomas de pacientes relativos à desnutrição, estes



podem ser introduzidos no sistema SEDIN. A variável de saída do sistema SEDIN também é um vetor de probabilidades que indica quão provável é cada uma das hipóteses diagnósticas para o paciente em questão. Da mesma como no sistema SISPAN, a base de conhecimento foi composta por variáveis que foi estabelecida através da definição de regras e/ou fatos.

#### **4.1.7 – Considerações Finais**

Nesta seção do capítulo 4 foi apresentado o protótipo de implementação do sistema SEDIN, onde foram descritas a sua arquitetura geral e a base de conhecimento do SISPAN. Na seqüência, também foi feita uma descrição da metodologia empregada na interface do SEDIN, bem como a sua base de conhecimento. No final foram apresentados os recursos necessários para a implementação do sistema SEDIN. Assim, partiu-se para a criação da aplicação com a utilização dos recursos disponíveis.

A próxima seção deste capítulo 4 apresenta aspectos importantes dos softwares utilizados bem como a implementação do sistema SEDIN.

### **4.2 – O SISTEMA SEDIN**

Nesta seção do capítulo 4 serão apresentadas as etapas realizadas para o desenvolvimento do SEDIN. Serão feitos comentários sobre a shell Netica e linguagem de programação utilizada. Bem como a implementação do sistema.

#### **4.2.1 – Softwares Utilizados**

##### **Shell Netica**

Para a construção do sistema, por questões práticas, funcional, econômicas, optou-se em utilizar o Netica 1.06. Esta shell é composta por dois ambientes de trabalho: 1) Netica Application e 2) Netica API. O Netica Application é a interface padrão Windows onde a base de

conhecimentos pode ser visualizada na forma de uma rede bayesiana. O Netica API é a DLL (Dynamic Link Library), uma biblioteca que contém todas as funções definidas na linguagem C para gerenciar a base de conhecimentos. Estas funções são para criar nós, adicionar links, compilar e gravar a rede, realizar as inferências na rede, dentre outras funções que podem ser observadas em [NETICA97].

## **Ambiente de Trabalho do Netica API**

Nesta pesquisa, inicialmente utilizaram-se os dois ambientes de trabalho do Netica para representar a base de conhecimento do SISPAN. Através do Netica *Application* foi criada a rede bayesiana com seus nós e links. Introduziram-se nela as probabilidades a priori e condicionais. Depois de estar definida, a rede bayesiana foi salva em um arquivo *.dnet*. Para acessar esta rede pré-gravada foi utilizada a linguagem C++Builder 3.0 juntamente com a DLL do Netica, o Netica API.

Como a utilização de uma rede bayesiana pré-gravada tinha um comportamento estático, isto é, não se podia alterar o número de nós na rede, abandonou-se então este ambiente de trabalho do Netica. Deste modo, a pesquisa seguiu apenas com a utilização do Netica API. Assim, a rede bayesiana com os seus nós, links e probabilidades a priori e condicionais foram introduzidos na própria aplicação.

## **Linguagem de Programação**

A linguagem de programação utilizada para desenvolver a interface gráfica do sistema foi o C++Builder 3.0 da Borland. Utilizou-se esta linguagem pela sua facilidade de lidar com objetos visuais e devido a DLL Netica API ser toda programada em C.

## 4.2.2 – Implementação

A interface gráfica foi projetada em várias partes. A seguir será apresentada uma descrição das principais telas do SEDIN:

- ♦ **Tela Principal:** Nesta tela do sistema o usuário visualiza as evidências do SISPAN (à esquerda), podendo alterar os seus estados livremente e em seguida fazer a atualização do vetor de probabilidades a posteriori correspondente (no centro da tela). Em paralelo, à direita está o vetor de probabilidades a posteriori do sistema SEDIN, o qual o usuário poderá ter acesso as evidências através do botão Variáveis. De modo semelhante, o usuário poderá fazer a atualização das probabilidades que estão neste vetor de probabilidades a posteriori. Com os dois sistemas atualizados o usuário pode aprender. Esta tela é considerada a tela principal e a partir dela o usuário poderá ter acesso a outras telas que compõe o sistema, como mostrado na Figura 12.

**SEDIN**  
Arquivo Formatar Alterar Ajuda  
Hora: 09:09:13 Data: 02/08/2000

**Diagnóstico da Desnutrição**

	Sispan	Sedin
<i>Grave Aguda</i>	0.0069	0.0002
<i>Grave Crônica Marasmo</i>	0.0259	0.0062
<i>Grave Crônica Kwashiorkor</i>	0.1035	0.0234
<i>Grave Crônica Marasmo-Kwashiorkor</i>	0.3451	0.1158
<i>Moderada Aguda</i>	0.0138	0.0000
<i>Moderada Crônica</i>	0.0699	0.2885
<i>Leve Aguda</i>	0.0414	0.0234
<i>Leve Crônica</i>	0.0966	0.0831
<i>Sem Desnutrição</i>	0.2968	0.4595

Atualizar      Atualizar

Variáveis

OK

**Evidências do Sispan**

Cabelo: Seco

Pele: [ ]

Unha: [ ]

Hipotrofia: [ ]

Evolução Hipotrofia: [ ]

Edema: [ ]

Peso: [ ]

Estatura: [ ]

Figura 12 – Visualização da tela principal do SEDIN

- ◆ **Tela Variáveis:** A segunda tela denomina-se *Variáveis*. Pode-se observar à esquerda da tela a lista das variáveis já existentes. Estas variáveis estão a princípio, desabilitadas, deste modo, não influenciam o sistema. Para ativá-las, isto é, introduzi-las no sistema, basta clicar sobre o nome delas. Se o usuário desejar introduzir uma nova variável no sistema ele deverá clicar no botão *Adicionar Nova Variável* que se encontra na parte inferior. Se uma nova variável for criada, o seu nome será visualizado à direita da tela abaixo de *Variáveis Novas* que constitui uma lista de novas variáveis do sistema. Qualquer variável poderá ser excluída do sistema com um simples clique sobre o seu nome. Na visualização da tela *Variáveis* pode-se observar, à direita, que a variável *Sexo* representa uma nova variável que foi introduzida no sistema conforme Figura 13.

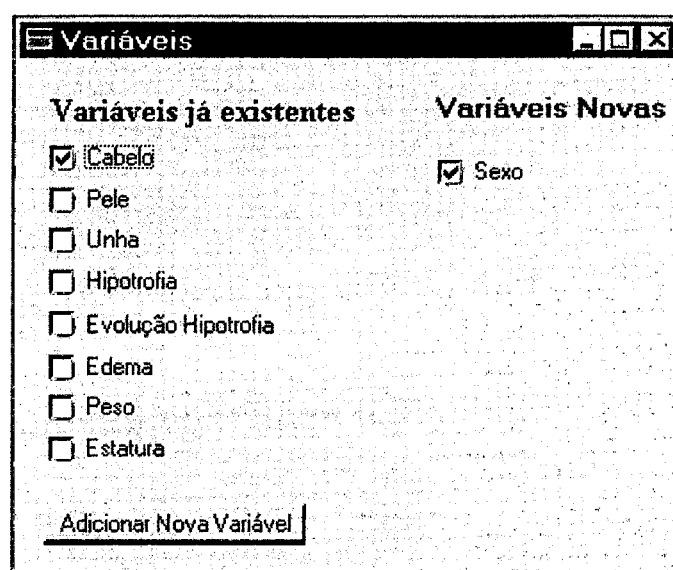


Figura 13 – Visualização da tela Variáveis

- ◆ **Tela Adicionar Nova Variável:** Ao clicar Adicionar Nova Variável, como visto na segunda tela, aparece uma terceira tela denominada *Adicionar Nova Variável*. Neste ponto o usuário optou em criar uma nova variável. Para continuar é necessário informar ao sistema o número de estados da nova variável, o nome da variável e o nome dos estados.

Adicionar Nova Variável

Nome |

Para adicionar uma variável preencha os campos abaixo.

Número de Estados

Nome

Nome dos Estados

Estado 1

Estado 2

**Figura 14 – Visualização da tela Adicionar Nova Variável – 1ª página**

O próximo passo é clicar no botão OK. Assim, o sistema abre uma nova página onde deverão ser informadas (digitados) as probabilidades condicionais da nova variável. O sistema permite alterar o número de estados da variável basta clicar na página *Nome*. Nota-se também na página *estados* que os totais das probabilidades condicionais dispostas em linhas devem ser iguais a 1. Ao terminar a digitação das probabilidades pode-se verificar os resultados clicando o botão *Verificar Total*. Esta tela pode ser visualizada na Figura 15.

Observação: As probabilidades condicionais apresentadas na Figura 15 são apenas ilustrativas.

Adicionar Nova Variável

Nome: 2 estados

Número de Estados: 2

Verificar Total

	Masculino	Feminino	Total	
Grave Aguda	0,0100	0,9900	1,0000	OK
Grave Crônica Marasmo	0,0200	0,9800	1,0000	OK
Grave Crônica Kwashiokor	0,3000	0,7000	1,0000	OK
Grave Crônica Marasmo-Kwashiokor	0,4000	0,6000	1,0000	OK
Moderada Aguda	0,0005	0,9995	1,0000	OK
Moderada Crônica	0,6000	0,4000	1,0000	OK
Leve Aguda	0,7000	0,3000	1,0000	OK
Leve Crônica	0,8000	0,2000	1,0000	OK
Sem Desnutrição	0,9000	0,1000	1,0000	OK

Teste OK

Figura 15 – Visualização da tela Adicionar Nova Variável – 2ª página

- ◆ **Tela Variável Sexo:** Ao clicar no botão OK após digitar todas as probabilidades condicionais, surge uma nova tela que caracteriza a variável criada como um objeto de forma reduzida e retangular. Pode-se movimentar com a tela sobre o sistema até encontrar uma posição mais desejável. Assim, o usuário pode selecionar um estado da variável e em seguida atualizar o sistema com um clique no botão *Atualizar* lá na tela principal. Para excluir a variável Sexo, por exemplo, basta fechar a tela do mesmo nome. A tela citada acima pode ser visualizada na Figura 16.

Sexo

Estado |

Masculino

Figura 16 – Visualização da tela Variável Sexo

- ◆ **Tela Probabilidades a priori:** Um outro recurso que o sistema oferece é a possibilidade de alteração do vetor das probabilidades a priori das hipóteses diagnósticas. Nota-se que estas probabilidades variam de uma região para outra. Por exemplo, o quadro da desnutrição das crianças de até 2 anos de idade da região Nordeste é diferente do quadro desnutrição da

região Sul. Assim, para introduzir um novo vetor de probabilidades no SEDIN basta digitar as probabilidades de cada hipótese diagnóstica nos espaços correspondentes e de preferência atribuir um nome a região para que estas informações sejam salvas, conforme Figura 17.

Hipótese Diagnóstica	Probabilidade
Grave Aguda	0,0050
Grave Crônica Marasmo	0,0050
Grave Crônica Kwashiorkor	0,0015
Grave Crônica Marasmo-Kwashiorkor	0,0050
Moderada Aguda	0,0100
Moderada Crônica	0,0135
Leve Aguda	0,0300
Leve Crônica	0,0700
Sem Desnutrição	0,8600

Base original

Total 1.0000 OK

Verificar Total

Região

Região Sul

Sair

**Figura 17 – Visualização da tela Probabilidades a Priori**

- ◆ **Ajuda do SEDIN:** A Ajuda está disponível no Menu Ajuda e permite o usuário acessar textos explicativos sobre o domínio de aplicação .

Inicialmente como proposta de pesquisa pensou-se em utilizar uma RB pré-gravada para a implementação do SEDIN assim como foi feito no SISPLAN. Mas, logo em seguida foi abandonada esta possibilidade pelo fato de que uma RB pré-gravada pelo Netica tinha um comportamento estático, a menos que se atualizasse a RB a cada mudança que ocorresse na base de conhecimento, o que no SEDIN se tornaria inviável. Conseguiu-se então, por outro lado, utilizar somente a Netica API (Application Program Interface) com sua DLL (Dynamic Link Library) que contém uma biblioteca com todas as funções definidas na linguagem C, como foi descrito anteriormente.

### **4.2.3 – Avaliação**

O SEDIN foi avaliado por um especialista da área médica que considerou a versão atual adequada para a utilização tanto para o diagnóstico de avaliação nutricional em crianças com até dois anos de idade quanto para o ensino. O SEDIN tem como referencial o conhecimento disponibilizado no sistema SISPAN. Portanto, o especialista avaliador também considerou que a rede bayesiana do sistema SISPAN tem uma base de conhecimentos válida para realizar o diagnóstico de avaliação nutricional. Dessa forma, conclui-se que o sistema SEDIN possui conhecimentos para realizar diagnóstico de avaliação nutricional pediátrica e pode ser uma ferramenta útil para o ensino de tal conhecimento médico.



## 5 – CONCLUSÃO

Neste trabalho fez-se uma homenagem especial ao Reverendo Thomas Bayes por meio de sua Biografia. O Reverendo Thomas Bayes nasceu em Londres em 1702. Foi o responsável pela formulação de um dos teoremas mais famosos da Teoria da Probabilidade, o Teorema de Bayes.

A Estatística Bayesiana surgiu por volta do ano 1763 com o primeiro trabalho a sugerir a representação da probabilidade condicional de Thomas Bayes. Em meados da década de 80, a Estatística Bayesiana na concepção de Rede bayesiana passou a ser aplicada em Sistemas Especialistas por se tratar de uma teoria consistente.

Nesta pesquisa foi desenvolvido um protótipo de Sistema Especialista Probabilístico, isto é, um sistema que explora o uso da Teoria da Probabilidade para representação da incerteza. O protótipo foi chamado de SEDIN - Sistema Especialista Probabilístico de Base de Conhecimento Dinâmica.

Sobre o SEDIN chegou-se a algumas conclusões:

- A base de conhecimento utilizada a partir da Rede Bayesiana do SISPAN é bem definida e fundamentada por ter sido extensivamente estudada. Representa para o SEDIN um agente inteligente. Outros agentes inteligentes foram considerados, como: a interface gráfica do SEDIN e o módulo de ajuda;
- O SEDIN é um Sistema Especialista Probabilístico com base de conhecimento dinâmica que possui uma rede bayesiana que pode ser facilmente modificada;
- O SEDIN pode ser considerado mais inteligente que o SISPAN porque pode continuar a aprender;

Finalmente, o SEDIN pode servir como fonte de consulta e material didático para apoiar o professor na transmissão do conhecimento de Sistemas Especialistas Probabilísticos. Do mesmo modo, especialistas em avaliação nutricional na aprendizagem e treinamento de alunos de graduação e/ ou residentes da área de saúde.

## **5.1 – TRABALHOS FUTUROS**

Com a conclusão deste trabalho, abrem-se novas possibilidades de pesquisas que podem ser feitas com a concepção de novas versões do SEDIN. Algumas das possibilidades sugeridas seguem abaixo:

- Medir a sensibilidade do sistema em relação às mudanças nos valores das probabilidades condicionais;
- Melhorar a “Ajuda” introduzindo figuras ilustrativas que permitem o usuário melhorar a aprendizagem sobre o sistema de um modo geral.
- Introduzir um módulo que seja capaz de fazer comparações, no que se refere a diagnósticos (probabilidades de saída), entre o SEDIN e o SISPAN.
- Criar um módulo “salvar” que seja capaz de salvar em um arquivo o trabalho desenvolvido pelo usuário.

## BIBLIOGRAFIA

1. AUER, K. **Agents**. 1995. [on-line] Disponível na Internet via WWW: <URL: <http://www.pcug.org.au/~kauer/project/main.htm>>, set/97.
2. BANTER, **Banter**. [on line] Documento disponível na internet via WWW: <URL: <http://www.cs.uwm.edu/~haddawy/>> , fev-98.
3. BNG, **Bayesian Network Generator**. [on line] Documento disponível na internet via WWW: <URL: <http://www.mcw.edu/midas/bng.html>> ,set-97.
4. BOL, Bayes-On-Line. [on line] Documento disponível na internet via WWW: <URL: <http://www.kbe.co.za/products/bol.htm>>, nov-97.
5. CARBONELL, J. g., Learning by analogy: Formulating and Generalizing Plans from Past Experience. In MICHALKI, R. S., CARBONELL, J. G., MITCHELL, T. M., editor, Machine Learning: Na Artificial Intelligence Approach, p. 137-161, Springer-Verlag, 1984.
6. CARBONELL, J., Machine Learning Paradigms and Methods, by Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1990.
7. DEMAZEAU, Y.; SICHMAN, J.; BOISSIER, O. “**When can Knowledgebased Systems be Called Agents?**”. IX Simpósio Brasileiro de Inteligência Artificial. pp172-185, Rio de Janeiro, 1992.

8. DIÉZ, F. J., J. Mira, E. Iturrande and S. Zubillaga. **DIAVAL, a Bauesian exert system for echocardiography**. Artificial Intelligence in Medicine, 1997. [on line] Documento disponível na internet via WWW: <URL: <http://www.dia.uned.es/tesis.html>>, set-97.
9. DOMBAL, F. T., LEAPER, D. J., STANILAND, J. R., McCANN, A. P. **Human and Computer-aided Diagnosis of Abdominal Pain: Further Report with Emphasis on the Performance of the Clinicians**. British Medical Journal, 4, 376-380, 1972.
10. DXPLAIN, **Ruth Lilly Medical Library – Test Connection to Dxplain**. [on line] Documento disponível na internet via WWW: <URL: <http://www.medlib.iupui.edu/dxplain.html>>, abr-97.
11. FLEISCHHAUER, L.I.A. **O uso da tecnologia de agentes na Programação da Produção**. Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina. Florianópolis, SC. Dez., 1996.
12. FRANKLIN,S.; GRAESSER, A. **Is it an Agent, or just a Program?: A taxonomy for Autonomous Agents**, University of Memphis, 1996. [on-line] Disponível na Internet via WWW: <URL: <http://www.mscl.memphis.edu/~franklin/AgentProg.html>>, out./97.
13. FROZZA, R.; ALVAREZ, L.O.C. **Um Ambiente para o Desenvolvimento de Sistemas Multiagentes Reativos**. ANAIS DO XXIV Seminário Integrado de Software e Hardware – SEMISH. Brasília, DF, 1997.pp. 375-386
14. GAAG, Linda C. Van Der. **Bayesian Belief Networks: Odds and Ends**. The Computer Journal, Vol. 39, N° 2, 97 – 113, 1996.
15. GRAPHICAL, **Graphical-Belief**. [on line] WWW: <URL: <http://www.bayes.stat.washington.edu/almound/gb/graphical-belief.html>> , dez-97.
16. HECKERMAN, David. **A Bayesian Approach to Learning Causal**. Technical Report MSR-TR-95-04, Microsoft Research, March, 1995. Documento disponível em: WWW: <URL: <http://www.research.microsoft.com/research/dtg/heckerman/TR-95-04.html>>
17. HUGIN, **Hugin**. [on line] Documento disponível na internet via WWW: <URL: <http://www.hugin.dk>>, set-97.

18. IDEAL, **Ideal – Influence Diagram Evaluaton and Analysis in Lisp**. [on line] Documento disponível na internet via WWW: <URL: <http://www.rpal.rockwell.com/ideal.html>>, set-97.
19. JENNINGS, N.; WOOLDRIDGE, M. **Software Agents**. IEE Review, Jan. 1996,pp 17-20
20. KIK, **About Knowledge Industries**. [on line] Documento disponível na internet via WWW: <URL: <http://www.kic.com.html>> , ago-97.
21. KOEHLER, Cristiane. **Uma Abordagem Probabilística Para Sistemas Especialistas**. Dissertação de mestrado apresentada ao Programa de pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina. Florianópolis, SC. Dez. , 1998.
22. MAES, P. **Agents that Reduce Work and Information Overload**. Communications of the ACM, Vol. 37, No. 7, Jul 1994. [On line] Disponível na Internet via WWW: <URL: <http://www.media.mit.edu/people/pattie/CACM-94/CACM-4.pl.html>, set./97.
23. MAMMO, **MammoNet – Mammography Decision Support System**. [on line] Documento disponível na internet via WWW: <URL: <http://www.mcw.edu/midas/mammo.html>>, jan-98.
24. MICHALSKI, R. S., Learning Strategies Automated Knowledge Acquisition. InLeonard Bolc, editor. **Computacional Models of Learning**. p. 1-19, Springer Verlag, Berlin, Heidelberg, 1987.
25. NASSAR, Sílvia Modesto. **Informática e Estatística: Uma Interação entre duas Ciências**. Trabalho submetido ao concurso de Professor Titular. Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis (Brasil), 1998.
26. Netica. [on line] Documento disponível na internet via WWW: <URL: <http://www.norsys.com.html>>, nov.-99.
27. O'CONNOR, D.O.; PRITKO, S.; SPAGNA, R.; WILSON, L **Intelligent Agent Strategy: White Paper**,. IBM Corporation, Researgy Triangle Park, NC. 1996.

28. PACHECO, Roberto C. S.. **Tratamento de Imprecisão em Sistemas Especialistas**. Dissertação de mestrado apresentada ao Programa de pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina. Florianópolis, SC. Fev. , 1991.
29. PATRICK, E., FATTU, J., **Artificial Intelligence with Statistical Patter Recognition**, Prentice-Hall, Inc. & Englewood Cliffs, New Jersey, 1986.
30. PEARL, J. **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference**. San Mateo, Calif,: Morgan Kaufmann, 1988.
31. PEREZ, Rocío Callupe. **Sistema Especialista de Apoio à Decisão Médica com Metodologia de Aprendizagem Simbólica**. Dissertação de mestrado apresentada ao Programa de pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina. Florianópolis, SC. Jan. , 1997.
32. SAITTA, l., Shift of Bias in Machine Learning. In Research Committee of the European Network of Excellence in Machine Learning (Mlnet), SAITTA, L., editor **State of the Art in Machine Learning**, p. 38-44, September 1995.
33. SIMON, H., Why Should Machines Learn?. In MICHALSKI, R. s., CARBONELL, J. G., MITCHELL, T. M., editor, **Machine Learning: Na Artificial Intelligence Approach**. p. 25-37, Springer-Verlag, 1984.
34. SOUZA, E.M.S. **Uma Estrutura de Agentes para Assessoria na Internet**. Dissertação de mestrado apresentada ao Programa de pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina. Florianópolis, SC. Fev. , 1997.
35. **Spirit**. [on line] Documento disponível na internet via WWW: <URL: <http://pawpaw.fernuni-hagen.de/BWLOR/forsch.html>>, nov.-99.
36. SRINIVAS, Srinivas, S. and Breese, J. **IDEAL: A software package for analys of influence diagrams**. Proceedings of the Sixth Uncertainty Conference in AI, Cambridge, MA, 1990.
37. STRATEGIT, **Strategist**. [on line] Documento disponível na internet via WWW: <URL: <http://www.prevision.com/strategist.html>>, dez-97.

38. TESSARI – Raciocínio Probabilístico em Sistemas Especialistas. Relatório de iniciação científica, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis (Brasil), 1998.
39. VAVASSORI, Fabiane Barreto. **Ferramentas e Agentes para um Ambiente de Aprendizagem na Web**. Dissertação de mestrado apresentada ao Programa de pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina. Florianópolis, SC. Abr. , 1998.
40. WOOLDRIDGE, M.; JENNINGS, N. **Intelligent Agents: Theory and Practice**. Knowledge Engineering Review v. 10 No 2, Jun. 1995. [*on-line*] Disponível na Internet via WWW: <URL: <http://www.doc.mmu.ac.uk/STAFF/mike/ker95/ker95-html.html>>, set./97.

## ANEXO 1 – Código Fonte do SEDIN

A seguir segue uma parte do Código Fonte do Sistema SEDIN.

(Parte correspondente a Base de Conhecimento do SISPAN)

```

net_bn *net;
node_bn *Desnu, *Cabelo, *Pele, *Unha, *Hipotrofia, *EvoluHipo, *Edema, *Peso,
*Estatura;

double belief;
char msg [MSG_LEN_ns];
int res;
char num[10];

//-----
env = NewNeticaEnviron_bn ("NassarS/UFSC/310-1W/23570");
res = InitNetica_bn (&env, msg);

if (res < 0) return;

net = NewNet_bn ("Desnu",env);

Desnu = NewNode_bn ("Desnu", 9, net);
Cabelo = NewNode_bn ("Cabelo",4, net);
Pele = NewNode_bn ("Pele" , 4, net);
Unha = NewNode_bn ("Unha" , 2, net);
Hipotrofia = NewNode_bn ("Hipotrofia" , 4, net);
EvoluHipo = NewNode_bn ("EvoluHipo" , 3, net);
Edema = NewNode_bn ("Edema" , 4, net);
Peso = NewNode_bn ("Peso" , 4, net);
Estatura = NewNode_bn ("Estatura" , 4, net);

CHKERR

SetNodeStateNames (Desnu,"A1","A2","A3","A4","A5","A6","A7","A8","A9");
SetNodeStateNames (Cabelo,"Sedoso","Seco","Quebradico","Descolorido");
SetNodeStateNames (Pele,"Sedosa","Seca","Descamativa","Lesoes");
SetNodeStateNames (Unha,"Normal","Quebradica");
SetNodeStateNames
(Hipotrofia,"Membros","Glutea","Facies_Senil","Sem_Hipotrofia");
SetNodeStateNames (EvoluHipo,"Ausente","Cronico","Agudo");
SetNodeStateNames
(Edema,"Mb_Inf","Mb_Inf_Sup","Sem_Edema","Generalizado");

```



```
SetNodeStateNames (Peso,"Normal","Escor_Z1","Escor_Z2","Escor_Z3");
SetNodeStateNames (Estatura,"Normal","Escor_Z1","Escor_Z2","Escor_Z3");
```

#### CHKERR

```
AddLink_bn (Desnu, Cabelo);
AddLink_bn (Desnu, Pele);
AddLink_bn (Desnu, Unha);
AddLink_bn (Desnu, Hipotrofia);
AddLink_bn (Desnu, EvoluHipo);
AddLink_bn (Desnu, Edema);
AddLink_bn (Desnu, Peso);
AddLink_bn (Desnu, Estatura);
```

#### CHKERR

```
SetNodeProbs(Desnu,0.005, 0.005, 0.0015, 0.005, 0.01, 0.0135, 0.03, 0.07, 0.86);

SetNodeProbs(Cabelo,"A1",0.950, 0.045, 0.004, 0.001);
SetNodeProbs(Cabelo,"A2",0.150, 0.830, 0.015, 0.005);
SetNodeProbs(Cabelo,"A3",0.002, 0.700, 0.200, 0.098);
SetNodeProbs(Cabelo,"A4",0.002, 0.780, 0.200, 0.018);
SetNodeProbs(Cabelo,"A5",0.970, 0.025, 0.004, 0.001);
SetNodeProbs(Cabelo,"A6",0.500, 0.480, 0.015, 0.005);
SetNodeProbs(Cabelo,"A7",0.980, 0.015, 0.004, 0.001);
SetNodeProbs(Cabelo,"A8",0.975, 0.020, 0.004, 0.001);
SetNodeProbs(Cabelo,"A9",0.990, 0.008, 0.001, 0.001);
```

#### CHKERR

```
SetNodeProbs(Pele,"A1",0.950, 0.048, 0.001, 0.001);
SetNodeProbs(Pele,"A2",0.050, 0.920, 0.020, 0.010);
SetNodeProbs(Pele,"A3",0.010, 0.390, 0.350, 0.250);
SetNodeProbs(Pele,"A4",0.020, 0.730, 0.150, 0.100);
SetNodeProbs(Pele,"A5",0.900, 0.070, 0.025, 0.005);
SetNodeProbs(Pele,"A6",0.810, 0.155, 0.030, 0.005);
SetNodeProbs(Pele,"A7",0.980, 0.015, 0.001, 0.004);
SetNodeProbs(Pele,"A8",0.980, 0.015, 0.001, 0.004);
SetNodeProbs(Pele,"A9",0.990, 0.008, 0.001, 0.001);
```

#### CHKERR

```
SetNodeProbs(Unha,"A1",0.995,0.005);
SetNodeProbs(Unha,"A2",0.40,0.60);
SetNodeProbs(Unha,"A3",0.15,0.85);
SetNodeProbs(Unha,"A4",0.25,0.75);
SetNodeProbs(Unha,"A5",0.995,0.005);
SetNodeProbs(Unha,"A6",0.60,0.40);
```

```

SetNodeProbs(Unha,"A7",0.995,0.005);
SetNodeProbs(Unha,"A8",0.70,0.30);
SetNodeProbs(Unha,"A9",0.995,0.005);
CHKERR

```

```

SetNodeProbs(Hipotrofia,"A1",0.87,0.10,0.01,0.02);
SetNodeProbs(Hipotrofia,"A2",0.879,0.10,0.02,0.001);
SetNodeProbs(Hipotrofia,"A3",0.60,0.03,0.00,0.37);

```

```

SetNodeProbs(Hipotrofia,"A4",0.88,0.10,0.01,0.01);
SetNodeProbs(Hipotrofia,"A5",0.80,0.05,0.00,0.15);
SetNodeProbs(Hipotrofia,"A6",0.75,0.05,0.00,0.20);
SetNodeProbs(Hipotrofia,"A7",0.40,0.01,0.00,0.59);
SetNodeProbs(Hipotrofia,"A8",0.35,0.01,0.00,0.64);
SetNodeProbs(Hipotrofia,"A9",0.02,0.005,0.00,0.975);
CHKERR

```

```

SetNodeProbs(EvoluHipo,"A1",0.00,0.01,0.99);
SetNodeProbs(EvoluHipo,"A2",0.00,0.99,0.01);
SetNodeProbs(EvoluHipo,"A3",0.00,0.99,0.01);
SetNodeProbs(EvoluHipo,"A4",0.00,0.99,0.01);

```

```

SetNodeProbs(EvoluHipo,"A5",0.00,0.01,0.99);
SetNodeProbs(EvoluHipo,"A6",0.00,0.99,0.01);
SetNodeProbs(EvoluHipo,"A7",0.00,0.01,0.99);
SetNodeProbs(EvoluHipo,"A8",0.00,0.99,0.01);
SetNodeProbs(EvoluHipo,"A9",0.99,0.005,0.005);
CHKERR

```

```

SetNodeProbs(Edema,"A1",0.010,0.002,0.987,0.001);
SetNodeProbs(Edema,"A2",0.010,0.002,0.987,0.001);
SetNodeProbs(Edema,"A3",0.900,0.070,0.000,0.030);
SetNodeProbs(Edema,"A4",0.900,0.098,0.001,0.001);
SetNodeProbs(Edema,"A5",0.040,0.009,0.950,0.001);

```

```

SetNodeProbs(Edema,"A6",0.040,0.009,0.950,0.001);
SetNodeProbs(Edema,"A7",0.010,0.009,0.980,0.001);
SetNodeProbs(Edema,"A8",0.010,0.009,0.980,0.001);
SetNodeProbs(Edema,"A9",0.003,0.001,0.995,0.001);
CHKERR

```

```

SetNodeProbs(Peso,"A1",0.002,0.048,0.700,0.250);
SetNodeProbs(Peso,"A2",0.001,0.009,0.390,0.600);
SetNodeProbs(Peso,"A3",0.020,0.130,0.650,0.200);
SetNodeProbs(Peso,"A4",0.005,0.015,0.630,0.350);
SetNodeProbs(Peso,"A5",0.005,0.075,0.919,0.001);
SetNodeProbs(Peso,"A6",0.010,0.200,0.789,0.001);

```

```

SetNodeProbs(Peso,"A7",0.899,0.100,0.001,0.000);

```

```
SetNodeProbs(Peso,"A8",0.930,0.069,0.001,0.000);
SetNodeProbs(Peso,"A9",0.960,0.039,0.001,0.000);
CHKERR
```

```
SetNodeProbs(Estatura,"A1",0.990,0.008,0.001,0.001);
SetNodeProbs(Estatura,"A2",0.000,0.005,0.295,0.700);
SetNodeProbs(Estatura,"A3",0.000,0.005,0.295,0.700);
SetNodeProbs(Estatura,"A4",0.000,0.005,0.295,0.700);
SetNodeProbs(Estatura,"A5",0.990,0.008,0.001,0.001);
SetNodeProbs(Estatura,"A6",0.025,0.680,0.290,0.005);
```

```
SetNodeProbs(Estatura,"A7",0.990,0.008,0.001,0.001);
SetNodeProbs(Estatura,"A8",0.600,0.380,0.015,0.005);
SetNodeProbs(Estatura,"A9",0.990,0.006,0.003,0.001);
CHKERR
```

```
CompileNet_bn (net);
CHKERR
```

```
//-----
```

```
if (ComboBox1->ItemIndex==0) //Cabelo
{
  EnterFinding("Cabelo","Sedoso",net);
}
```

```
if (ComboBox1->ItemIndex==1) //Cabelo
{
  EnterFinding("Cabelo","Seco",net);
}
```

```
if (ComboBox1->ItemIndex==2) //Cabelo
{
  EnterFinding("Cabelo","Quebradico",net);
}
```

```
if (ComboBox1->ItemIndex==3) //Cabelo
{
  EnterFinding("Cabelo","Descolorido",net);
}
```

```
if (ComboBox2->ItemIndex==0) //Pele
{
  EnterFinding("Pele","Sedosa",net);
}
```

```
if (ComboBox2->ItemIndex==1) //Pele
{
```

```
EnterFinding("Pele","Seca",net);
}

if (ComboBox2->ItemIndex==2) //Pele
{
EnterFinding("Pele","Descamativa",net);
}

if (ComboBox2->ItemIndex==3) //Pele
{
EnterFinding("Pele","Lesoes",net);
}

if (ComboBox4->ItemIndex==0) //Unha
{
EnterFinding("Unha","Normal",net);
}

if (ComboBox4->ItemIndex==1) //Unha
{
EnterFinding("Unha","Quebradica",net);
}

if (ComboBox5->ItemIndex==0) //Hipotrofia
{
EnterFinding("Hipotrofia","Membros",net);
}

if (ComboBox5->ItemIndex==1) //Hipotrofia
{

EnterFinding("Hipotrofia","Glutea",net);
}

if (ComboBox5->ItemIndex==2) //Hipotrofia
{
EnterFinding("Hipotrofia","Facies_Senil",net);
}

if (ComboBox5->ItemIndex==3) //Hipotrofia
{
EnterFinding("Hipotrofia","Sem_Hipotrofia",net);
}

if (ComboBox6->ItemIndex==0) //Evolucao_Hipotrofia
{
EnterFinding("EvoluHipo","Ausente",net);
}
```

```
if (ComboBox6->ItemIndex==1) //Evolucao_Hipotrofia
{
    EnterFinding("EvoluHipo","Cronico",net);
}

if (ComboBox6->ItemIndex==2) //Evolucao_Hipotrofia
{
    EnterFinding("EvoluHipo","Agudo",net);
}

if (ComboBox7->ItemIndex==0) //Edema
{
    EnterFinding("Edema","Mb_Inf",net);
}

if (ComboBox7->ItemIndex==1) //Edema
{
    EnterFinding("Edema","Mb_Inf_Sup",net);
}

if (ComboBox7->ItemIndex==2) //Edema
{
    EnterFinding("Edema","Sem_Edema",net);
}

if (ComboBox7->ItemIndex==3) //Edema
{
    EnterFinding("Edema","Generalizado",net);
}

if (ComboBox8->ItemIndex==0) //Peso
{
    EnterFinding("Peso","Normal",net);
}

if (ComboBox8->ItemIndex==1) //Peso
{
    EnterFinding("Peso","Escor_Z1",net);
}

if (ComboBox8->ItemIndex==2) //Peso
{
    EnterFinding("Peso","Escor_Z2",net);
}

if (ComboBox8->ItemIndex==3) //Peso
```

```

    {
    EnterFinding("Peso","Escor_Z3",net);
    }

if (ComboBox9->ItemIndex==0) //Estatura

    {
    EnterFinding("Estatura","Normal",net);
    }

if (ComboBox9->ItemIndex==1) //Estatura
    {
    EnterFinding("Estatura","Escor_Z1",net);
    }

if (ComboBox9->ItemIndex==2) //Estatura
    {
    EnterFinding("Estatura","Escor_Z2",net);
    }

if (ComboBox9->ItemIndex==3) //Estatura
    {
    EnterFinding("Estatura","Escor_Z3",net);
    }

//wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww

belief = GetNodeBelief("Desnu", "A1", net);   CHKERR

sprintf(num,"%9.4f",belief);
if (Form3->ComboBox3->ItemIndex==0) {sprintf(num,"%9.1f",belief);}
if (Form3->ComboBox3->ItemIndex==1) {sprintf(num,"%9.2f",belief);}
if (Form3->ComboBox3->ItemIndex==2) {sprintf(num,"%9.4f",belief);}
if (Form3->ComboBox3->ItemIndex==3) {sprintf(num,"%9.6f",belief);}
if (Form3->ComboBox3->ItemIndex==4) {sprintf(num,"%9.1f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==5) {sprintf(num,"%9.2f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==6) {sprintf(num,"%9.4f%",belief*100);}

if (Form3->ComboBox3->ItemIndex==7) {sprintf(num,"%9.6f%",belief*100);}
Edit1->Text=num; ProgressBar1->Position=(belief*100);

belief = GetNodeBelief("Desnu", "A2", net);   CHKERR
sprintf(num,"%9.4f",belief);
if (Form3->ComboBox3->ItemIndex==0) {sprintf(num,"%9.1f",belief);}
if (Form3->ComboBox3->ItemIndex==1) {sprintf(num,"%9.2f",belief);}
if (Form3->ComboBox3->ItemIndex==2) {sprintf(num,"%9.4f",belief);}
if (Form3->ComboBox3->ItemIndex==3) {sprintf(num,"%9.6f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==4) {sprintf(num,"%9.1f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==5) {sprintf(num,"%9.2f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==6) {sprintf(num,"%9.4f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==7) {sprintf(num,"%9.6f%",belief*100);}
Edit2->Text=num; ProgressBar2->Position=(belief*100);

```

```

belief = GetNodeBelief("Desnu", "A3" , net);  CHKERR
sprintf(num,"%9.4f",belief);
if (Form3->ComboBox3->ItemIndex==0) {sprintf(num,"%9.1f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==1) {sprintf(num,"%9.2f",belief);}
if (Form3->ComboBox3->ItemIndex==2) {sprintf(num,"%9.4f",belief);}
if (Form3->ComboBox3->ItemIndex==3) {sprintf(num,"%9.6f",belief);}
if (Form3->ComboBox3->ItemIndex==4) {sprintf(num,"%9.1f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==5) {sprintf(num,"%9.2f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==6) {sprintf(num,"%9.4f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==7) {sprintf(num,"%9.6f%",belief*100);}

```

```

Edit3->Text=num; ProgressBar3->Position=(belief*100);

```

```

belief = GetNodeBelief("Desnu", "A4" , net);  CHKERR
sprintf(num,"%9.4f",belief);
if (Form3->ComboBox3->ItemIndex==0) {sprintf(num,"%9.1f",belief);}
if (Form3->ComboBox3->ItemIndex==1) {sprintf(num,"%9.2f",belief);}
if (Form3->ComboBox3->ItemIndex==2) {sprintf(num,"%9.4f",belief);}
if (Form3->ComboBox3->ItemIndex==3) {sprintf(num,"%9.6f",belief);}
if (Form3->ComboBox3->ItemIndex==4) {sprintf(num,"%9.1f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==5) {sprintf(num,"%9.2f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==6) {sprintf(num,"%9.4f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==7) {sprintf(num,"%9.6f%",belief*100);}
Edit4->Text=num; ProgressBar4->Position=(belief*100);

```

```

belief = GetNodeBelief("Desnu", "A5" , net);  CHKERR
sprintf(num,"%9.4f",belief);
if (Form3->ComboBox3->ItemIndex==0) {sprintf(num,"%9.1f",belief);}
if (Form3->ComboBox3->ItemIndex==1) {sprintf(num,"%9.2f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==2) {sprintf(num,"%9.4f",belief);}
if (Form3->ComboBox3->ItemIndex==3) {sprintf(num,"%9.6f",belief);}
if (Form3->ComboBox3->ItemIndex==4) {sprintf(num,"%9.1f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==5) {sprintf(num,"%9.2f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==6) {sprintf(num,"%9.4f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==7) {sprintf(num,"%9.6f%",belief*100);}
Edit5->Text=num; ProgressBar5->Position=(belief*100);

```

```

belief = GetNodeBelief("Desnu", "A6" , net);  CHKERR
sprintf(num,"%9.4f",belief);
if (Form3->ComboBox3->ItemIndex==0) {sprintf(num,"%9.1f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==1) {sprintf(num,"%9.2f",belief);}
if (Form3->ComboBox3->ItemIndex==2) {sprintf(num,"%9.4f",belief);}
if (Form3->ComboBox3->ItemIndex==3) {sprintf(num,"%9.6f",belief);}
if (Form3->ComboBox3->ItemIndex==4) {sprintf(num,"%9.1f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==5) {sprintf(num,"%9.2f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==6) {sprintf(num,"%9.4f%",belief*100);}
if (Form3->ComboBox3->ItemIndex==7) {sprintf(num,"%9.6f%",belief*100);}
Edit6->Text=num; ProgressBar6->Position=(belief*100);

```

```

belief = GetNodeBelief("Desnu", "A7", net);   CHKERR

```

```

sprintf(num,"%9.4f",belief);

```

```

if (Form3->ComboBox3->ItemIndex==0) {sprintf(num,"%9.1f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==1) {sprintf(num,"%9.2f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==2) {sprintf(num,"%9.4f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==3) {sprintf(num,"%9.6f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==4) {sprintf(num,"%9.1f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==5) {sprintf(num,"%9.2f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==6) {sprintf(num,"%9.4f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==7) {sprintf(num,"%9.6f%",belief*100);}

```

```

Edit7->Text=num; ProgressBar7->Position=(belief*100);

```

```

belief = GetNodeBelief("Desnu", "A8", net);   CHKERR

```

```

sprintf(num,"%9.4f",belief);

```

```

if (Form3->ComboBox3->ItemIndex==0) {sprintf(num,"%9.1f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==1) {sprintf(num,"%9.2f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==2) {sprintf(num,"%9.4f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==3) {sprintf(num,"%9.6f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==4) {sprintf(num,"%9.1f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==5) {sprintf(num,"%9.2f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==6) {sprintf(num,"%9.4f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==7) {sprintf(num,"%9.6f%",belief*100);}

```

```

Edit8->Text=num; ProgressBar8->Position=(belief*100);

```

```

belief = GetNodeBelief("Desnu", "A9", net);   CHKERR

```

```

sprintf(num,"%9.4f",belief);

```

```

if (Form3->ComboBox3->ItemIndex==0) {sprintf(num,"%9.1f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==1) {sprintf(num,"%9.2f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==2) {sprintf(num,"%9.4f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==3) {sprintf(num,"%9.6f",belief);}

```

```

if (Form3->ComboBox3->ItemIndex==4) {sprintf(num,"%9.1f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==5) {sprintf(num,"%9.2f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==6) {sprintf(num,"%9.4f%",belief*100);}

```

```

if (Form3->ComboBox3->ItemIndex==7) {sprintf(num,"%9.6f%",belief*100);}

```

```

Edit9->Text=num; ProgressBar9->Position=(belief*100);

```



```
//www
```

```
CompileNet_bn(net);  
CHKERR
```

```
end:
```

```
FreeNet_bn (net);
```

```
res = CloseNetica_bn (env, mesg);  
return;
```

```
error:
```

```
fprintf (stderr, "Desnu: Error in %s\n",  
         ErrorMessage_ns (GetError_ns (env, ERROR_ERR, NULL)));  
goto end;
```

## **ANEXO 2 – Manual do Usuário**

Este Manual do Usuário é parte integrante do software **SEDIN 2000**.

### **MANUAL DO USUÁRIO SEDIN 2000**

#### **Dois Sistemas na Tela Principal do SEDIN 2000**

Ao inicializar o programa **SEDIN 2000**, o usuário poderá visualizar a Tela Principal onde aparecem as evidências da desnutrição infantil do **SISPAN**, à esquerda. E um quadro comparativo com o Diagnóstico da Desnutrição do **SISPAN** e do **SEDIN**, à direita.

Nota-se que no início os campos de probabilidades estão vazios.

#### **O Botão Atualizar da Tela Principal**

Para se visualizar as probabilidades dos diagnósticos é necessário clicar sobre o botão *Atualizar*. Se nenhum estado for previamente ativado nas evidências do **SISPAN**, ao clicar no botão *Atualizar* será exibido o vetor de probabilidades a priori do **SISPAN**.

Do mesmo modo, ao clicar no botão *Atualizar* do **SEDIN** pode-se notar o mesmo vetor de probabilidades a priori. Essas probabilidades a priori fazem parte do sistema, por isso são exibidas somente ao se clicar no botão *Atualizar*, no início .

#### **Alterar Probabilidades a Priori das Hipóteses Diagnósticas no SEDIN**

No caso do **SEDIN**, a base de probabilidades a priori das hipóteses diagnósticas pode ser facilmente modificada. Para fazer essas alterações o usuário deverá selecionar o item *Sedin* no menu Alterar/Probabilidades a Priori. Ao selecionar esse item, surgirá uma tela em que o usuário poderá fazer as alterações nas probabilidades a priori do sistema **SEDIN** e dar um nome a essa nova situação. Ainda poderá salvar essas probabilidades em uma tabela para utilizar futuramente.

Observação: As probabilidades ao serem inseridas no campo de entrada deverão ser digitadas na **forma decimal** e com a utilização da **vírgula**. O botão *Verificar Total*, situado abaixo dos campos de entrada, ao ser clicado, oferece o total das probabilidades introduzidas (digitadas). Observa-se que esse total deverá ser igual a 1. Caso contrário, o sistema acusará erro.

### **Realizar um Diagnóstico da Desnutrição com o SISPAN**

Para alterar o vetor de probabilidades do diagnóstico da desnutrição do **SISPAN**, o usuário deverá selecionar os estados das evidências do **SISPAN**. Essas evidências se encontram no lado esquerdo da tela principal em forma de combo box. Estão disponíveis 8 (oito) evidências e para se obter um diagnóstico é necessário selecionar pelo menos uma evidência (um combo box). Em seguida, basta clicar no botão *Atualizar* do **SISPAN** para que as probabilidades sejam calculadas.

### **Realizar um Diagnóstico da Desnutrição com o SEDIN**

Para alterar o vetor de probabilidades do diagnóstico da desnutrição do **SEDIN** o usuário deverá criar as variáveis para disponibilizar o conhecimento. Essas variáveis podem ser as mesmas do **SISPAN** e/ ou outras que o usuário deseja explorar. Veja mais informações no tópico: A Tela: Variáveis do **SEDIN**.

### **A Tela: Variáveis do SEDIN**

Para que o usuário possa acrescentar variáveis no sistema **SEDIN** ele deverá clicar no botão *Variáveis*. Esse botão abre uma nova tela chamada Variáveis. Veja os tópicos: Inserir/Excluir Variáveis no **SEDIN**.

### **Inserir Variáveis no SEDIN**

À esquerda estão relacionadas as variáveis já existentes que são as mesmas do **SISPAN**. O usuário poderá inseri-las no **SEDIN** clicando diretamente em cima dos nomes das variáveis. Conforme o usuário vai clicando sobre os nomes as variáveis serão criadas e introduzidas sobre a

tela principal em forma de pequenas telas retangulares que serão empilhadas umas sobre as outras. Para visualizar todas as variáveis abertas no sistema basta arrastar as pequenas telas colocando-as uma próxima da outra de modo que se consiga visualizá-las.

### **Excluir Variáveis no SEDIN**

O usuário pode, também, excluir variáveis abertas bastando clicar novamente em cima dos nomes das variáveis desmarcando-as. Ou clicar no x que aparece no canto superior - direito da pequena tela que representa a variável.

### **Botão Adicionar Nova Variável**

Na tela: Variáveis, têm um botão chamado *Adicionar Nova Variável* que o usuário poderá usar para criar novas variáveis e introduzi-las no SEDIN. Ao clicar no *botão Adicionar Nova Variável* aparece uma nova tela com o mesmo nome do botão.

### **A Tela Adicionar Nova Variável do SEDIN**

Para inserir uma nova variável no SEDIN, o usuário deverá selecionar, inicialmente, o número de estados que deseja criar, que varia entre 2 a 5 estados. Ao selecionar o número de estados da variável que deseja criar, aparecerão os campos para serem preenchidos. Um para o nome da variável e os outros para o nome dos estados da variável. Após ter preenchido os campos é só clicar no botão OK. Uma nova página será acrescentada nesta tela com o nome de *Estados* onde o usuário deverá digitar as probabilidades condicionais. Após serem digitadas as probabilidades condicionais no formato de **número decimal** com **vírgula**, deverá ser verificado que, o total das probabilidades em linha deverá ser igual a 1. Caso contrário, o sistema emitirá uma mensagem de erro. Em seguida é só clicar em *OK* e a variável estará criada. Surgirá, então, uma pequena tela com o nome da variável criada. Essa tela poderá ser arrastada sobre a tela principal de modo que ocupe uma posição adequada. Como foi descrito anteriormente, o usuário poderá, também excluir a variável criada bastando clicar no x que aparece no canto superior – direito da pequena tela nomeada que representa a variável.

### **Atualizar o Sistema SEDIN Para Obter Novo Conhecimento**

Sempre que o usuário quiser avaliar o novo conhecimento no **SEDIN** deverá clicar no botão *Atualizar* do **SEDIN** que está na tela principal.

### **Formatar os Números das Probabilidades dos Diagnósticos da Desnutrição**

No menu *Formatar/Números* o usuário poderá formatar os números das probabilidades dos diagnósticos da desnutrição dos dois sistemas: o **SEDIN** e o **SISPAN**.

Os formatos disponíveis são:

Exibir probabilidades: com 1 casa decimal: 0,0

Exibir probabilidades: com 2 casas decimais: 0,00

Exibir probabilidades: com 4 casas decimais: 0,0000

Exibir probabilidades: com 6 casas decimais: 0,000000

Exibir probabilidades: em porcentagem com 1 casa decimal: 0,0%

Exibir probabilidades: em porcentagem com 2 casas decimais: 0,00%

Exibir probabilidades: em porcentagem com 4 casas decimais: 0,0000%

Exibir probabilidades: em porcentagem com 6 casas decimais: 0,000000%

---

---

**UFSC – UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**CENTRO TECNOLÓGICO**  
**CPGCC – Curso de Pós-Graduação em Ciência da Computação**

**Projeto: SEDIN 2000**

Copyright 2000

Sistema Especialista Probabilístico de Base de Conhecimento Dinâmica

**Autor:** Carlos Efrain Stein

**e-mail:** [efrain@furb.rct-sc.br](mailto:efrain@furb.rct-sc.br)

**Telefone:** 0\*\*47 342-1366

Florianópolis, agosto, 2000

**Orientadores do projeto:**

Professora Dr. Sílvia Modesto Nassar

**e-mail:** [silvia@inf.ufsc.br](mailto:silvia@inf.ufsc.br)

Professora Dr. Maria Marlene de Souza Pires

**e-mail:** [marlene@ccs.ufsc.br](mailto:marlene@ccs.ufsc.br)

---

---

## ANEXO 3 – Instalação

O texto que segue abaixo constitui o arquivo **Leia-me** que pode ser encontrado do quarto disco de instalação do **SEDIN 2000**.

### COMO INSTALAR O SEDIN 2000

(WINDOWS 95/98/NT/2000)

Para instalar o **SEDIN 2000** insira o disco 1 no drive A e clique no botão **Iniciar/Executar** do Windows. Selecione o arquivo Setup.exe que está no disco 1. A seguir clique no botão **Abrir**. Para dar início a instalação do **SEDIN 2000** clique no botão **OK**. Assim o programa de instalação irá instalar o **SEDIN 2000** no seu computador.

Nota: Você também poderá selecionar o arquivo setup.exe através do Windows Explorer.

Ao surgir a tela WELCOME (Bem Vindo) clique em Next (avançar).

Na próxima tela, User Information (Informação do Usuário), digite o seu nome e local de trabalho. Não esqueça de clicar em next para avançar!

Na tela Choose Destination Location (Escolher Local de Destinação), o **SEDIN 2000** deverá ser instalado no diretório **c:\Netica\Sedin**. Atenção! Se outro diretório for selecionado, o programa apresentará problemas ao ser executado.

Na tela Select Program Folder (Selecionar Pasta de Programas) você poderá manter a pasta Sedin2000 por default. Também poderá selecionar uma pasta já existente ou criar uma nova pasta. Clique em Next para que o instalador prossiga com a instalação dos arquivos requeridos pelo sistema.

Nesta fase da instalação, o instalador irá pedir os disquetes numerados de 2 a 4. Pois o programa de instalação do **SEDIN 2000** é composto por 4 disquetes de 1.44MB.

Para posteriormente carregar o programa **SEDIN 2000** clique no botão **Iniciar** do Windows, selecione o item **Programas** e procure pela pasta **Sedin2000** (ou pasta com outro nome atribuído durante a instalação) e selecione o arquivo **SEDIN**.