


**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA**

Um Modelo de Integração do Conhecimento Aplicado à Diagnose

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA
TESE DEFENDIDA POR ITAMAR SHINDI SUNIURA
E APROVADA PELA
COMISSÃO JULGADORA EM 22 / 03 / 96.


ORIENTADOR

Aluno: **Itamar Shindi Suniura**
Orientador: **Oswaldo Luis Agostinho**

03/96

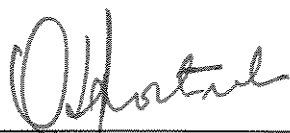


**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA**

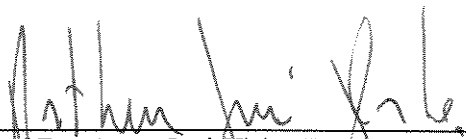
Um Modelo de Integração do Conhecimento Aplicado à Diagnose

Dissertação de Mestrado

**Aluno: Itamar Shindi Suniura
Orientador: Oswaldo Luis Agostinho**



Prof.Dr. Oswaldo Luis Agostinho
Universidade Estadual de Campinas



Prof.Dr. Artur José Vieira Porto
Universidade Estadual de São Paulo - São Carlos



Prof.Dr. Paulo Corrêa Lima
Universidade Estadual de Campinas

Campinas, 22 de Março de 1996

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA**

Um Modelo de Integração do Conhecimento Aplicado à Diagnose

Aluno: **Itamar Shindi Suniura**
Orientador: **Oswaldo Luis Agostinho**

Curso: Engenharia Mecânica
Área de Concentração: Fabricação

Dissertação de Mestrado apresentado à comissão de Pós-Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Mestre em Engenharia Mecânica.

Campinas, 1996
S.P. - Brasil

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Su73m Suniura, Itamar Shindi
Um modelo de integração do conhecimento aplicado à
diagnose / Itamar Shindi Suniura.--Campinas, SP: [s.n.],
1995.

Orientador: Oswaldo Luis Agostinho.
Dissertação (mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Mecânica.

1. Inteligência artificial. 2. Sistemas especialistas
(Computação). 3. Máquinas - Manutenção. I. Agostinho,
Oswaldo Luis. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Mecânica. III. Título.

Dedicatória

Dedico esse trabalho a meus pais - Sussumo e Mitico Suniura

Agradecimentos

Agradeço a minha esposa Lúcia pelo grande companheirismo, compreensão e suporte.

Agradeço ao meu orientador Prof.Oswaldo Luiz Agostinho pelo incentivo, acompanhamento e por acreditar no trabalho aqui desenvolvido.

Agradeço ao amigo José Roberto Favilla, pela amizade, pelo intercâmbio de conhecimento técnico durante a elaboração dessa tese. E também pelas oportunidades que me abriu no desenvolvimento desse trabalho.

Agradeço aos amigos Joca, Cleber, Marcelo Carlos e Marilene pelas reflexões sempre estimuladas.

Agradeço a IBM Brasil pelo patrocínio e apoio, particularmente ao gerente Ivan Moroni pela confiança sempre depositada.

Agradeço a UNICAMP pela oportunidade e auxílio às pesquisas, sem os quais não seria possível a concretização desse trabalho.

Agradeço a Deus, que tornou tudo possível.

Índice

1.0 Introdução	1
1.1 Cenário	1
1.2 Objetivo desse Trabalho	7
2.0 Inteligência Artificial, um Estudo	9
2.1 Inteligência Artificial	10
2.1.1 Histórico	11
2.2 Sistemas Especialistas	23
2.2.1 Introdução	23
2.2.2 Definição	27
2.2.3 Classificação	29
2.2.4 Estrutura	30
2.2.5 Construção	31
2.2.6 Aquisição do Conhecimento	33
2.2.7 Representação do Conhecimento	36
2.2.8 Avaliação dos Sistemas Especialistas	48
2.2.9 Áreas de Aplicação	52
2.3 Sistemas Especialistas Voltados à Diagnose	56
2.3.1 Introdução	56
2.3.2 Diagnóstico	56
2.3.3 Abordagens dos Sistemas Especialistas para Tarefas de Diagnose	57
2.4 Resumo	63
3.0 Um modelo de Integração do Conhecimento Aplicado à Diagnose	64
3.1 Modelo proposto	65
3.2 Proposta da Forma de Implementação do Modelo Proposto	68
3.2.1 Formas de Implementação	68
3.2.2 Forma de Implementação Proposta	71
3.3 Representação do Conhecimento	72
3.4 Resumo	73
4.0 Aplicação e Avaliação do Modelo Proposto	74
4.1 Introdução	75
4.1.1 O Armazenamento Magnético	75
4.1.2 Particularidades do Produto	75
4.2 Caracterização do Ambiente de Aplicação do Modelo	76
4.2.1 Introdução	76
4.2.2 Características do ambiente e do processo de fabricação	76
4.2.3 Características Gerais do Equipamento e do Processo	80
4.3 Aplicação do Modelo Proposto	82
4.3.1 Aquisição do Conhecimento	82
4.3.2 Definição da Estratégia de Diagnóstico	84
4.3.3 Seleção da Ferramenta de Implementação	90
4.3.4 Implementação do Sistema de Diagnóstico	93
4.4 Avaliação do Sistema Implementado	98
4.4.1 Verificação	98
4.4.2 Validação	99
4.4.3 Aceitação do Usuário	100
4.5 Resultados da Implementação do Sistema	101
4.6 Resumo	103

5.0 Conclusões	104
5.1 Sugestões para Próximos Trabalhos	108
6.0 Bibliografia Utilizada	109
6.1 Referências Bibliográficas	109
6.2 Bibliografia Adicional Consultada	114
Apêndice A. Teste de casos	117
Apêndice B. Estruturas Funcionais do ESE	127
B.1 Objetos da base de conhecimento	127
B.1.1 Parâmetros	127
B.1.2 Regras	129
B.1.3 FCB	131
B.1.4 Grupos	134
B.1.5 Telas	134
B.2 Métodos de inferência	135
B.2.1 Encadeamento para trás	135
B.2.2 Encadeamento para frente	135

Figuras

1.	O sistema de manufatura integrado por computador - CIM	5
2.	Exemplos apresentados ao programa ARCS	18
3.	Um exemplo da analogia de Evans	20
4.	Arquitetura típica de um Sistema Especialista	31
5.	Representação de uma rede semântica	45
6.	Exemplo de uma rede semântica	46
7.	Paradigma de Davis e Hamscher.	57
8.	Estrutura de inferência da classificação heurística	61
9.	Arquitetura funcional proposta para o sistema especialista.	66
10.	Representação esquemática de um HDA	77
11.	Esquema do processo de montagem do HDA-3380X	79
12.	Esquema da máquina de limpeza KLN	81
13.	Processo de diagnose humano.	83
14.	Organização do conhecimento experimental	85
15.	Organização do conhecimento fundamental.	86
16.	Integração do conhecimento orientado ao aspecto funcional.	87
17.	Integração do conhecimento conforme o tipo de conhecimento.	88
18.	Combinação das alternativas (1) e (2).	89
19.	Implementação do modelo da base de conhecimento integrada.	91
20.	Relação entre o ESE e o usuário	92
21.	Estrutura dos FCBs do presente trabalho.	94
22.	Tempo médio para reparo.	102
23.	Exemplo de estrutura hierárquica FCB.	132

Tabelas

1.	Importantes fases da Inteligência Artificial.	15
2.	Uma comparação entre as formas de processamento.	24
3.	Programas convencionais versus Sistemas Especialistas	24
4.	Tabela verdade	37
5.	Tabela comparativa	76
6.	Teste de casos	100

Glossário

C

CAD. Projeto Assistido por Computador. “Computer Aided in Design”

CAPP. Planejamento de Processos Assistidos por Computador. “Computer Aided Process Planning”

CAPSC. Programação e Controle da Produção Assistidos por Computador. “Computer Aided Process ”

CICS. Sistema de comunicação de propósito geral que pode suportar uma rede de computadores. “Customer Information Control System”

CIM. Manufatura Integrada por Computador. “Computer Integrated Manufacturing”

CMS. “Conversational Monitor System”. É um sistema operacional iterativo, consersacional, projetado para operar sob o VM/XA em maquina virtual.

D

DASD. Dispositivo de armazenamento de acesso direto, ou unidade de disco magnético. “Direct Access Storage Device”

E

ESE. Sistema para desenvolvimento de sistemas especialistas. Ambiente de Sistemas Especialistas. “Expert System Environment”

ESCE. Ambiente de consulta do ESE. “Expert System Consultation Environment”

F

FCB. Unidade de trabalho do ESE. Bloco de controle. “Focus Control Block”

G

GPS. Programa com propósito de ser utilizado na resolução de problemas de qualquer natureza. “General Problem Solver”

H

HDA. Montagem do conjunto disco/cabeça. Unidade de armazenamento magnético. “Head Disk Assembly”

I

IA. Inteligência Artificial

IBM. Industry Business Machine Corporation

IMS. Sistema de gerenciamento de informação. "Information Management System"

K

KLN. KLN Gmbh - Empresa alemã fabricante de máquinas e equipamentos.

M

MTBF. Tempo médio entre falhas. "Mean Time Between Failure"

MTTR. Tempo médio de reparo. "Mean Time To Repair"

MVS. Sistema operacional para computadores de grande porte. Múltiplo armazenamento virtual. "Multiple Virtual Storage"

P

PLC. Controlador lógico programável. "Programable Logic Controller"

T

TSO. Operação de compartilhamento de tempo. "Time Sharing Operation"

V

VCM. Conjunto atuador da unidade de disco magnético. Subconjunto do HDA. Conjunto bobina motor "Voice Coil Motor"

VM. Sistema operacional para computadores de grande porte. Máquina virtual. "Virtual Machine"

VM/XA. "Virtual Machine / Extended Architecture". É um sistema de controle que gerencia uma arquitetura estendida (XA) e seus sucessores (IBM4341,4381,etc), permitindo múltiplos usuários tenham uma simulação funcional de uma máquina real (máquina virtual).

Resumo

Esse trabalho relaciona-se a utilização de Sistemas Especialistas, uma área da Inteligência Artificial, como uma das tecnologias contribuidoras no solucionamento de problemas relativos ao uso da experiência e do conhecimento humano. Particularmente, tenta responder as necessidades da manutenção de máquinas e equipamentos, onde no processo de diagnose:

- existe uma grande quantidade de informações para uma só pessoa manipular,
- as soluções incluem muitas alternativas para uma só pessoa considerar, ou ainda
- rápidas decisões são necessárias de múltiplas fontes.

Nesse trabalho é proposto a utilização de uma estratégia de diagnóstico integrada, baseada em classificação heurística e em modelo causal, através da integração dos conhecimentos experimental e fundamental, tendo em vista a manutenção de máquinas e equipamentos voltados à produção.

O modelo foi aplicado e avaliado em uma situação real de fábrica, através do desenvolvimento de um Sistema Especialista para auxiliar o processo de manutenção de um equipamento de limpeza e descontaminação de partes metálicas. A implementação utilizou uma ferramenta do tipo casca com representação do conhecimento através de regras de produção e um método de inferência para trás. Esse processo compreendeu as etapas de aquisição do conhecimento, definição da estratégia de diagnóstico, seleção da ferramenta, codificação do conhecimento e testes do sistema. Sendo que avaliação do sistema deu-se a nível de verificação, validação e aceitação do usuário.

O resultado obtido da aplicação do modelo foi a redução do tempo médio de reparo do equipamento, da ordem de 50%, obtidos através da diminuição do tempo gasto no processo de identificação da causa dos problemas - o diagnóstico do problema.

Com base no resultado obtido, conclui-se que a utilização de um sistema automático de diagnóstico voltado a manutenção que utilize uma abordagem integrada baseada em modelo causal e em classificação heurística é válido e eficaz. E pode ser considerado um avanço no processo de diagnóstico automático, particularmente para aqueles aplicados a classe de máquinas e equipamentos de natureza mais complexa.

Abstract

This work is related to the Expert System application in the Manufacturing Process. The Expert System is an Artificial Intelligence field that is trying to bring the computer systems to solve problems that requires the human experience and knowledge. One identified area is the diagnose process, where usually there is a large quantity of knowledge to be handled by only one person, or when many alternatives must be considered, or when many alternatives must be taken from multiple sources, during a diagnostic process.

The proposal of this work is the integration of the diagnose strategy based in heuristic classification and based in model, obtained by the experimental and fundamental knowledge integration, to help production machines and equipments maintenance process.

The proposed model was applied and tested in a real manufacturing environment, where a diagnostic Expert System was developed and tested, to help the maintenance activities of a cleaning parts machine. The developed system was accomplished in a shell tool, with production rules for knowledge representation and a back-tracking inference process. The overall process phases were: knowledge acquisition, diagnose strategy definition, shell tool selection, knowledge coding, and tests. The developed Expert System was evaluated in terms of verification, validation and user acceptance or human factors.

The application result was that the mean time to repair was reduced in 50%, due to the reduction of the time spent in the problem identification task, or the diagnostic process itself.

Based on the obtained result, the conclusion of this work, is that an automatic diagnostic system for maintenance purpose based in a integrated strategy - heuristic classification and model, is valid and effective. And it can be considered an advance in the automatic diagnostic systems, particularly for those systems that deal with a more complex equipment's class.

1.0 Introdução

Esse capítulo apresenta um cenário que é o contexto do presente trabalho e a identificação do problema que levou a motivação desse estudo.

1.1 Cenário

Entender a história e a evolução do processo de produção industrial é uma abordagem essencial para uma compreensão mais ampla desse trabalho e do seu contexto.

Raramente na história da industrialização cenários tão diversificados tem sido exercitados para o desenvolvimento do futuro das fábricas. Em nenhuma outra era industrial as idéias tem sido levadas tão a frente e encorajadas como em debates correntes sobre o futuro da fábrica. E raramente uma discussão sobre tecnologia e o desenvolvimento do trabalho no processo produtivo assumiu os caracteres de uma cruzada (Broedner,1990).

Mas afinal porque é tão importante os fabricantes em todo o mundo buscarem um modo mais eficiente de produção ? A resposta é que a adoção de uma produção mais eficiente, na medida em que inevitavelmente se expanda, resultará em mudanças globais em quase todas as indústrias, no destino das companhias, nas alternativas para os consumidores, na natureza do trabalho, e - em última instância - no destino das nações (Womack,et al.,1992). Para viver bem, uma nação deve produzir bem (Dertouzos,et al.,1990).

Olhando para o sistema produtivo, inicialmente tem-se a produção artesanal, onde o produtor artesanal lança mão de trabalhadores altamente qualificados e ferramentas simples mas flexíveis, para produzir exatamente o que o consumidor deseja: um item de cada vez. Móveis sob encomenda, trabalhos de arte decorativa e alguns poucos e exóticos carros esportivos constituem exemplos atuais. Todos nós adoramos a idéia da produção artesanal, mas seu problema é óbvio: bens produzidos pelo método artesanal custam caro demais para a maioria de nós, dado que é executada por uma força de trabalho altamente qualificada onde muitos trabalhadores progridem através de um aprendizado abrangendo todo um conjunto de habilidades artesanais. Quando a produção artesanal deparou com problemas que não conseguiu superar, como por exemplo devido ao alto custo de seu produto, a produção em massa foi desenvolvida no século XX como alternativa (Womack,et al.,1992).

O processo de produção em massa utiliza profissionais excessivamente especializados para projetar produtos manufaturados por trabalhadores semi ou não-qualificados, utilizando máquinas dispendiosas e especializadas em uma única tarefa, que processam produtos padronizados em altíssimos volumes (Womack,et al.,1992). Quando essas máquinas entraram no mundo de produção capitalista, elas precisavam ser aperfeiçoadas para satisfazer seus dois propósitos principais. Um foi o de aumentar o valor excedente, e o outro para garantir a autonomia do processo produtivo forçando as leis do comércio à persistência no trabalho humano. Sendo que o progressivo aperfeiçoamento das máquinas visava atingir seus principais propósitos que eram de: aumentar a produção, incrementar a precisão e assim facilitar a produção de partes intercambiáveis e, finalmente, possibilitar que as operações fossem processadas mais independentemente. E esse processo de introdução e aperfeiçoamento das máquinas separa o trabalho manual do trabalho intelectual, e separa seu planejamento de sua implementação; sendo essa uma das características fundamentais desse novo sistema incorporado dos princípios de Taylor e Babbage (Broedner,1990).

Essa indústria manufatureira vai desenvolver-se a partir dos primeiros decênios desse século sobre um paradigma central que pode ser resumido em poucas palavras: a engenharia produtiva taylorista e fordista constrói a eficácia de suas linhas e partir de uma organização em postos de trabalho parcelados e encadeados, tanto na fabricação quanto na montagem. Do ponto de vista analítico, esse paradigma baseia se no parcelamento, especialização e intensificação do trabalho. E é fato que esse paradigma foi historicamente

eficaz. A partir dele funda-se a produção em grande série de mercadorias padronizadas e de baixo custo, na qual o sistema moderno de produção em massa encontra um de seus fundamentos essenciais (Drucker,1994). Esses princípios centrais de concepção e organização das fábricas expostos inicialmente por Taylor e depois por Ford estão resumidos e codificados no que se chama a Organização Científica do Trabalho (Corbert,et al.,1991). As fábricas desse período caracterizam-se principalmente por (Agostinho,1989):

- produzir uma pequena variação de produtos, que também era aliada a pequena exigência do consumidor.
- alta divisão do trabalho, através de rotas de fabricação com grande número de operações individuais, tais como torneamento, fresamento, retificação, etc.; que resultam em uma complexidade continuamente crescente da programação e do controle da manufatura.
- a responsabilidade do controle de produção é assumida pelo encarregado/mestre/supervisor, com informações de todos os detalhes da operação de sua área em particular. Devido a essa característica, os resultados da produção e quantidade de produção são inteiramente dependentes do desempenho dos operadores das máquinas e dos departamentos de suporte.

E por ser essa maquinaria tão cara e pouco versátil, o produtor em massa adiciona várias folgas - suprimimentos adicionais, trabalhadores e espaço extra para assegurar a continuidade da produção. Por ser a mudança para um novo produto tão dispendiosa, o produtor em massa mantém o modelo padrão em produção o maior tempo possível. O resultado: o consumidor obtém preços mais baixos, mas a custa de variedade, e com métodos de trabalho que muitos trabalhadores julgam monótonos e sem sentido (Womack,et al.,1992).

A partir de meados dos anos 60, esse paradigma entra em crise relativa, ou, passa a ser objeto de um reexame de conjunto (Coriat,1988). Notadamente também devido a pressão de lutas operárias que contestavam de maneira cada vez mais aberta e eficaz, o tipo de organização da produção que resultava da aplicação do princípio dos processos taylorista e fordista de trabalho basearem-se em grandes massas de trabalhadores não qualificados submetidos a ritmos de trabalho, por vezes intensos (Hobsbawn,1994). A esse aspecto, somou-se outro fato que devido ao grau de sofisticação alcançado, a linha taylorista ou fordista, tornou-se em parte “contraproducente” na medida em que uma grande quantidade de tempos “mortos” e de tempos improdutivos eram gastos com técnicas complexas de balanceamento das cadeias de produção (Coriat,1988).

E enquanto os mercados eram globalmente regidos pela demanda até os anos 60 - significando para as empresas que a saída dos produtos estava garantida com a demanda superior a oferta, nos anos 70 a situação inverte-se e os mercados tornam-se globalmente regidos pela oferta: as capacidades instaladas são superiores a demanda (Garegnani,et al.,1980).

A crise do petróleo na primeira metade dos anos 70 pode ser considerada como um marco histórico que se caracteriza como um questionamento da sociedade industrial, cujos sintomas gerais associados a esse período, entre outros são (Carvalho,1988):- custos altos de produção, forte competição no mercado mundial, fortalecimento das associações de trabalhadores, instabilidade mundial, surgimento de novos produtores no terceiro mundo, etc. Sendo que dois fenômenos de importância consideráveis podem ser identificados (Coriat,1988):

- o modo central de organização do trabalho, fundado no taylorismo entra em crise,
- a crise econômica se instala e se afirma como permanente.

Em termos práticos, isso significou uma mudança nos entraves que pesavam sobre o sistema produtivo. Ao objetivo de quantidades e volumes ao menor custo possível, conjuga-se o objetivo da qualidade, no sentido de que ganhar mercado supõe a capacidade de adaptar-se rapidamente a um tipo particular de produto, obedecendo as suas normas e especificações. Em síntese, isso significa que satisfazer a demanda supõe, atualmente, menos a geração de séries muito grandes de produtos padronizados, e mais a obtenção de uma capacidade de produzir, em séries mais restritas, bens diversificados dirigidos para atender demandas

particulares, elas próprias variando em quantidade e qualidade. Pode-se acrescentar que a demanda torna-se diferenciada devido ao perfil mais complexo dos assalariados, a multiplicação das categorias e dos novos segmentos no mercado de trabalho. Para um mesmo produto, tornam-se diferentes as características particulares do consumo esperado ou realizado, em cada categoria do consumidor. Tem-se um exemplo dessa situação na indústria automobilística, onde para cada modelo base existem inúmeras variações segundo as combinações possíveis de opções, modelos, especificações ou normas para exportação (Drucker,1994).

Essas novas características da demanda - instável, volátil e diferenciada, relacionam-se com as novas modalidades de competição e são expressas tanto ao nível dos custos quanto dos produtos. Torna-se necessário fazer as linhas de produção corresponderem a nova demanda volátil e diferenciada, assegurando que a oferta adquira essa mesma capacidade de se adaptar e de se diferenciar, exigidas pelo caráter essencialmente aleatório do mercado (Carvalho,1988). Em resumo, com a emergência de novas normas de consumo, encontra-se em jogo novas normas de concorrência. É esta mutação notável que gera a necessidade de linhas "flexíveis" de produção. Flexíveis porque são capazes de fabricar, sem exigir uma reorganização maior, diferentes produtos, a partir da mesma organização básica de equipamentos e com um intervalo reduzido de adaptação. Portanto essa dupla característica do período - limite da eficácia do taylorismo e mutação das normas de concorrência provocada pela crise econômica - influencia e determina profunda e persistentemente as direções impressas às inovações tecnológicas, ainda que as tecnologias de informação, por si só, conheçam progressos expressivos visando a sua utilização produtiva nas fábricas. E a segunda metade dos anos 70 marca um processo claramente irreversível na indústria, relativo ao deslançar de novas formas de automação (Coriat,1988).

Faz-se importante caracterizar nesse momento a diferença entre a mecanização e a automação. A mecanização representa a substituição direta do homem pela máquina, onde geralmente o processo não é afetado e os benefícios não são determinados pela redução do custo do trabalho, ou o custo dado dos ambientes de alto risco. Nessas aplicações, as máquinas são incapazes ou fracamente capazes de se adaptar as alterações. A automação inclui as mesmas máquinas encontradas na mecanização, porém o controle das máquinas é capaz de se adaptar a conhecidas e limitadas mudanças no ambiente que é feito pelo uso de controles realimentados. Sendo que dessa distinção entre a automação e a mecanização torna claro porque a mecanização não atingiu os níveis de tratamento da automação (Lenz,1988).

A novidade, da automação dos anos 70, do ponto de vista tecnológico consiste em que os novos meios de trabalho apresentam a particularidade de serem programáveis, vale dizer, serem capazes de incluir instruções para séries alternativas e diferentes de operações. Os controles informatizados dos equipamentos operam segundo as informações que lhe são fornecidas pelos sensores, sendo capazes de ativar automaticamente o programa de operação correspondente a peça a ser produzida (Toffler,1980). A presença de máquinas-ferramentas de comando numérico, ao lado de robôs e computadores, controlando a produção modifica em profundidade o processo de fabricação. A fábrica muda e a automação microeletrônica possibilita grandes incrementos do trabalho e a qualidade do produto é melhorada pelas características do equipamento automatizado que uniformiza a produção, eliminando perdas e refugos (Feldmann,1988).

Cabe aqui situar a complexidade e a abrangência do desenvolvimento da tecnologia microeletrônica e da informática, como uma verdadeira revolução tecnológica, a qual está levando a redefinição da base técnica de produção do capitalismo: "A segunda metade do século XX tem sido palco de profundas transformações na maneira pela qual os agentes econômicos interagem - quer na esfera da produção, quer na esfera da circulação de mercadorias - provocadas pela mudança na base técnica do sistema. Acelerados desenvolvimentos no campo da (micro) eletrônica e sua subsequente incorporação ao aparato produtivo tem propiciado: o surgimento e o crescimento de um complexo microeletrônico, a destruição, a recriação e a criação de novas atividades produtivas, setores e ramos industriais; a conseqüente modificação dos padrões de acumulação e das formas de trabalho e de concorrência."(Tauile,1984). A expressão Revolução da Microeletrônica, que serve de título a um livro organizado por Tom Forester (Forester,T.(org.), The Microelectronic Revolution, Cambridge, The MIT Press,1981) tem sido empregada no sentido de demonstrar que a enorme evolução tecnológica ocorrida na produção dos circuitos integrados, por permitir a massificação do uso computadores e equipamentos assemelhados, está por promover uma larga reestruturação na sociedade,

e daí o termo revolução, qual seja, a passagem da sociedade industrial para a sociedade do conhecimento (Drucker,1994).

A indústria microeletrônica efetivamente iniciou-se após a segunda guerra mundial, pois até então era desprezível toda a produção de semicondutores, limitando-se a produção de cristais de galena, utilizados nos primórdios da indústria do rádio. A atividade econômica está cada vez mais associada a criação, processamento e transmissão de informação, indo de encontro ao que muitos já denominam de sociedade da informação. A base para essa evolução é a indústria eletrônica, a qual está cada vez mais baseada em componentes derivados da microeletrônica (Toffler,1980).

O campo de aplicação da microeletrônica não se restringiu a novos produtos, mas se estendeu aos próprios processos produtivos, nas mais diversas áreas. No caso da indústria, a tecnologia proporcionou uma nova onda de automação, considerada por muitos autores como a mudança tecnológica mais importante e revolucionária desde a Revolução Industrial. Na automação eletromecânica, as instruções de comando estão incorporados nos próprios componentes mecânicos e elétricos que compõe a máquina. O curso das ferramentas está definido mecanicamente. Sua rigidez significa que qualquer alteração importante no comando (no tipo ou sequência de operações realizadas) implicaria um amplo rearranjo dos seus componentes, isto é, implicaria a construção de uma outra máquina eventualmente. Tal rigidez explica os limites a automação eletromecânica nas indústrias de forma, que são muito mais de natureza econômica do que técnica: ela só é viável economicamente quando aplicada em operações que implicam um alto grau de repetibilidade (Carvalho,1988).

Historicamente, portanto, esse tipo de automação teve de se restringir a produção em grande série, por exemplo, nas indústrias automobilística e têxtil. Ocorre, no entanto, que uma parte substancial das indústrias de forma operam a base da produção de pequenos e médios lotes de peças, ou mesmo de produtos industrializados. A grande revolução da tecnologia microeletrônica à automação industrial consistiu em *aliar a flexibilidade a automação* (Tauile,1984; Coriat,1985). Isso se deu através do acomplamento, às máquinas, de microprocessadores eletrônicos (controladores programáveis), que detem as informações necessárias ao seu comando. O comando das máquinas automatizadas com base na microeletrônica não se encontra “desenhado” no corpo mecânico das máquinas, e sim “editado” ou “impresso” nos programas (da palavra inglesa “software”) que alimenta os microprocessadores a ela acomplados. Esse tipo de comando é passível de reprogramação, isto é, os controladores programáveis podem receber novos programas com as instruções devidamente modificadas e necessárias para a variação dos produtos e serem fabricados, sem qualquer alteração mecânica do equipamento (Carvalho,1988).

Uma nova onda de produtividade torna-se viável com uma nova tecnologia, ou seja, a integração de sistemas individuais existentes (Agostinho,1989). Mas porque focar na integração da manufatura e não em outros parâmetros como produtividade e qualidade ? Isso é respondido quando se declara que o objetivo geral da manufatura é produzir um produto na forma do custo mais eficiente, e isto pode ser interpretado como - comprar um montante mínimo de capacidade (capital investido) e utilizá-lo na sua forma mais eficiente. Em adição, isso requer obter capacidade de produção ajustada a intercorrências por falha em equipamentos, parada do operador, falta de material, etc. E igualmente importante a identificação dos efeitos da capacidade perdida é a transferência desse efeito a outras estações. Os efeitos da integração fornecem um meio de se medir os ajustes da capacidade produtiva para toda a instalação fabril (Lenz,1988).

O modelo de integração da manufatura fornece um meio pelo qual a relação de ganhos e perdas de uma estratégia de manufatura pode ser avaliada. Isso inclui o relacionamento entre o nível de inventário, equilíbrio do balanceamento, a disponibilidade da estação e a flexibilidade. E esse modelo de integração da manufatura implica em controle computadorizado e em novas tecnologias que estão evoluindo para ajudar a esse processo (Wright & Bourne,1988).

A esse processo de integração da manufatura através do uso do computador denomina-se: Manufatura Integrada por Computador - CIM (do termo inglês “Computer Integrated Manufacturing”), sendo que através dela a automação flexível da manufatura pode ser obtida pela aplicação dos sistemas computacionais na automação do sistema de manufatura por completo - num sistema que se inicia com o modelamento

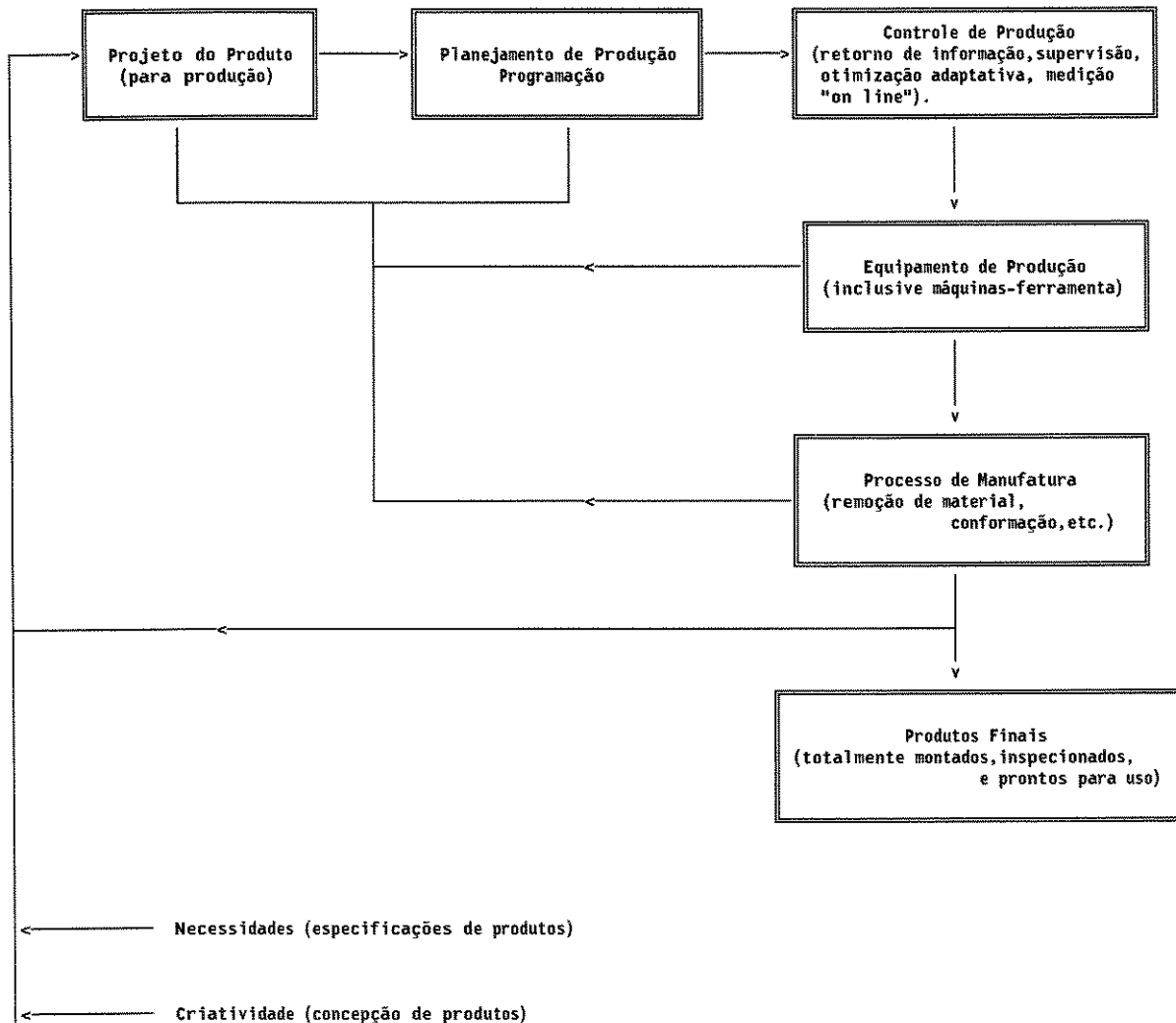


Figura 1. O sistema de manufatura integrado por computador - CIM. (Agostinho,1989)

geométrico e projeto do produto e envolve todos os estágios da produção. Uma conceituação genérica de manufatura integrada por computador, de acordo com o Dr.Eugene Merchante do "International Institution for Production Engineering Research - CIRP" é expressa na Figura 1.

Sendo que algumas denominações de sistemas computacionais já se tornam-se familiares como (Agostinho,1989):

- Projeto Assistido por Computador - CAD (da sentença inglesa "Computer Aided in Design"),
- Planejamento de Processos Assistidos por Computador - CAPP (da sentença inglesa "Computer Aided in Process Planning"),
- Programação e Controle da Produção Assistidos por Computador - CAPSC (da sentença inglesa "Computer Aided in Production Scheduling on Control"),
- Automação de Fábrica - (da sentença inglesa "Shop Floor Automation"),
- Integração dos Sistemas Individuais - (da palavra inglesa "Integration").

Outras tecnologias estão-se revelando gradativamente para auxiliar esse processo da integração da manufatura. Uma dessas novas técnicas é a Inteligência Artificial que vem ajudar a solucionar os problemas decorrentes dos complexos problemas industriais, onde as técnicas da computação ditas convencionais encontram suas limitações, dentro do ambiente de fabricação, particularmente na solução de problemas de otimização, e de problemas relativos ao uso da experiência e conhecimento humano (Kusiak,1988). Podendo citar-se:

- na solução de problemas relativos a otimização: projeto automático de processos de fabricação, especificação de produto e de escalonamento da produção (Tsutsui,1993)
- e relativos ao conhecimento humano: especificamente o processo de controle adaptativo, onde pode-se prover capacidade para o controle responder a situações desconhecidas durante a concepção da automação, através de um processo onde o raciocínio do especialista é incorporado ao programa do computador (Lenz,1988). E vem também ajudar a resolver o problema de manutenção das unidades de fabricação de natureza mais complexa e seus custos, associadas a nova dinâmica de mercado. Para usuários de maquinários de alta tecnologia, a eficiência dos serviços de manutenção tem-se tornado um crescente fator importância, e para os fabricantes a facilidade da manutenção dos equipamentos que comercializam, um elemento, que tem sido tomado em conta na batalha por segmentos de mercado (Bouche,et al.,1990).

Na diagnose ou no auxílio a diagnose de problema desses equipamentos, os Sistemas Especialistas ou Sistemas Baseados em Conhecimento são áreas da Inteligência Artificial que incorporam a abordagem baseada no conhecimento e experiência humana. Tal abordagem vem obtendo significativo sucesso de implementação junto aos meios de produção industrial dada as suas características que bem se adequam a esse meio, onde a solução de problemas requer uma quantidade muito grande de informações para uma só pessoa manipular, ou quando as soluções incluem muitas alternativas para uma só pessoa considerar, ou quando rápidas decisões são necessárias de múltiplas fontes (Kusiak,1988).

A utilização da Inteligência Artificial no auxílio a manutenção de máquinas de natureza mais complexa através do uso de um Sistema Especialista, é o contexto do presente trabalho; que através da abordagem proposta vem tentar facilitar o desenvolvimento desses sistemas pelo uso de ferramentas do tipo casca (da palavra inglesa "shell") e vem tentar tornar mais completo o processo de diagnose através da integração da base de conhecimento experimental e funcional. Dado que a maior parte dos atuais sistemas especialistas voltados a diagnose utilizam somente o conhecimento experimental (Ferreira,1991) (Fiol,1993). E em última instância esse trabalho vem contribuir ao modelo de integração da manufatura através do uso do computador - CIM, em busca de respostas a uma fábrica mais eficiente na natureza mais dinâmica do mercado.

1.2 Objetivo desse Trabalho

O rápido avanço tecnológico responsável pelo projeto e construção de dispositivos e sistemas cada dia mais complexos, e portanto mais difíceis de serem diagnosticados, aliados ao grande desenvolvimento e disponibilidade dos recursos computacionais e a escassez de especialistas humanos em diagnose, tem demandado o desenvolvimento de Sistemas Especialistas visando o diagnóstico automatizado (Clegg,1988).

Sistemas Baseados em Conhecimento ou Sistemas Especialistas são um ramo da Inteligência Artificial que se utilizam de programas de computadores (da palavra inglesa "software"), para resolver e/ou auxiliar na resolução de problemas que normalmente solicitariam a presença e inteligência humana. Esses programas podem elaborar perguntas de forma a obter dados que permitam gerar conclusões e recomendações de modo a imitar o processo do raciocínio humano (Rich,1988).

E dentro do contexto dos Sistemas Especialistas voltados à diagnose, constata-se os seguintes *fatos*:

1. os sistemas de diagnose baseados em conhecimento implementados através da abordagem única de estratégias de diagnóstico baseado em classificação heurística, é na maioria das vezes uma abordagem necessária porém não suficiente, particularmente para máquinas e equipamentos de maior complexidade tecnológica (Germond,1992),
2. a implementação de estratégias de diagnóstico baseada em modelos permite obter informações mais completas do que as informações encontradas nas estratégias de classificação; porém requer formação computacional ou experiência computacional, por parte do implementador (Goldstein,1994),
3. a maioria dos sistemas especialistas correntemente utilizados empregam a estratégia da abordagem de classificação heurística (Ferreira,1991) (Fiol,1993). Recentes estudos especialmente relativos a diagnose e reparos vem enfatizando a utilização de estratégias baseadas em modelo (Yoon & Hammer,1988) (Germond,1992). Entretanto, muito pouco tem sido feito para combinar ambos os modelos (Fink & Lush,1987), (D'Ambrosio,1990), (Bowen,1994).
4. simplicidade e modularidade na implementação de sistemas com representação do conhecimento através de sistemas de regras de produção (Chandrasekaran,1993),
5. disponibilidade no mercado de programas de computadores do tipo casca (da palavra inglesa "shell") para a construção de Sistemas Especialistas (Germond & Niebur,1993)

Com base nesse contexto define-se como *objetivo* desse trabalho:

- *Propor a utilização de uma estratégia de diagnóstico integrada, baseada em classificação heurística e em modelo causal, através da integração dos conhecimentos experimental e fundamental, tendo em vista o auxílio a manutenção de máquinas e equipamentos voltados à produção.*

A *contribuição esperada*, está na proposta de um processo automático de diagnóstico mais eficiente, visando auxiliar o processo de manutenção de máquinas e de equipamentos voltados a produção. E ainda, confirmar o potencial das ferramentas do tipo casca no desenvolvimento de Sistemas Especialistas voltados a diagnose, e o potencial das regras de produção como forma de representação do conhecimento.

Este trabalho foi dividido em seis capítulos que são:

- Capítulo 1 - Introdução
- Capítulo 2 - Inteligência Artificial - Um estudo
- Capítulo 3 - Um modelo de Integração do Conhecimento Aplicado a Diagnose
- Capítulo 4 - Aplicação e Avaliação do Modelo Proposto
- Capítulo 5 - Resultados e Conclusões
- Capítulo 6 - Bibliografia

2.0 Inteligência Artificial, um Estudo

O Capítulo 1 desse trabalho apresentou uma pequena parte da história e das idéias do processo produtivo industrial e a maneira como a indústria vem reagindo e se adequando as mudanças do mercado e do cenário mundial.

Essa reação advém da identificação do esgotamento do modelo de produção em massa frente as necessidades de um mercado com novas características - instável, volátil e diferenciado num universo mais competitivo tanto ao nível dos custos quanto dos produtos. E também advém da busca de uma fabricação mais eficiente i.e. com um mínimo montante de capacidade investido deve-se ser capaz de utilizá-lo da forma mais eficiente possível.

Sendo que o processo de busca da fabricação mais eficiente, contou com um novo e poderoso aliado - a (micro) eletrônica, que vem determinar uma nova forma de automação às fábricas - a automação programável, e vem contribuir com os sistemas de informação - os computadores. Enquanto a automação programável vem permitir uma produção mais flexível, os computadores que inicialmente limitavam-se a aplicação para processamento administrativo, vem participar de forma decisiva no modelo de integração da manufatura. Desde a concepção até o produto totalmente acabado e pronto para uso, com a utilização de sistemas como CAD, CAPP e outros.

Devido as limitações de eficiência e de custo do processamento computacional frente a dois problemas específicos da fabricação:- uso da experiência humana e resolução de problemas de otimização. Uma outra forma de participação do computador nesse processo de integração, vem sendo a utilização da Inteligência Artificial que entre outras técnicas vem empregando os Sistemas Baseados em Conhecimento ou Sistemas Especialistas.

O presente capítulo faz um estudo da Inteligência Artificial através de uma análise histórica, seus trabalhos expoentes e áreas de aplicação. Particularmente apresenta os Sistemas Especialistas nos processos automatizados de diagnose. Sendo que o capítulo 2, para um entendimento mais claro, apresenta-se em dividido em três partes como a seguir:

1. Inteligência Artificial :- apresenta-se os conceitos, um histórico, e um levantamento dos trabalhos expoentes.
2. Sistemas Especialistas :- estuda-se em detalhes os Sistemas Especialistas e suas principais aplicações.
3. Sistemas Especialistas voltados à Diagnose :- uma das aplicações especializadas dos Sistemas Especialistas é a diag e nesse contexto são discutidas as particularidades dos sistemas de diagnose.

2.1 Inteligência Artificial

A Inteligência Artificial é uma Área de estudo da Ciência da Computação, relativa ao projeto e a programação de máquinas que visam realizar tarefas, em que as pessoas utilizariam sua inteligência para realizá-las. E também dedica-se ao entendimento de como pensa o ser humano, através do estudo do comportamento e programas que modelam as hipóteses correntes e as conjecturas de alguns dos aspectos do processo cognitivo humano (Schutzer,1987).

Ou ainda, em (Kirsh,1991) a Inteligência Artificial é um ramo da Ciência da Computação cujo objetivo é o de desenvolver sistemas computacionais que exibam características que nós associamos com inteligência no comportamento humano, tais como: a linguagem, aprendizagem, raciocínio, visão e capacidade de resolver problemas. E cujos objetivos fundamentais podem ser considerados como:

- desenvolver novas aplicações para os computadores.
- entender princípios que possibilitem os atos inteligentes.

Nosso entendimento sobre o que constitui a inteligência é vago, e definições sobre a inteligência humana são imprecisas. O mais aceito padrão de medição da inteligência humana é o teste de QI (Quociente Intelectual), mas a validade desse teste como um medidor verdadeiro da inteligência intrínseca de uma pessoa (potencial ou atingida) é ainda contestada por muitos. Consequentemente, não seria tão surpreendente verificarmos que a definição de inteligência de máquina (artificial) é igualmente vaga (Boose,1985).

Muitas tentativas tem sido feitas para se definir e demonstrar mais precisamente o entendimento sobre a Inteligência Artificial. Porém a Inteligência Artificial relaciona-se menos à questões filosóficas sobre a constituição da inteligência e mais a questões mais estreitas sobre quais teorias, conceitos, ferramentas, técnicas e modelos produzidos no desenvolvimento dessa área podem ser aplicados para a construção de melhores sistemas baseados em computadores (Schank,1987).

Acima de tudo, a Inteligência Artificial como um modo particular de se solucionar problemas, difere-se bastante da abordagem tradicional orientada a algoritmos. Muito da diferença traçada, é dado a natureza dos problemas escolhidos pelos pesquisadores da Inteligência Artificial. A pesquisa em Inteligência Artificial tem-se concentrado na resolução de problemas que são demonstradamente solucionáveis por seres humanos em que não existem atualmente metodologias bem formuladas e computacionalmente tangíveis. Essa classe de problemas inclui (Rich,1988):

- como aprendemos,
- como jogamos jogos (xadrez, quebra-cabeça, enigma e outros)
- como nós nos comunicamos,
- como percebemos (ver,ouvir,falar,escutar,escrever),
- como criamos.

Dessa forma, a Inteligência Artificial tem se preocupado com atividades mentais humanas que estão entre aquelas poucas entendidas (Deutsch-McLeish,1993).

As abordagens adotadas para solucionar tais problemas podem ser melhor caracterizadas como informais, simbólicas, e conceituais, do que como quantitativas (Chandrasekaran,1993).

Dado que a Inteligência Artificial tem se desenvolvido principalmente na comunidade científica e porque as metodologias necessárias para solucionar problemas de interesse desse grupo são fracamente entendidas, a Inteligência Artificial tem sido, em sua maioria, uma ciência de tentativas e erros (Gardner,1985). Consequentemente as técnicas desenvolvidas construíram programas que se sujeitam a alterações e evoluções. Os métodos empregados tendem a ser não procedurais, flexíveis e adaptativos em natureza; sendo

que conhecimento adicional pode ser acrescentado, apagado ou modificados facilmente sem nenhuma modificação no sistema projetado (Newell & Simon,1988).

Muitos pesquisadores atraídos pela Inteligência Artificial são psicólogos cognitivos pela prática ou aqueles interessados no entedimento de como o ser humano pensa. Eles observam e estudam pessoas em várias situações de solucionamento de problemas e tentam formular teorias e hipóteses em um nível macro (não ao nível de conexões neuronais) para explicar o comportamento observado. E desenvolvem modelos computacionais que testam essas hipóteses na tentativa de se duplicar os comportamentos esperados. Tais modelos e teorias tem fortemente influenciado a área da Inteligência Artificial, inspirando muitos conceitos e técnicas de Inteligência Artificial (Kattan,1994).

Em suma, mesmo que a Inteligência Artificial tenha muito em comum com outras disciplinas de Ciência da Computação, ela se difere das áreas mais convencionais da Ciência da Computação nos seguintes aspectos (Thakore,1993):

- ponto de vista (raciocínio lógico e plausível, ao invés de cálculo quantitativo),
- assunto relacionado (atividade mental - muito intenso em conhecimento),
- tolerância para erros e dados imprecisos,
- manipulação simbólica (invés de orientação numérica),
- princípios de projeto evolucionário (não procedural, antecipação a alterações e adições),
- projeto baseado em conhecimento,
- capacidades de inferência e dedução (tem uma linha de raciocínio e pode se auto explicar),
- abordagem heurística ou aproximada a solução de problemas.

Sem dúvida, alguns desses atributos individualmente podem ser encontrados individualmente em mais de uma das disciplinas da Ciência da Computação convencional. Porém todos juntos, esses atributos caracterizam uma forma própria de abordagem, muito relacionada ao estilo humano de abordar o solucionamento de problemas. Diferenciando-se substancialmente da abordagem central dos sistemas computacionais convencionais (Bradac,1994).

2.1.1 Histórico

A Inteligência Artificial - IA, pode ter uma curta história, mas o sonho de coisas artificiais já leva há um passado mais distante, que talvez iniciaram-se com a criação dos deuses gregos - aqueles seres com um único significado de existência como: o amor, a guerra, a fertilidade, a beleza e outros. A aspiração humana não tem se resumido em controlar o ambiente, mas expande-se no desejo de criar seres artificiais que podem ser expressos pelo desejo de máquinas que pensam e na engenharia genética (McCorduck,1979). Em grandes módulos apresenta-se a seguir, alguns dos mais importantes momentos e idéias que compuseram o esteio inicial da Inteligência Artificial.

Mais recentemente no século XIX - Inglaterra, depara-se com alguns pesquisadores que perseguiram a mecanização do pensamento com uma forma mais próxima da forma atual. Um desses elementos foi Charles Babage que devotou muitos anos no projeto de realizar uma calculadora automática que fosse capaz de executar complexos cálculos necessários à navegação e balística. Enquanto Babage dedicava-se as suas aspirações, outro matemático britânico, George Boole com base nos trabalhos de Kurt Goedel envolveu-se na representação das leis básicas do pensamento e de como encontrá-los em princípios lógicos. De forma a eliminar as ambiguidades da linguagem natural, Boole utilizou-se de conjuntos de símbolos arbitrários. Esses elementos simbólicos poderiam ser combinados ou dissociados através de operações como adição, subtração ou multiplicação bem como novas formas de expressão ou concepções envolvendo esses mesmos elementos. Mais importante que tudo, Boole observou que sua lógica era um sistema de dois valores ou um

sistema verdadeiro-falso, onde qualquer expressão lógica, qual fosse sua complexidade, poderia ser expresso por 1 - um (sendo "tudo" ou "verdade") ou 0 - zero (sendo "nada" ou "falso") (Barr & Feigenbaum,1981).

A importância do trabalho de Boole foi apreciado meio século após por Alfred North Whitehead e Bertrand Russell quando produziram o trabalho chamado - *Principia Mathematica (1910-1913)* . O objetivo desse trabalho foi o de demonstrar que as raízes da matemática residem nas leis básicas da lógica, exercendo profundas influências sobre toda uma geração de cientistas matemáticos, inclusive Norbert Wiener e John von Neumann.

Nos anos 30, um dos trabalhos de lógica-matemática da maior importância foi desenvolvido pelo matemático, Alan Turing. Em 1936 (Turing,1988), ele desenvolveu a noção de uma máquina *Máquina de Turing* que em princípio seria capaz de operar qualquer tipo de cálculo. A idéia baseava-se em passar um fita, previamente dividida em segmentos idênticos (celas) marcados ou não com um sinal, através da máquina que seria capaz de interpretar as marcações e executar ações de acordo com os sinais lidos da fita. Com essas simples operações, a máquina poderia executar qualquer tipo de programa ou plano, que pudesse ser expresso em código binário. A demonstração e a prova da teoria de Turing - sobre a realidade de se construir máquinas operadas com um número indefinido de programas compostos simplesmente de zeros e uns - código binário, foi de importância fundamental para os pesquisadores.

Nessa mesma década o matemático Claude Shannon desenvolveu a noção chave sobre a teoria da informação onde sugeriu que a informação fosse desvinculada de seu conteúdo. Dessa forma ela seria resumida em uma simples decisão entre duas alternativas igualmente plausíveis. A unidade básica de informação é o bit - abreviação da sentença inglesa "binary digit" que é a quantidade de informação requerida para selecionar uma mensagem de duas alternativas igualmente prováveis. E em 1938, publicou o importante trabalho - *A Symbolic Analysis of Relay on Switching Circuits*, onde mostrou que reles e chaves do tipo encontrado nos circuitos elétricos de equipamentos eletrônicos poderiam ser expressos em termos das equações de Boole, onde para sistemas verdadeiro-falso haveriam chaves do tipo liga-desliga ou os estados fechado e aberto de um circuito elétrico. No trabalho de Shannon está a base para construções de máquinas que executassem operações lógicas (Cohen & Feigenbaum,1982). Ele também sugeriu novas maneiras de como circuitos elétricos poderiam ser projetados e simplificados. E no nível teórico, indicou que a programação de um computador deveria ser pensado como um problema de lógica formal invés de aritmético. Portanto Shannon introduziu o assunto de interesse puramente acadêmico para o mundo prático.

Finalmente John von Neumann desenvolveu a idéia de programas armazenados, onde as operações do computador podem ser direcionados ou controlados por meio de um programa ou de um conjunto de instruções alojados na memória interna do computador. Ele demonstrou como a lógica binária e a aritmética poderiam trabalhar juntas na formação de programas armazenáveis, onde um pode codificar instruções para a máquina na mesma linguagem que a dos dados que processa e, o outro então, misturar instruções e dados em um programa e guardá-los junto ao computador (McCorduck,1979).

Contemporâneo de Turing, outro matemático - Norbert Wiener envolveu-se com projetos fundamentais, entre outros, seu trabalho com servo-mecanismos em artilharia anti-aérea, mísseis guiados e aviões. Wiener iniciou o pensamento sobre a natureza dos sistemas realimentados (do inglês - "feedback"), auto-corretivos e auto-reguladores sejam eles mecânicos ou humanos. E introduziu com Bigelow, em 1943, o pensamento sobre a legitimidade das máquinas exibirem o processo de realimentação, através da diferença entre seus objetivos e seu desempenho real, e então trabalhar para reduzir essas diferenças. E Wiener em 1948 introduziu sua nova ciência chamada Cibernética responsável pela área da teoria de Controle e Comunicação em máquinas ou em animais. Reforçando então as idéias de McCulloch, Pitts e de von Neumann sobre a forte correlação entre a operação, das então novas máquinas de comunicação, e os organismos.

Warren McCulloch e Walter Pitts (1943) mostraram que operações de células nervosas e suas conexões com outras células nervosas - então chamadas redes neuronais, podiam ser modeladas em termos da Lógica. Esse modelo permitiu pensar na ativação de um neurônio ativando um outro neurônio, da mesma forma que um elemento ou uma proposição em uma sequência lógica pode implicar em outra proposição. Nervos poderiam ser pensados como declarações lógicas e a propriedade passa/não passa dos nervos, disparando ou não disparando poderia ser comparada a operações do cálculo proposicional, onde as declarações seriam ou verdadeiras ou falsas. Mais ainda, a analogia entre os neurônios e lógica poderia ser pensado em termos elétricos - como sinais do tipo passa / não-passa através de um circuito.

Nesse momento dois fatos permitiram a clara visualização do potencial do computador (McCorduck 1979):

- Turing e Von Neumann demonstraram, em princípio, o grande poder das máquinas computacionais.
- McCulloch e Pitts demonstraram que um dos princípios de operação do pensamento do cérebro humano - poderia ser o princípio lógico.

E formava-se então uma das importantes bases da Ciência da Computação e os princípios da Inteligência Artificial.

2.1.1.1 Marco

Os primeiros resultados expressivos da Inteligência Artificial começaram a ser obtidos na década de 50. Em 1952, Newell e Simon trabalhando com a capacidade de manipulação simbólica do computador, iniciaram a exploração desse recurso na resolução de problemas como jogos de xadrez, solução de problemas geométricos e a prova de teoremas de Lógica. Para tal, trabalharam no desenvolvimento de linguagens de alto-nível como o IPL - "Information Processing Languages" que em agosto de 1956 permitiu a implementação do programa chamado **Logic Theorist Machine - LT** que produziu, de fato, no computador Johniac da Rand Corporation, a primeira prova completa de um teorema do Principia Mathematica de Whitehead e Russel (Newell & Simon,1972). E através da implementação e execução desse programa, Newell, Simon e Cliff Shaw provaram, de maneira prática, a viabilidade da Inteligência Artificial.

Assume-se historicamente que o termo Inteligência Artificial foi cunhado e recebe seu impulso oficial na conferência que se realizou no verão de 1956 no campus do Dartmouth College de Hanover, Estado de New Hampshire, dos Estados Unidos da América, quando então um grupo de estudantes, pesquisadores e profissionais de várias áreas reuniram-se para discutir seus interesses na potencial utilização de computadores na resolução de problemas, ou mais especificamente, debater sobre o desenvolvimento de programas de computador que pudessem comportar-se ou pensar inteligentemente. Esse encontro é conhecido como **Conferência de Dartmouth** e destaca quatro elementos: John McCarthy, Marvin Minsky, Allen Newell e Herbert Simon. Pessoas as quais é atribuída a "paternidade" da Inteligência Artificial (McCorduck,1979).

Dando continuidade, Newell, Simon e seus colegas envolveram-se em um grande projeto chamado - **General Problem Solver - GPS**, um programa com o propósito de ser utilizado na resolução de problemas de qualquer natureza. O GPS deveria ser capaz de resolver problemas como: solução de teoremas, jogos de xadrez, quebra-cabeças do tipo missionário e canibal e a torre de Hanoi, de uma forma que pudesse assemelhar-se a maneira humana de resolvê-los. O GPS empregou a técnica chamada de análise de meios-e-fins para guiar a procura da solução que consiste na subdivisão de problemas em subproblemas mais fáceis. A análise de meios-e-fins consiste em tomar a diferença entre o que é dado e o que é desejado, por exemplo:- entre dois objetos, ou entre um objeto e a classe de objetos que o operador aplicar. A diferença é então utilizada para selecionar o desejado operador, ou seja, aquele que é relevante na redução dessa diferença e então aplicá-lo sobre essa diferença para minimizá-la. Esse procedimento é repetido até que o objetivo seja atingido ou que seja demonstrado que ele não pode ser atingido com as informações fornecidas ou com os operadores disponíveis no programa. A genialidade desse programa está na utilização do método para reduzir a distância e as manipulações simbólicas dos estados permitiu dizer que o GPS foi o primeiro programa que simulou um espectro do comportamento humano. A primeira versão desse programa foi a GPS-1 e foi codificada em IPL-IV para o computador Johniac em 1961 (Newell & Simon,1988).

A partir de então iniciam-se esforços explícitos e conjugados para o desenvolvimento dessa nova área da Ciência da Computação.

2.1.1.2 Eventos Chaves

A Tabela 1 abaixo, mostra em linha gerais, uma relação cronológica entre importantes fases da Inteligência Artificial - I.A. com alguns eventos chaves.

Tabela 1. Importantes fases da Inteligência Artificial.	
Período	Eventos Chaves
Raízes	<ul style="list-style-type: none"> • Lógica formal • Psicologia cognitiva
Pré Inteligência Artificial (1945-1954)	<ul style="list-style-type: none"> • Desenvolvimento do computador • H.Simon, <i>Administrative Behavior</i> • N.Wiener, <i>Cybernetics</i> • A.M.Turing, "Computing Machinery and Intelligence" • Conferência de Macy em cibernética
Formação e Iniciação da Pesquisa em I.A. (1955-1960)	<ul style="list-style-type: none"> • Crescimento da disponibilidade de computadores • IPL-I (Information Processing Language) • Conferência de Dartmouth • GPS (General Problem Solver)
Desenvolvimento e Redireção (1961-1970)	<ul style="list-style-type: none"> • A.Newel e H.Simon, <i>Human Problem Solving</i> • LISP (LISt Processing) • Heurística • Robótica • Programas de xadrez • DENDRAL (Stanford)
Sucesso dos Sistemas Baseados em Conhecimento (1971-1980)	<ul style="list-style-type: none"> • MYCIN (Stanford) • HEARSAY II (Carnegie-Mellon) • MACSYMA (MIT) • Engenharia do Conhecimento • EMYCIN (Stanford) • PROLOG
Corrida para novos Desenvolvimentos e Aplicações (1980-)	<ul style="list-style-type: none"> • Projeto Japonês - Quinta Geração • Consórcio Europeu - ESPRIT

A seguir, apresenta-se uma pequena amostra de alguns marcos fundamentais que, através de idéias e implementações, contribuíram significativamente para o desenvolvimento da Inteligência Artificial.

2.1.1.2.1 PERCEPTRON: Por volta de 1960, Cornell Frank Rosenblatt, desenvolveu um mecanismo de reconhecimento de padrões utilizando um sistema baseado no modelo de redes neuronais chamado - PERCEPTRON. O mecanismo consistia de uma rede de quatrocentas fotocélulas correspondentes aos neurônios da retina. Essas células eram conectadas aos elementos associativos, cuja função era a de coletar impulsos elétricos produzidos pelas fotocélulas. Um elemento associativo produziria um sinal somente se o estímulo resultante fosse superior a um pré-determinado valor limite.

2.1.1.2.2 Teste de Turing: Alan Turing em 1963 propôs um método para determinar se uma máquina poderia pensar. Seu método desde então tornou-se conhecido como - *Teste de Turing*. Para esse teste são necessárias duas pessoas e a máquina a ser avaliada. Uma das pessoas interpretará o papel do interrogador. O interrogador deve ficar em uma sala distinta da sala em que encontra-se o computador e da outra sala em que encontra-se a outra pessoa, com ele ou ela poderá se comunicar ao digitar perguntas e receber respostas digitadas, através de um terminal. O interrogador pode fazer perguntas para a pessoa ou para computador mas não sabe qual deles é quem. O interrogador apenas os conhece como A e B, e tem por finalidade determinar qual deles é a pessoa e qual a máquina. O objetivo da máquina é levar o interrogador a acreditar que ele seja a pessoa. A máquina tem permissão para fazer o que puder para “enganar” o interrogador. Assim, por exemplo, se lhe fosse feita a pergunta *Quanto é 12.324 vezes 73.981 ?* o computador poderia aguardar vários minutos e depois dar a resposta errada. Algumas pessoas acreditam que computador algum jamais passará no Teste de Turing.

2.1.1.2.3 ELIZA: Joseph Weizenbaum e Kenneth Colby trabalharam em programas para simularem o diálogo entre duas pessoas. Colby, que era psiquiatra interessou-se em distúrbios psiquiátricos desenvolveu um programa cujas respostas eram fornecidas por uma pessoa com uma personalidade com uma particular obsessão sobre a Máfia.

Weizenbaum em 1966, desenvolveu o programa chamado - ELIZA, que simulava o diálogo de um terapeuta da linha de Roger com um paciente (usuário). Esse programa implementa a técnica de reconhecimento de padrões. Cada sentença no texto de entrada é vasculhado em busca de palavras chaves como: eu, você, gostar, pai. Se uma dessas palavras é encontrada, a sentença é transformada de acordo com a regra associada a essa palavra. Mas se nenhuma palavra chave é encontrada no texto de entrada então:

- ELIZA responde através expressões sem um conteúdo aparente como: *Porque você pensa isso ?*, ou
- faz alguma referência neutra sobre alguma sentença anterior, ou ainda
- retoma a discussão sobre um assunto anterior.

Segue um trecho da conversa entre um “paciente/usuário” (apresentado em letras minúsculas) e o terapeuta - ELIZA (apresentado em letras maiúsculas):

¹ (Boden 1977,pp.106-7; Weizenbaum,1966)

homens são todos iguais.
DE QUE FORMA ?
estão sempre nos incomodando de uma forma ou de outra.

PODE PENSAR EM UM EXEMPLO ESPECÍFICO ?
bem, meu namorado me fez vir aqui.

SEU NAMORADO FEZ VOCÊ VIR ATE AQUI
ele acha que estou deprimida na maioria das vezes

SINTO MUITO SABER QUE VOCÊ ESTA DEPRIMIDA
é verdade. Sou infeliz.

VOCÊ ACHA QUE VINDO ATÉ AQUI AJUDARÁ VOCÊ A SER A NÃO SER INFELIZ ?
eu preciso de alguma ajuda, com certeza.

O QUE SIGNIFICARIA PARA VOCÊ TER ALGUMA AJUDA ?
talvez, eu pudesse aprender a ficar melhor com a minha mãe.

FALE MAIS SOBRE SUA FAMÍLIA

1

2.1.1.2.4 ARCS: Patrick Winston, trabalhou no aprendizado do significado das descrições estruturais. E o seu programa ARCS mostra o excelente resultado desse trabalho. Inicialmente são fornecidos ao programa alguns conceitos básicos de blocos e suas formas, bem como fatos estruturais básicos como:- blocos podem se suportar entre si e tocar-se entre si. Então o conceito de arco é ensinado ao sistema através da apresentação de uma série de exemplos. Desses exemplos, é construído uma rede semântica descritiva de um arco, que pode ser modificada até que se torne suficientemente robusto para atribuir uma resposta com precisão, conforme as ilustrações da Figura 2.

2.1.1.2.5 STUDENT: Em 1968, Daniel Bobrow aluno de Marvin Minsky, pesquisou a resolução de problemas no domínio da linguística e desenvolveu o programa chamado **STUDENT** que foi projetado para solucionar problemas de álgebra (que são encontrados em livros de matemática equivalentes ao nível de segundo grau das escolas brasileiras). Como um exemplo de um dos tipos de problemas tratados tem-se o seguinte:

O consumo de combustível do meu carro é de 15 milhas por galão. A distância entre Boston e Nova York é de 250 milhas. Quantos galões de combustível são necessários para uma viagem entre Nova York e Boston ?

O programa assume que cada sentença é uma equação. Por exemplo a palavra é foi codificada para significar - igual a por significa divisão. Nas palavras de Marvin Minsky:

Da palavra matemática “por” na primeira sentença “milhas por galão”, pode-se dizer que o número 15 pode ser obtido pela divisão de um certo número x de milhas, por outro número, y de galões. Além disso o programa não tem a mínima idéia do que milhas por galão significam ou sobre que assunto de carro estamos falando. A segunda sentença aparece para dizer algo igual a duzentos e cinquenta milhas - logo a frase “distância entre” é um bom candidato para ser o x . A terceira sentença pergunta sobre o número de galões - então a frase “combustível utilizados em uma viagem” é um candidato a ser y . Então propõe-se que uma das equações seja $x = 250$, e outra equação, $x/y = 15$. Logo, a parte matemática do programa pode facilmente encontrar $y = 250/15$.

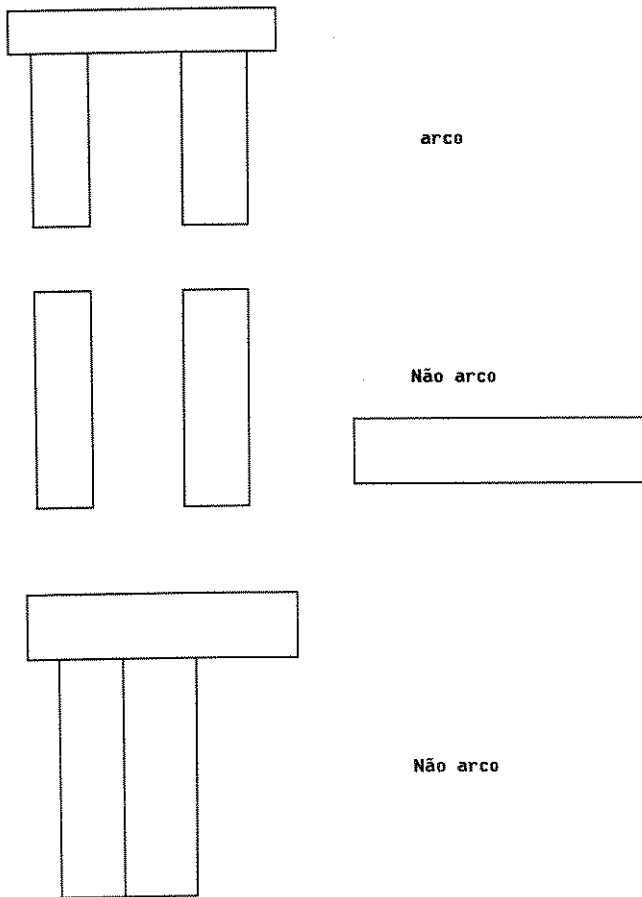


Figura 2. Exemplos apresentados ao programa ARCS. (Cohen & Feigenbaum, 1982)

2.1.1.2.6 LISP: Enquanto isso, John McCarthy estava empenhado trabalhando em uma de suas mais significativas realizações - a linguagem **LISP** (derivada da sentença inglesa: "LISt Processing"), que se tornou a linguagem de computador mais utilizada na área da Inteligência Artificial. A linguagem, basicamente, está relacionada com a apresentação e manipulação de listas, de itens de listas, e listas de listas nomeadas. Programas e dados são estruturados como listas. O poder da linguagem LISP deriva do fato de ser uma linguagem de programação recursiva, que é hierárquica. E assim pode ser descrita e pode operar em muitos níveis de detalhes. LISP também é uma linguagem muito flexível, pois permite que o programa mova-se entre níveis aninhados entre si, pode referir-se a si mesmo, pode operar-se tanto quanto necessário e pode automaticamente alocar memória de forma dinâmica.

2.1.1.2.7 MACSYMA: Originalmente desenhado em 1968 por Carol Engleman, Willian Martin e Joel Moses no Massachusetts Institute of Technology - MIT. MACSYSMA é um enorme sistema interativo que auxilia matemáticos, cientistas e engenheiros a solucionarem complexos problemas matemáticos. O sistema desempenha simbolicamente cálculo integral e diferencial através de centenas de regras adquiridas por especialistas em matemática aplicada representadas por mais de 300.000 linhas de código LISP e suportado por mais de 500 páginas de documentação. Basicamente, cada regra expressa um modo de transformar uma expressão em expressão equivalente; a solução do problemas está em encontrar a corrente de regras que transforme a expressão original na expressão simplificada.

2.1.1.2.8 Analogia de Evans: T.G.Evans, aluno de Marvin Minsky, desenvolveu em 1968 um programa do tipo:- solucionador de problemas (da sentença inglesa “problem solver”) através de analogias. Ao programa é dado um par de figuras que trazem alguma relação entre si e é pedido que se selecione um outro conjunto de figuras que complete a analogia visual. Por exemplo o programa caracteriza as analogias e traça as diferenças entre as figuras A e B, a seguir a regra da diferença é aplicada em C, obtido de um grupo de cinco candidatos devidamente identificados. Cabe observar que a resolução da analogia visual desse programa não ocorre através da percepção física, mas através da conversão da descrição em formas simbólicas - analogia geométrica, da maneira em que seria utilizada em um problema de analogia numérica, conforme Figura 3.

2.1.1.2.9 DENDRAL: Em 1971, Feigenbaum, Joshua Lederberg e Buchanan uniram-se para desenvolver o programa chamado - DENDRAL que foi projetado para identificar, de uma imensa quantidade de dados de um espectrógrafo de massa, qual o componente orgânico estava sendo analisado. Após o processamento dos dados relevantes, o programa formulava hipóteses sobre a estrutura molecular do componente e então testava as hipóteses formuladas de modo a tentar prever sua identificação. O resultado final era uma lista de possíveis componentes moleculares em ordem probabilisticamente decrescente. Esse programa armazenou e fez uso de grande quantidade de conhecimentos químicos, porque ele foi baseado no princípio de que um especialista retém e faz uso de uma grande quantidade de conhecimentos, e na medida do possível, o mesmo deveria ocorrer com um programa que se propõe a comportar-se inteligentemente.

2.1.1.2.10 SHRDLU: O programa SHRDLU foi desenvolvido por Terry Winograd em 1972. SHRDLU propõe ser um especialista trabalhando em um domínio muito limitado - o denominado micro-mundo dos blocos. O programa simula o braço de um robô que pode mover um conjunto de blocos das mais variadas formas, e permite que uma pessoa (usuário) envolva-se em um diálogo com o computador - perguntando, declarando, e comandando esse micro-mundo de objetos. O programa é um sistema integrado que faz uso da sintaxe, semântica e fatos sobre os blocos.

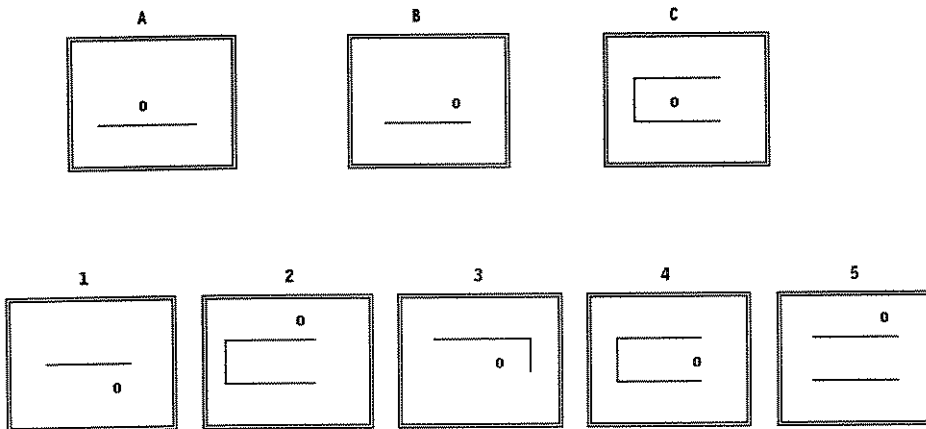
SHRDLU foi um importante marco para a Inteligência Artificial porque incorporou com sucesso duas novas idéias:

1. Utilizar o conceito de micro-mundo - um domínio que pode ser isoladamente analisado com muito sucesso.
2. Explorar a idéia de representar o conhecimento na forma de procedimentos, que resultou em diálogos muito naturais entre o usuário e o programa.

Segue um exemplo da interação de SHRDLU (apresentado em letras maiúsculas) e o usuário (em letras minúsculas). Entre parênteses estão os comentários de Winograd:

1. apanhe o bloco vermelho
OK
(- O sistema responde “OK” quando ele atende ao comando. Para pegar o bloco vermelho, ele tem que arrumar espaço movendo o bloco verde)
2. pegue a pirâmide
EU NÃO ENTENDO DE QUE PIRÂMIDE VOCÊ FALA
(- Há três pirâmides em cena, então o programa não faz nada)
3. encontre o bloco que é mais alto do que esse que você segura e coloque-o na caixa
POR “o”, ESTOU ASSUMINDO QUE VOCÊ QUE DIZER - O BLOCO QUE MAIS É MAIS ALTO QUE AQUELE QUE SEGURO.
OK
(- O sistema tem que concluir o que você quer dizer com “aquele” e “o”, e também, usando dedução decidir qual bloco o sistema está

- A figura A está para a figura B,
para qual das figuras a seguir está figura C ?



resposta: figura 4

Figura 3. Um exemplo da analogia de Evans. (Evans,1984)

segurando)

4. o que a caixa contém ?

A PIRÂMIDE AZUL E O BLOCO AZUL

5. o que suporta a pirâmide ?

A CAIXA

(- Anteriormente o sistema deu a seguinte resposta para uma pergunta sobre a pirâmide: "Eu não entendo de que pirâmide você fala". Agora o sistema assume aquele que foi acabado de mencionar).

6. a mesa pode apanhar os blocos ?

NÃO

(- Através das regras semânticas essa questão não faz sentido, logo deve ser falsa)

7. pode uma pirâmide ser suportada por um bloco ?

SIM

(- O sistema dedutivo encontra um exemplo real, logo ele sabe que é possível)

8. pode uma pirâmide suportar uma pirâmide ?

EU NÃO SEI

(- A questão tem sentido, mas não há exemplo na cena, logo não há um modo fácil de dizer sem ser capaz de examinar o programa do robô)

9. empilhe duas pirâmides

NÃO POSSO

(- O robô tenta e falha)

² (Winograd,1972; pp.8-11)

2.1.1.2.11 HEARSAY I e II: No final da década de 70 na Universidade de Carnegie-Mellon foi desenvolvido o HEARSAY que endereça o entendimento da fala. A sua arquitetura teve um papel muito importante para o desenvolvimento dos Sistemas Especialistas subsequentes. Em linhas gerais o sistema faz a representação em várias formas de conhecimento:- semântico, pragmático, sintático, léxico, fonêmico e fonético. Cada conhecimento fonte olha a hipótese gerada por todos os outros e mostrados no quadro-negro (da sentença inglesa “blackboard”) e então faz a melhor tentativa para descrever o que foi falado. É como se o HEARSAY fosse composto de vários diferentes outros Sistemas Especialistas, cada um analisando o dado e fazendo sua contribuição. Essa programa introduziu o conceito de controle por modelo de “blackboard”

HEARSAY II foi um importante marco pois demonstrou a superioridade dos métodos simbólicos-heurísticos sobre os métodos estatísticos em problemas envolvendo significado. E também, demonstrando como múltiplas fontes de conhecimento poderiam ser controladas e integradas, de forma a gradualmente, acumular a solução de um problema muito complexo.

2.1.1.2.12 SOPHIE: Desenvolvido por Brown e Burton em 1975, SOPHIE é um programa que comporta-se como um instrutor de um laboratório de eletrônica interagindo com estudantes que se dedicam a identificar o mal funcionamento de um equipamento. SHOPHIE emprega a simulação de modelos eletrônicos e também possui o conhecimento declarado de um particular circuito codificado em redes semânticas e rotinas que raciocinam através dessa forma de conhecimento. Quando a situação em um circuito se altera, por exemplo uma falha é inserida, o programa muda da condição real para a simulação, determinando os detalhes das tensões e correntes em vários pontos do circuito e no final atualiza a sua rede semântica com os novos valores. Dessa forma ele evita o cálculo de todos os valores através da elaboração do raciocínio simbólico utilizando as leis de Kirchhof e outros. Assim o programa incorpora técnicas que combinam os resultados de simulações com informações obtidas pelo raciocínio de conhecimento declarativo, de forma a responder a muitos tipos de perguntas.

2.1.1.2.13 MYCIN: Foi desenvolvido na Universidade de Stanford (Shortliffe,1976). MYCIN auxilia na diagnose e terapia de infecções sanguíneas. Seu conhecimento compreende aproximadamente quatrocentas regras relacionando possíveis condições às interpretações associadas. No solucionamento do problema, MYCIN testa as regras de condição contra o dados disponíveis, ou seja, faz o encadeamento para trás das hipóteses de diagnósticos para os dados, guiado por regras de inferência. Se apropriado, o programa tenta inferir sobre a condição de verdade ou de falsidade das outras regras. O conhecimento é representado na forma de regras de produção e a aquisição de conhecimento envolve a adição de novas regras ao sistema. O uso de regras simples do tipo SE-ENTÃO estimularam uma variedade de sistemas relacionados como:

1. EMYCIN (Van Melle,1979), proveniente da expressão inglesa:- “Empty Mycin (EMycin)” onde o mecanismo de inferência foi separado do domínio do conhecimento. Assim, trata-se de um programa do tipo casca - “shell” que pode tratar com conhecimentos de outros campos.
2. TERESIAS (Davis,1982), uma interface interativa para auxiliar na aquisição de conhecimentos.
3. NEOMYCIN, um programa onde o banco de conhecimentos original do Mycin foi reescrito para atuar como um sistema de treinamento.

2.1.1.2.14 PROSPECTOR: Desenvolvido no final de década de 70 no Stanford Research Institute por um grupo composto entre outros por: P.Hart, R.Duda, R.Reboh, K.Konolige, P.Barret e M.Einandi. PROSPECTOR foi projetado para auxiliar geólogos na investigação, em estágios iniciais, de possíveis depósitos geológicos minerais. Os dados são, primeiramente, observações geológicas da superfície e são assumidas como incertas e incompletas. O programa alerta o usuário sobre as possíveis interpretações e identifica observações adicionais que podem ser valiosas para se chegar a um conclusão mais definida. A sua base de dados baseia-se em redes semânticas organizada em torno de cinco modelos diferentes. Cada modelo descreve a informação e as relações pertinentes a um tipo particular de depósito mineral. As proposições são considerados nós na rede e cada asserção pode ser: desconhecida, verdadeira, falsa ou verdadeira com alguma probabilidade - ponderação. Os arcos de influência da rede são as regras de inferência. Cada regra especifica como a probabilidade de uma proposição/asserção afetará a probabilidade de outra probabilidade. As regras de inferência são as mesmas regras de produção do MYCIN.

2.1.1.2.15 PROLOG: PROLOG somou-se recentemente a lista de linguagens de programação da Inteligência Artificial. Foi inventada em 1972 por Alain Colmerauer e seus associados na Universidade de Marselha - França, que escreveu o primeiro interpretador PROLOG em linguagem ALGOL-W. Em 1975, P.Roussel, um colega de Colmerauer, implementou um interpretador mais eficiente em FORTRAN. A programação de computadores em PROLOG consiste na declaração de fatos, definindo regras e perguntando questões sobre objetos e suas relações. Por exemplo, supondo que dissessemos que um garoto chamado João é filho de Maria; e supondo-se que também fornecêssemos ao PROLOG uma regra que relacione marido e esposa. Então poderíamos perguntar: "Quem é o pai de João ?" PROLOG então poderia retornar *não* significando que não foi encontrado um pai para João ou forneceria o nome do pai de João. Programar em PROLOG significa fornecer fatos e regras e então fazer perguntas sobre eles. PROLOG utiliza a sintaxe de lógica de predicado na procura da base de conhecimento dos fatos e regras e faz as inferências necessárias para responder as questões de uma forma transparente ao usuário. A lógica, que fornecido pelo programador usuário, é mantida completamente separada do controle, que faz do interpretador/compilador PROLOG.

2.2 Sistemas Especialistas

2.2.1 Introdução

O maior uso de computadores nos últimos 40 anos tem sido em aplicações numéricas, e no processamento de dados administrativos e contábeis. Esses sistemas são capazes de coletar e processar um enorme volume de dados através de complexos algoritmos que executam procedimentos do tipo passo a passo de forma a garantir a correta conclusão, quando os dados de entrada estão corretos (Warkentin,1993). Por exemplo, diariamente complexos algoritmos são executados em instituições financeiras. A movimentação em contas correntes pode implicar em alteração no saldo, aplicação de juros, correção monetária, caderneta de poupança, investimento em bolsa de valores, e uma infinidade de outras operações que ao final do dia tem o objetivo de fornecer o resultado pré-determinado esperado - o saldo bancário geral da empresa. Portanto, algoritmo é um procedimento que garante encontrar a correta solução de um problema em um dado período de tempo ou dizer que não há uma solução. Sendo que os programas que implementam esses algoritmos comportam-se conforme o pensamento do programador, isto é, o processamento computacional resume-se na rápida e correta execução dos procedimentos explicitados pelo programador à máquina (Gray,1993).

Por outro lado, os métodos heurísticos não garantem sempre uma solução, e baseiam-se no fato de que a maior parte da inteligência humana não é matemático ou quantitativo, isto é, normalmente um especialista humano não faz uso de algoritmos na resolução de problemas (Fulda,1993). E problemas intelectuais não demandam respostas corretas, mas respostas adequadas. Esse fato requer ponderações entre as diferentes partes da evidência de forma a selecionar um caminho entre os muitos disponíveis, onde o potencial resultado dos diferentes caminhos precisam ser acessados e comparados com o objetivo, e o caminho com a melhor "aparência" resulta na escolha (Hieb,1992). Dessa forma os métodos para a resolução de problemas são usualmente técnicas de raciocínio qualitativo que relacionam itens através de regras de julgamento, ou heurística, como também através de leis teóricas e definições. Dessa forma, a criação de sistemas especialistas difere significativamente da abordagem dos programas convencionais porque a representação desses conceitos é extremamente difícil utilizando-se a programação convencional (Kattan,1994).

(Harmon & King,1985) descrevem alguns contrastes entre um Sistema Especialista e um programa dito convencional:

1. A tarefa realizada por um Sistema Especialista foi previamente desempenhada por um especialista humano.
2. Enquanto um programa convencional é mantido por um programador, um Sistema Especialista é mantido pelo(s) especialista(a) ou por um engenheiro do conhecimento.
3. A base de conhecimento de um Sistema Especialista é facilmente legível e fácil de ser modificada.
4. Programas convencionais tem a sua estrutura geral baseada em algoritmos, enquanto Sistemas Especialistas tem a estrutura geral baseada na heurística.

Além disso, os sistemas especialistas, estão em contínuo processo de desenvolvimento interativo para o seu sucessivo refinamento e crescimento da base de conhecimento, o que não ocorre nos programas ditos convencionais (Fiol & Nicolau,1993). Uma outra diferença, está na presença do Engenheiro do Conhecimento. Este trabalha no desenvolvimento do programa e na análise de como o especialista humano resolve o problema auxiliando-o na descrição de seu conhecimento e das estratégias de inferência na forma que permita o conhecimento a ser codificado.

(Waterman,1985),(Harmon & King,1985) e (Laufmann,1990), enfatizam as diferenças entre essas duas formas de programas, conforme Tabelas 2 e 3.

Tabela 2. Uma comparação entre as formas de processamento.

Processamento de Dados	Processamento Simbólico
Algorítmico	Heurístico
Banco de dados endereçado numericamente. Representação e uso de dados.	Base de conhecimento simbolicamente estruturada. Representação e uso de conhecimento.
Processamento numericamente orientado.	Processamento simbolicamente orientado.
Processamento em grupo (da palavra inglesa "batch") sequencial.	Processamento altamente interativo.
Processo repetitivo.	Processo inferencial.
Impossível explicação do processamento.	Fácil explicação da linha de raciocínio ou resposta.

Nos programas convencionais baseados em algoritmos, o conhecimento é inserido dentro do programa código como uma sequência de operações determinadas sobre os dados. Nos sistemas especialistas, existe uma separação entre o conhecimento utilizado para resolver o problema e o programa código que o manipula para a obtenção da resposta desejada. Esta separação facilita a modificação do conhecimento, por exemplo, adição de novos fatos, sem interferir com programa utilizado para manipulá-lo. A idéia revolucionária sobre o Sistema Especialista está na nova maneira de se utilizar o computador na resolução de problemas (Arruda,1988). Ver Tabela 3.

Tabela 3. Programas convencionais versus Sistemas Especialistas. (Laufmann,1990)

Programas Convencionais	Sistemas Especialistas
Para cada conjunto de dados há somente uma solução	Mais de uma solução possível
Não há incerteza. Mesmo quando a solução é dada em probabilidade, há certeza na probabilidade.	Não há certeza nas soluções e algumas respostas erradas são toleradas.
Se faltam dados não pode se chegar a uma conclusão. Dados quantitativos.	Dados são incompletos,incertos,subjetivos,inconsistentes, sujeitos a mudança (dados conceituais)
Decisões e dados podem ser reduzidos a números, fórmulas e algoritmos.	Lidam com conhecimento conceitual que não pode ser reduzido a números.
Relativamente fáceis de se modificar e atualizar.	Relativamente fáceis de se modificar,atualizar e aumentar.
Trabalham primariamente com processamento numérico.	Trabalham primariamente com processamento simbólico.
Informações e estrutura de controle integrados no corpo do programa.	Estrutura de controle separada do conhecimento utilizado pelo sistema.
Soluções através de algoritmos (passos da solução estão explícitos).	Utilizam procura heurística (passos da solução estão implícitos).
Desenvolvimento linear e direto.	Desenvolvimento cíclico e incremental.

Dentro da Inteligência Artificial, algumas áreas mais se desenvolveram e se destacaram em relação a outras, em função de fatores históricos e de necessidades específicas. Sendo que as áreas que melhor se beneficiaram e se adequaram foram (Hayes-Roth,1983):

1. **Resolução de Problemas** - programas interativos em forma de diálogo, explicando o problema, executando testes, perguntando, auxiliando na resolução de problemas na: Medicina, Geologia, Química, Educação, Engenharia, Sistemas Computacionais e outras áreas.
2. **Robótica** - Manipulação do robô através de movimentos, planejamento e execução de ações.
3. **Visão** - Programas para reconhecimento de padrões, objetos, cenas visuais, alterações de figuras e análise de fotos aerométricas
4. **Processamento de Linguagem Natural** - Compreensão da linguagem escrita e falada. Geração de textos.
5. **Educação** - Desenvolvimento de ambientes e programas direcionados ao ensino e aprendizagem em diferentes domínios do conhecimento.
6. **Novas Arquiteturas e Sistemas Computacionais** - Paralelismo, Redes Neurais e Conexão. Ambientes altamente interativos de desenvolvimento de programas (da sentença inglesa - "framework development environment").
7. **Jogos** - xadrez, enigmas, exercícios de combate militar e outros.
8. **Provas de Teoremas Matemáticos**

Tais áreas melhor se beneficiaram com o desenvolvimento da Inteligência Artificial, porque as soluções computacionais tradicionais não apresentaram alternativas eficientes. Por exemplo, áreas onde são necessárias aplicações de heurística sem uma definição algorítmica das estratégias de solução e onde se fazem necessárias análises de dados que incluam uma parcela de incerteza em conjunto com a aplicação de conhecimentos adquiridos e armazenados (Goldstein 1994).

Na área de **Resolução de Problemas** os programas de resolução de problemas de propósito geral exemplificado através do GPS, provaram ser um trabalho muito difícil e de certa forma infrutífero (Newell & Simon,1988). E historicamente, provou-se que quanto mais classes de problema um único programa deseja manipular, mais pobremente ele manipulará cada uma dessas classes individualmente, com um maior grau de dificuldade e menor eficiência (Burton,1993).

Assim, a abordagem de desenvolvimento desses programas mudou, e observa-se nos anos 70, esforços concentrados no desenvolvimento das técnicas de representação do problema -(como formular o problema para simplificar a resolução) e na técnica da busca - (como controlar a busca da resolução de forma a não ser muito longa ou absorver muitos recursos como memória). E no final da década de 70, iniciaram-se os desenvolvimentos de programas de Resolução de Problemas onde o poder desses programas reside no conhecimento possuído e acumulado, e não somente da forma de representação ou da busca da solução. Essa realização levou ao desenvolvimento de programas de computadores com propósitos específicos, sistemas especializados em áreas de problemas muito específicos. Esses programas são chamados de Sistemas Especialistas ou Sistemas Baseados em Conhecimento (Germond,1993).

Sistemas especialistas baseados em conhecimento, ou sistemas de conhecimento, empregam o conhecimento humano para solucionar problemas que comumente requerem a presença e a inteligência humana (Hayes-Roth,1983). Sistemas de conhecimento representam e aplicam o conhecimento eletronicamente. Essa capacidade torna os sistemas baseados em conhecimento mais poderosos que as tecnologias anteriores para armazenamento e transmissão de conhecimento. Os livros, por exemplo, armazenam o maior volume de conhecimento existente, que entretanto, é retida passivamente na forma de símbolos. Antes que o conhecimento armazenado possa ser aplicado, um ser humano precisa recuperá-lo, interpretá-lo e decidir como explorá-lo na solução de problemas (Kuipers,1975).

Os programas convencionais consistem de duas partes distintas, algoritmo e dado. O algoritmo determina como resolver tipos específicos de problemas, e os dados caracterizam os parâmetros no particular problema. Entretanto o conhecimento humano não se encaixa nesse modelo. Porque muito do conhecimento consiste de fragmentos elementares de como fazer (do inglês "know-how"). Os sistemas baseados em conhecimento, coletam esses fragmentos em uma base de conhecimento e então acessa essa base de conhecimento para raciocinar sobre o problema específico. Como consequência, um sistema de conhecimento difere-se dos programas convencionais na forma como são organizados, na forma como incorporam o conhecimento, na forma de execução, e na impressão que criam quando da interação com o usuário. Sistemas baseados em conhecimento simulam o desempenho de um especialista humano (Weerahandi,1994).

Uma forma de se pensar sobre o solucionamento de problemas é em termos do espaço do problema, que consiste de um conjunto de estados de conhecimento e uma seleção de operadores que podem ser usados para transformar estados. Um problema é definido em termos de um estado inicial e um estado objetivo (um estado ou conjunto de estados). O solucionamento de problemas envolve encontrar um sequência de operadores -um caminho- pelo qual se possa transformar o estado inicial no estado objetivo. Logo o solucionamento de problemas é um caso de procura no espaço do problema, baseado em um conhecimento parcial de como proceder (que operador selecionar); esse conhecimento de busca controlada guia a busca no espaço. E uma vez que o conhecimento especializado ou o processo cognitivo é aprendido, o especialista devota um pequena ação consciente para a estratégia de controle de busca e portanto torna-se pouco preparado para explicar a estratégia de controle para outros (Mitchel et all,1985).

A aplicação de sistemas baseados em conhecimento pode trazer as seguintes vantagens (Turban,1988):

- retenção do conhecimento crítico e experiência,
- maior disponibilidade do conhecimento especializado,
- redução da influência da aprendizagem na estabilização do processo,
- redução da carga de trabalho do especialista humano,
- maior autonomia do usuário,
- redução do tempo necessário para se chegar a uma conclusão,
- menor dependência da experiência do especialista,
- melhor qualidade,
- menor custo,
- maior competitividade.

2.2.2 Definição

Um especialista humano é alguém que é reconhecido como sendo capaz de resolver uma complexa classe particular de problemas, de uma forma especialmente bem e eficiente. Um especialista, também, geralmente conhece os limites de seu conhecimento especializado (Waterman,1985).

Paul E.Johnson um cientista que devotou muitos anos estudando o comportamento de especialistas humanos descreve o que queremos dizer quando nos referimos ao termo especialista (Gammack,1987):

Especialista é a pessoa que, por causa de seu treinamento e experiência em sua área de domínio, é capaz de realizar tarefas que o resto de nós não consegue. Especialista é aquele que é competente não somente pela eficiente aplicação de seus conhecimentos e habilidades quando da resolução de problemas, como também devido a sua capacidade de:

1. pesquisar e decodificar informações que aparentam ser irrelevantes e obter a informação básica desejada,
2. agrupar, associar e reconhecer os problemas e situações que lhe são familiares,
3. ser eficiente nas ações que toma.

O comportamento do especialista é o corpo operante do conhecimento. É razoável supor então que, especialistas, sejam aqueles a quem devemos perguntar, quando gostaríamos de representar o conhecimento especializado - "expertise" (Waterman,1986).

Os especialistas humanos possuem duas importantes qualidades em comum:

1. considerável conhecimento em um domínio específico e,
2. estratégias efetivas para um rápida seleção, classificação e aplicação de seu conhecimento quando da resolução de problemas - estratégias consistidas de regras de dedução, heurística formal e intuição ou sentimento (da palavra inglesa "feeling").

Essas qualidades, em todos os casos, são eficientemente aplicados pelos especialistas humanos na resolução dos problemas de seu domínio. Dessa forma, pode-se adotar a definição de Sistema Especialista como a seguir (Bowen,1994):

Sistema Especialista baseado em computadores é uma coleção de programas de computador projetados para simular o comportamento de um especialista humano na resolução de problemas do domínio de seu conhecimento especializado. Seu processo de aproximação ao raciocínio de um especialista humano deve-se a dois principais componentes:

1. uma base de conhecimento que contém uma larga base de informações do especialista humano,e
2. uma máquina de inferência que aplica esses conhecimentos eficientemente.

É chamado de sistema, ao invés de programa, porque é composto por um módulo de resolução de problemas e também, usualmente, por um módulo de suporte. O ambiente de suporte auxilia o usuário a interagir com o programa principal e pode contar com sofisticadas ferramentas como: auxílio a depuração no teste e avaliação do código, facilidades de edição para auxiliar o especialista a alterar a base de conhecimento e de dados, e avançada interface gráfica para auxiliar o usuário na entrada de dados e leitura das informações durante o processamento do sistema (Harmon & King,1985).

E. Feigenbaum definiu Sistema Especialista como sendo (Cohen & Feigenbaum,1982):

Um programa inteligente de computador que utiliza conhecimentos e procedimentos de inferência para solucionar problemas que, sendo difíceis o suficiente, requerem um especialista humano para a sua solução. O conhecimento necessário para desempenhar em tal nível mais os procedimentos de inferência utilizados, podem ser pensados como um modelo do melhor especialista da área. O conhecimento de um Sistema Especialista consiste de *fatos e heurísticas*. Os fatos constituem o corpo da informação que é largamente difundida, publicamente disponível e geralmente acordada entre os especialistas da área. A heurística é em sua maior parte um conhecimento, restrito, de poucas e boas regras de bom julgamento -(regras de bom senso, ou boas regras de inicialização - “chute”), que caracterizam o nível do especialista e das decisões de um especialista de uma dada área. O nível de desempenho de um Sistema Especialista, é em princípio, função de seu tamanho e da qualidade do conhecimento possuído.

2.2.3 Classificação

De acordo com (Hayes-Roth, Waterman, Lenat, 1983) a maioria das aplicações da Engenharia do Conhecimento enquadram-se em uma das áreas abaixo:

1. **Interpretação** - Sistemas que inferem descrições de situações. Um sistema de interpretação explica os dados observados designando significados simbólicos, descrevendo situações. Nessa categoria incluem-se: vigilância, entendimento da fala, análise de imagens, elucidação de estruturas químicas, interpretação de sinais, e outros
2. **Previsão** - Sistemas que inferem consequências sobre uma dada situação. Nessa categoria incluem-se: previsão meteorológica, previsão demográfica, previsão de tráfego, estimativas de colheita, previsão militar. Ignorando-se as estimativas probabilísticas, sistemas de previsão podem gerar um grande número de possíveis cenários.
3. **Diagnose** - Sistemas que relacionam o comportamento irregular observado com as possíveis causas. Nessa categoria incluem-se entre outras, as áreas: médica, mecânica, eletrônica, diagnose de programas - "software".
4. **Projeto** - Sistemas que desenvolvem configurações de objetos que satisfaçam a determinadas especificações. Esses programas incluem lay-out de circuitos eletrônicos, projetos de construção e orçamentos.
5. **Planejamento** - Sistemas que planejam ações, onde incluem a programação automática, robótica, projetos, comunicação, roteamento, planejamento de problemas militares e outros.
6. **Monitoramento** - Sistemas que observam e analisam o comportamento de um outro sistema. Os aspectos cruciais ou vulnerabilidades, correspondem a potenciais falhas do sistema observado. Incluem-se sistemas de monitoramento computadorizada em usinas nucleares, tráfego aéreo, epidemias, monitoramento fiscal e outros.
7. **Depuração** - Sistemas que identificam um mau funcionamento. Sistemas computadorizados de depuração existem para programas de computadores na forma de uma base de conhecimentos inteligente e editores de texto, mas não se qualificam como totalmente sendo Sistemas Especialistas.
8. **Reparos** - Sistemas que desenvolvem e executam planos para administrar soluções de problemas diagnosticados. Esses sistemas incorporam a capacidade de depuração, planejamento, e execução. Sistemas auxiliados por computadores estão em domínios das áreas automotivas, redes, aviação e manutenção de computadores.
9. **Instrução** - Sistemas de instrução diagnosticam e depuram o comportamento de iniciantes/novatos. Tipicamente esses sistemas iniciam-se pela construção de uma descrição hipotética do conhecimento do iniciante através da interpretação de seu comportamento, esse fato permite a possível identificação de uma área de deficiência do conhecimento e o sistema indica um treinamento apropriado que pode ser aplicado através da interação entre o sistema e o estudante.
10. **Controle** - Sistemas que adaptativamente controla todo o ambiente de um outro sistema. Para isso o sistema de controle deve repetidamente interpretar a situação corrente, prever, diagnosticar antecipadamente causas de problemas, formular planos de solução e monitorar a execução dos mesmos. Problemas endereçados a sistemas de controles incluem: controle de tráfego aéreo, gerenciamento de negócios, gerenciamento de batalhas e controle de missões.

2.2.4 Estrutura

A noção central de um solucionador de problemas inteligente é aquela em que o sistema deve construir a solução de forma seletiva e eficiente a partir do espaço de alternativas disponíveis. Dentro desse contexto o termo arquitetura refere-se a ciência e ao método de se projetar o que determina a estrutura de um Sistema Especialista (Hayes-Roth et al,1983). Essa definição permite uma grande variedade de implementações exemplificadas por MYCIN (Shortliffe,1976) e HEARSAY-II (Erman et al,1980), sendo que os métodos de busca da solução são uma das mais variáveis características dessas estruturas.

A estrutura de um Sistema Especialista caracteriza-se pela separação entre as estratégias de solução do problema e os algoritmos de controle do processamento da informação. Uma arquitetura típica é composta pelos seguintes elementos (Kowalski,1993):

1. *Base de conhecimento*, onde reside o conjunto de informações que codificam a estratégia de solução e representação do conhecimento.
2. *Base de dados*, que contém as características e parâmetros relativos ao problema a ser resolvido.
3. *Módulo de inferência*, que controla o processamento do sistema e aplica a base de conhecimentos à base de dados, para obter novos dados e conhecimentos.
4. *Módulo de explicação*, que permite ao sistema explicar como uma conclusão foi obtida através do rastreamento e da reconstrução da cadeia de raciocínio que levaram a uma dada resposta. A habilidade de se explicar uma resposta ou linha de raciocínio é uma importante e diferenciadora característica de um programa inteligente.
5. *Módulo de aquisição do conhecimento*, que através de técnicas de aquisição, aplicadas manualmente ou com auxílio de ferramenta, adquire o conhecimento de um ou mais especialistas e implementa a base de conhecimento utilizando-se das estruturas de dados definidas pelas técnicas de representação.
6. *Interface homem-máquina*, que permite a comunicação e interação entre o sistema e o usuário.

Uma arquitetura típica de um Sistema Especialista é representada na Figura 4.

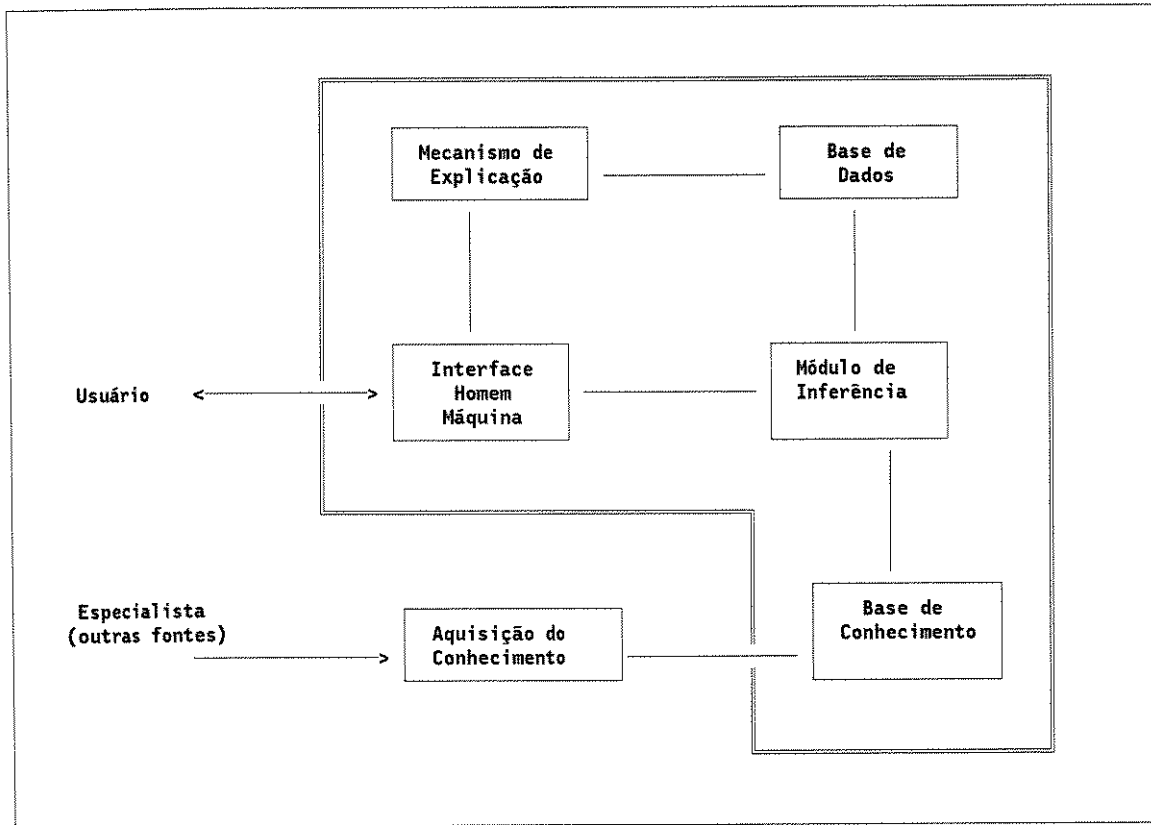


Figura 4. Arquitetura típica de um Sistema Especialista. (Waterman,1985)

2.2.5 Construção

O engenheiro do conhecimento extrai do especialista humano seus procedimentos, estratégias e “regras de bom senso” para a solução de problemas; e constrói esse conhecimento em um sistema computacional.

O procedimento de extração do conhecimento do especialista humano e a codificação do mesmo na forma de um programa é chamada de *Aquisição do Conhecimento*. Essa transferência e transformação da “expertise” de solucionar problemas da fonte de conhecimento para o programa é o coração do processo de desenvolvimento do Sistema Especialista. Através de uma extensa série de interações o engenheiro do conhecimento e o especialista definem o problema a ser atacado, descobrem os conceitos básicos envolvidos e desenvolvem regras que expressam as relações existentes dos conceitos obtidos (Dubas,1990).

A ferramenta de construção é a linguagem de programação utilizada pelo engenheiro do conhecimento ou programador para construir o Sistema Especialista. Em Inteligência Artificial o termo ferramenta, normalmente refere-se a linguagem de programação e ao ambiente de suporte utilizados (Estevanik,1994).

Os principais estágios da construção de um Sistema Especialista são descritos em (Kowalski,1993) como sendo:

1. **Identificação** - Determinação das características do problema. O engenheiro do conhecimento e o especialista, juntos, identificam a área problema e definem seu escopo. Também identificam os participantes no processo de desenvolvimento - especialistas adicionais, determinam os recursos necessários - cronogramas e facilidades computacionais, e decidem sobre os objetivos da construção do sistema.
2. **Conceituação** - Identificação dos conceitos para representação do conhecimento. Nessa fase o especialista e o engenheiro do conhecimento explicam os conceitos chaves, relações e o fluxo de informação característico necessários para descrever o processo de resolução de problemas do dado domínio. Também são especificadas sub-tarefas, estratégias e restrições relacionadas a atividade da resolução dos problemas.
3. **Formalização** - Projeto das estruturas para organização do conhecimento. Este estágio envolve o mapeamento dos conceitos chaves e das relações na representação formal sugerida por uma ferramenta de construção ou de uma linguagem.
4. **Implementação** - Formulação das estruturas que incorporam o conhecimento. Durante a implementação o engenheiro do conhecimento combina e reorganiza o conhecimento formalizado tornando-o compatível com fluxo de informação característico do problema. O conjunto resultante, por exemplo das regras, e das estruturas de controle associadas definem o programa protótipo capaz de ser executado e testado.
5. **Teste** - Validação da representação do conhecimento. A fase de teste envolve a avaliação do desempenho do programa. E a fase de revisão para verificar a conformidade com os padrões e definições estabelecidas pelos especialistas do domínio. Tipicamente, o especialista avalia o desempenho do programa e auxilia o engenheiro do conhecimento nas revisões.

Esses estágios não são necessariamente modulares, bem definidos ou mesmo independentes, mas caracterizam claramente as fases do processo de construção de um Sistema Especialista.

2.2.6 Aquisição do Conhecimento

Aquisição do conhecimento pode ser entendido como o processo de transferência e transformação do conhecimento especializado - "expertise" através da extração, estruturação, organização e representação desse conhecimento. Sendo que as potenciais fontes de conhecimento incluem:- especialista humano, livros, banco de dados e a própria experiência (Hayes-Roth et al,1983).

Neste tópico será focado a aquisição do conhecimento de especialistas humanos. Um dos grandes problemas no desenvolvimento e construção de um Sistema Especialista é atribuído a extração e caracterização do conhecimento especializado do especialista humano - "expertise". Isto porque o conhecimento eliciado constitui a estrutura básica de um Sistema Especialista. E a qualidade, precisão e a integridade do conhecimento adquirido tem um grande efeito na qualidade do sistema, em termos de: desempenho, eficiência e precisão na resolução de problemas por esse sistema (Janardan & Salvendy,1988).

De acordo com (Warkentin,1993) as razões básicas que tornam o conhecimento humano ser expresso de forma incompleta, imprecisa ou irrelevante tem como principais motivos:

1. a limitada habilidade humana de verbalizar o conhecimento e as estratégias usadas na resolução de problemas.
2. ao longo de muito tempo de vivência e experiência, geralmente:
 - a. os especialistas usam o conhecimento de uma maneira automática para resolver os problemas,
 - b. os conhecimentos do especialista tornam-se extremamente detalhados e interconectados de forma que os métodos que utilizam na resolução de problemas não são totalmente compreendidos.

Assim, a aquisição do conhecimento revelou-se um processo com características multidisciplinares, envolvendo problemas cujas origens estendem-se da lógica a psicologia cognitiva, incluindo a ciência da computação e a linguística.

Aquisição do Conhecimento compreende a eliciação, análise e transformação do conhecimento em estruturas necessárias a construção do Sistema Baseado em Conhecimento (Breuker & Wielinga,1987).

Os métodos existentes para a extração do conhecimento podem ser classificadas em: técnicas de entrevistas manual e metodologias interativas. Sendo que as correntes abordagens para aquisição do conhecimento podem ser agrupadas em três áreas (Hart,1985) (Kobayashi & Nakamura,1991):

1. **Engenharia do conhecimento tradicional** : - É o método tradicional e convencional, onde o engenheiro do conhecimento centraliza a incorporação do conhecimento e constrói o programa, baseado na interação direta com o(s) especialista(s) do domínio (Burton,1993).
2. **Ferramentas interativas para aquisição do conhecimento** :- Fornecem uma interface estruturada para especialistas e engenheiros do conhecimento construir as bases do conhecimento dentro de uma arquitetura pré-determinada (Arciszewski et all,1992). As ferramentas mais avançadas tem um trabalho ativo auxiliando o especialista a identificar o conhecimento necessário a ser adquirido, como TEIRESIAS (Davis,1976) e AQUINAS (Boose & Bradshaw,1988). A chave do sucesso das ferramentas interativas está na utilização e conhecimento sobre essa tarefa e no método particular para resolução de problemas suportado pela arquitetura definida. Esse fator trás dois benefícios imediatos na aquisição do conhecimento:
 - a. A interface entre o homem e a máquina pode ser estruturada para adquirir conhecimento relevante a tarefa e somente eliciar conhecimento na forma útil para o método.

- b. Dado que é projetada para trabalhar com um específico método de solução de problemas e assim saber como o método aplica-se ao domínio do conhecimento, a ferramenta pode analisar as entradas do usuário em termos de completude e consistência.

As ferramentas de aquisição de conhecimento reduzem a tarefa de construção da base de conhecimento para a tarefa de instanciação e representação do específico domínio.

3. **Máquina de aprendizado** :- Onde programas de aprendizado, produzem representações operacionais do conhecimento à partir de ações desempenhados pelo(s) especialista(s). A máquina pode facilitar a aquisição do conhecimento transformando o conhecimento prontamente disponível, como na forma de exemplos, em uma forma mais prática, como regras de diagnóstico (Kattan,1994).

De imediato, as máquinas de aprendizado, podem facilitar a aquisição do conhecimento para um Sistema Especialista sendo capazes de (Shalin,Wieniewski e Levi,1988):

- a. Contribuir para construção inicial da base de conhecimentos, como em (Michalski & Chilausky,1980) e em (Wisniewski,Winston,Smith e Klein,1987).
- b. Refinar a base de conhecimento existente como em (Davis,1979) e em (Mitchel,Mahadevan e Steinberg,1985).
- c. Auxiliar na adaptação de um sistema baseado em conhecimento. Por exemplo ajustar-se ao conhecimento ou ao estilo do usuário final.
- d. Prover um mecanismo formalizado para o processo de aquisição do conhecimento, substituindo a não completa definição da relação engenheiro-especialista na interação para aquisição do conhecimento.

Máquina de aprendizado é uma forma promissora de se ter computadores comportando-se de forma inteligente sem a necessidade de programá-los ou sem a necessidade de se construir e depurar complexas bases de conhecimento.

Entende-se por **Eliciação do Conhecimento**, a interação direta entre o engenheiro do conhecimento e o especialista, onde o objetivo do primeiro é determinar o conhecimento utilizado pelo segundo no processo de resolução de problemas (Giorno et al,1988).

O tempo tomado e os problemas encontrados dependem da complexidade do domínio e do tamanho do problema tomado. Alguns métodos funcionam bem em certos domínios e não em outros. As técnicas de eliciação são classificadas em função do tipo de conhecimento a ser adquirido, a saber (Ferreira,1991):

1. Técnicas de propósito geral :- onde são levantadas questões de caráter genérico sem um objetivo específico, apenas para a familiarização dos conceitos. Entre outros estão:
 - entrevista,
 - questionário,
 - método das hipóteses terminais - HITE (Gonçalvez,1986),
 - lista de fatos - LIFA (Gonçalvez,1986),
 - codificação por lista (Rocela et al,1987),
 - método de "brainstorm" adaptado - BRADA (Gonçalvez,1986)

2. Técnicas para determinação de objetos, estruturas e relações . Entre outros estão:

- grades de repertório (Kelly,1955) (Boose,1985) e (Gammack,1987),
- desenho de curvas fechadas (Schiavini,1989),
- traçado de diagramas (Schiavini,1989),
- árvores por recordação (Schiavini,1989),
- criação e separação por grupos (Gammack,1987),
- agrupamento hierárquico de Johnson (Schiavini,1989)

3. Técnicas para descrição de procedimentos e tarefas. Entre outros estão:

- análise de protocolo e análise por interrupção (Waterman & Newell,1971),
- observação da realização da tarefa (Hoffman,1989)

2.2.7 Representação do Conhecimento

Uma das áreas mais importantes da Inteligência Artificial é a representação do conhecimento. A representação do conhecimento se dá na combinação de estruturas de dados e de procedimentos explicativos, que se utilizados da forma correta por um programa, o levará a um comportamento inteligente (Barr & Feigenbaum,1982).

A escolha da representação do conhecimento é um fator muito importante para o sucesso do sistema especialista, dado que pode afetar o número de estados do problema domínio e assim aumentar ou diminuir a complexidade do problema. As bases para se fazer a escolha da apropriada forma de representação ainda é fracamente entendida (Schutzer,1987). Não existe uma metodologia para, sistematicamente, iniciando a representação de um conhecimento de um dado problema - sucessivamente refinar e melhorar a representação. Pelo contrário, a seleção satisfatória da forma de representação parece basear-se no ganho de experiência de problemas similares e na experiência de se usar as várias e diferentes formas de representação (Hayes-Roth,1983). O trabalho com representação do conhecimento em Inteligência Artificial tem trazido o desenvolvimento de muitas classes de estruturas de dados para armazenamento de informação em computadores, assim como o desenvolvimento de procedimentos que permitem uma manipulação inteligente dessas estruturas de dados para fazer inferências (Deutch-McLeish,1993). Atualmente, formas mais importantes de representação do conhecimento para sistemas especialistas são (Chandresakaran,1993):

- **Lógica** :- cálculo proposicional e cálculo predicado,
- **Regras de produção,**
- **Redes semânticas,**
- **Quadros e roteiros** (das palavras inglesas “frames” e “scripts”).

Dado que os Sistemas Especialistas manipulam uma expressiva quantidade de conhecimento especializado e distinto, alguns aspectos fundamentais devem ser levados em consideração na análise das técnicas para a representação do conhecimento (Ferreira,1991) e (Favilla,1988):

1. **Adequabilidade lógica** :- o formalismo adotado deve ser capaz de representar o conhecimento a ser expresso.
2. **Poder de representação heurística** :- considerando-se que existem sintaxe e semântica definidos, devem existir meios de utilizar representações previamente definidas para a solução dos problemas do domínio.
3. **Conveniência de notação** :- as expressões produzidas devem ser fáceis de ler e escrever e deve ser possível entender os significados sem o prévio conhecimento de como o computador irá processar as informações.
4. **Manuseabilidade** :- capacidade de manipular as estruturas de representação existentes, para obter novas estruturas que corresponderão ao conhecimento inferido.
5. **Crescimento** :- é importante que a estrutura de conhecimento seja capaz de incorporar informações adicionais, que aumentem a eficiência da máquina de inferência.

Descreve-se a seguir as formas mais comuns de representação do conhecimento em Sistemas Especialistas:

2.2.7.1 Lógica

Historicamente o desenvolvimento e o uso de sistemas formais de lógica para a investigação da natureza do raciocínio e do conhecimento tem tido uma importante participação no pensamento ocidental. O desenvolvimento de uma matemática lógica mais formal iniciou-se com os trabalhos de Boole, Frege e Russell; e foi estendido no século XX por Quine, Carnap e Tarski (Kirsh,1992).

Lógica de primeira ordem e cálculo predicado são atualmente o sistema lógico matemáticos mais utilizados. Ambas as lógicas são relativas a forma (ou sintaxe) das declarações e a determinação da verdade através da manipulação sintática dessas bem formadas fórmulas de lógica (Burton,1993).

O conceito mais fundamental da lógica é o da verdade. Uma declaração (ou proposição) apropriadamente formada tem somente um de dois possíveis valores: verdadeiro ou falso (Kattan,1994). A lógica proposicional adiciona a esse conceito de verdade vários conectivos, ou operadores lógicos. Entre eles:

- o e lógico, em geral denotado pelo símbolo & chamado de conjunção de duas proposições lógicas
- o ou lógico, em geral denotado pelo símbolo v e chamado de disjunção de duas proposições lógicas
- a negação, ou não conectivo, em geral denotado pelo símbolo !
- a implicação conectiva, em geral denotado pelo símbolo \rightarrow
- o conectivo equivalente, em geral denotado pelo símbolo = ou \leftrightarrow

Esses conectivos são definidos pela tabela verdade abaixo, extraída do “Handbook of Artificial Intelligence” editado por A.Barr e E.Feigenbaum.

Tabela 4. Tabela verdade					
P	Q	P&Q	$P \rightarrow Q$! P	$P \leftrightarrow Q$
v	v	v	v	f	v
v	f	f	v	f	f
f	v	f	v	v	f
f	f	f	f	v	v

Nota: Onde v está para verdade, f para falso e, P e Q representam duas proposições.

No cálculo proposicional, encontramos as primeiras regras de inferência. Uma regra de inferência permite a dedução de novas sentenças de sentenças previamente dadas. A mais conhecida regra de inferência, *modus ponens* declara que, se conhecemos que duas sentenças na forma p e $p \rightarrow q$ são verdadeiras, podemos inferir que a sentença q também é verdadeira. Logo, *modus ponens* permite-nos substituir a sentença p e $p \rightarrow q$ pela simples declaração q , eliminando-se o conectivo \rightarrow

Cálculo predicado é uma extensão da lógica proposicional que proporciona ser utilizada em Inteligência Artificial permitindo a especificação de objetos, a postulação de relações entre objetos, e a generalização dessas relações sobre classes de objetos (Turnban,1988). Declarações sobre objetos, sejam sobre si próprios ou em relação a outros objetos são chamados *predicados*. Um predicado é aplicado a um número específico de argumentos e tem um valor verdadeiro ou falso. Considere os seguintes exemplos a seguir:

O predicado *cor* é escrito como:

(cor objeto valor)

ou aternativamente como:

cor(objeto,valor)

Constitui-se de um predicado de dois argumentos que é verdadeiro se o objeto tem o valor associado a sua cor. Mais especificamente,

(cor bola vermelho)

é um predicado que é verdadeiro se o objeto, uma bola, tem o valor da cor vermelho.

Predicados podem ser logicamente conectados. Por exemplo, o predicado:

(ou(cor bola vermelho) (cor bola branco))

é verdadeiro se a bola tem a cor valor vermelho ou a cor valor branco.

Lógica de primeira ordem estende o cálculo predicado introduzindo a noção de *operadores* ou *funções*. Funções, como predicados ou funções verdade, tem um número fixo de argumentos. Diferente dos predicados, entretanto, eles não tem somente os valores verdadeiro ou falso, mas podem também retornar objetos relacionados aos seus argumentos para o qual a função é verdadeira. Cada um dos argumentos de uma função podem ser uma variável, uma constante, ou uma outra função. De fato, uma função pode incluir-se a si mesma - uma circunstância chamada *recursão*, por exemplo, a função definida como:

(pai (pai (Carlos)))

seria o avô paterno de Carlos (Schultzer,1987).

Algumas características atrativas da representação do conhecimento através da lógica são (Barr & Feigenbaum,1981):

- é em geral uma forma natural de se expressar o conhecimento sobre um problema,
- é precisa. Utiliza métodos padronizados de expressão,
- tem a flexibilidade para representar um fato particular de uma forma única sem ter que considerar seus possíveis usos,
- é modular. Asserções lógicas podem ser colocadas independentemente umas das outras e podem crescer incrementalmente.

Suas desvantagens são (Keuneke,1991):

- obstrue severamente especificações do usuário e controle sobre a porção procedural do solucionamento do problema. Por causa do esquema separado da representação e do processamento, o usuário não pode facilmente direcionar o sistema baseado em lógica em como usar os fatos acumulados e como proceder com a solução do problema.
- é incapaz de expressar incerteza e relações aproximadas.

2.2.7.2 Regras de Produção

Desde a proposição por Prost em 1943, sobre regras de produção como um mecanismo computacional geral, a metodologia tem sido um objeto de grande desenvolvimento e vem sendo aplicada em uma diversificada coleção de problemas (Berghel & Roach,1993).

Um sistema de regras de produção pode ser visto como composto de três componentes básicos (Buchanam & Shortliffe,1985): um conjunto de regras, um banco de dados e um interpretador de regras.

- **Regras** :- Em seu modo mais simples, uma regra é um par ordenado de conjunto de dados (da palavra inglesa "string"), com um lado esquerdo e um lado direito. São sentenças apresentadas na forma de pares do tipo condição-ação, por exemplo na forma: *SE essa condição existir, ENTÃO execute aquela ação* . Um lado da regra é avaliado como referência ao banco de dados, e se ocorrer o sucesso (i.e., de alguma forma for considerada como VERDADE), a ação especificada é tomada como uma operação passiva de "percepção" ou "uma operação envolvendo somente a identificação e a detecção" (Newell &

Simon,1972), enquanto que a ação é consistida geralmente de uma ou mais operações. Regras podem ser representadas através da seguinte notação:

$$F \& B \rightarrow Z,$$

onde a seta indica as partes SE e ENTÃO da regra. Uma possível interpretação da representação acima poderia ser - SE: ambas as situações A e B ocorrerem ENTÃO: execute a ação Z

Um conjunto de regras de produção específicas sobre uma área constitui uma base de conhecimento sobre esse domínio, sendo que tanto o conhecimento factual quanto o conhecimento heurístico são descritos através de regras do tipo condição-ação.

- **Banco de dados** :- Nos sistemas que exploram os aspectos do processamento simbólico da cognição humana, o banco de dados é interpretado como um modelamento do conteúdo de algum mecanismo de memória com cada símbolo representando um dado conhecimento - uma coleção de símbolos. Para sistemas baseados em conhecimento, o banco de dados contém fatos e afirmações sobre o domínio, é tipicamente de tamanho arbitrário e, em princípio, não tem restrições quanto a complexidade na sua organização. Por exemplo, o sistema MYCIN utiliza uma coleção de quádruplas, consistido de uma tripla associativa e um fator de certeza - FC, que indica, em uma escala de (-1 a 1) o grau de certeza do fato confirmado ($FC > 0$) ou desconfirmado ($FC < 0$):

(identidade	organismo-1	E.Coli	0,8)
(local	cultura-2	sangue	1,0)
(sensitivo	organismo-1	penicilina	-1,0)

Qualquer que seja a organização do banco de dados, uma importante característica que deve ser notada é que o banco de dados é a forma e o meio de armazenamento das variáveis em qualquer estado do sistema.

- **Interpretador** :- Em sua concepção mais simples, o interpretador pode ser visto como uma sequência repetitiva (da palavra inglesa "loop"). A sequência consiste em procurar o lado esquerdo de cada regra até que seja encontrada aquela que com sucesso identifique-se com a informação do banco de dados. Nesse ponto, os símbolos identificados no banco de dados são substituídos por aqueles no lado direito da regra; e a procura continua ou com a próxima regra ou inicia-se outra vez com a primeira.

O ciclo de controle básico do sistema pode ser quebrado em duas fases chamadas:- *reconhecimento e ação*. E dá-se através das etapas:

1. verificar a validade das premissas das regras com os dados presentes na base de dados.
2. se existir mais de um regra que possa ser disparada, então escolher uma delas utilizando critérios estabelecidos para o desempate - Resolução de Conflito.
3. aplicar a regra escolhida, alterando se necessário os objetos presentes na base de dados.

Se o processamento ainda pode prosseguir então retorne ao passo (a).

A fase de reconhecimento envolve a seleção de uma única regra para execução e pode ser subdividida em:- *seleção e resolução de conflito* No processo de seleção, uma ou mais potenciais regras são escolhidas de um conjunto e passadas ao algoritmo de resolução de conflito que escolhe uma delas. A **Resolução de Conflito** requer alguns critérios para a escolha de uma única regra desse conjunto chamado - Conjunto de Conflito. Muitos critérios tem sido sugeridos incluindo (Hieb et all,1992):

- **Ordem da regra**:- existe um completo ordenamento de todas as regras do sistema, e a regra do conjunto de conflito com a maior prioridade é a regra escolhida.
- **Ordem do dado**:- os elementos da base de dados também são ordenados, e a regra escolhida é aquela que se identifica com o elemento de maior prioridade no banco de dados.

- Ordem geral:- a regra mais específica é a escolhida.
- Regra de precedência:- a rede de precedência determina a hierarquia.
- Ordem de tempo:- a regra mais recentemente executada ou a regra contendo o mais recente elemento de atualização da base de dados é a escolhida.

Existem dois importantes modos de utilização das regras no sistema de produção. Um é chamado Encadeamento para Frente e outro Encadeamento para trás:

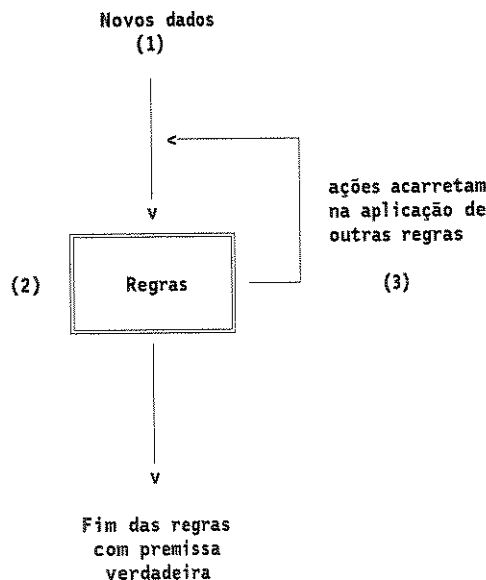
- Encadeamento para frente:- O encadeamento para frente é orientado a dados. As regras de encadeamento para frente são exaustivamente executadas baseadas nos dados conhecidos. E as principais características do encadeamento para frente são (Trinzic,1994):

- é orientado a dados,
- analisa as regras a partir de sua premissa,
- o processo é continuado até que todas as possíveis ocorrências sejam inferidas.

O encadeamento para frente é disparado quando novos dados tornam-se conhecidos, ou seja novos valores são atribuídos aos dados, daí a utilização do termo: *orientado a dados*. Durante o processo de encadeamento para frente, a máquina de inferência processa as regras na medida em que suas premissas tornam-se verdadeiras. A execução das regras, por sua vez, inferem outros novos dados. E assim o processo de encadeamento continua até que todo o possível conhecimento tenha sido inferido e todas as possíveis conclusões sejam alcançadas.

A execução do encadeamento para frente dá-se como na ilustração do ciclo a seguir:

1. entrada de dados novos
2. disparo de regras quando suas premissas são verdadeiras,
3. ações das regras atribuem novos valores aos dados,
4. execute o passo (2).



O exemplo a seguir ilustra uma sequencia típica de encadeamento para frente.

- SE
um motorista está designado E
(1) um operador de empilhadeira está designado E
um encarregado está designado
ENTÃO
todos os funcionários do segundo turno foram designados
- (2)
- SE
todos os funcionários do segundo turno foram designados E
(3) o dia é sábado E
foi feita uma provisão para trabalho em horário extraordinário
ENTÃO
o trabalho pode começar

1. a primeira regra dispara quando sua premissa torna-se verdadeira.
2. a atribuição de um valor na ação da primeira regra na base de conhecimento acarreta que a premissa da segunda regra torne-se verdadeira.
3. se as outras partes da premissa da segunda regra tornarem-se verdadeira, então dispara-se a segunda regra.

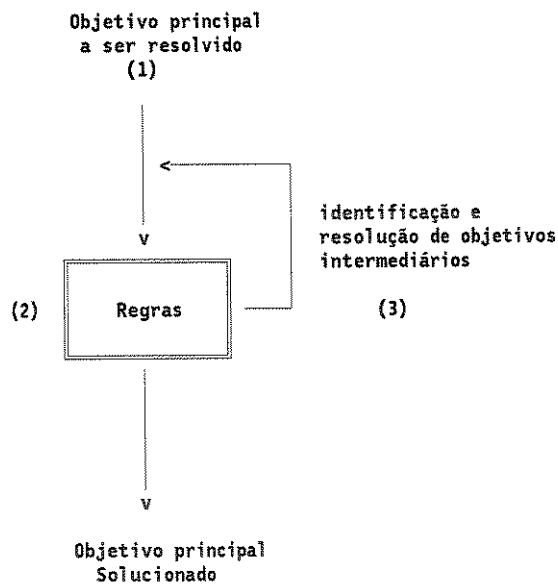
O mecanismo de inferência seleciona para a resolução de conflito todas as regras da base de conhecimento cujas premissas são verdadeiras, o processo continua enquanto houver regras que possam ser disparadas. Dessa forma, ao final do processamento, todos os dados que poderiam ser inferidos em função dos dados conhecidos estarão presentes na base de dados.

- Encadeamento para trás:- No encadeamento para trás a máquina de inferência tem um objetivo previamente estabelecido a ser determinado, e trabalha para trás através das regras para determinar valores que possam resolver o objetivo. As principais características do encadeamento para trás são (Trinzic,1994):

- é orientado a objetivo,
- analisa as regras a partir de sua ação,
- somente as informações necessárias são acessadas da base de conhecimento.

O ciclo de execução da máquina de inferência no encadeamento é:

1. identifica o objetivo,
2. procura pela regra que resolva o objetivo na sua cláusula de ação,
3. identifica objetivos intermediários, ("subgoals") nas premissas das regras,
4. para cada objetivo, obtém um resultado: seja o objetivo resolvido ou não. Faça perguntas e/ou acione regras adicionais (passo 2) necessárias para tentar resolver cada objetivo.



O exemplo a seguir ilustra uma sequência típica de encadeamento para frente.

- SE
 um motorista está designado E
 (3) um operador de empilhadeira está designado E
 um encarregado está designado
 ENTÃO
 todos os funcionários do segundo turno foram designados
- (2)
- SE
 todos os funcionarios do segundo turno foram designados E
 (1) o dia é sábado E
 foi feito uma provisão para trabalho em horário extraordinário
 ENTÃO
 o trabalho pode começar

1. um objetivo (o trabalho pode começar ?) é colocado para a máquina de inferência. Se nenhuma solução final é encontrada na base de conhecimento, a máquina de inferência procura por uma regra cuja ação corresponda ao do objetivo a ser resolvido.
2. se alguma das partes da premissa da regra selecionada é desconhecida para ser verdadeira, a máquina de inferência procura por regras adicionais cujas ações possam resolver esses novos objetivos intermediários ("subgoals").
3. se a premissa de uma regra que a máquina de inferência está testando for verdadeira, a regra é aplicada, e adiciona os novos dados à ação da regra.

O processamento é finalizado assim que o objetivo ficar determinado, ou quando for possível concluir que o seu valor não pode ser inferido a partir das informações presentes no sistema.

Apesar da representação por Regras de Produção constituir-se basicamente de procedimentos, o formalismo das regras de produção apresenta várias vantagens dos esquemas declarativos, principalmente a modularidade (Cohen et al,1987). E a simplicidade da alternativa aliada ao modo bastante natural para se representar as estratégias de resolução de um grande número de problemas fez com que a grande maioria dos Sistemas Baseados em Conhecimento implementados utilizassem o sistema de regras de produção como a estrutura básica de representação do conhecimento (Germond & Niebur,1993).

Dessa forma, as vantagens imediatas associadas aos sistemas de regras de produção incluem (Bruchanam & Shortliffe,1985):

- modularidade:- regras de produção podem ser adicionadas, alteradas ou eliminadas independentemente umas das outras e não dependem da ordem.
- uniformidade:- regras de produção impõe uma estrutura uniforme no conhecimento baseado em regras, tornando a informação fácil de ser entendida por terceiros.
- naturalidade:- regras de produção provêm uma sintaxe muito adequada para exprimir certos tipos de conhecimento.

2.2.7.3 Redes Semânticas

Rede semântica foi originalmente desenvolvida por Quillian em 1968, para ser utilizado como modelo psicológico da memória associativa humana. A rede semântica é um grafo direcionado onde a informação é representada por um conjunto de pontos chamados **nós** conectados por ligações chamadas **arcos** que descrevem as relações entre os nós (Tsutsui,1993). É definida como uma tripla: (NC, AC, IC) , onde NC está para o conjunto de nós, AC para a categoria de arcos e IC é a categoria de restrições para cada elemento de AC, e existe uma função:

$$f = NC \times NC \rightarrow (\text{verdade, falso})$$

de forma que $f(n_i, n_j)$ é verdade se existe um arco de classe f entre n_i e n_j , e falso em caso contrário.

1. Os nós da rede semântica podem representar objetos, conceitos e eventos. Os objetos podem ser físicos, que podem ser vistos e tocados, como também entidades como atos, eventos ou categorias abstratas.
2. Os arcos podem ser definidos em uma variedade formas, dependendo do tipo de conhecimento a ser representado. Arcos comuns utilizados para a representação de hierárquias incluem ligações do tipo:
 - **é_um**, são geralmente usados para representar a relação entre classes e instâncias.
 - **composto_por**, identificam nós que são propriedades de outros nós, mostrando uma relação de parte e subparte ou conjunto e subconjunto.

Muitos conceitos da Teoria de Grafos foi importada para a Ciência da Computação e para a Inteligência Artificial para descrever estruturas de dados. Na engenharia do conhecimento, grafos simples são usados para representação de estruturas espaciais e temporais, ou relações ainda mais abstratas como as relações causais (Block et al,1993).

A utilização de redes semânticas na representação do conhecimento apresenta de imediato as seguintes vantagens (Yasi & Ziarko,1988):

1. a representação é muito próxima da linguagem natural,
2. emprega um número de conceitos abstratos que ajudam a dar mais significado sobre os objetos observados,
3. pode ser usado para obter implementações mais eficientes,
4. pode ser estendido de forma que tenha o mesmo poder de expressão da lógica de predicados.

Uma representação de uma Rede Semântica é mostrada na Figura 5.

As relações do tipo **é_um** estabelecem a importante propriedade de herança hierárquica em uma rede. O que significa que os itens inferiores da rede herdam propriedades dos itens superiores da rede, ou seja, a herança de propriedades significa que são assumidas que as instâncias de uma classe tem todas as propriedades de uma classe mais geral da qual são membros. Entre outros benefícios, esse fato permite uma economia de espaço, dado que a informação sobre nós similares não tem que ser repetidos a cada nó, ao contrário, pode ser armazenada em uma localização central (Thakore,1993).

Por exemplo, na rede semântica do navio da Figura 6, as partes do navio como motor, caldeira e leme são armazenadas no mesmo nível do navio, ao invés de repeti-los nos níveis inferiores para cada tipo de navio ou para um navio em particular.

É útil imaginar as redes semânticas utilizando-se uma notação gráfica como mostrado, embora esteja claro que elas não podem ser representadas em programa com essa forma. Normalmente são representadas utilizando algum tipo de estrutura de memória de valor de atributo. Assim, por exemplo, na linguagem

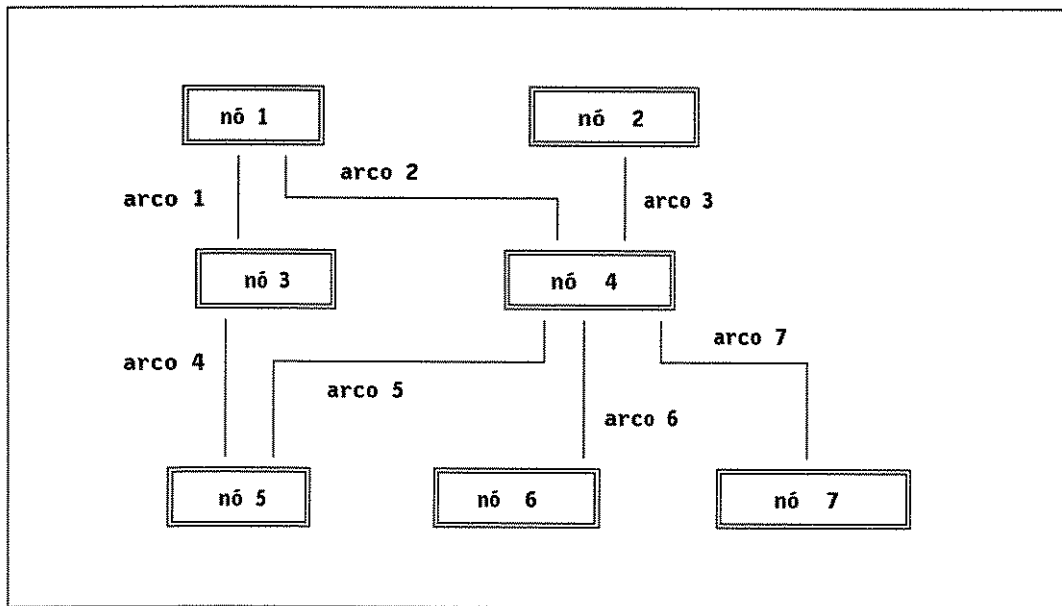


Figura 5. Representação de uma rede semântica. (Waterman,1985)

LISP, cada nó seria um átomo, os elos seriam propriedades e os nós nas outras extremidades dos elos seriam os valores.

2.2.7.4 Quadros e Roteiros

Quadros e roteiros organizam o conhecimento em objetos prototipados e eventos estereotipados, apropriados a situações específicas (Schulzter,1987). A organização desse conhecimento facilita os processos orientados a expectativa. Evidências psicológicas sugerem que as pessoas utilizam uma grande coleção de conhecimentos de experiências prévias para interpretar novas situações em atividades cognitivas cotidianas (Keuneke,1991).

Como as redes semânticas, os quadros são um outro mecanismo de representação do conhecimento. Eles partilham a noção de que entidades complexas podem ser descritas como uma coleção de atributos e valores associados. Assim também muitas vezes são denominadas de *estruturas de preenchimento*. Existem muitas evidências de que as pessoas não analisam situações novas “á partir do zero” e depois constróem novas estruturas de conhecimento para descrevem essas situações. Em vez disso, tem disponíveis na memória, uma grande coleção de estruturas representando sua experiência anterior com objetos, locais, situações e pessoas. Para analisar uma nova experiência elas evocam estruturas apropriadamente armazenadas e depois as preenchem com detalhes do evento atual. O mecanismo geral projetado para a representação computacional de tal conhecimento comum é o quadro (Krish,1991).

A palavra quadro (da palavra inglesa - “frames”) foi aplicada a uma série de estruturas de representação de preenchimento, em sua maior parte seguindo a teoria apresentada em (Minsky,1975) e discutidas em (Kuipers,1975). Em um certo sentido, essas estruturas podem ser vistas como complexas redes semânticas; contudo terão muita estrutura interna tipicamente projetada para torná-las úteis em tipos específicos de tarefas para solução de problemas (Rich,1988).

Um quadro é composto por uma rede de nós e relações hierarquicamente organizadas, onde os nós superiores representam os conceitos gerais e os nós inferiores instâncias mais específicas desses conceitos.

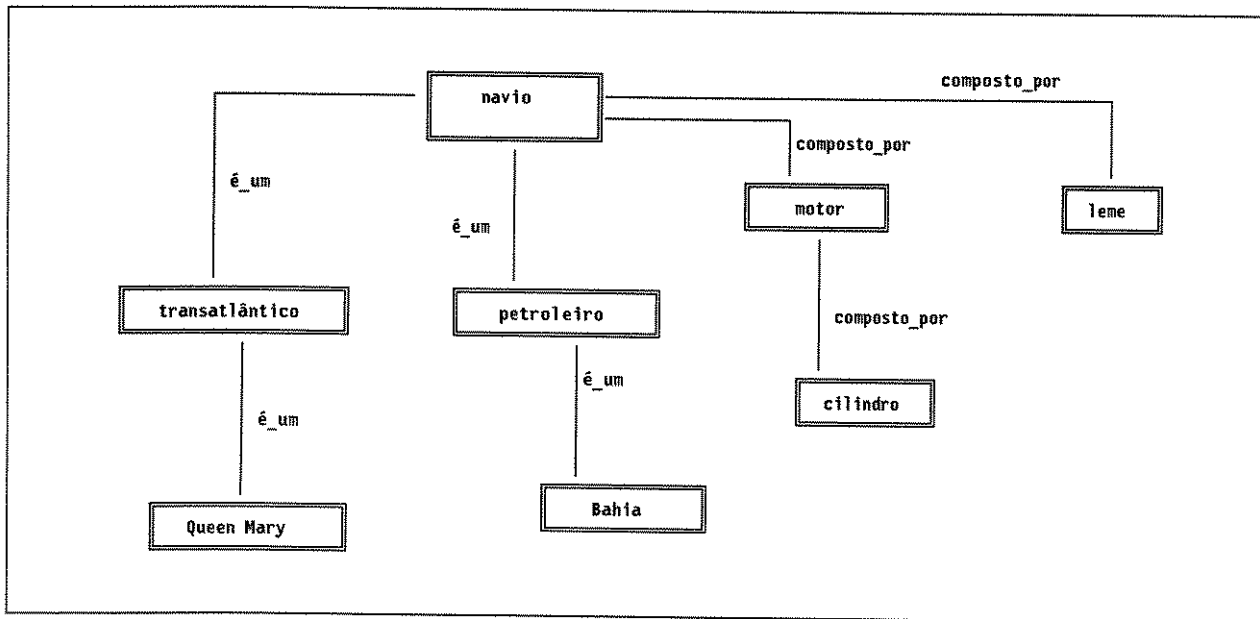


Figura 6. Exemplo de uma rede semântica. (Golçalves,1986)

Basicamente um quadro é uma estrutura de dados que inclui informações procedurais e declarativas em relações pré-definidas. E tipicamente, um quadro, descreve uma classe de objetos.

No sistema de quadros uma maneira de se representar o conceito de cada nó é fornecer algum tipo de memória associativa onde são armazenadas triplas ordenadas, normalmente consideradas na forma: *objeto x atributo x valor*. A informação pode ser recuperada de uma memória especificando valores para dois campos quaisquer. Todas as triplas com os valores especificados são encontradas e o conjunto de valores que ocorrem no campo restantes daquelas triplas é devolvido.

A tripla associativa pode ser interpretada como: *objeto tem atributo com valor*. O nome geral das triplas associativas é uma característica do domínio do problema, e atributo é um nome geral do conjunto de valores. Portanto a descrição de um objeto é uma combinação única dos valores dos atributos, no conjunto de todas as possíveis descrições - espaço de descrição. Assim por exemplo poderíamos ter os atributos: nome, cor e tamanho com valores como: João, branco e grande. Os atributos nos quadros são chamados de escaninho (da palavra inglesa "slot"), assim, no sistema de quadros o conceito de quadros pode ser visto como uma coleção de atributos (Favilla,1988).

Cada escaninho pode ter informações na forma de procedimentos anexados a ele, que são executados quando a informação no escaninho (valor dos atributos) é alterada. Dessa forma tem-se que escaninhos podem ter qualquer número de procedimentos anexados a ele e três tipos de procedimentos que com frequência são anexadas as esses escaninhos são (Waterman,1985):

1. *procedimento se_adicionado* :- executado quando nova informação é colocada em um escaninho.
2. *procedimento se_removido* :- executado quando alguma informação é apagada do escaninho.
3. *procedimento se_necessário* :- executado quando uma informação é necessária de um escaninho, mas o mesmo está vazio.

(Schank & Abelson,1977) projetaram uma estrutura semelhante aos quadros, um tipo especializado de quadro que são usados para representar eventos estereotipados chamados de - roteiros (do ingles "script"), para assisti-los no desenvolvimento de sistemas capazes de entender estórias e textos. Por exemplo, quando ouvimos uma estória sobre uma pessoa envolvendo um restaurante, antecipamos que serão ouvidas coisas como: cardápio e garçons. E quando ouvimos que essa pessoa pediu uma refeição e mais tarde deixou o restaurante, concluímos que aquela pessoa comeu a refeição e pagou a conta, mesmo que não as ouçamos explicitamente.

O exemplo a seguir ilustra de forma representativa o parte do esquema de quadros e roteiros utilizados no entendimento de estórias sobre cenas em restaurante (Barr & Feingenbaum,1981):

Quadro restaurante genérico

Especialização_de: estabelecimento de negócio

Tipos:

- .variação: (de_auto_serviço, lanchonete, aguarde_para_ser_acomodado)
- .padrão: aguarde_para_ser_acomodado
- .se_necessário: SE balcão de plástico ENTÃO lanchonete
SE pilha_de_bandejas ENTÃO de_auto_serviço
SE há avisos aguarde_para_ser_acomodado ou
peça_reservas ENTÃO aguarde_para_ser_acomodado
CASO CONTRÁRIO acomode-se

Localização:

- .variação: endereço
- .se_necessário: (olhe o cardápio)

Nome:

- .se_necessário: (olhe o cardápio)

Estilo:

- .variação: (chines, francês, frutos_do_mar, pizza)
- .padrão: pizza
- .se_adicionado: (atualizar alternativas de restaurante)

Tipo de Operação:

- .variação: horário de funcionamento
- .padrão: aberto à noite exceto as segundas-feiras

Forma de Pagamento:

- .variação: (dinheiro, cartão_de_crédito, cheque, vale_refeição)

Sequência de Evento:

- .padrão: script comendo_no_restaurante

Alternativas:

- .variação: todos os restaurantes com o mesmo estilo de comida
- .se_necessário: (encontre todos os restaurantes com o mesmo estilo de comida)

Anexo a cada quadro existem vários tipos de informações, incluindo a forma de se usar o quadro, o que pode ser esperado a seguir, e o que fazer se as expectativas não forem cumpridas.

2.2.8 Avaliação dos Sistemas Especialistas

Sistemas especialistas são avaliados, primeiramente, para se testar a precisão e a utilidade do sistema (Burton,1993). Sendo que o processo de avaliação também é crucial para a melhoria do projeto e do desempenho do sistema especialista (Bradac,1994). A avaliação pode ser considerada como um agrupamento de termos que cobrem a:- verificação, validação e teste de aceitação do usuário (Adrion et al,1982).

- **verificação**:- de acordo com a engenharia de software convencional, a verificação é o método que determina se o sistema implementado satisfaz completamente as especificações (Adrion et al,1982). Nos sistemas baseados em conhecimento, cada componente deve ser analisado a nível de consistência interna e a nível de conjunto. Como consistência interna tem-se que não devem existir informações conflitantes no sistema, ou que o sistema não forneça nenhuma informação conflitante (M.White & J.Goldsmith,1990). A procura por inconsistências pode ser feita pela verificação de:

1. regras redundantes:- duas regras são redundantes se ocorrem em uma mesma situação (premissa) e tomam as mesmas conclusões (ação).
2. regras conflitantes:- duas regras são conflitantes se ocorrem em uma mesma situação (premissa) com conclusões conflitantes (ação).
3. regras inclusas:- uma regra é incluída por outra se duas regras tem as mesmas conclusões (ações), mas uma contém restrições adicionais nas situações em que for disparada.
4. condições "IF" desnecessárias:- duas regras contém condições desnecessárias IF, se as regras tem as mesmas conclusões, porém a premissa em uma regra está em conflito com uma premissa da outra regra, sendo que as outras demais premissas das duas regras são equivalentes.
5. cadeia circular de regras:- um conjunto de regras é circular se o encadeamento do conjunto dessas regras forma um círculo.

E a verificação a nível de um conjunto da base de conhecimento, usualmente pode ser dada pelos seguintes indicativos:

1. valor de atributo não referenciado:- quando alguns valores de um conjunto de valores possíveis dos atributos de um objeto não são valorizados por nenhuma condição IF das regras.
 2. atribuição ilegal de valores:- quando uma regra refere-se a um valor de atributo que não pertence ao conjunto de tipos legais.
 3. premissas "sem saída":- se os atributos de um objetivo não podem ser providos pelo usuário, através da solicitação do sistema, ou o objetivo deve ser valorizado pela conclusão de uma das regras do conjunto de regras aplicadas ao objetivo, então o objetivo não pode ser atingido.
 4. objetivos "sem saída":- as premissas deve atender as condições o caso anterior (premissas "sem saída") de forma a evitar os objetivos "sem saída".
 5. conclusões inatingíveis:- a conclusão de uma regra deve atingir um objetivo ou atingir a premissa de uma outra regra do mesmo conjunto de regras. Se não houverem conclusões a mesma é inatingível.
- **validação**:- validação de sistemas de software, incluindo sistemas especialistas, é conduzido para se determinar, se o sistema implementado desempenha satisfatoriamente nas tarefas do mundo real, para o qual foi criado (Hollnagel,1989). Ou seja, o sistema correto foi construído ? ele desempenha em níveis aceitáveis ? As seguintes diretrizes gerais para validação são apresentadas em (M.White & J.Goldsmith,1990):
 - o que validar:- os resultados intermediários, conclusão, raciocínio, ou qualquer desses podem ser validados. Em qualquer dos estágios de desenvolvimento, se o desempenho de uma parte do sistema pode ser medida para um dado conjunto de entradas, então essa parte deve ser medida de forma a identificar e corrigir os erros antecipadamente.

- validar contra o que:- um sistema baseado em conhecimento pode ser validado tanto contra resultados conhecidos como contra o desempenho do especialista.
- validar com o que:- teste de casos podem ser utilizados. Idealmente os testes de caso devem cobrir a distribuição de todos os casos esperados, de forma que usuário final encontre-se desempenhando efetivamente um caso típico de sua tarefa.
- quando validar:- a validação deve ocorrer durante os pontos críticos das revisões de projetos.
- controlar o custo da validação:- a resposta para esse caso é renomear essa questão para: quais os riscos e custos incorridos da não validação, pois esse item está relacionado ao domínio da aplicação, por exemplo: recomendação financeira, diagnóstico médico.
- como controlar interferências:- o julgamento do desempenho de um sistema especialista não deve ser interpretado como conselhos do(s) desenvolvedor(es) do sistema. Dessa forma, idealmente, deve-se empregar um assessor independente que não tenha relações, preferências ou dependências no uso do sistema no ambiente em questão.
- como se resolver com múltiplos resultados:- suponha que um sistema de diagnóstico médico prescreva duas drogas para um mesma doença. O tratamento será válido para cada uma das drogas individualmente, porém a interação das duas drogas poderá ser danosa. O conselho do sistema é inválido. Idealmente uma análise estatística multivariada deveria ser utilizada.
- **aceitação do usuário:-** a aceitação do usuário é uma noção difícil para se definir, pois tende a encapsular os aspectos ergonômicos e organizacionais do sistema (Pierce,1990). E também está relacionada ao estudo da interação entre pessoas e máquinas. A aceitação do usuário é o motivo da tecnologia denominada de Fatores Humanos (da sentença inglesa “Human Factors”) que se preocupa com o relacionamento entre os sistemas e os seres humanos (Rubinstein & Hersh,1984). O termo “usabilidade” é com frequência usado para expressar a aceitação do usuário.

(Shackel,1986) tem proposto uma definição operacional de usabilidade incorporando quatro fatores: eficiência, aprendizado, flexibilidade e atitude. Eficiência é definida como desempenho acima de um dado grau (medido por exemplo em termos de velocidade e erros). Aprendizado refere-se a critérios de desempenho alcançado em um dado tempo, com base em um dado treinamento e suporte ao usuário. Flexibilidade é definida como a adaptação a uma determinada faixa de variação nas tarefas do usuário. E atitude refere-se ao nível de aceitação de determinados fatores como: fadiga, desconforto, frustração ou sucesso perante o sistema.

(Clegg et al,1988) sugere seis características de usabilidade a serem levadas em consideração na avaliação de usabilidade, que são:

1. facilidade de aprendizado:- fatores que contribuem para facilitar o aprendizado, e que se estendem ao entendimento dos conceitos chaves, a extensão do aprendizado e também o grau de consistência.
2. estar no controle:- é importante que os usuários se sintam no controle das interações. Os sistemas não devem se comportar de formas imprevisíveis. Por exemplo, se o sistema contém diferentes modos de utilização, esse fato deve ser facilmente distinguível na tela. E também os sistemas devem permitir que o usuário retorne aos primeiros passos da interação com o sistema.
3. grau de esforço:- que inclui o esforço físico e mental. Esforços excessivos e desnecessários devem ser evitados. É necessário assegurar que o número de informações a serem tecladas ou outras ações físicas não sejam excessivas e/ou grandes de se memorizar.
4. velocidade do sistema:- a maioria dos sistemas requer que o usuário aguarde um dado tempo, em algum ponto de interação. Não é somente a frequência e a duração dos atrasos/pausas que são importantes, mas também o quão previsíveis eles/elas o são. Por exemplo, um atraso/pausa é mais aceitável quando alguma mensagem é mostrada ao usuário informando o que está se passando e qual o tempo médio provável de processamento.

5. entrada e saída de informações:- é necessário certificar-se de que os dispositivos de entrada e saída são aqueles apropriados para a tarefa em questão. Não deve haver discrepâncias entre o modo pelo qual o sistema trabalha e a tarefa desempenhada pelo usuário. Dificuldades de entrada, por exemplo, podem surgir quando o sistema solicita informações em uma dada sequência ou formato incompatível com a fonte de informação do usuário. Problemas similares podem ocorrer com a saída de dados.
6. erros e correção de erros:- grande parte das pessoas cometem erros ocasionais. É necessário projetar sistemas que tenham características que não facilitem os erros e que permitam a correção dos mesmos facilmente quando ocorrerem. Inconsistência, arbitrariedade e ambiguidade são fatores que podem muito bem conduzir a erros. Há um estudo em andamento sobre erros humanos (Norman,1981) entre deslize e engano. Um deslize pode ser pensado como um erro durante a execução de uma intenção enquanto que um engano é um erro na formação de uma intenção. Muitos dos erros frequentes dos usuários de sistemas computacionais podem ser classificados como deslize.

A proposta acima foi feita para sistemas computacionais em geral. Em muitos aspectos os sistemas especialistas são menos claramente definidos que os sistemas de informação, seu conjunto é menos preciso, são abertos e necessitam ser capazes de responder as alterações do ambiente, manuseiam informações incompletas e imprecisas, não podem ser totalmente testadas e é difícil de se definir exatamente como se comportam após sua construção. Portanto os fatores particulares importantes de usabilidade podem variar para cada domínio e para cada aplicação (Burton,1993).

Tendo identificado os vários aspectos da usabilidade, o próximo passo é considerar quais os métodos a serem usados para avaliá-los. Como ponto de partida podem ser incluídos os questionários e conjuntos de telas propostos em (Clegg et al,1988) e (Ravden e Johnson,1989). Vários e diferentes métodos de avaliação são disponíveis, cada um com diferentes vantagens e desvantagens. Os métodos variam consideravelmente em termos de interrupções ao usuário e dos tipos de dados fornecidos.

Um conjunto de técnicas são descritos a seguir. Mesmo que estejam descritas em seções distintas, não devem ser pensadas como sendo mutuamente exclusivas. Pelo contrário, devem ser vistas como complementares, possibilitando à coleta de diferentes tipos de informações (Liebowitz,1986):

- **entrevistas:-** O entrevistador deve preparar uma lista de questões consideradas fundamentais e também preparar-se para possíveis desvios ou complementações quando necessário. Entrevistas podem servir para uma variedade de propósitos. E podem ser usadas como uma técnica preliminar a outros métodos específicos de obtenção de informações mais profundas. Entrevistas são geralmente conduzidas em base individual, entretanto é possível que seja realizada em grupo.
- **questionários:-** Existem dois tipos básicos de questões: nomeadas e fechadas, e abertas. Para questões nomeadas e fechadas duas ou mais possibilidades de respostas são fornecidas. Podem ser alternativas simples como: “sim” ou “não”, ou podem ser de múltipla escolha com um leque de opiniões do tipo: “muito insatisfeito”, “moderadamente insatisfeito”, “nem insatisfeito nem satisfeito”, “moderadamente satisfeito” e “muito satisfeito”. Para cada alternativa pode haver um peso associado de forma a permitir um sistema de classificação ou cálculo percentual. Para o caso de questões em aberto nenhuma resposta é sugerida ao usuário. As pessoas simplesmente são questionadas com perguntas como: “Quais as partes do sistema são as mais difíceis ?”
- **revisões de sistemas:-** Nesse método um avaliador trabalha em algumas tarefas selecionadas no sistema, atentando-se a ver o conjunto: programa e máquina, da perspectiva do usuário. Sugere-se que esse método seja utilizado em novos sistemas ou sistemas protótipos, onde por definição não há usuários com experiência direta.
- **observação formal:-** O avaliador pode observar o usuário diretamente e tomar notas detalhadas ou completar um “checklist”, ou mesmo observar o usuário através de um espelho unidirecional. Sendo que esse último método tem a vantagem de causar menos interrupções ou desconforto ao usuário.

- **diário do usuário**:- O relato em um diário pode fornecer oportunidade de se detectar problemas de longo termo. Ele pode destacar problemas raros ou aqueles que ocorrem em um determinado período ou padrão de tempo. O usuário pode ser solicitado a fornecer vários tipos de informações, como por exemplo, detalhes de um erro, detalhes muito específicos, ou dificuldades em geral. Em alguns casos o diários pode tomar a forma de um livro aberto, em outros casos uma forma estruturada pode ser a mais apropriada.
- **registro no sistema**:- Esse método preve que o sistema automaticamente registre as interações com o usuário final. Ele pode, por exemplo, simplesmente compilar a gravação que o usuário digitou e como o sistema respondeu. Ou pode ser programado para gravar os erros, tempos, e outras informações.

E como os sistemas de conhecimento são frequentemente utilizados para diferentes tipos de problemas e diferentes tipos de usuários (Berry & Hart,1990) sugerem que os sistemas especialistas também sejam avaliados de acordo com:

1. qualidade das decisões e conselhos do sistema,
2. nível de acerto das técnicas de raciocínio utilizadas,
3. qualidade da interação homem/máquina,
4. eficiência do sistema, e
5. efetividade de custo.

Apesar das dificuldades para se encontrar todos os critérios adequados para a avaliação desses itens.

Testes de sistemas baseados em conhecimento são extremamente importantes por causa de sua complexidade e da natureza de suas funções nas tarefas de tomada de decisão e de suportes a decisão. Assim sendo, um considerável número de trabalhos tem sido feitos nessa área (Gaschnig et al,1983). Nos últimos anos, a literatura sobre avaliação de sistemas especialistas tem expandido rapidamente e muitos métodos alternativos tem sido propostos para verificação, validação e teste de aceitação do usuário - (para pesquisas mais recentes, ver (Suen et al,1990) (Burton,1993).

A experiência tem demonstrado que sem um claro entendimento das necessidades do usuário e das especificações do sistema, os engenheiros do conhecimento ou os desenvolvedores de sistema podem falhar em prover as necessidades cruciais, e o sistema especialista resultante terá utilidade limitada (Berry & Broadbent,1987).

2.2.9 Áreas de Aplicação

Atualmente as áreas identificadas, que comumente utilizam-se dos Sistemas Especialistas, são (Germond & Niebur,1993):

- Agricultura
- Química
- Sistemas Computacionais
- Engenharia
- Geologia
- Gerenciamento de Informações
- Direito
- Fabricação
- Matemática
- Medicina
- Meteorologia
- Ciências Militares
- Física
- Controle de Processo
- Aeroespacial
- Projeto Mecânico
- Controle de Chão de Fábrica

A seguir são apresentadas alguns Sistemas Especialistas desenvolvidos com sucesso (Waterman,1986) (Ferreira,1991), nas seguintes áreas:

Química

- Interpretação
 - CRYSLIS: Infere estruturas tridimensionais de proteínas através do mapa de densidade de elétrons.
 - DENDRAL: Infere estruturas de compostos moleculares através dos espectros de massa e dos dados de reações nucleares.
 - TQMSTONE: Seleção fina de um quadripólo triplo de um espectômetro de massa (TQMS) através da interpretação dos sinais de TQMS.
- Projeto/Planejamento
 - CLONER: Auxilia biólogos moleculares a projetarem e criarem novas moléculas.
 - MOLGEN: Auxilia geneticistas moleculares a planejarem experiências com clones de gens.
 - SECS: Auxilia químicos a sintetizarem complexas moléculas orgânicas.
 - SPEX: Auxilia cientistas a planejarem complexas experiências em biologia molecular, a nível de laboratório.
 - SYNCHEM2: Sintetiza moléculas orgânicas complexas sem assistência ou auxílio humano.

Sistemas Computacionais

- Previsão
 - PTRANS: Auxilia o gerenciamento e distribuição de sistemas computacionais da DEC.
- Diagnose
 - BDS: Auxilia na localização de módulos problemáticos em redes com elevado número de interrupções de sinal.
 - DART: Auxilia na diagnose de falhas em computadores.
 - IDT: Auxilia na localização de unidades defeituosas em computadores VAX.
- Projeto
 - XCON: Configura computadores da família VAX.

Eletrônica

- Diagnose
 - ACE: Diagnostica falhas em redes telefônicas.
 - IN-ATE: Auxilia na diagnose de falhas em osciloscópios.
 - NDS: Auxilia na diagnose de falhas em redes de comunicação.
- Projeto
 - EURISKO: Auxilia no projeto de dispositivos microeletrônicos tridimensionais.
 - PALLADIO: Auxilia no projeto e teste de novos circuitos VLSI (Very Large Scale Integration).
 - REDESIGN: Auxilia no redesenho de circuitos digitais para atender a novas especificações.
 - TALIB: Sintetiza o lay-out de circuitos integrados para células do tipo nMOS.
- Instrução
 - CADHELP: Ensina a utilização de um subsistema de CAD para o projeto de circuitos digitais.
 - SOPHIE: Ensina a diagnosticar falhas em circuitos elétricos.

Engenharia

- Interpretação
 - REACTOR: Auxilia operadores a diagnosticarem e tratarem acidentes em reatores nucleares.
- Diagnose / Deputação
 - DELTA: Auxilia a identificação e correção de mau funcionamento em locomotivas.
 - SACON: Auxilia engenheiros a encontrarem estratégias para problemas de análise estrutural.
- Instrução
 - STEAMER: Ensina a operação de uma usina de propulsão a vapor.

Geologia

- Interpretação
 - ELAS: Auxilia geólogos utilizarem o INLAN, - um complexo e extenso programa de análise.
 - DIPMETER ADVISOR: Auxilia geólogos a interpretar os relatórios de inclinação.
 - LITHO: Auxilia geólogos na análise de relatórios sobre petróleo.
- Diagnose
 - DRILLING ADVISOR: Auxilia a diagnosticar e corrigir os problemas de aderência das brocas, quando da operação de descaixe.
 - HYDRO: Auxilia geólogos a utilizarem um programa de simulação de disponibilidade de água chamado de HSPF.
 - MUD: Auxilia a diagnose e tratamento de problemas relacionado com fluídos de perfuração.
 - PROSPECTOR: Auxilia geólogos a avaliarem o potencial mineral de uma região.

Medicina

- Interpretação
 - PUFF: Diagnostica infecções pulmonares através da interpretação de dados dos testes de funções pulmonares.
 - SPE: Diagnostica condições inflamatórias pela interpretação de dados do densitômetro - “scanning densitometer”.
 - VM: Monitora unidades de terapia intensiva - UTI, através da interpretação de dados dos equipamentos da UTI.
- Diagnose
 - ABEL: Auxilia a diagnose de desarranjos no sistema ácido-base e eletrolítica.
 - AI/COAG: Auxilia na diagnose de infecções de hemostase.
 - AI/RHEUM: Auxilia na diagnose de doenças de reumatologia clínica.
 - CADUCEOS: Auxilia a diagnose em medicina geral interna.
 - BLUE BOX: Auxilia na diagnose e tratamento de várias formas de depressão clínica.
 - CASNET/GLAUCOMA: Auxilia na diagnose e tratamento de doenças relacionados a glaucoma.
 - MYCIN: Auxilia na diagnose e tratamento de infecções bacterianas.
 - ONCOCIN: Auxilia o tratamento e acompanhamento de pacientes com câncer sob tratamento quimio-terápico.

Militar

- Interpretação
 - ADEPT: Desempenha situações de avaliação através da interpretação de relatórios dos serviços de inteligência.
 - ASTA: Auxilia a identificar tipos de sistemas de radar, através da análise dos sinais de radar.
 - HANNIBAL: Auxilia em situações avaliação através da análise de rádio comunicação.
 - HASP/SIAP: Detecta e identifica embarcações pela interpretação de dados provenientes de sonares.
 - RTC: Classifica navios pela interpretação de imagens de radar.
- Previsão
 - I&W: Auxilia prever quando e onde maior os próximos principais conflitos armados irão ocorrer.
- Projeto
 - ACES: Desempenha trabalhos de cartografia rotulando mapas.
- Planejamento
 - KNOBS: Auxilia no planejamento de missões de comando tático aéreo e centros de controle.
 - TATR: Auxilia o esquadrão da força aérea desenvolver planos de ataque nas linhas aéreas inimigas.

2.3 Sistemas Especialistas Voltados à Diagnose

2.3.1 Introdução

A diagnose tem sido uma aplicação emergente de maior interesse e uso dentro dos sistemas baseados em conhecimento (Bowen,1994). Sendo que os benefícios obtidos com esses sistemas como ferramentas de auxílio a diagnose de equipamentos são muitos (Krishnamurt & Phillips,1992). E as principais vantagens diretas são (Favilla,1988):

- maior disponibilidade do conhecimento; o conhecimento pode ser aplicado sem a presença do(s) especialista(s),
- maior consistência na análise do problema, pois evita que fatores que influenciam a ocorrência sejam omitidos durante o diagnóstico,
- auxílio na aprendizagem dos processos de diagnóstico, facilitado pelos mecanismos de explicação presentes na maioria desses sistemas, e pela disponibilidade do sistema para consultas independente da presença do especialista,
- retenção e aprimoramento dos conhecimentos adquiridos por meio de experiências práticas, através da constante evolução do sistema, evitando a perda do conhecimento quando os especialistas não estão mais disponíveis para aquela tarefa.

As vantagens indiretas são:

- maior precisão nos diagnósticos, diminuindo-se o número de erros, e
- menor dependência direta dos especialistas

Essas vantagens adicionadas as dificuldades na formação de especialistas em diagnose de equipamentos, à facilidade de implementação utilizando-se de ferramentas ou linguagens orientadas, e ao crescimento do número de aplicações na área, vem motivando fortemente a aplicação dos sistemas baseados em conhecimento à diagnose (Mussi & Morpurgo,1990).

2.3.2 Diagnóstico

Um dispositivo pode deixar de funcionar ou funcionar inadequadamente por diversos motivos, que vão desde a discrepância entre o projetado e o contruído, falhas em componentes, até a ocorrência de eventos externos como erro de operação, surtos de energia e outros (Chandresakaran,1993). O objetivo da diagnose é o de determinar a falha ou as falhas responsáveis, dado um conjunto de sintomas, especialmente aqueles sintomas que se desviam do esperado. E o diagnóstico de uma falha pode ser considerado completo se a descrição do mesmo contiver informações suficientes para a prova (Fiol & Nicolau,1993). Em alguns casos, esses sintomas podem indicar a ocorrência de diversas causas, tornando necessária a execução de testes, sendo que o propósito do teste de diagnóstico é o de obter informações suficientes para completar a diagnose. É claro que algumas falhas não são diagnosticáveis, pois não há um conjunto atingível e observável de dados que permitam distingui-los uns dos outros (Genesereth,1984).

Tipicamente, um diagnóstico inicia-se com a observação de algum comportamento reconhecido como um desvio daquele esperado ou desejado, i.e., observa-se um mau funcionamento do comportamento esperado. Uma análise das discrepâncias obtidas a partir de uma comparação entre o comportamento esperado e o comportamento observado é utilizada para determinar as falhas apresentadas pelo dispositivo (Sekine,1992). Esta abordagem segue um paradigma proposto por Davis e Hamscher,1988 e é baseada na interação entre a observação e a predição (expectativa). Sendo que a discrepância resultante entre ambos é

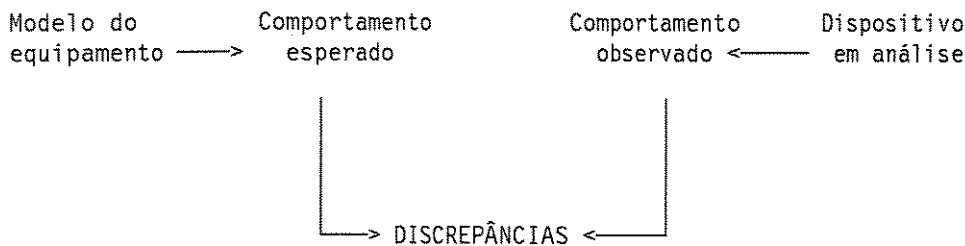


Figura 7. Paradigma de Davis e Hamscher.. (Davis & Hamscher,1988)

que levará a geração das hipóteses. O processo de raciocínio identifica essas discrepâncias, determina a razão da ocorrência e refina sua localização. A Figura 7 apresenta uma ilustração do processo de detecção de discrepâncias segundo o paradigma de Davis e Hamscher.

O solucionador, neste ponto, precisa gerar hipóteses sobre a possível, ou as possíveis causas do mau funcionamento, por exemplo uma hipótese de diagnóstico do tipo: *componente A e/ou conexão B estão incorretos e causam um conjunto de desvios observados*. Nesse estágio, é levantado um conjunto de hipóteses candidatas ou diferenciais mais plausíveis com base no conhecimento e sua organização, que levam a estratégias que possam ser empregadas na identificação das causas.

É útil visualizar a tarefa de diagnóstico como sendo composta das seguintes principais etapas (Giorno,1990):

1. **Geração de hipóteses** :- gera hipóteses candidatas baseadas nas informações disponíveis.
2. **Teste das hipóteses** :- reúne informações advindas de testes buscando a confirmação ou a rejeição das hipóteses geradas.
3. **Discriminação das hipóteses** :- reúne informações adicionais para a verificação da consistência entre as hipóteses geradas que restaram da etapa de teste.

O coração do processo de diagnose é o inter-relacionamento entre a geração das hipóteses e o teste dessas hipóteses (Fulda,1993).

2.3.3 Abordagens dos Sistemas Especialistas para Tarefas de Diagnose

Existem muitas dimensões para estratégias de diagnósticos. Porém a idéia central é a de que o conhecimento básico utilizado e sua organização conduzam às estratégias que deverão ser empregadas (Chanderasekaran & Milne,1985).

Um ser humano especialista em diagnose e manutenção de um particular dispositivo tem muitos e diferentes tipos de conhecimento a respeito de problemas:

1. como qualquer outro ser humano, o especialista terá o conhecimento do senso comum. O que inclui as propriedades físicas que regem o nosso mundo, por exemplo, quando um objeto é derrubado, ele cairá no chão. E também inclui os procedimentos básicos da melhor abordagem sobre um tipo particular de problema (Warkentin,1993).
2. um especialista também é possuidor do que se denomina de domínio do conhecimento fundamental, chamado na literatura de conhecimento "profundo", "causal", "funcional", ou "físico" (Fiol & Nicolau,1993). Esse é o conhecimento que o especialista adquire nos primeiros estágios do

desenvolvimento da "expertise". O que inclui o conhecimento básico relativo a como e porquê o dispositivo funciona da forma que funciona, bem como os procedimentos para a investigação de um problema (da palavra inglesa "troubleshooting"); obtendo dessa forma as informações de causa e efeito necessárias quando da diagnose de um mau funcionamento (Lirov,1993).

3. um terceiro tipo de conhecimento, com frequência referido na literatura como conhecimento: "superficial", "raso", "compilado", "empírico" ou "experimental", é conhecimento que determina a diferença entre um especialista e um novato (Sekine,1992). Esse conhecimento é baseado na experiência adquirida ao longo de repetidos processos de diagnose de problemas com o dispositivo e pode ser denominado de "domínio do conhecimento experimental". Ele fornece atalhos através do conhecimento, permitindo então que o especialista solucione os problemas rapidamente, e forma mais eficiente e precisa que faria um novato (Fink & Lusth,1987).

Obviamente que não existe uma linha distinta delimitando cada um desses conhecimentos. Eles são simplesmente pontos discretos em um contínuo do senso comum ao conhecimento experimental. Esse conhecimento contínuo com frequência tem uma forma dinâmica, por exemplo, uma dada experiência torna-se tão profundamente enraizada que passa a ser considerada de "senso comum" (Deutsch,McLeish et all,1993). Um especialista humano traz a tona todo o seu conhecimento quando diagnostica um problema e o utiliza de uma forma altamente integrada. Entretanto ele/ela poderá iniciar a diagnose tentando utilizar o conhecimento experimental dado que provê uma forma mais rápida e eficiente de solucionar um problema. Se nada da experiência se aplica, o especialista pode facilmente buscar entre outros diferentes tipos de conhecimento que julgar necessário e apropriado (Roth & Wood II,1993).

Por exemplo, na tentativa de se determinar porque um carro não funciona, um especialista que suspeita de que a bateria é o problema poderá tentar dar umas pancadas nos conectores e terminais dos cabos da mesma, com uma chave. Baseada na experiência de situações similares do passado, essa ação funcionou algumas vezes. Entretanto, a base dessa ação está no conhecimento de que os conectores dos cabos da bateria tendem-se a corroer e a oxidar, sendo que dar umas batidas com uma chave poderá prover uma melhor conexão. A conexão permitirá a eletricidade fluir, que por sua vez permitirá a partida do carro quando a chave tentar a ignição. Esse método é mais rápido e fácil do que desconectar, retirar, inspecionar e ajustar os cabos. Como base desse conhecimento mais profundo está o senso comum que diz que se uma conexão entre fios é ruim, então a eletricidade não pode fluir e logo o carro não pode funcionar (Chandrasekaran & Milne,1985).

A atual tecnologia de sistemas especialista, baseia-se profundamente no conhecimento superficial (Ferreira,1991) (Fiol,1993). O raciocínio empregado nesse tipo de conhecimento consiste de regras de bom senso, muito frequentemente apresentadas na forma de pares do tipo condição-ação, adquiridos pelo especialista através de muita vivência experimental. Essas regras dizem ao usuário o que concluir em uma dada situação, mas não fornecem nenhum tipo de razão para a relação entre as condições e a conclusão, além de estatísticas (Rauch-Hindin,1985). Essa abordagem usa a heurística ou o conhecimento experimental como uma árvore finita. E o sistema de diagnóstico baseado nessa abordagem geralmente captura as regras de bom senso, intuições e experiências passadas de um ser humano considerado especialista de diagnóstico de uma determinada área (Morales & Garcia,1990). Elas tendem a ser experimentais e modelam o processo de tomada de decisão - como o especialista raciocina sobre o problema do domínio, ou seja o processo de diagnóstico que utiliza a estratégia do raciocínio superficial, emula a habilidade humana de utilizar a experiência passada na diagnose de sintomas conhecidos. O especialista humano, inicialmente, tenta diagnosticar o problema comparando-o com um problema similar previamente vivido, e aplica heurísticas simples bem sucedidas em casos anteriores (Fink & Lusth,1987).

Os sistemas especialistas de conhecimento compilado são rápidos e podem auxiliar no processo de diagnóstico desde as Indústrias de Forma até as Indústrias de Processo (Chandresakaran et all,1993). O sucesso de um sistema baseado nessa abordagem depende da robustez da experiência, do conhecimento do especialista humano e da correta captura desse conhecimento (Rooth & Wood II,1993). Pode-se citar alguns exemplos clássicos de sistemas baseados em conhecimento raso como: MYCIN (diagnóstico e tratamento de doenças sanguíneas), REACTOR (diagnóstico para acidentes em reatores nucleares e ARBY (diagnóstico

para falhas eletrônicas). Entretanto como a base de conhecimento é baseada em heurística, existem problemas quando são tratadas falhas não usuais ou modos inesperados de operação, e há dificuldades em se verificar se o correto diagnóstico foi feito (Arciszewski & Wnek,1992).

O raciocínio superficial, mesmo sendo fácil de ser representado, é incapaz de emular o real processo cognitivo do conhecimento do especialista humano porque não faz uso do conhecimento causal do problema domínio (Kattan,1994). O raciocínio raso funciona melhor nos casos onde é impossível ou desnecessário a representação completa das descrições de projeto de um problema domínio ou onde não existe a necessidade de se diagnosticar novas falhas ou adquirir novas informações (Mussi & Morpurgo,1990).

Pesquisas mais atuais em Sistemas Especialistas, especialmente com respeito a diagnóstico e manutenção, tem enfatizado a utilização do conhecimento profundo (Goldstein,1994). O raciocínio baseado no conhecimento profundo, por outro lado, fornece o nível inferior, causal, funcional e informações físicas em uma situação de solucionamento de problema. Modela como o domínio trabalha, em termos de funcionamento (Lirov,1993). Sistemas especialistas de diagnóstico baseados em modelo examinam os modelos para determinar a causa ou causas da falha ou falhas (Weerahandi & Hausman,1994). De uma forma geral, os modelos podem ser abordados como *modelo de dispositivo centralizado* onde descrevem o equipamento utilizando-se das relações entre os valores de entrada e de saída, enquanto que os *modelos de processos descentralizados* descrevem em mínimos detalhes o processo que ocorre dentro de uma parte do equipamento (Wnek & Michalski,1992). Um sistema especialista baseado no raciocínio profundo faz a análise de que, qualquer discrepância entre o comportamento esperado e observado é uma falha, e utiliza o conhecimento da estrutura, função e do comportamento de um dispositivo para determinar as possíveis causas da falha (Reiter,1987). A estratégia de raciocínio profundo é basicamente uma estratégia de busca eficiente que emula o raciocínio humano no descobrimento através dos vários componentes interconectados e de seus possíveis caminhos causais num mecanismo de identificação do componente problemático (Krishnamurthi & Phillips,1992). O modelamento desse raciocínio humano como estratégia de raciocínio profundo requer a identificação e a modularização das tarefas que os especialistas trazem durante o processo de diagnose (Hieb et al,1992).

Os trabalhos fundamentais nessa área incluem a pesquisa de Davis em diagnose de circuitos eletrônicos utilizando modelos estrutural e causal (Davis & Hamscher,1988). A pesquisa de Chandrasekaran e Mitall,1989 na compilação de conhecimento de diagnóstico proveniente do conhecimento profundo na área de diagnóstico médico, no trabalho de Kleer e William,1983 na visualização e na física qualitativa como meios para prover explicações causais do comportamento de dispositivos, no trabalho de Forbus,1990 na teoria do processo qualitativo como meio de se atingir o raciocínio do senso comum sobre os processos físicos. E no trabalho de Genesereth,1984 no diagnóstico de falhas em circuitos eletrônicos baseados nos seus projetos. A maioria sendo relacionados com o desenvolvimento de modelos físicos ou causais do dispositivo, como em circuitos eletrônicos. Esses sistemas podem conter um entendimento muito profundo de como o dispositivo funciona. Outro trabalho que tem enfatizado o estudo do processo de diagnóstico em si é o trabalho no sistema ARBY por McDermott et al.. Alguns sistemas, como a pesquisa de Patil em 1989, no sistema ABEL na área de diagnóstico médico, contêm diferentes pontos de vista do dispositivo ou processo. Entretanto, pouco tem sido feito na combinação de como o dispositivo funciona com a experiência do especialista de como melhor diagnosticar o dispositivo (Bradley,1993). Sem o mesmo conhecimento disponível no especialista humano, não se pode esperar que um Sistema Especialista comporte-se de uma maneira tão precisa ou robusta (Fink & Lusth,1987).

Tanto a abordagem do raciocínio raso como a abordagem do raciocínio profundo apresentam as suas vantagens e desvantagens. O raciocínio profundo é muito robusto devido a completa representação do conhecimento básico de um domínio de problema, que habilita ao sistema especialista a diagnosticar novas falhas e apropriadamente capturar o real processo de raciocínio de diagnóstico do especialista humano (Steels,1989). Entretanto, essa abordagem pode ser muito cara e impraticável se o modelo causal do problema domínio for difícil de ser desenvolvido ou representado (Bouche et al.,1990).

As duas das mais representativas abordagens de estratégias de diagnóstico utilizadas nos Sistema Especialistas são (Giorno,1990):

- *Abordagem baseada em classificação,*
- *Abordagem baseada em modelos*

2.3.3.1 Abordagem Baseada em Classificação

Um sistema de diagnóstico automatizado desenvolvido de acordo com a abordagem baseada em classificação ou baseada em reconhecimento:- classifica e explica falhas em equipamentos através da representação de associações empíricas que relacionam sintomas a falhas e a procedimentos de reparo, contando também com um processo de raciocínio que conduz a manipulação dessas associações, dado os sintomas observados (Giorno,1990).

O processo de classificação mais simples consiste na identificação de algum objeto desconhecido ou fenômeno como um membro de uma classe de objetos, eventos, ou processos. Tipicamente, essas classes são estereótipos hierarquicamente organizados, e o processo de identificação consiste na observação da correspondência e igualdade entre a entidade desconhecida e as classes conhecidas (Thakore,1993).

A característica essencial da classificação é que o solucionador do problema selecione um conjunto de soluções pré-enumeradas. O que não significa que a “resposta correta” está necessariamente em uma dessas soluções, mas que o solucionador do problema atentar-se-á para casar o dado contra as soluções conhecidas ao invés de construir uma nova. As evidências podem ser incertas e os casamentos parciais, e portanto a saída pode ser uma lista de hipóteses de forma ordenada (Clancey,1992).

Uma das estratégias de classificação de maior utilização nos desenvolvimentos de Sistemas Especialistas voltados à diagnóstico é a **Classificação Heurística** (Roth & Wood II,1993). A classificação heurística é uma estratégia de resolução onde dados e soluções relacionam-se de forma direta e não hierárquica através de relações predominantemente heurísticas (Clancey,1989); nos problemas de diagnose essas relações associam os sintomas às falhas diretamente ou através de abstrações. Uma relação heurística é incerta, baseada em hipóteses genéricas e, às vezes, com um fraco entendimento dessas correlações. A heurística é quase sempre empírica, advinda da vivência e experiência na resolução de problemas (Wnek & Michalski,1992).

O diagrama da Figura 8, mostra esquematicamente como um conjunto composto de: dados, abstração do dado, abstração da solução e solução - se relacionam e compõe a estrutura de inferência da classificação heurística.

Para muitos problemas, as opções de solução não são fornecidas como dados, mas inferidas pela abstração de dados. Há três relações básicas para dados abstraídos em programação heurística (Kirsch,1992):

1. abstração de definição:- baseada nas características necessárias e essenciais de um conceito. (“Se o animal é mamífero então é um animal homeotérmico”)
2. abstração qualitativa:- uma forma de definição envolvendo dados quantitativos, geralmente com respeito a um valor normal ou a um valor esperado. (“Se o paciente é um adulto e a contagem de glóbulos brancos é menor do que 2500, então o valor da contagem de glóbulos brancos está baixa”)
3. generalização:- é uma abstração em um subtipo hierárquico. (“Se o cliente é um juiz, então ele é uma pessoa educada”)

A idéia de disparar uma relação entre dados e soluções é fundamental em quase todas as inferências de classificação heurística. É dito que um dado problema dispara uma solução. E para uma dada classificação heurística, a rede de dados e a solução hierárquica no processo de disparo da solução podem ser interpretadas de três formas (Steels,1989):

1. *Busca orientada à dados* :- o programa trabalha no sentido dos dados para as abstrações, associando soluções até que todas as inferências possíveis, ou não redundantes, tenham sido feitas.

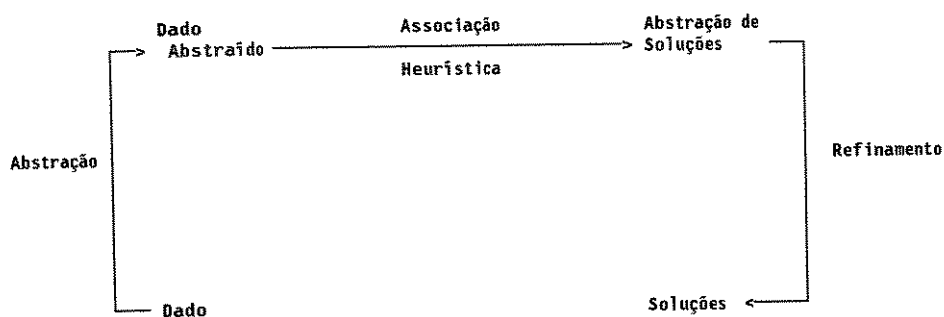


Figura 8. Estrutura de inferência da classificação heurística. (Clancey,1983)

2. **Busca orientada à solução** :- o programa trabalha a partir das soluções, coletando evidências para suportá-las, através das relações heurísticas para as abstrações dos dados e os dados necessários para solução do problema.
3. **Busca oportunística** :- o programa combina o raciocínio orientado aos dados e as hipóteses. As regras de abstração de dados tendem a serem aplicadas imediatamente à disponibilidade dos dados. As regras heurísticas disparam as hipóteses seguidas por uma busca com o enfoque das hipóteses. Novos dados podem ser focados.

O grau de busca depende do nível de dependência da implementação e de como ela é explorada (Clancey,1985).

A abordagem baseada em classificação tem o apelo a simplicidade e a modularidade, porém tem como limitações (Giorno,1990):

- incapacidade de reconhecer quando o problema está além do escopo do conhecimento,
- conhecimento específico limitado na diagnose de novas falhas,
- fraca capacidade de gerar informações úteis,
- baixa capacidade de aquisição de conhecimento, em função da dificuldade de aquisição do conhecimento experimental,
- capacidade inadequada para validação do conhecimento, especialmente em grandes bases de conhecimento,
- cobertura incompleta de todas as falhas.

2.3.3.2 Abordagem Baseada em Modelos

Representar o conhecimento de como coisas funcionam tem sido um importante problema da Inteligência Artificial. Ainda que muito esforço de pesquisa tenha sido focado na determinação do modelo de representação apropriado, não tem havido um consenso real sobre que informações essenciais esses modelos devam incluir (Bradley,1993).

A representação de dispositivos envolve teorias sobre linguagens de representação, compromissos para representação do comportamento e tipos de causas necessárias para representar o comportamento. Diferentes abordagens focalizam esses aspectos em vários diferentes níveis de detalhes (Clancey,1989).

Na Inteligência Artificial, genericamente falando, pode-se dizer que dois tipos de modelos se distinguem (Vinson,1992):

1. um onde uma descrição matemática embasada é utilizada numericamente ou qualitativamente para simular o dispositivo, e
2. e outro onde uma representação de sequências causais são utilizadas para a simulação, através de meios apropriados, de forma que o agente entenda a função do dispositivo.

Em (Davis,1984) tem-se que existem duas formas de visualizarmos a organização da informação, que são particularmente relevantes no processo de diagnóstico: funcional e físico. O modo funcional nos dá a visão da máquina organizada de como os módulos se interagem; o modo físico nos diz como tudo isso está empacotado. Dessa forma a organização funcional e a organização física das quais cada dispositivo a ser diagnosticado será descrito, produzirão duas descrições distintas e interconectadas.

Em contraste a abordagem experimental, onde conhecimento suficiente para o desempenho de uma tarefa é fornecida sem qualquer representação de mecanismos causais, a abordagem baseada em modelo captura estruturas causais explicativas de onde as inferências são feitas (Bradley,1993).

Um sistema de diagnóstico automatizado desenvolvido de acordo com a abordagem baseada em modelo, dado os sintomas observados, isola as falhas do dispositivo diagnosticado através de uma representação simbólica e de um processo de raciocínio que o manipule (Chandresakaran,1993).

Dado que problemas de diagnóstico podem ser vistos e entendidos sob diferentes aspectos pode-se ter diferentes tipos de modelos. (Steels,1990) indica os seguintes tipos de modelos qualitativos para a tarefa de diagnóstico:

1. *modelo geométrico* :- representa relações espaciais entre os componentes.
2. *modelo estrutural* :- descreve relações parte-todo entre os componentes e subsistemas.
3. *modelo funcional e comportamental* :- representa como a função do todo segue o funcionamento das partes.
4. *modelo causal* :- representa possíveis falhas e componentes para cada função que possa ser responsável pela falha.
5. *modelo associativo* :- relata propriedades observadas com estados do sistema.

Em (Goldstein,1994) tem-se que os modelos: estruturais, comportamentais e causais são os modelos básicos para a abordagem baseada em modelos. *Modelo estrutural e modelo comportamental* envolvem a identificação e a representação de descrições de projeto, como estruturas e descrições do comportamento do dispositivo a ser diagnosticado. Davis (Davis et al.,1982,1984,1985 e 1988) propõe uma linguagem para expressar a estrutura de um dispositivo composto por um conjunto de comandos indicando quais subconjuntos são necessários e como são interconectados. Para a descrição do comportamento propõe a utilização de redes de Petri no modelamento de eventos paralelos, regras de codificação de entrada/saída. E no modelamento do comportamento simples um conjunto de códigos e restrições. *Modelo causal* baseia-se em um modelo qualitativo físico do dispositivo. Ele consiste na representação dos possíveis estados do dispositivo e no relacionamento entre esses estados. Modelos causais podem ser utilizados para representar o comportamento correto ou falho, e a decisão de uma dessas formas de representação do comportamento tem um grande impacto de como o diagnóstico é desempenhado (Torasso e Console,1989).

A chave da utilização de modelos qualitativos reside em uma representação de conhecimento apropriada e de um algoritmo para a manipulação da representação desse conhecimento (Kobayashi & Nakamura,1991). Modelos, sejam comportamentais, funcionais ou causais são os mais poderosos sistemas de diagnósticos dentre os mecanismos centrais de organização (Milne,1987). Eles provêm mecanismos de representação do conhecimento de uma grande quantidade de informações e suas relações, que vem a habilitar um raciocínio mais complexo para um poderoso sistema de diagnóstico (Steels,1989). Os modelos são primariamente utilizados como mecanismos de simulação de encadeamento para frente ou previsão. Ou seja, dado um conjunto de entrada de um determinado nível, o modelo produz a descrição da saída do próximo nível superior ou subsequente. Modelos são grandemente utilizados como meios de se perseguir possíveis candidatos

que se identifiquem como as falhas de um sistema (Fiol & Nicolau,1993). Utilizando-se encadeamentos causais ou estruturais é na grande maioria das vezes possível a identificação de quais os componentes que podem ou não falhar em um sistema.

Com relação as suas limitações, a abordagem baseada em modelo, não modela a experiência do especialista (Rodi et al,1987) e tem um elevado custo computacional em função da complexidade envolvida. Dependendo, entre outras coisas, do número de possíveis falhas a considerar, do número de componentes envolvidos e da conectividade entre os mesmos (Morales e Garcia,1990). Também, para equipamentos derivados da combinação de diferentes áreas de conhecimento, como sistemas eletro-mecânicos, torna-se mais difícil de se formular uma descrição conveniente do equipamento (Hieb et all,1992).

2.4 Resumo

Esse capítulo apresentou um estudo da Inteligência Artificial através dos trabalhos expoentes, conceituações, e áreas de aplicação. Destacou a área de resolução de problemas, entre outros, como uma das áreas onde são necessárias aplicações de heurística sem uma definição algorítmica das estratégias de solução. E também onde se faz necessário a análise de dados que inclua parcelas de incerteza em conjunto com a aplicação de conhecimentos adquiridos e armazenados. Sendo que esses fatores conduziram ao desenvolvimento de programas computacionais com propósitos específicos e em áreas de problemas de domínio restrito. Esses programas são chamados de Sistemas Especialistas.

Esse capítulo ainda estudou os Sistemas Especialistas em termos de sua estrutura, classificação, construção, aquisição e formas de representação do conhecimento, implementação e avaliação. Particularmente analisou-se a aplicação desses sistemas no processo de diagnose e as formas de abordagem para essa tarefa, por ser esse o propósito desse trabalho.

3.0 Um modelo de Integração do Conhecimento Aplicado à Diagnose

Nos Capítulos 1 e 2 deste trabalho, apresentou-se a questão da utilização dos computadores no modelo de integração da manufatura, particularmente focou a contribuição da Inteligência Artificial através dos Sistemas Especialistas aplicados a tarefas de diagnose.

Sendo que nesse estudo, os seguintes fatos foram identificados:

- as vantagens da utilização de Sistemas Especialistas como resposta aos problemas relativos a experiência e ao conhecimento humano, comparado as abordagens dos sistemas computacionais ditos convencionais,
- a adequação dos Sistemas Especialistas nos processos de diagnose,
- a necessidade de uma abordagem de diagnóstico baseada em classificação heurística, e as limitações de um Sistema Especialista de abordagem única baseada em classificação,
- a necessidade e as dificuldades da implementação de um sistema de diagnose baseado em modelo,
- a simplicidade e modularidade da representação e da implementação do conhecimento - base de conhecimento, através de regras de produção,
- e finalmente a disponibilidade, no mercado, de programas do tipo casca (da palavra inglesa “shell”) como ferramenta de implementação de Sistema Especialistas.

Com bases nesses fatos, o autor entende que um processo de diagnóstico baseado em uma abordagem integrada baseada em modelo e em classificação heurística, pode proporcionar um diagnóstico mais eficiente. Sendo o modelo causal, aquele que melhor se adequa ao processo de diagnóstico de falhas de máquinas e equipamentos, pela natural associação entre essa forma de se organizar o conhecimento e o processo de diagnóstico propriamente dito.

O autor ainda entende que a implementação do modelo proposto através de programas do tipo “shell” com representação do conhecimento via regras de produção, pode proporcionar uma implementação mais rápida e simples, e principalmente sem comprometer a qualidade do sistema. E portanto, além da proposta de um modelo de integração do conhecimento, esse trabalho ainda propõe a utilização de programas do tipo “shell”, como ferramenta para a instrumentação desse modelo.

A contribuição esperada do presente trabalho, está em propor um processo automático de diagnóstico mais eficiente, visando auxiliar o processo de manutenção de máquinas e de equipamentos voltados a produção. E ainda, confirmar o potencial das ferramentas do tipo casca no desenvolvimento de Sistemas Especialistas voltados a diagnose, e o potencial das regras de produção como forma de representação do conhecimento.

A materialização desse conjunto de propostas é apresentada nesse capítulo, através dos seguintes tópicos:

1. Modelo proposto: discute o modelo da abordagem de diagnóstico proposta, e o modelo de integração do conhecimento em uma arquitetura típica de um Sistema Especialista.
2. Proposta da forma de implementação do modelo proposto: apresenta o “shell” como uma ferramenta para a implementação.
3. Representação do Conhecimento: discute a utilização de regras de produção como forma de representação da base de conhecimento do modelo proposto.

3.1 Modelo proposto

A maioria dos sistemas especialistas correntemente utilizados empregam a estratégia da abordagem de classificação heurística (Ferreira,1991) (Fiol,1993). Recentes estudos especialmente relativos a diagnose e reparos vem enfatizando a utilização de estratégias baseadas em modelo (Yoon & Hammer,1988) (Germond,1992). Entretanto, muito pouco tem sido feito para combinar ambos os modelos (Fink & Lusth,1987), (D'Ambrosio,1990), (Bowen,1994), i.e., a visão de como funciona o dispositivo combinada com a forma com que um especialista diagnosticaria o problema de acordo com a sua experiência. (Torasso e Console,1989) também indicam que o problema de solucionamento do diagnóstico é geralmente uma tarefa tão complexa que o uso de uma única forma de representação do conhecimento e um único mecanismo de raciocínio não é o modo mais apropriado para se abordar e solucionar o problema.

Analisando-se as vantagens e desvantagens associadas a cada abordagem apresentadas no Capítulo 2, torna-se claro que uma abordagem integrada baseada em classificação e baseada em modelo torna-se a mais indicada, de forma a aproveitar o melhor de cada uma das abordagens. Dessa forma, esse trabalho tem como uma de suas propostas a utilização de uma estratégia de diagnóstico de abordagem integrada baseada em classificação heurística e em modelo onde a base de conhecimento contempla dois tipos distintos de conhecimento: o conhecimento experimental, e o conhecimento fundamental.

A abordagem da estratégia de diagnóstico baseada em classificação heurística é uma estratégia de resolução onde os dados e as soluções relacionam-se de forma não direta e não hierárquica, ou seja através de relações predominantemente heurísticas, sendo que essa heurística é quase sempre empírica advinda da vivência experimental da resolução de problemas que é denominada de conhecimento experimental ou conhecimento empírico - que se adquire com o passar de muitas e várias experiências ao longo do tempo de vivência na área de domínio do conhecimento. Nesse tipo de conhecimento quase não existe uma ligação extensiva entre a causa e o efeito que associe condições com conclusões, além do fato da experiência passada conectá-las através de observações empíricas. As conexões entre a causa e o efeito podem existir, mas não são as razões principais para essa associação. E o presente trabalho propõe que esse conhecimento experimental seja obtido do especialista do processo e do equipamento, dos técnicos de manutenção e dos operadores de máquina.

Em contraste a abordagem experimental, onde o conhecimento suficiente para o desempenho de uma tarefa é fornecida sem qualquer representação de mecanismos causais, a abordagem baseada em modelo captura estruturas causais explicativas de onde as inferências são feitas. Um sistema de diagnóstico automatizado desenvolvido de acordo com a abordagem baseada em modelo, dado os sintomas observados, isola as falhas do dispositivo diagnosticado através de uma representação simbólica e de um processo de raciocínio que os manipula. E a chave da utilização de modelos qualitativos reside em uma representação de conhecimento apropriado que tende a ser baseada nas estruturas e funções dos dispositivos. Esse conhecimento é denominado de fundamental, onde causa e efeito são as razões primeiras para associações nesse nível. E esse trabalho propõe que o conhecimento fundamental seja organizado segundo a classificação de Steels, denominada de *modelo causal*. Pois o autor entende que esse modelo de organização do conhecimento fundamental é um dos mais adequados para aplicações voltadas ao processo de diagnóstico, devido a natural associação entre a sua definição e o processo de diagnóstico humano. Esse trabalho ainda entende que o processo de modelamento causal pode ser facilitado pela escolha mais adequada da forma de representação do conhecimento. Especificamente, esse trabalho propõe a utilização de regras de produção, por julgá-lo como uma das formas mais adequadas de representação para esse modelo, dentre aquelas estudadas pelo autor.

A proposta de utilização de uma estratégia de diagnóstico integrada, resulta em um sistema especialista que além de maximizar as vantagens e minimizar as desvantagens de cada uma das abordagens, ainda resulta em um sistema que pode solucionar problemas mesmo quando não existam situações de conhecimento experimental, resultando em uma maior capacidade do sistema.

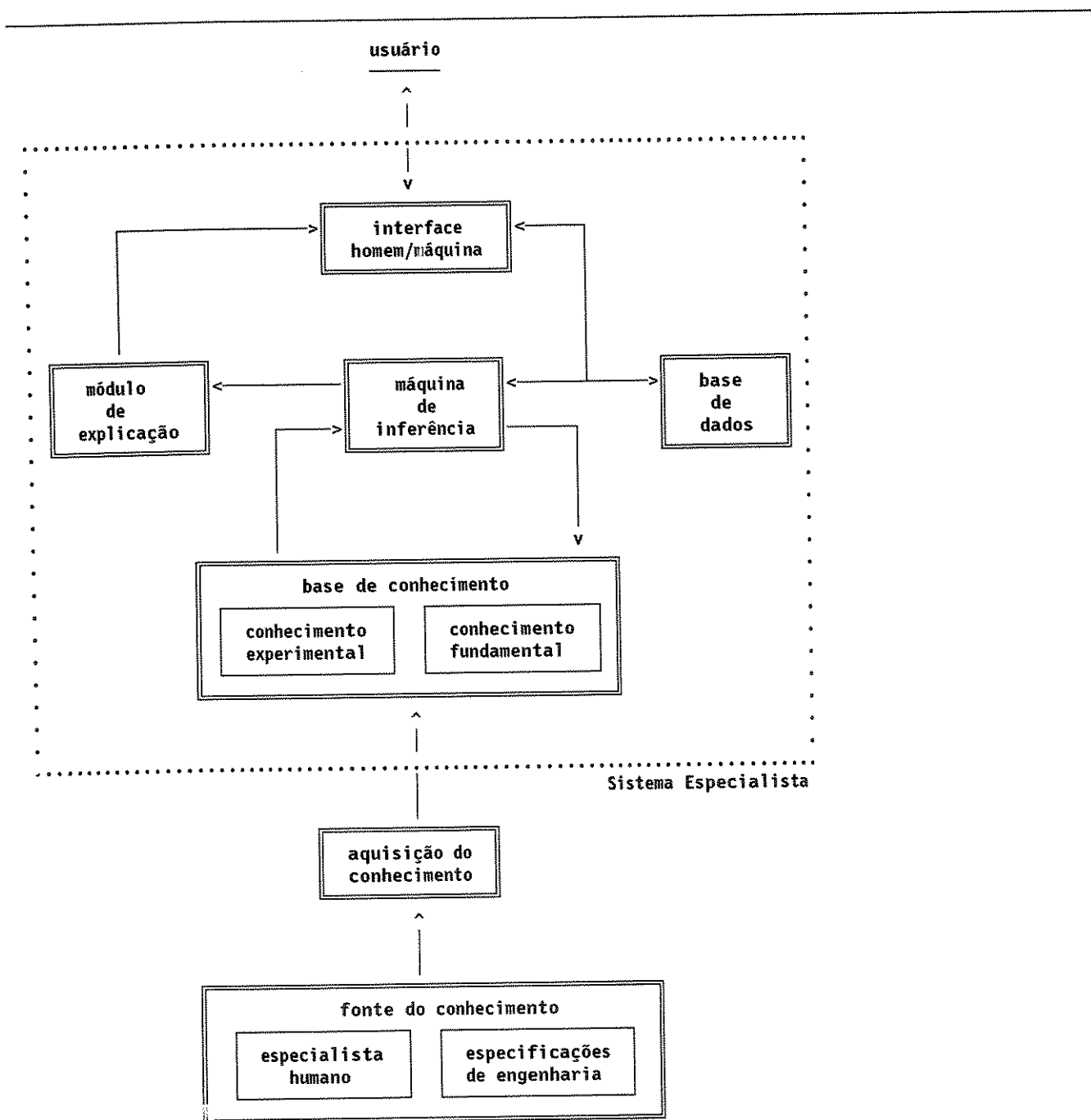


Figura 9. Arquitetura funcional proposta para o sistema especialista.

Essas especificações funcionais permitiram a definição do modelo do sistema especialista de diagnóstico conforme a Figura 9.

Sendo esse modelo proposto, composto dos seguintes elementos:

1. **Base de conhecimento**, onde reside o conjunto de informações que codificam a estratégia de solução e a representação do conhecimento, composta dos conhecimentos experimental e fundamental.
2. **Base de dados**, que contém as características e parâmetros relativos ao problema a ser resolvido.
3. **Módulo de inferência**, que controla o processamento do sistema e aplica a base de conhecimentos à base de dados, para obter novos dados e conhecimentos.
4. **Módulo de explicação**, que permite ao sistema explicar como uma conclusão foi obtida através do rastreamento e da reconstrução da cadeia de raciocínio que levaram a uma dada resposta. A habilidade de explicar uma resposta ou linha de raciocínio é uma importante e diferenciadora característica de um programa inteligente.
5. **Módulo de aquisição do conhecimento**, que através de técnicas de aquisição, aplicadas manualmente ou com auxílio de ferramenta, adquire o conhecimento de um ou mais especialistas e implementa a base de conhecimento utilizando-se das estruturas de dados definidas pelas técnicas de representação.
6. **Interface homem-máquina**, que permite a comunicação e interação entre o sistema e o usuário.

3.2 Proposta da Forma de Implementação do Modelo Proposto

3.2.1 Formas de Implementação

O processo de implementação do sistema consiste no desenvolvimento de um programa de computador, sendo que esse programa pode ser implementado através das formas a seguir:

1. Através de linguagens de programação procedural direta.

As linguagens de programação utilizadas para a implementação de sistemas especialistas são geralmente: *linguagem orientada a problemas* como Fortran, Pascal e "C", ou *linguagem de manipulação simbólica* como Lisp e Prolog. Sendo que as linguagens de manipulação simbólica são mais largamente utilizadas e adequadas a programação em Inteligência Artificial e para a construção de sistemas especialistas (Sekine et al,1992).

- LISP Inventado em 1962 por John McCarthy, basea-se na capacidade de análise e manipulação de listas, através de uma estrutura hierárquica e recursiva, além da capacidade de realocação dinâmica de memória. Diversas versões foram desenvolvidas como: MacLisp proveniente do laboratório de Inteligência Artificial do MIT e o InterLisp de Bolt, Beranek, Newman e do centro de desenvolvimento da empresa Xerox de Palo Alto - CA, e existe uma tendência de padronização para o chamado Common Lisp.
- PROLOG Linguagem inicialmente desenvolvida entre 1970 e 1980 por A.Colmerauer, R.Kowalski e P.Roussel, com o seu compilador/ interpretador desenvolvidos por D.Warren, F.Pereira e L.Byrd em 1975 na Universidade de Edinburgo na Inglaterra, baseia-se na similaridade de três tipos de declarações da língua inglesa: fatos, regras e questões; bem como a manipulação desses elementos.

2. Através de ferramentas específicas de desenvolvimento de sistemas especialistas.

Usualmente trata-se de um ambiente integrado de desenvolvimento cuja estrutura principal reside em um programa do tipo casca (da palavra inglesa "shell"). Que são programas derivados das primeiras implementações de sistemas baseados em conhecimento para um dado propósito específico, geralmente desenvolvidos em Lisp ou Prolog. Num segundo momento abstrai-se a estrutura do sistema implementado e removem-se os dados e os conhecimentos, obtendo-se então uma estrutura onde o conhecimento pode ser intercambiável de forma a permitir ao usuário uma implementação em um formato de linguagem de alto nível, evitando-se assim as "dificuldades" das programações procedurais (Butler & Hodil, 1988).

A grande contribuição da conceituação de programas do tipo shell foi o entendimento e a praticalização da abstração e da generalização do processo de raciocínio do especialista humano, e cujas investigações mostraram claramente a possibilidade de separação entre as descrições de aplicações, e as estratégias de diagnóstico (Gisolfi & Balzano, 1993).

3.2.1.1 Shell

O termo sistema especialista "shell" é derivado de sistemas especialistas cuja base de conhecimento é intercambiável e que tem seus dados removíveis. Um shell constitui-se de três partes principais (Gisolfi & Balzano, 1993):

1. A base de conhecimento, contendo ambos dados e métodos de representação do conhecimento,
2. Máquina de inferência, que mecanicamente aplica os métodos para inferir novo conhecimento a partir do dado conhecimento,
3. Interface homem/máquina, com características de adquirir/mostrar e de procurar/explicar o conhecimento.

As partes (1) e (2) constituem o que é denominado de núcleo (da palavra inglesa "kernel") do shell.

A identificação de requisitos específicos de um shell voltados a diagnóstico é essencial para se chegar a uma arquitetura apropriada. Os requisitos a seguir foram identificados como desejados, da perspectiva de desenvolvimento e uso de um shell de diagnóstico e da perspectiva da aplicação resultante gerada pela ferramenta de desenvolvimento (Krishnamurt & Phillips,1992), (Arciszewski,1993):

1. a aplicação do sistema de diagnose gerado deve ser capaz de diagnosticar falhas simples e complexas,
2. a estratégia de diagnóstico capturada deve permitir o diagnóstico de falhas desconhecidas,
3. o sistema de diagnóstico gerado deve ser capaz de explicar seu processo de raciocínio, bem como ter meios de armazenar um histórico de falhas de diagnóstico,
4. a aplicação gerada deve poder ser utilizada como ferramenta de treinamento,
5. a interface homem/máquina deve ser a mais padronizada possível,
6. o shell deve manter a representação interna do conhecimento adquirido transparente ao usuário, e aliviá-lo da carga de ter que aprender a linguagem de representação do shell,
7. o shell deve facilitar o processo de aquisição interativa das descrições de projeto e do conhecimento heurístico, para a aplicação a ser gerada.

(Boley,1994), faz uma contributiva análise através de diferentes cenários e critérios de programas do tipo shell:

1. shells acadêmicos e comerciais:

O desenvolvimento dos programas shell iniciou-se no meio acadêmico, principalmente através dos esforços coroados de MYCIN, EMYCIN E OPS5. Esse desenvolvimento teve rápida continuidade no meio comercial. Hoje a quantidade de shells comerciais vem crescendo rapidamente, incentivado pelo sucesso das estações de trabalho e do mercados de computadores pessoais, ou microcomputadores. Alguns shells iniciados em meio acadêmico foram posteriormente estendidos e comercializados por empresas emergentes em Inteligência Artificial.

2. shells para computadores de grande porte, estações de trabalho e computadores pessoais.

Os primeiros programas shell eram executados em computadores de grande porte. O rápido crescimento no aumento de memória e características de velocidade dos computadores pessoais dos circuitos da série 68030 Motorola e 386,486 da Intel tem permitido a execução dos shell em computadores pessoais de pequenos sistemas especialistas. Atualmente as principais plataformas para os programas shell estão na classe de computadores intermediários das estações de trabalho, principalmente entre as marcas Sun, Apollo, IBM-RT e HP-9000/350. Competindo com as estações de trabalho estão as máquinas Lisp e Prolog que fornecem shells baseados na linguagem Lisp e Prolog com alto poder computacional, utilizando processadores especiais como o TI Explorer II cujos circuitos integrados foram especialmente projetados para suportar sistemas especialistas de grande porte.

3. shells baseados em linguagem Prolog, Lisp e "C"

Enquanto a maior parte dos shells ainda são escritos em Lisp e alguns em Prolog, existe atualmente, uma forte tendência na direção de implementações de programas produto - do tipo shell através da linguagem "C". Tal linguagem é favorecida pela tendência dos centros de processamento de dados convergirem para uma padronização do sistema operacional UNIX. A linguagem Lisp tem boas chances de se manter na tendência da linguagem preferencial, se o CommonLisp for rapidamente padronizado como uma norma ISO, bem como, se houver um maior e melhor desenvolvimento da interface entre o padrão Lisp e os padrões dos centros de processamento, ou seja, melhorar a comunicação e a conectividade entre os diferentes sistemas.

4. Usuários finais, engenheiros do conhecimento e programadores de shells de Inteligência Artificial

Um shell pode ser utilizado por um grande mercado de pessoas não treinadas, usuários finais não experientes (pessoas usualmente não relacionadas com a Ciência da Computação) ou pelo pequeno e altamente qualificado mercado dos programadores experientes em Inteligência Artificial. Para o usuário final, o shell é restrito ao domínio da aceitação do conhecimento e não ao programa de Inteligência Artificial, enquanto que os programadores dos shells utilizam-se da linguagem básica para a programação da plataforma. E entre o usuário final e o programador de Inteligência Artificial existe geralmente o engenheiro do conhecimento, que é habilitado na aquisição de domínios de conhecimento de um especialista humano e formalizar esse conhecimento, para inseri-lo no programa shell na forma de conhecimento representado. Em muitos shells existem interfaces para todos os três tipos de usuários.

5. shell de propósito geral e shell de propósito específico

Existem duas formas principais para a construção de um sistema especialista do tipo shell. A primeira forma conduz a um shell de propósito especial (para construir sistemas especialistas para a mesma categoria em diferentes domínios). Isto é possível pegando-se um sistema especialista, implementado através de uma linguagem básica do tipo Lisp, e esvaziá-lo de parte ou de todo o conhecimento armazenado na base de conhecimento, deixando o conhecimento reutilizável, as formas de representação, a máquina de inferência e a interface homem/máquina. Esse estado está relacionado com a concepção de programa tipo casca, e assim pode ser preenchido com conhecimento que descreve outro domínio ou subdomínio de forma a se obter outro sistema baseado em conhecimento para a mesma categoria de tarefa. O clássico exemplo desse procedimento é o programa MYCIN e seus derivados, e o shell EMYCIN, que foram posteriormente comercializados. A segunda forma conduz a um programa do tipo shell de propósito geral (para construir sistemas especialistas para diferentes categorias em diferentes domínios). Isto é possível utilizando-se uma linguagem base para se construir vários componentes para representação de conhecimento, inferência (para frente, para trás, herança) e interface homem/máquina, como necessários para a construção rápida de um sistema especialista. O uso desse shell pode ser empregado para a construção de sistemas especialistas para quaisquer categorias em um domínio arbitrário qualquer. Atualmente os mais conhecidos exemplos comerciais, desenvolvidos em Lisp, são os programas ART, KEE e KnowledgeCraft.

6. shells uniformes e híbridos

Em um programa do tipo shell, a representação do conhecimento e conseqüentemente a máquina de inferência pode ser uniforme ou híbrida. Uma representação uniforme emprega um formalismo canônico como por exemplo o uso único ou de regras de produção ou de quadros. Uma representação híbrida utiliza-se de dois ou mais formalismos em várias partes do conhecimento base, como o uso de ambos quadros e regras de produção. As vantagens claras da uniformização são:

- menor carga de aprendizado, se o usuário deseja conhecer todo o programa shell,
- não necessidade de tradução das inter-representações, tanto para a máquina como para o ser humano,
- evita a evolução de representações redundantes e repetitivas do conhecimento,
- não necessidade de uma metodologia de seleção para as representações

Complementarmente as vantagens dos programas híbridos são:

- menor carga de aprendizado se o usuário puder encontrar uma forma de representação familiar dentre as oferecidas,
- melhorar o sistema através da representação do conhecimento de múltiplas perspectivas,
- a mais natural e eficiente representação pode ser escolhida para cada problema.

7. shell de conhecimento superficial e de conhecimento profundo

A máquina de inferência de um shell pode executar inferências (raciocínio) superficiais ou profundas sobre a base de conhecimento de regras ou modelos respectivamente. Uma inferência profunda consiste de um encadeamento de regras que associam premissas (e.g. observados os sintomas) com uma conclusão (e.g. um diagnóstico hipotético). Uma inferência profunda consiste da prova (e.g. dados os fatos) através de um modelo que geralmente coleciona propriedades (e.g. causais, temporais e/ou espaciais) do domínio.

3.2.2 Forma de Implementação Proposta

Esse trabalho, propõe a implementação do sistema de diagnóstico através de uma ferramenta de desenvolvimento do tipo "shell", essencialmente em função dos seguintes fatores:

1. menor tempo de implementação,
2. menor necessidade de recursos iniciais,
3. boa disponibilidade de programas do tipo "shell" no mercado.

que se devem basicamente a:

1. facilidade do uso das linguagens de alto nível,
2. alta integração do ambiente de trabalho,
3. menor tempo de aprendizado, e
4. da alta disponibilidade no mercado de ferramentas do tipo "shell".

Além das considerações técnicas, esse trabalho entende ainda, que em **linhas gerais** os seguintes requisitos desejáveis sejam contemplados no Sistema Especialista proposto:

- O programa deve ser *útil*; i.e. deve responder satisfatória e adequadamente as necessidades.
- O programa deve ser *utilizável*; i.e. deve ser capaz de ser utilizado - ser rápido, acessível, fácil de ser aprendido, e simples para um usuário novato.
- O programa deve ser *educativo quando apropriado*; i.e. deve permitir aos técnicos ou usuários acessarem sua base de conhecimento e deve ser capaz de transportar as informações pertinentes de uma forma que possam entender sem dificuldade.
- O programa deve ser capaz de *explicar sua orientação*; i.e. deve prover informações suficientes ao usuário sobre o seu raciocínio, de forma que o técnico possa decidir sobre as ações recomendadas.
- O programa deve ser capaz de *responder a simples questões*; i.e. deve ser possível ao usuário requerer justificativas de uma inferência específica, idealmente utilizando uma linguagem natural.
- O programa deve ser capaz de *aprender novos conhecimentos*; i.e. deve ser possível dizer ao programa novos fatos e tê-los incorporados facilmente para uso futuro, ou idealmente, deve ser capaz de aprender da experiência utilizada de um grande número de casos.
- O conhecimento do programa de ser *facilmente modificável*; i.e. a adição de novo conhecimento ou a correção de erros deve ser direta, idealmente, deve ser realizada sem qualquer alteração explícita no programa (código fonte propriamente dito).

3.3 Representação do Conhecimento

A proposta de se utilizar regras de produção como forma de representação do conhecimento, é justificada por um conjunto de características que o autor entende como determinantes:

1. adequação

A representação do conhecimento através de regras de produção tem uma forte associação natural com a organização do modelo causal, através do seu formato do tipo par condição-ação.

2. modularidade:

Pode-se relacionar a modularidade de um programa ao grau de separação de suas unidades funcionais em partes isoladas. Um programa é altamente modular se qualquer unidade funcional pode ser alterada (adicionada, eliminada ou substituída) sem qualquer alteração, antecipada, de outras unidades funcionais. A modularidade de programas escritos com de regras de produção advém do importante fato de que a próxima regra a ser invocada é determinada unicamente pelo conteúdo do banco de dados, e nenhuma regra é chamada diretamente, logo, a inserção (ou eliminação) de uma regra não requer a modificação de qualquer outra regra.

3. regras como ações primitivas:

Em um sistema de regras de produção *puro*, a menor unidade de comportamento é a invocação de uma regra. Na sua forma mais simples, envolve na identificação literal do lado esquerdo da regra, seguido pela substituição desses símbolos no banco de dados com aqueles encontrados no lado direito da regra.

4. formato restrito / simplicidade:

Mesmo que haja uma grande variação no formato permitido pelos vários sistemas de regras de produção, em todos eles a sintaxe é um tanto quanto restritiva, e geralmente segue a convenção aceita para as regras de produção. Mais comumente isso significa que: -primeiro, que o lado da regra a ser encontrado deve ser um simples predicado importado de uma combinação Booleana que envolve somente identificação e detecção, -segundo, isso significa que o lado da regra a ser executado deve desempenhar operações simples no banco de dados. Dessa forma, qualquer que seja o formato, a convenção adotada conduz a que: -primeiro, como um predicado, o lado da identificação da regra deve retornar somente a indicação de sucesso ou falha de identificação, -segundo, como uma simples expressão a operação de identificação traz a prevenção de se utilizar estruturas de controle mais complexas como iterações e recursões. E finalmente, como uma operação de identificação e detecção, o sistema deve somente observar o estado do banco de dados e alterá-lo na operação do teste. Para o lado da regra a ser executado pode haver uma simples ação primitiva, uma simples coleção de ações independentes, uma cuidadosa sequência ordenada de ações ou mesmo estruturas complexas de controle.

5. visibilidade do fluxo do sistema:

Sistemas de regras de produção permitem que a visibilidade do fluxo do sistema seja uma característica natural, quando da leitura e representação do domínio do conhecimento.

6. facilidade de leitura:

A facilidade de leitura também desperta outras capacidades quando as regras são examinadas. Por exemplo, permite uma fácil verificação de consistência das regras.

7. consenso:

A totalidade das ferramentas "shell" analisadas utilizam regras de produção como formas de representação do conhecimento.



3.4 Resumo

Fundamentado nos estudos do Capítulo 2, o presente capítulo apresentou a proposta do autor. Essa proposta constitui-se da integração dos conhecimentos experimental e fundamental como uma abordagem mais eficiente no processo de diagnóstico de máquinas e equipamentos.

A proposta ainda inclui implementação desse modelo através de programas do tipo "shell", que represente o conhecimento via regras de produção. Isso em função da simplicidade, modularidade e rapidez de implementação, sem o comprometimento da qualidade do trabalho proposto.

4.0 Aplicação e Avaliação do Modelo Proposto

O objetivo principal deste capítulo é o de aplicar e avaliar o desempenho do modelo proposto no capítulo anterior.

E dado que o objetivo desse modelo é de auxiliar o processo de manutenção de máquinas e equipamentos, sua instrumentação resultou num programa de computador para auxiliar o processo de diagnóstico de problemas do equipamento de limpeza e descontaminação, denominado de "Metal Cleaner KLN".

Sendo que os motivos que justificaram a escolha do equipamento de limpeza e descontaminação para a aplicação do modelo foram:

- dificuldade no diagnóstico das falhas, em função da natureza complexa do equipamento e seu processo.
- necessidade de retenção do conhecimento, devido ao deslocamento do especialista do processo para outro departamento.

Esse equipamento faz parte do processo de fabricação de unidades de disco magnéticos presente nas instalações da IBM Brasil, que permitiu o desenvolvimento desse trabalho na linha de fabricação do produto 3380X.

Para uma melhor compreensão, o Capítulo 4 apresenta-se sub-dividido em cinco tópicos:

1. Introdução :- faz uma breve ilustração sobre o produto fabricado, uma vez que suas especificações de engenharia impõe a complexidade ao processo e ao equipamento.
2. Caracterização do Ambiente de Aplicação do Modelo:- apresenta o contexto do processo de fabricação, onde o modelo foi aplicado.
3. Aplicação do Modelo :- descreve a aplicação do modelo proposto, através das várias etapas do desenvolvimento e da implementação do Sistema Especialista.
4. Avaliação do Sistema Implementado:- apresenta os resultados da avaliação do sistema implementado, em termos de validação, e de aceitação do usuário,
5. Resultados da implementação do Sistema:- expõe os resultados obtidos e os fatos decorrentes dessa implementação.

4.1 Introdução

O objetivo dessa introdução é o de apresentar o cenário mais amplo do contexto de aplicação do modelo proposto. Particularmente, comentar sobre as razões das restritas especificações de engenharia que impuseram a complexidade ao processo e aos equipamentos.

Esse trabalho foi desenvolvido nas instalações do Centro Industrial da IBM Brasil, localizado em Hortolândia (SP), e motivado no processo de fabricação do conjunto disco-cabeça - HDA (do inglês Head Disk Assembly) da unidade de disco de armazenamento magnético IBM 3380X.

4.1.1 O Armazenamento Magnético

Aplicações de processamento de dados em computadores tem crescido muito rapidamente nos últimos trinta anos. No princípio dos tempos dos sistemas de processamento, o principal meio de armazenamento de grandes quantidades de informações magnéticas era a fita. A maioria do processamento era feito em grupo de forma sequencial e em base tarefa a tarefa. Sendo que essa forma de processamento tinha apenas um impacto secundário nos negócios a que se aplicavam. Esses primeiros sistemas computacionais são um forte contraste aos atuais sistemas de processamento, que permitem com que muitas e diferentes tarefas sejam executadas concorrentemente e com uma elevada necessidade de armazenamento de informações em meios magnéticos acessíveis diretamente sem a intervenção humana (“on-line”).

Esses meios de armazenamento magnético de informação de acesso direto, vem fortemente contribuindo para o crescimento e consolidação da computação, porque:

- armazenam grandes quantidades de dados,
- são acessíveis diretamente sem a intervenção humana (*on-line*),
- permitem acesso randômico direto (menor tempo de acesso aos dados),
- tem armazenamento não volátil, e
- são de menor custo.

O progresso da indústria de discos magnéticos tem sido incentivado pela crescente e continua demanda por armazenamento “on-line” dados os aperfeiçoamentos como:- redução de custo, aumento da capacidade e de desempenho, e aumento da confiabilidade. O mais determinante dos fatores que vem ao encontro da demanda tem sido o aumento real da densidade de armazenamento permitido pelos avanços na tecnologia básica de gravação. Mas para se obter as vantagens de custo dos avanços tecnológicos tem-se o contínuo desafio da fabricação em face do sempre crescente aumento de precisão, redução das dimensões geométricas e requisitos de ambientes mais livres de contaminantes em processos e produtos (Mittal,1987).

4.1.2 Particularidades do Produto

O principal componente de uma unidade de disco magnético é denominado de:- conjunto cabeça-disco - HDA (abreviatura da expressão inglesa “Head Disk Assembly”).

As características magnéticas da cabeça de escrita/leitura e a sua distância com o disco determinam em alto grau, a densidade de informação por unidade de área, a capacidade de armazenamento de uma unidade de disco, e o suporte para a sua sustentação e deslizamento através de um rolamento aerostático,

A mínima distância entre o disco e a cabeça, aliada a necessidade de grande precisão mecânica, determinaram rigorosas especificações de engenharia mecânica e de limpeza, na fabricação de HDAs.

Devido a altura de vôo das cabeças ser bastante reduzida - ver Tabela 5, partículas presentes (no ambiente de fabricação, nos componentes do produto, ou gerados pelo próprio produto) podem ser fatais para a(s) cabeça(s), ou provocar a destruição da superfície do disco, e conseqüentemente a perda da informação armazenada.

Item	Dimensão
Distância média de vôo entre cabeça de leitura/gravação e o disco	0.0005 mm
Diâmetro médio de uma partícula de fumaça	0.003 mm
Diâmetro médio de uma partícula de pó típica	0.012 mm
Diâmetro médio de um fio de cabelo humano	0.076 mm

Partículas de frações de milésimos de milímetro podem se interpor entre a cabeça e o disco ou colidir com a cabeça, e sendo arrastadas, podem destruir a superfície de leitura e gravação do disco. Podendo danificar a unidade de disco e as informações nela contidas.

Por isso, o processo de fabricação do HDA se dá em um ambiente com classificação 100 e 10.000, de acordo com os padrões do *Federal Standard 209C*, e os componentes devem atingir graus de limpeza baseado nas normas do *Military Standard C-81302*, também dos Estados Unidos da América. Garantindo dessa forma as especificações de engenharia do produto.

4.2 Caracterização do Ambiente de Aplicação do Modelo

4.2.1 Introdução

O objetivo desse tópico é o de apresentar os fatos e o cenário que motivaram a escolha e a aplicação do modelo proposto para esse caso. São apresentados o contexto do processo de fabricação, bem como as características gerais do equipamento e do respectivo processo.

Sendo que dados aqui apresentados permitirão uma análise comparativa com os resultados obtidos da aplicação do modelo.

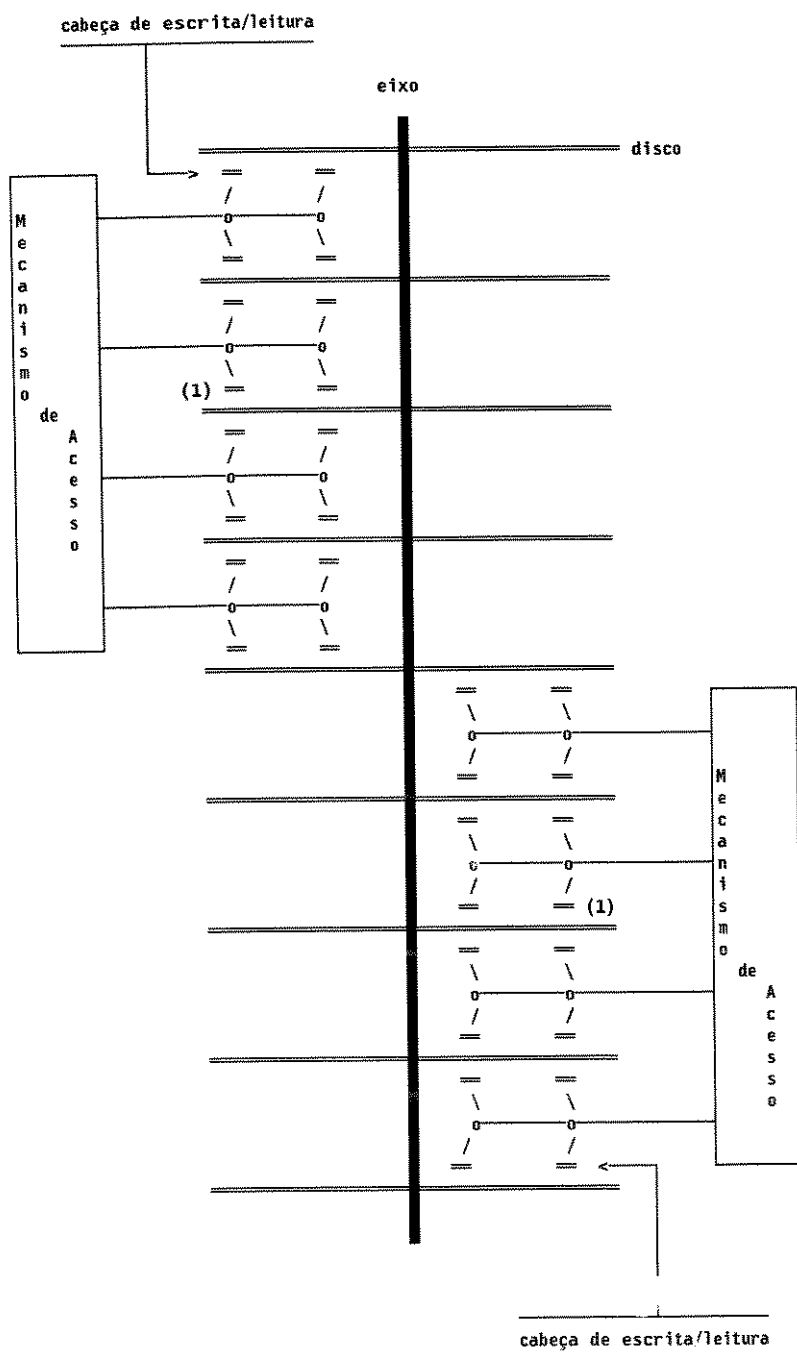
4.2.2 Características do ambiente e do processo de fabricação

O primeiro estágio da montagem de um HDA, consiste no processo de limpeza e descontaminação de seus componentes - Figura 11. Esse processo utiliza-se de tecnologia de limpeza fina de superfícies.

Para o processo de limpeza final de partes metálicas dos HDAs 3380 modelos J e K produzidos no Brasil, um equipamento foi especialmente projetado e construído com o objetivo de garantir as especificações de limpeza e descontaminação requeridos pelo produto e processo.

E de acordo com a engenharia de fabricação, esse processo é considerado crítico à produção, devido aos seguintes fatores:

1. uma quebra do equipamento pode implicar na paralisação da linha de fabricação, dado que é o primeiro processo do roteiro de fabricação do produto.



(1) cabeça de servo

Figura 10. Representação esquemática de um HDA. (Favilla,1988)

2. uma falha no processo pode acarretar em retrabalhos nos componentes ou a necessidade de uma completa descontaminação do ambiente de fabricação, significando um longo período de paralisação.
3. o conjunto equipamento/processo conta com uma significativa quantidade e diversidade de conhecimentos integrados.

Sendo que um problema no processo de limpeza pode ser identificado através da estação de controle e medição do grau de limpeza das partes ou através de uma quebra no equipamento de limpeza.

E a possível origem desses problemas pode estar em:

1. o equipamento de limpeza propriamente dito,
2. os produtos químicos utilizados no processo que estão com o seu grau de pureza fora da especificação de processo,
3. o grau de limpeza e descontaminação da peça, encontrar-se fora do especificado, antes do processo de limpeza final,
4. os parâmetros do processo encontrarem-se fora do especificado.

A análise histórica identificou o equipamento de limpeza como aquele que mais frequentemente foi a causa dos problemas nesse processo.

E como fator agravante, constatou-se ainda, um problema de performance do departamento de manutenção, que dispndia um tempo de reparo maior que o esperado, durante a manutenção desse equipamento. Mais precisamente os seguintes fatos foram identificados:

1. utilização de 1056 horas anuais de atuação de mão obra de engenheiros ou outros técnicos indiretos.
2. tempo médio de reparo - MTTR, (da expressão inglesa "Mean Time To Repair") do equipamento em aproximadamente 100% acima dos compromissos da engenharia de fabricação. MTTR esperado = 2.0 horas.

Sendo que os elementos analisados e julgados como responsáveis por esse quadro foram:

- natureza complexa do equipamento e processo,
- presença de novos e diferentes domínios técnicos integrados,
- limitação da formação técnica dos funcionários,
- rotatividade em função dos turnos de trabalho,
- limitações para retreinamento,

As consequências desse conjunto de fatores podem ser citadas como:

1. aumento no custo do produto,
2. atrasos no cronograma de produção,
3. insatisfação do setor de produção.
4. desgaste na relação entre os setores de fabricação, engenharia e manutenção,
5. sobrecarga de atividades do engenheiro de processo,
6. exposição do equipamento a manutenções incorretas,
7. elevada dependência de muitos e diferentes profissionais, dado a pulverização do conhecimento,
8. riscos no cumprimento das metas de produção.

Sendo que nesse cenário identificou-se a oportunidade para a aplicação e teste do modelo proposto. E ainda, o autor entende que entre outros resultados desse trabalho a contribuição no nível prático será o de permitir:

- ter uma base de conhecimento sempre disponível para consulta,
- maior autonomia da linha de fabricação e do departamento de manutenção,

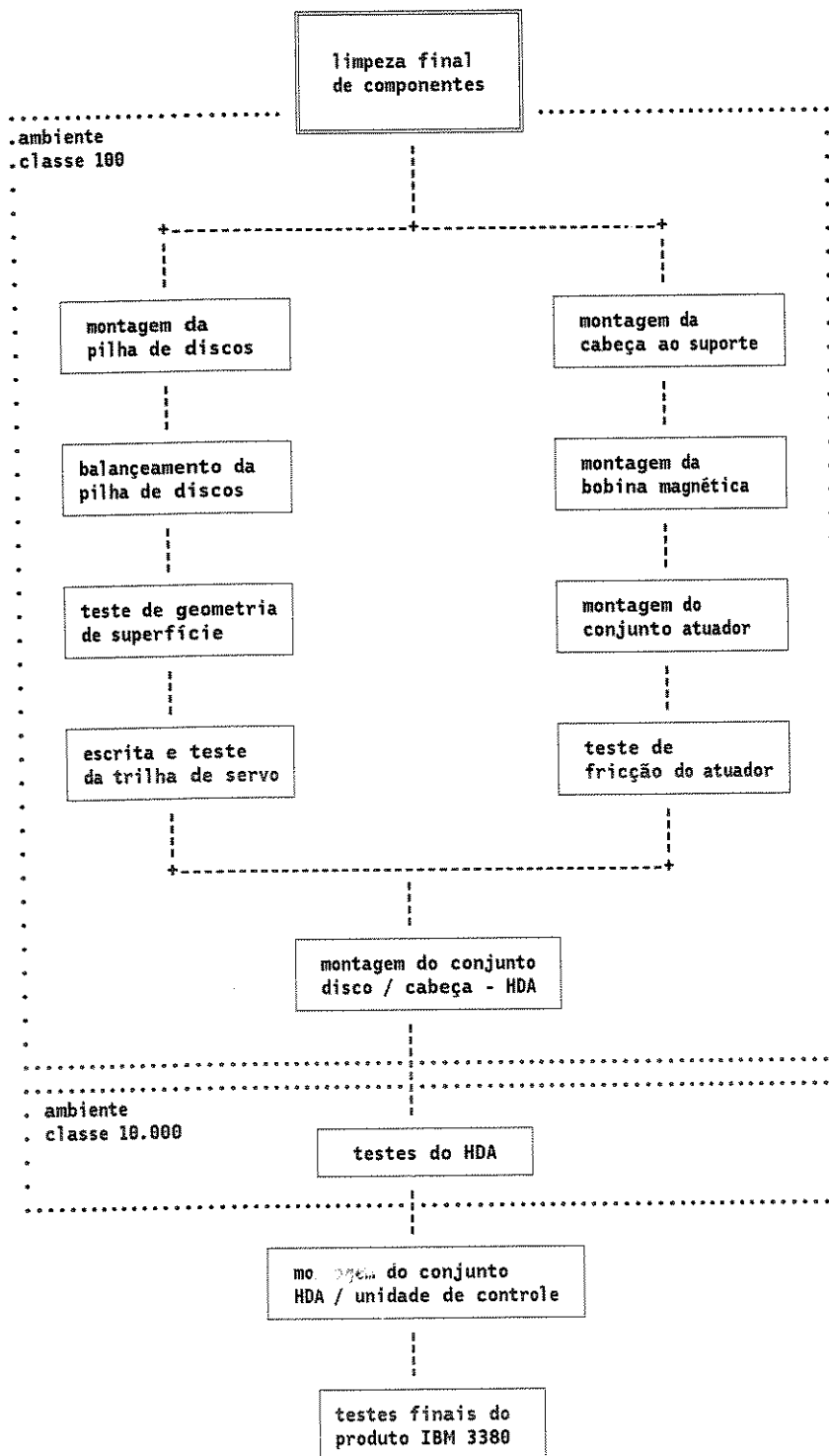


Figura 11. Esquema do processo de montagem do HDA-3380X. (Mulvany & Thompson,1981)

- menor dependência do processo, ao grupo de engenharia de fabricação,
- melhoria da qualidade,

A idéia é que os conhecimentos sobre o equipamento e o processo estejam disponíveis o tempo todo e a todos através do sistema de diagnóstico implementado. De forma que o setor de produção em conjunto com a manutenção, consigam na maioria das vezes, diagnosticar e resolver um problema com sucesso.

4.2.3 Características Gerais do Equipamento e do Processo

Os processos de limpeza e descontaminação compreendem as operações de:

1. limpeza orgânica,
2. limpeza inorgânica,
3. enxague e
4. secagem.

Esse processo é executado em um equipamento que foi desenvolvido e construído em conjunto com a empresa alemã: KLN GmbH. Ver Figura 12. Sendo o regime de trabalho do equipamento é de 2 turnos durante cinco dias por semana.

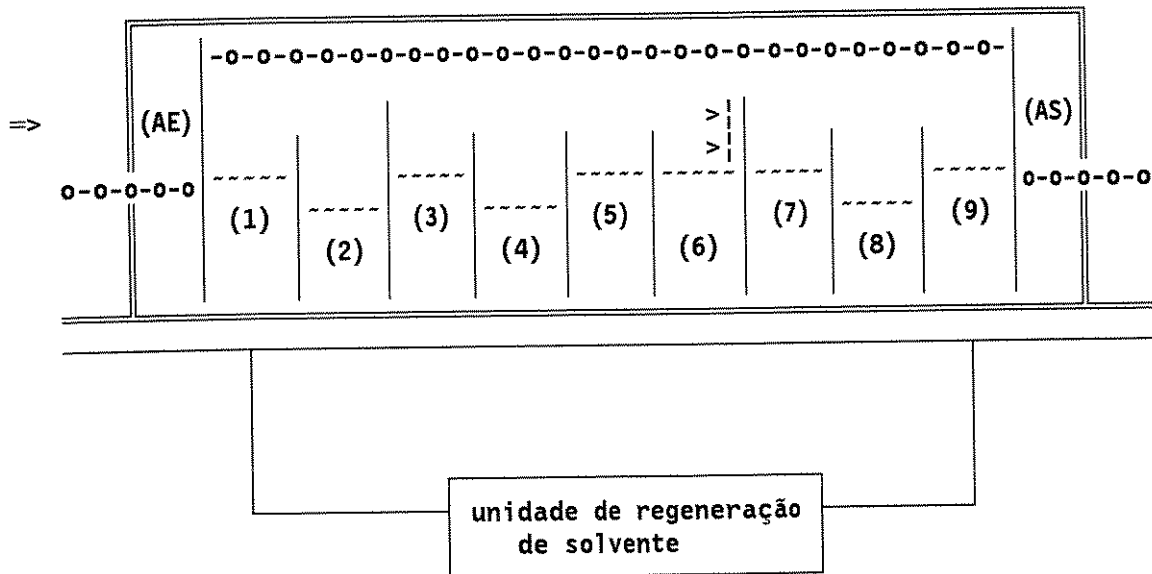
O equipamento tem as seguintes **características gerais**:

- Dimensões:
 1. máquina:
 - comprimento = 14,1 metros
 - altura = 4,0 metros
 - largura = 2,0 metros
 2. painel de controle:
 - comprimento = 4,6 metros
 - altura = 0,6 metros
 - largura = 2,2 metros
- Potência elétrica = 100 kVA (254/440 v; 60 Hz)
- Componentes estruturais:
 - 14 transportadores verticais - elevadores,
 - 06 transportadores horizontais de rolo,
 - 09 tanques,
 - 03 botabas de calor termo-controladas,
 - 06 geradores de ultrassom programáveis,
 - 324 transdutores de ultrassom,
 - 03 bombas de diafragma de alta vazão e pressão,
 - 02 separadores gravitacionais,
 - 01 unidade filtro laminar,
 - 01 unidade de regeneração de solvente em circuito fechado,
 - 01 controlador lógico programável.

METAL CLEANER KLN

(9 - estágios)

| AE | ← Processo de Limpeza e Secagem → | AS |



AE => Ante-câmara de entrada

(1) => solvente tipo (a) + ultrassom + calor
(2) => solvente tipo (a) + vapor

(3) => água desmineralizada + surfactante + ultrassom + calor
(4) => água desmineralizada + ultrassom + calor
(5) => água desmineralizada
(6) => água desmineralizada - imersão + spray

(7) => solvente tipo (b) - imersão + vapor
(8) => solvente tipo (a) + vapor
(9) => solvente tipo (a) + vapor

AS => Ante-câmara de saída

Figura 12. Esquema da máquina de limpeza KLN

- Manipulação de solventes orgânicos de alto grau de pureza, e de água desmineralizada ultra-pura com resistividade mínima de 18 Mohms.
- Componentes mecânicos em aço inoxidável ISO 14541 e ISO 14301.

4.3 Aplicação do Modelo Proposto

O resultado esperado da aplicação do modelo, é um programa de computador que seja capaz de auxiliar o processo de diagnóstico de falhas do equipamento de limpeza apresentado no tópico 4.2.3

Sendo que os trabalhos de desenvolvimento e implementação foram executados respectivamente conforme as etapas a seguir:

1. Aquisição do Conhecimento
2. Definição da Estratégia de Diagnóstico
3. Seleção da Ferramenta de Implementação
4. Implementação do Sistema de Diagnóstico.

4.3.1 Aquisição do Conhecimento

A fase de aquisição do conhecimento especializado compreende as etapas de interação com o especialista para a determinação do conhecimento específico, da análise e da transformação desse conhecimento em estruturas apropriadas, necessárias à construção do Sistema Especialista de diagnóstico. O resultado esperado dessa etapa foi o de identificar, entender, interpretar, organizar e representar adequadamente o conhecimento específico. Bem como, localizá-lo no contexto do processo de diagnóstico.

O processo de aquisição do conhecimento consistiu das etapas de extração (entendimento e interpretação), organização e da representação adequada do conhecimento especializado.

No presente trabalho o engenheiro do conhecimento e o especialista do domínio foram desempenhados pela mesma pessoa, motivado pelo interesse no aprendizado em Inteligência Artificial e de suas aplicações.

Dado que a aquisição do conhecimento é a transformação da capacidade de solucionar problemas de uma fonte de conhecimento em um programa, tem-se que o conhecimento adquirido constitui a estrutura básica do sistema. E a primeira etapa desse processo consistiu da síntese do processo de diagnóstico adotado pelo especialista. Esse trabalho foi fruto da observação e análise do especialista durante as diferentes etapas de vários processos de diagnóstico executados. E o resultado desse trabalho encontra-se expresso no fluxograma apresentado pela Figura 13.

Esse processo ainda permitiu a identificação e a definição dos conhecimentos utilizados, bem como as fontes desses conhecimentos, que foram:

- fontes do conhecimento experimental ou heurístico:
 1. da experiência do especialista:
 - a. através de uma descrição e análise do processo de diagnose de problemas no equipamento, formulados por exemplo, através de fluxogramas.
 - b. através da elaboração de uma lista de fatos ocorridos, contendo a maior quantidade de ocorrências possíveis com o equipamento. Após listados, os problemas foram classificados de acordo com os critérios de:
 - frequência de ocorrência do problema,
 - severidade do problema, e
 - grau de dificuldade da diagnose.
 2. dos registros de ocorrências de manutenção. No ambiente de fabricação do HDA, o procedimento de solicitação de manutenção corretiva varia de acordo com a severidade do problema. Esse fato

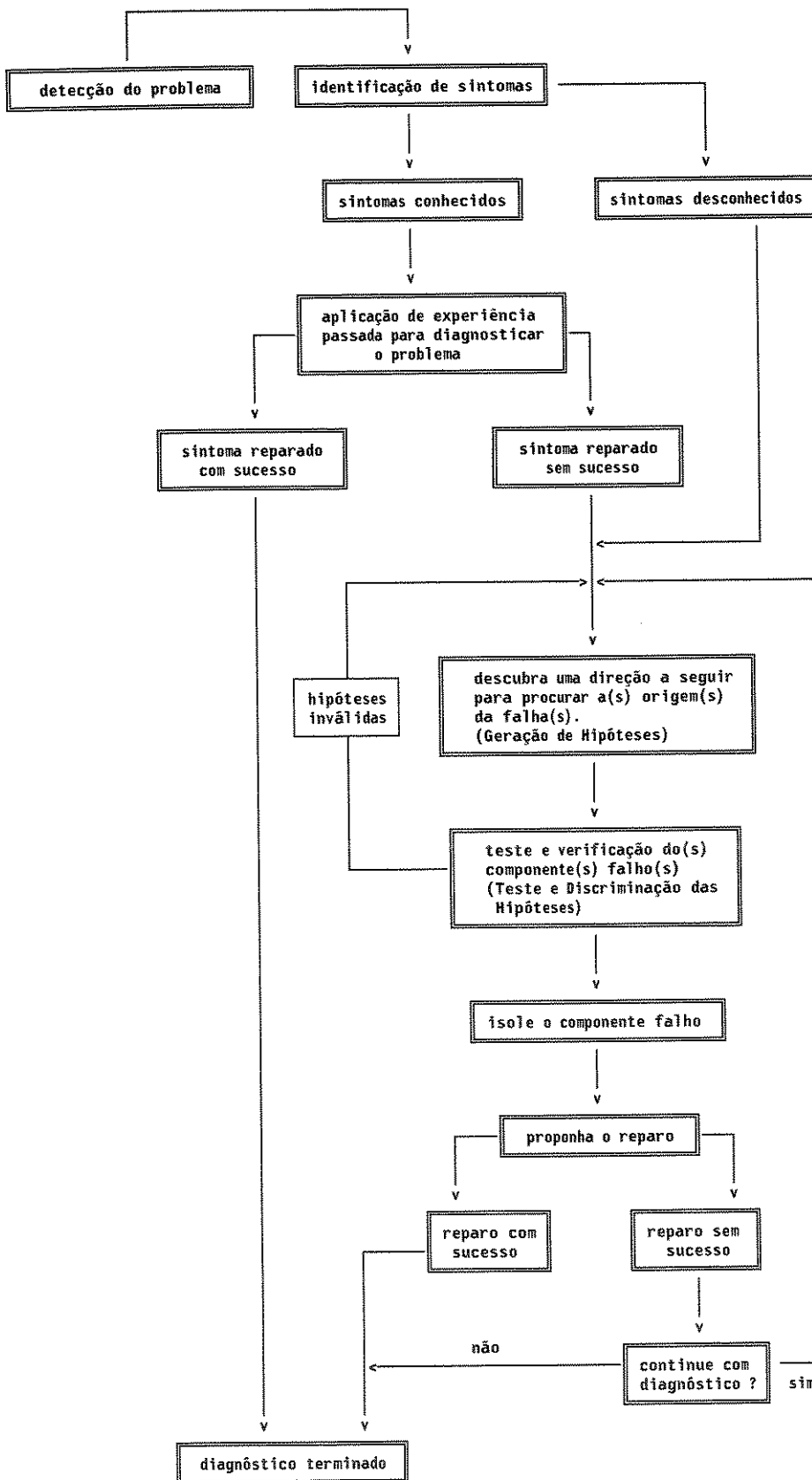


Figura 13. Processo de diagnose humano.

define se a chamada será imediata ou através do preenchimento e entrega de uma requisição padrão. Como na prática esses procedimentos não são sempre seguidos, fez-se necessário complementar a aquisição do conhecimento através de uma série de entrevistas com os operadores da máquina, e com os técnicos de manutenção. Os relatórios de manutenção preventiva executados, também foram analisados e contribuíram no processo de aquisição do conhecimento experimental.

- modelo

De acordo com modelo proposto de diagnóstico, o modelo causal é aquele que deve reger a organização do conhecimento fundamental. Ele se baseia na representação das possíveis falhas e componentes para cada função que possa ser responsável pela falha do sistema.

A aquisição desse conhecimento foi feita de duas fontes básicas:

1. do especialista do equipamento e do processo,
2. das especificações de engenharia e manuais técnicos.

Durante essa fase, decidiu-se que alguns conhecimentos não seriam cadastrados ou seriam parcialmente cadastrados. Essa decisão baseia-se no fato de que tais conhecimentos pertencem a domínios de conhecimentos muito particulares que requerem usuários com formação técnica específica como pré-requisito para a manipulação desses domínios; dessa forma tais conhecimentos continuaram acumulados por especialistas humanos.

Estatisticamente constatou-se uma baixa frequência de problemas nas áreas excluídas, e portanto é possível inferir que o nível de comprometimento da qualidade do sistema será igualmente pequeno no tocante a esses conhecimentos. Os domínios parcialmente cadastrados ou não cadastrados são:

1. Controlador lógico programável,
2. Sistema termodinâmicos das bombas de calor,
3. Geradores de ultrassom.

4.3.2 Definição da Estratégia de Diagnóstico

Essa etapa foi responsável pela estruturação dos conhecimentos, identificados e adquiridos na etapa anterior, de forma a definir uma forma de manipulação que permita a melhor utilização desses conhecimentos.

E conforme apresentado no Capítulo 3, o sistema proposto aborda uma estratégia integrada de diagnose baseada em classificação heurística e baseada em modelo causal através de uma base de conhecimento que integre o conhecimento experimental e o conhecimento fundamental.

O **conhecimento experimental** adquirido através da vivência do especialista com o equipamento adicionado dos relatórios de manutenção e das experiências dos vários profissionais permitiram classificar esse conhecimento experimental, em:

1. conhecimento experimental geral:- caracterizado pelo conhecimento heurístico não relacionado, ou fracamente relacionado a alguma função do equipamento, sendo que esse conhecimento resultou em 07 regras de produção.
2. conhecimento experimental específico:- caracterizado por heurísticas fortemente relacionadas as funções do equipamento, que foram mapeadas em 72 regras de produção distribuídas pelos sistemas de: ultrassom, transporte, refrigeração, regeneração de solvente, elétrico e outros.

A Figura 14, ilustra a organização do conhecimento experimental identificado.

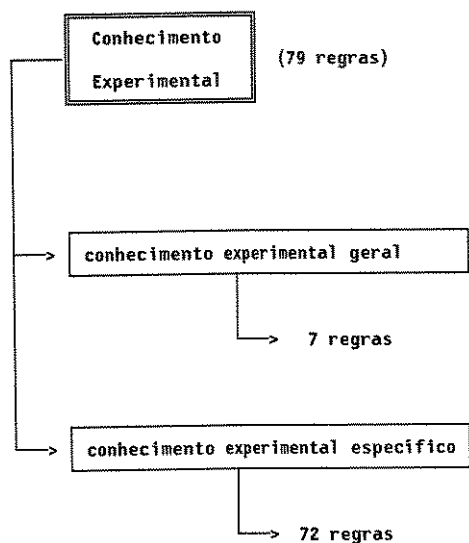


Figura 14. Organização do conhecimento experimental

O **conhecimento fundamental**, organizado de acordo com o modelo causal, procurou modelar o conhecimento através das mesmas funções identificadas pelo conhecimento experimental específico. Dado a adequação dos sistemas identificados às entidades funcionais procuradas e propostas pelo modelo causal, o conhecimento fundamental buscou identificar os principais possíveis candidatos em caso de uma falha (para cada uma das funções). Tal trabalho resultou em 217 regras de produção que expressam o conhecimento fundamental sob o prisma do modelo causal, para as seguintes funções:

1. sistema de ultrassom,
2. sistema de transporte,
3. sistema de refrigeração,
4. sistema de regeneração de solvente,
5. sistema elétrico,
6. outros sistemas.

A Figura 15, ilustra a organização do conhecimento fundamental organizado de acordo com o modelo funcional causal.

De posse dessas estruturas experimental e causal funcional, o próximo passo destinou-se a integrá-los de forma a encontrar a combinação mais adequada entre a organização e a estratégia. Considerando-se ainda, as características do problema, do equipamento e da otimização do processo diagnóstico.

Nesse estágio, a estratégia de diagnóstico visualizada permitiu com que as duas bases de conhecimento fossem agrupadas de três diferentes formas como se seguem:

1. **modelo (1)**:- a base de conhecimento tem o conhecimento fundamental e experimental agrupados separadamente entre si e organizados de acordo com a função a que se aplicam, conforme ilustração da Figura 16.

Nesse arranjo a base de conhecimento do sistema é organizada em módulos distintos em tantos grupos quantos sejam as funções definidas para o equipamento. Sendo que os conhecimentos exper-

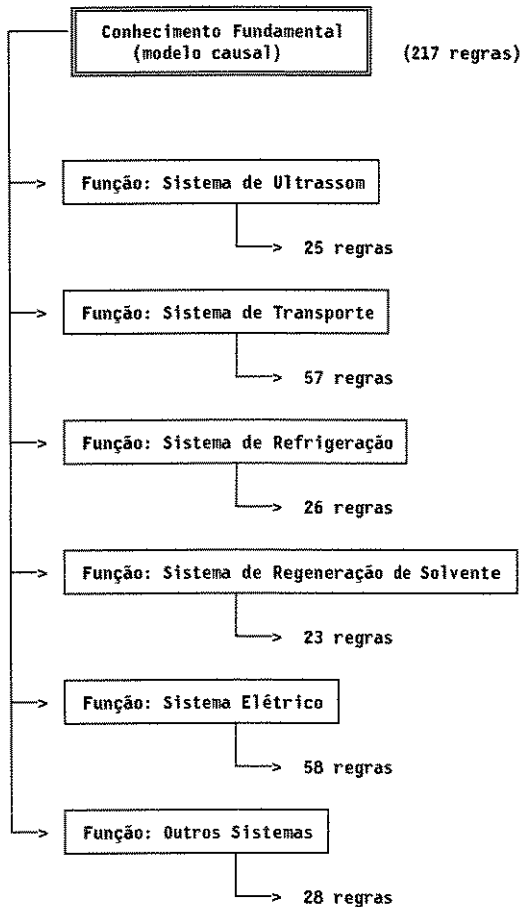


Figura 15. Organização do conhecimento fundamental.

imental e funcional estão organizados separadamente entre si, e de acordo com os grupos funcionais a que estão submetidos.

Uma pequena ilustração é dada no exemplo a seguir:

a. Função 1: sistema termodinâmico

1) conhecimento experimental:

- se o compressor está “suando frio” então aumente a vazão de água valvula 17Y6.
- se não é possível visualizar a zona de vapor então aumente a temperatura desse tanque.
- etc.

2) conhecimento fundamental:

- se a temperatura do fluido de compressão for de aproximadamente 17 oC então compressor comprimirá o fluido nas fases líquido e gasoso alterando as características termodinâmicas do meio e colocando em risco a integridade do compressor.
- se a temperatura dos tanques de solvente for diferente de 28 oC então haverá uma evaporação suficiente que permita a troca de calor eficiente dos condensadores da bomba de calor da zona fria.

b. Função 2: sistema de transporte

1) conhecimento experimental:

2) conhecimento fundamental:

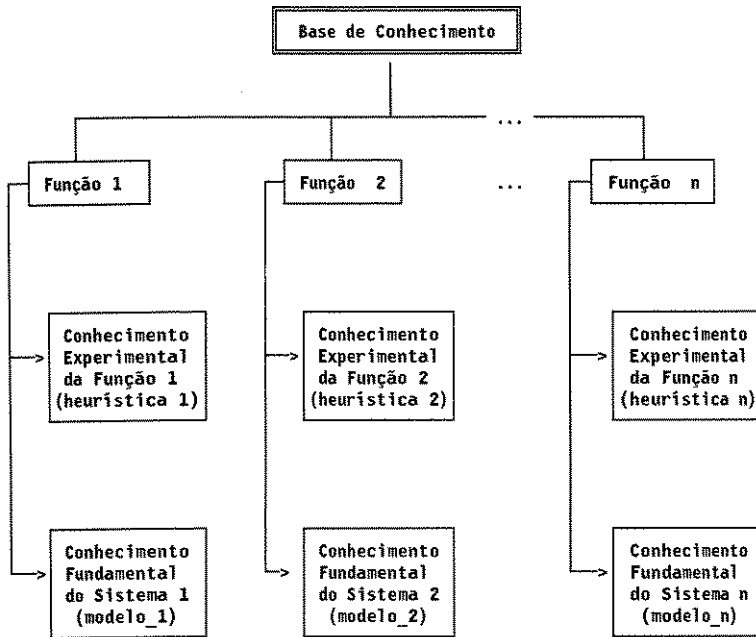


Figura 16. Integração do conhecimento orientado ao aspecto funcional.

c. Função 3: etc.

2. *modelo (2)*:- a base de conhecimento é classificada de acordo com o conhecimento experimental e o conhecimento fundamental distintamente, e estes por sua vez sub-classificados de acordo com a aplicação, conforme a ilustração da Figura 17.

Nesse modelo (2), a forma de organização dos conhecimentos dá-se de acordo com o tipo de conhecimento. Ou seja, a base de conhecimento é organizada em dois módulos somente, que são o conhecimento experimental e o conhecimento fundamental. Sendo que o conhecimento experimental é composto por todas as heurísticas relativas ao equipamento e ao processo, e o conhecimento experimental é composto pelos modelos funcionais dos vários sistemas componentes. Por exemplo:

a. conhecimento experimental:

- se a guilhotina nr.1 da ante-câmara de saída está amassando os containers, então alinhe o sensor 40K9 com a régua de apoio do transportador.
- se está havendo perda excessiva de solvente, então está faltando água no separador gravitacional de solvente.
- etc

b. conhecimento fundamental

- se a chave contatora K301 for desligada então a solenóide 1Y6 da válvula atuará, cortando a alimentação de água do circuito de refrigeração da bomba TYP13.
- se a chave de proteção 12Q0 tiver a corrente máxima ajustada para um valor menor do que 7.0A então é possível que o compressor TYP11 do circuito menor dos tanques 1 e 2 desliguem-se frequentemente ocasionando perda de solventes e alteração no processo.

3. *modelo (3)*:- esse modelo faz uma combinação dos modelos (1) e (2) anteriores, onde a base de conhecimento é organizada conforme o modelo (2) porém cada um dos conhecimentos é organizado internamente conforme o modelo (1), conforme apresentado na Figura 18.

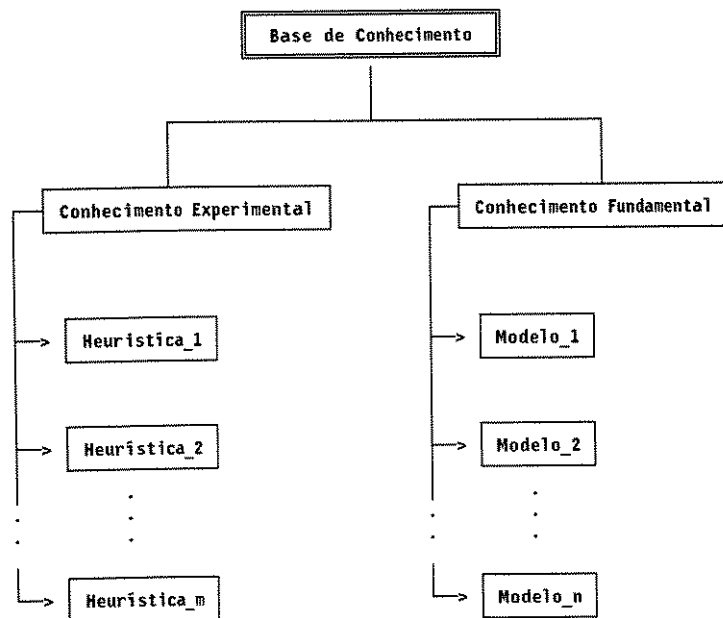


Figura 17. Integração do conhecimento conforme o tipo de conhecimento.

Apesar do modelo (3) ser semelhante ao modelo (2) na forma de organização da base de conhecimento, esses conhecimentos são estruturados de uma forma diferente pois o conhecimento experimental contém as heurísticas genéricas do equipamentos, ou seja, aquelas que não estão associadas ou estão fracamente associadas a alguma função específica do equipamento ou do processo. É o conhecimento fundamental é organizado como o modelo (1), ou seja de acordo com os módulos funcionais da máquina, sendo que esses módulos por sua vez são compostos de heurísticas referentes a essa função e também pelos conhecimentos fundamentais desse módulo. O modelo (3) pode ser ilustrado pelo seguinte exemplo:

a. conhecimento experimental:-

- se a máquina parou de repente, sem qualquer alarme então verifique se o fornecimento de ar comprimido está normal.
- se a temperatura do tanque 3 não está aumentando, então verifique a bomba de recirculação número 4B0.

b. conhecimento fundamental:-

1) função 1:- sistema de transporte

- heurística do sistema de transporte
 - se o sistema de transporte horizontal parar, procure mexer no alinhamento do carro horizontal trazendo-o mais próximo da porta.
- modelo funcional do sistema de transporte
 - as chaves 39K4.1, 50K2, 50K3, e 50K4 fazem o acionamento do movimento lento para baixo do elevador 4.

2) função 2:- sistema de regeneração de solvente

- heurística do sistema de regeneração de solvente
 - se a cor do solvente está muito alaranjada então pare de adicionar aditivo secante.

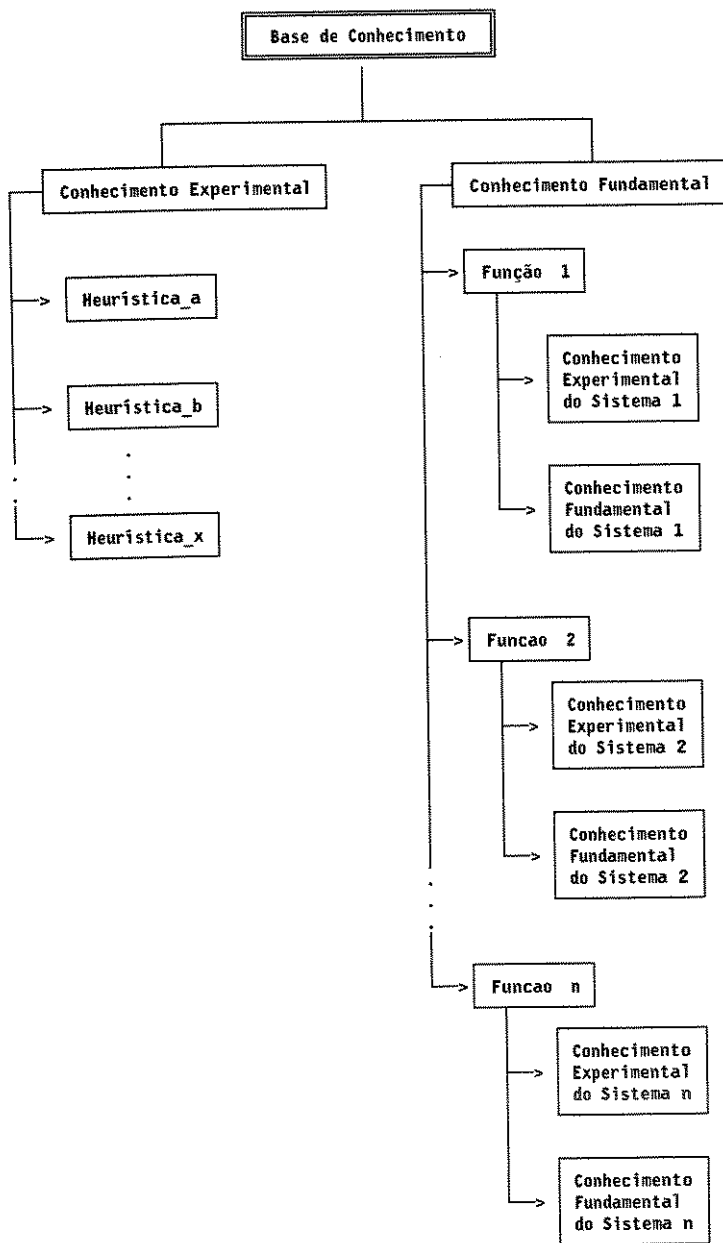


Figura 18. Combinação das alternativas (1) e (2).

- modelo funcional do sistema de regeneração de solvente
 - se o rele térmico Q301 deve ter a corrente ajustada para um valor diferente de 7.5 A então proceda o ajuste do mesmo para 7.5 A.

A decisão sobre a mais adequada organização do conhecimento e estratégia, para esse caso, foi tomada analisando-se o processo de diagnose adotado pelo especialista humano - ver Figura 13.

A avaliação do processo de diagnóstico do especialista humano permitiu concluir que a estratégia de raciocínio superficial é utilizada como o passo inicial para o processo de raciocínio profundo. Se o componente falho não puder ser identificado utilizando-se as estratégias de raciocínio superficial, então a estratégia de raciocínio profundo é iniciada. A identificação desse processo permitiu desenvolver um projeto de sistema em que o usuário não identifique uma mudança explícita da estratégia de raciocínio raso para o

racíonio profundo desde que as estratégias de diagnóstico sejam mantidas transparentes para o usuário; e as interações entre o usuário e o sistema sejam conduzidas de forma natural e contínua. Tem-se nesse exato ponto, a definição da estratégia de manipulação do conhecimento, ou estratégia diagnóstico do sistema.

A identificação desse processo também consolidou a escolha do modelo (3) como o modelo mais adequado para a estruturação da base de conhecimento.

E portanto, tem-se nesse estágio a materialização do modelo integrado de diagnóstico, proposto no Capítulo 3, na forma da estrutura apresentada pela Figura 19. Tem-se como próximo passo a implementação do resultado aqui obtido.

4.3.3 Seleção da Ferramenta de Implementação

O propósito dessa etapa é o de selecionar uma ferramenta do tipo "shell" para a implementação do modelo integrado de diagnóstico. Assim sendo, foram estudadas algumas ferramentas disponíveis no mercado brasileiro; a saber: G2 da empresa Gensyn, Nexpert da empresa Neuron Data, AionDS da empresa Trinzic e ESE da empresa IBM.

Sendo que no desenvolvimento do presente sistema optou-se pelo uso ferramenta denominada: ESE - Expert System Environment. Trata-se de um produto da empresa IBM, para a implementação de sistemas baseados em regras, que foi desenvolvido no Centro Científico da IBM de Palo Alto, nos Estados Unidos da América.

E os motivos para a escolha dessa ferramenta basearam-se nos seguintes fatos:

1. imediata disponibilidade do ESE no sistema de computadores da IBM,
2. domínio da ferramenta por outros profissionais da empresa,
3. adequação do ESE para aplicações em sistemas de diagnose,
4. a existência de documentação e suporte adequados.

4.3.3.1 Descrição da Ferramenta de Desenvolvimento

O Expert System Environment é um programa computacional utilizado na criação de sistemas baseados em conhecimento voltados principalmente as áreas de identificação, seleção, diagnose e planejamento. Sendo que sua base de conhecimento ainda pode auxiliar na tomada de decisões nas seguintes áreas: comercial, comunicação, computação, educação, engenharia, finanças, fabricação, medicina, militar e de transportes (Application Programming Guide, 1990).

O Expert System Environment consiste de dois módulos:

1. ESDE - Expert System Development Environment

Trata-se do ambiente de desenvolvimento, que provê os recursos necessários para a construção da base de conhecimento de um dado domínio específico. Para se criar essa base de conhecimento, o engenheiro do conhecimento ou o desenvolvedor do sistema, reúne as informações ou extrai os conhecimentos necessários - do especialista humano e/ou de outras fontes, que uma vez organizados usa o ESDE para se criar o domínio do conhecimento. Esse conhecimento é inserido em uma base de conhecimento composta das seguintes entidades ou objetos: parâmetros, regras, blocos focais de controle - FCB (do Inglês "Focus Control Block"), grupos e telas (Expert System Development Environment User's Guide, 1989).

2. ESCE - Expert System Consultation Environment

É o ambiente de consulta que permite que os usuários finais consultem o sistema baseado em conhecimento. Durante o processo de consulta o sistema especialista faz perguntas ao usuário final

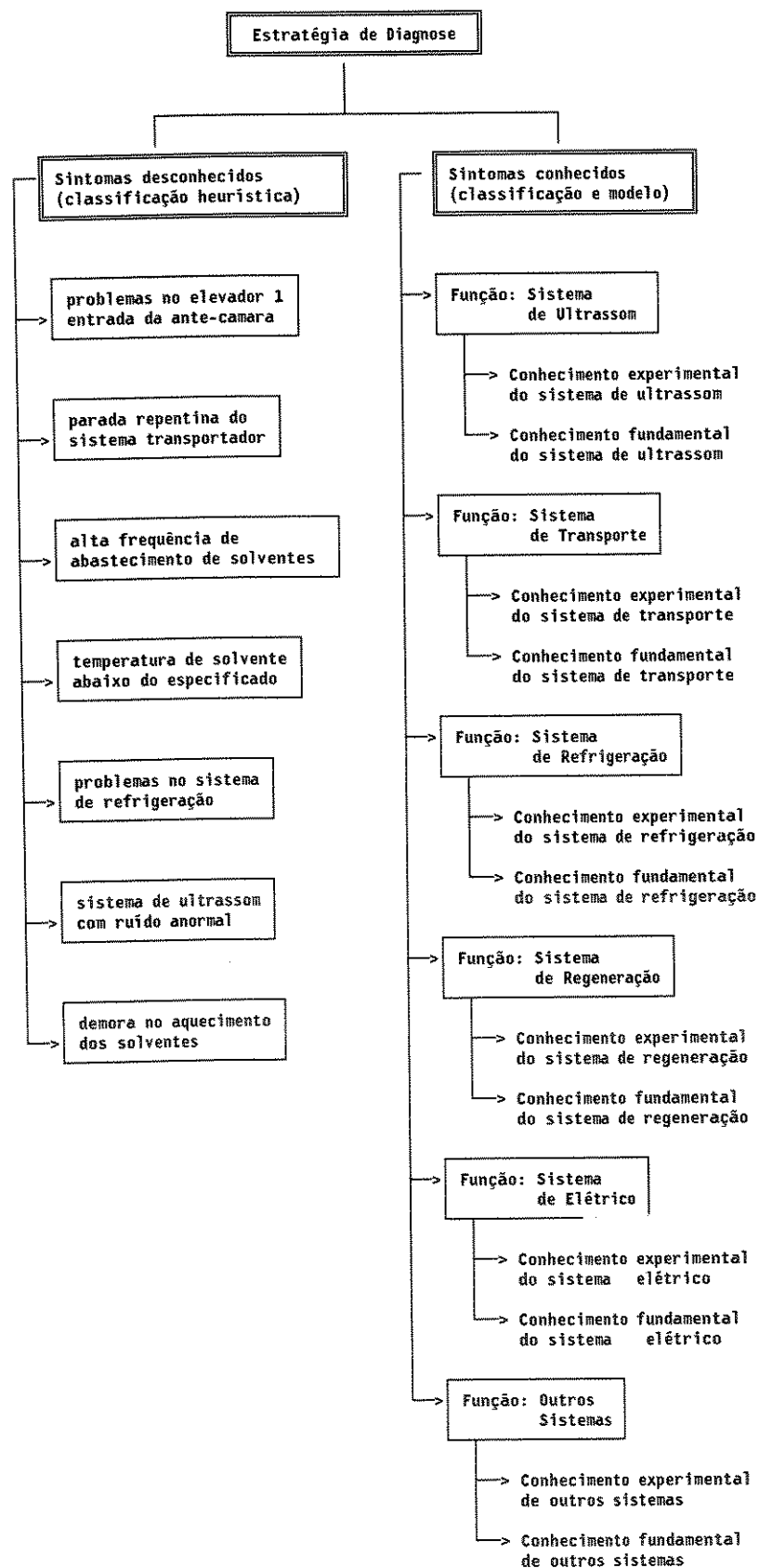


Figura 19. Implementação do modelo da base de conhecimento integrada.

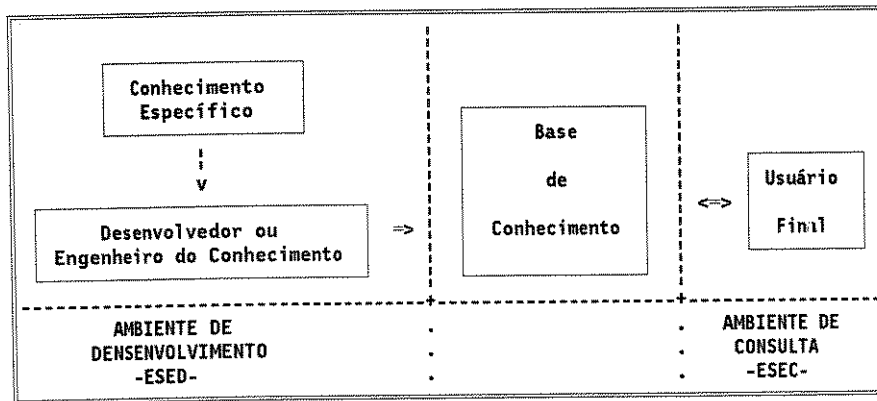


Figura 20. Relação entre o ESE e o usuário. (Application Programming Guide,1990)

sobre um dado problema pertinente ao domínio da base de conhecimento, e então fornece ao mesmo uma solução ou resposta (Expert System Consultation Environment User's Guide,1990).

A ilustração da Figura 20, mostra o relacionamento entre as pessoas e os conhecimentos envolvidos na criação e uso da base de conhecimento.

- A fonte de conhecimento é o conhecimento do domínio específico usualmente fornecidos por um ou mais especialista humano que ajudam a desenvolver e a manter a base de conhecimento.
- O desenvolvedor ou engenheiro do conhecimento assiste o especialista na tradução do conhecimento para o formato de entendimento do computador e especifica técnicas de controle necessárias que conduzem a solução do problema.
- O usuário final interativamente consulta a base de conhecimento para solucionar problemas.

4.3.3.1.1 Características Operacionais: A ferramenta ESE é executada em um sistema IBM/370 e pode ser utilizada nos sistemas operacionais:

- VM/CMS - "Virtual Machine" / "Conversational Monitor System"
- MVS/TSO - "Multiple Virtual Storage" / "Time Sharing Operation"

Sendo que apenas o sistema operacional VM/CMS suporta o desenvolvimento de aplicações.

O ESE ainda permite que sua capacidade de trabalho seja incrementada pela criação de programas que podem:

1. Acessar informações externas dinamicamente:- permite, através de programas (rotinas externas), quando a base de conhecimento está operando em modo interativo com um dos seguintes sistemas operacionais:
 - VM - "Virtual Machine", ou
 - MVS/TSO - "Multiple Virtual Storage" / "Time Sharing Operation", ou
 - CICS - "Customer Information Control System", ou
 - IMS. - "Information Management System".

As rotinas externas podem ser escritas em uma das seguintes linguagens: assembler, cobol, fortran, PL/I, pascal ou rexx.

2. Embutir a base de conhecimento:- permite, no modo embutido, que um programa aplicativo consulte a base de conhecimento e os resultados da consulta sejam retornados a aplicação. Nesse modo de

operação, uma série de diálogos ocorrem entre o ESCE - ambiente de consulta, e o programa aplicativo. Para se embutir o ESCE os programas aplicativos devem ser escritos em: cobol II, PL/I ou assembler. É permitido que o programa aplicativo embuta o ESCE executando em um dos ambientes de MVS:

- CICS - "Customer Information Control System", ou
- IMS - "Information Management System", ou
- TSO - "Time Sharing Operation".

4.3.3.1.2 Estruturas Funcionais: A base de conhecimento dessa ferramenta é composta por:

1. Objetos da base de conhecimento

Objetos da base de conhecimento são os elementos que definem a base de conhecimento. Esses elementos são: parâmetros, regras, FCBs (do inglês "Focus Control Block"), grupos e telas.

Sendo o domínio do conhecimento é composto pelos seguintes objetos:

- fatos do domínio, chamados de parâmetros.
- relacionamento entre os fatos do domínio, chamado de regras.
- unidades de trabalho, chamados de FCBs - blocos de controle.
- coleção de parâmetros, regras ou FCBs, chamado de grupos.
- apresentador, chamado de telas.

2. Métodos de inferência

Método de inferência é a lógica que define como o problema é desenvolvido na base de conhecimento. As duas técnicas de lógica que podem ser utilizados no método de inferência são conhecidos por: *encadeamento para trás*, que é direcionado a um objetivo e trabalha para trás - a partir de um dado objetivo, até encontrar os dados que o suportem. E o *encadeamento para frente*, que é direcionado a dados e trabalha para frente - a partir um dado conhecido até encontrar uma nova informação.

4.3.4 Implementação do Sistema de Diagnóstico

O propósito dessa etapa foi o de codificar o conhecimento especializado e a estratégia de manipulação desse conhecimento na ferramenta de desenvolvimento selecionada, e de acordo com o modelo proposto por esse trabalho. Sendo que o resultado esperado foi o desenvolvimento do Sistema Especialista de diagnóstico propriamente dito.

A implementação do problema modelado, conforme o modelo da figura 19 - pag.91, dentro da arquitetura proposta e através da ferramenta ESE, resultou em dois níveis de FCB (da expressão inglesa "Focus Control Block"). Onde o primeiro nível formado pela FCB raiz tem por objetivo determinar a possível função que apresentou o problema, bem como encaminhar o processamento ao FCB adequado do segundo nível - se necessário. Nesse primeiro nível encontram-se os conhecimentos experimentais de natureza geral. E o segundo nível é composto por vários FCBs que tem por objetivo a análise mais detalhada para a diagnose do problema. Que contemplam o conhecimento experimental específico e o conhecimento fundamental. Um esquema da estrutura dos FCBs no ESE pode ser visualizada na Figura 21. O sistema utilizou como método de inferência, a técnica de encadeamento para trás

O FCB raiz que contém a maior parte do conhecimento experimental sem uma relação causa e efeito. E tem por objetivos:

- informar ao usuário a finalidade do sistema especialista, através de uma breve descrição e de orientações sobre o programa.

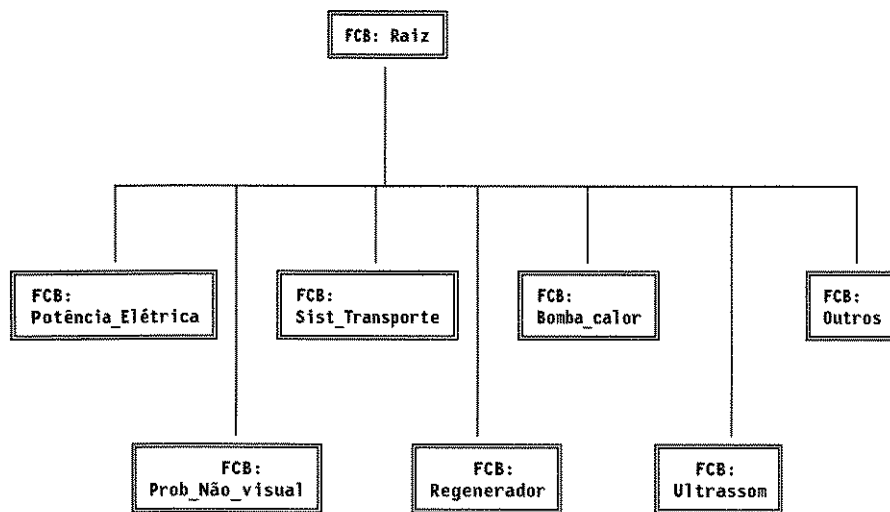


Figura 21. Estrutura dos FCBs do presente trabalho.

- auxiliar o usuário na determinação, de qual a função da máquina que está apresentando o problema.
- chamar o FCB adequado para proceder a análise adequada do sistema.

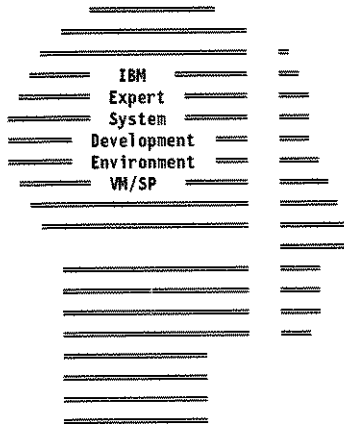
Os demais FCBs, por sua vez, tem por objetivo interagir com o usuário através de perguntas para diagnosticar o problema.

O sistema especialista implementado, sofreu seis revisões até a versão final disponibilizada, e implementou:

- 247 parâmetros
- 296 regras
- 009 FCBs
- 016 telas

que refletiram a implementação do modelo proposto.

Tem-se a seguir uma pequena amostra da implementação, através da sua interface com usuário.



(C) Copyright
IBM Corp. 1985, 1990

Release 3.0 Date 03/30/92

PF1 - Help PF2 - List PF3 - End
Enter Knowledge Base Name, or press PF2 for selection list.
=>

S I S T E M A E S P E C I A L I S T A

P A R A

D I A G N Ó S T I C O D A C L E A N E R K L N

Versão 1.0 - 07/02/92

*** Tecla <ENTER> para continuar ***

=>

SISTEMA ESPECIALISTA PARA DIAGNÓSTICO DA CLEANER KLN

Este Sistema Especialista tem por objetivo auxiliá-lo no diagnóstico de problemas com a Metal Cleaner - KLN.

Por tratar-se de um equipamento com diversificadas funções, por favor, siga todos os procedimentos de segurança IBM quando trabalhar com: Alta Tensão, Produtos Químicos, Alta Pressão, Sistemas em Movimento (conveyors e elevadores), Sistemas de Refrigeração, etc...

*** Teclle <ENTER> para continuar ***

⇒

SISTEMA ESPECIALISTA PARA DIAGNÓSTICO DA CLEANER KLN

Para iniciarmos o nosso trabalho, gostaria que você conversasse com a pessoa que solicitou o seu auxílio, e respondesse a seguinte pergunta:

Através de qual das formas ao lado problema se manifestou ?

Coloque um 'x' na resposta apropriada

_ Visual através do painel de controle
_ Não visual

PF1 Help PF2 TogglePF PF3 End
PF4 What PF5 Question PF6 Unknown
PF7 Up PF8 Down PF9 Tab
PF10 How PF11 Why PF12 Command

15:06:55

⇒

SISTEMA ESPECIALISTA PARA DIAGNÓSTICO DA CLEANER KLN

Visual, quer dizer: Você sabe dizer que existe um problema na cleaner por que existe uma lâmpada ou led que está indicando esse problema no painel de controle da cleaner. Não visual, quer dizer: A cleaner parou e não existe nenhum aviso luminoso no painel de controle ou nenhuma indicação visual no quadro sinóptico. A cleaner simplesmente parou de funcionar.

x Visual através do painel de controle
- Não visual

PF1 Help PF2 TogglePF PF3 End
PF4 What PF5 Question PF6 Unknown
PF7 Up PF8 Down PF9 Tab
PF10 How PF11 Why PF12 Command

15:06:59



4.4 Avaliação do Sistema Implementado

O tempo total dispendido para a aplicação do modelo proposto, foi de seis meses e compreendeu as etapas de:

1. aquisição do conhecimento,
2. definição da estratégia de diagnóstico,
3. seleção da ferramenta de implementação, e
4. implementação do sistema.

Sendo que o trabalho final para disponibilização do sistema ao setor de produção e de manutenção compreendeu na consolidação das tarefas de:- treinamento do usuário, verificação, validação e teste de aceitação do usuário. Essas tarefas tiveram a duração de um mês e contou com as seguintes atividades:

1. apresentação geral do sistema:- fez-se uma apresentação geral da idéia e dos objetivos desse trabalho. E também sobre o produto ESE e a forma de implementação do Sistema Especialista de diagnóstico.
2. treinamento:- foram fornecidos materiais de instrução para o uso do sistema, e exercícios de manipulação do programa.
3. elaboração dos testes de caso:- o desenvolvedor do sistema em conjunto com todos os usuários, elaboraram um conjunto de testes de caso, utilizando-se da experiência individual com o equipamento, da documentação de projeto e da consulta ao equipamento em si.
4. aplicação dos testes de caso:- aplicação dos testes gerados e verificação *in loco*, dos problemas diagnosticados (quando possível),
5. análise dos testes de caso:- através validação ou não dos resultados obtidos dos testes de caso aplicados; da correção de falhas identificadas e da implementação de melhorias e sugestões,
6. avaliação da aceitação do usuário:- durante a elaboração e execução dos testes de caso, os usuários foram observados e requisitados a fazerem comentários sobre a interface de comunicação do sistema.

E o sistema especialista ora implementado foi denominado de:- Sistema Especialista para Diagnóstico da Metal Cleaner KLN. Durante essa última etapa, os usuários finais (operador líder da máquina, mecânicos e eletricitas de manutenção e o supervisor de manutenção), juntamente com o desenvolvedor do sistema participaram da etapa final de avaliação. As avaliações do sistema por parte do especialista do domínio ajudaram a determinar a precisão do conhecimento embutido e a precisão de qualquer conselho ou conclusão fornecido pelo sistema. E as avaliações por parte dos usuários ajudaram a determinar a utilidade do sistema, a extensão das capacidades do sistema, a facilidade de interação, a intelegibilidade e a credibilidade dos resultados, a eficiência e velocidade, a confiabilidade, e as limitações do sistema.

O objetivo dessa fase foi o de avaliar a validade do modelo proposto, através de sua aplicação. Sendo que para uma avaliação mais completa do modelo foram cobertos os aspectos de verificação, validação e aceitação do usuário:

4.4.1 Verificação

A verificação que tem por objetivo determinar se o processo implementado satisfaz as especificações, foi efetuada pelo especialista do domínio, que reviu cada componente a nível de consistência interna e a nível de conjunto. Como consistência interna tem-se que informações conflitantes não devem existir no sistema, ou ainda que o sistema não forneça nenhuma informação conflitante através da verificação da base de conhecimento implementada.

O processo de verificação deu-se durante todas as fases de desenvolvimento do sistema com o objetivo de sempre assegurar o cumprimento dos requisitos pré-estabelecidos de cada fase e da fase anterior. Essa verificação se deu através da busca de resposta para questões simples como:

- Existem erros no processo em questão ?
- Existem resultados errôneos sendo gerados ?
- O sistema falha, sob determinadas condições ?
- Quão confiável e consistentes são as respostas do sistema ?
- Como estão as respostas da fase anterior ?

e uma ênfase particular foi prestada durante a fase de teste funcional do sistema de diagnóstico quando não se encontrou nenhum problema de inconsistência ou conflito de informações na base de conhecimento ou na sua manipulação.

4.4.2 Validação

A validação, que visa determinar se o sistema implementado desempenha satisfatoriamente nas tarefas do mundo real, para o qual foi criado, contou com a técnica de avaliação através de métodos empíricos. O método de avaliação denominado:- avaliação através de métodos empíricos, é decorrente do uso experimental do sistema baseado em conhecimento, em situações controladas. De uma forma geral, os métodos empíricos envolvem a execução e análise de testes de casos (do inglês “test cases”), ou testes previamente elaborados que simulam e exercitam o sistema em vários e diferentes modos e situações. Os resultados obtidos são verificados pela sua precisão. Ou de uma forma mais direta, o teste consiste em simular uma dada e conhecida situação no sistema especialista e comparar o resultado obtido com o resultado esperado, ou resultado padrão. Os métodos empíricos também podem ser utilizados no auxílio dos testes de aceitação do usuário, através da observação dos mesmos durante a execução dos testes de caso, quando torna-se possível detectar problemas na interface do sistema com o usuário e outros problemas ergonômicos. Os usuários também podem ser questionados sobre o ver e sentir (do inglês “look and feel”) sobre o sistema.

Os testes de casos aplicados para a validação do sistema desenvolvido encontram-se em anexo a esse trabalho. Sendo que os resultados dos testes de casos aplicados encontram-se sumarizados na Tabela 6 e a análise desses resultados permitiu a identificação dos seguintes problemas:

1. identificação de áreas não diagnosticáveis pelo sistema:- As áreas identificadas pelos testes de caso que não são contempladas na base de conhecimento do presente sistema especialista são:
 - Controlador Lógico Programável: porque a manutenção desse sistema exige um conhecimento muito específico do equipamento, da linguagem de programação (STEP 5), e do programa da máquina.
 - Sistema Termodinâmico das Bombas de Calor: esses sistemas tem um diagnóstico parcial que se restringem aos módulos de alimentação elétrica e de segurança, porém um diagnóstico mais completo exige equipamentos específicos para manutenção de sistemas frigoríficos bem como conhecimentos da termodinâmica desses sistemas.
 - Gerador de Ultrassom: esse módulo também tem um diagnóstico parcial em função do conhecimento específico necessário referente as tecnologias de geração e controle de ultrassom.
2. identificação de problemas não contemplados na base de conhecimento:- Quatro problemas identificados pelos testes foram incluídos no FCB - OUTROS; que são: bomba de recirculação e filtragem do tanque 7, ventiladores das placas transdutoras do tanque 1, exaustor sobre os tanques 3 e 4, bomba de abastecimentos dos tanques 1 e 2.
3. Melhoria do conteúdo de apresentação da interface homem/máquina principalmente através do ajuste da linguagem utilizada, ou seja, usar a linguagem mais simples e clara possível de forma a não gerar interpretações duvidosas e permitir que o usuário responda de forma adequada e precisa.

4. Utilização do comando “WHAT” do ESE, de forma mais intensa. Essa função permite com que a interação entre o usuário e o sistema, através das telas, seja incrementada através de uma complementação ou de uma extensão da informação que o sistema apresenta para uma solicitação de entrada de dados ou de uma ação para verificação.

Tabela 6. Teste de casos	
Total de testes executados	53 testes
diagnósticos corretos	47 casos
diagnósticos incompletos	06 casos

4.4.3 Aceitação do Usuário

A aceitação do usuário que possibilita avaliar a aceitação do usuário ao sistema adotou as técnicas de:

- revisões de sistema:- durante a fase desenvolvimento e implementação.
- entrevistas e observação formal:- durante a fase de treinamento dos usuários e durante a fase de geração e execução dos teste de casos.
- registro no sistema:- além da capacidade do ESE registrar todos os passos das consultas feitas ao sistema, posteriormente implementou-se um pequeno questionário, executado ao término de cada consulta, de forma a permitir uma avaliação do desempenho do programa, em termos de: utilidade do sistema, registro de novos problemas (aumento da base de conhecimento), tempo ganho com a utilização do sistema.

Os resultados obtidos das técnicas de avaliação da aceitação do usuário permitiram identificar os seguintes aspectos sobre o sistema implementado:

1. o sistema tem um bom nível de comunicação com o usuário através da interface oferecida,
2. as telas geradas, de uma forma padronizada, facilitam o aprendizado do usuário com relação a interação com o sistema, devido a manipulação de telas quase que idênticas em todos os passos,
3. as informações, solicitadas pelo sistema, que exigem como resposta entradas com escolha do tipo “sim” ou “não” foram melhor aceitas pelo usuário pela sua simplicidade e objetividade,
4. uma interface gráfica de alta resolução, ou a aplicação de multimídia seriam extremamente desejados para uma maior e melhor explicação das ações durante o processo de diagnóstico e de reparo do problema.
5. o processo repetitivo de consulta permite com o que usuário tenha um aprendizado sobre o processo de diagnóstico.
6. a qualidade de comunicação do sistema varia de usuário para usuário, baseado nas diferenças das características pessoais e dos diferentes níveis de formação de cada usuário.

Pode-se ainda avaliar os resultados da implementação desse sistema de uma forma mais geral quando consideramos que os maiores usuários são os funcionários do Departamento de Manutenção e os operadores da máquina, ou seja, sob o aspecto desses usuários a implantação desse sistema significou:- um maior e melhor conhecimento da máquina e de seu funcionamento, um maior e melhor conhecimento do processo de limpeza e descontaminação, introdução a educação de tecnologia de informação, e desmistificação de sistemas computacionais. Esse processo de educação foi mais explicitamente constatado durante os exercícios de consultas ao sistema, onde inúmeras dúvidas foram esclarecidas referentes ao projeto da máquina, e a concepção do processo de limpeza e descontaminação.

Significou também uma desmitificação com relação aos sistemas computacionais. Via de regra, os operários de chão de fábrica apresentam-se céticos e temem os sistemas automatizados. Em sua maior parte fundamentada no desconhecimento e na incompleta interpretação dos papéis a serem executados por esses sistemas, como por exemplo: a total substituição do trabalho humano por sistemas automatizados do tipo: robôs, máquinas automáticas, e outros que eventualmente possam ocasionar o desemprego. Nesse sentido, essa implementação também pode melhorar o nível de educação, permitindo reduzir a distância entre o homem e o computador, bem como promover discussões que melhoraram o nível de entendimento do papel da automatização, de seus benefícios e limitações nos processos produtivos.

4.5 Resultados da Implementação do Sistema

Durante a utilização desse sistema no nível de chão de fábrica, os seguintes resultados foram obtidos:

1. Conforme os relatórios de manutenção, o sistema de diagnóstico permitiu a redução do tempo médio de reparo da ordem de 50%; onde grande parte do tempo de reparo, era destinado ao diagnóstico dos problemas e não ao reparo do problema propriamente dito. Figura 22.
2. Conforme os relatórios de engenharia, o sistema de diagnóstico permitiu a redução da utilização de mão de obra indireta, ou suporte da engenharia de processo a manutenção, em aproximadamente 47% em tempo. O que significa que o tempo de suporte da engenharia de processo a manutenção, foi reduzido de 1056 horas/ano para 496,3 horas/ano.
3. Os relatórios da manutenção, o sistema de diagnóstico reportaram os seguintes resultados do sistema automático de diagnóstico:
 - diagnósticos completos: 89.3%,
 - diagnósticos incompletos: 10.7%

Onde os resultados incompletos referem-se a a base de conhecimento propositadamente não cadastrada.

4. O aumento da disponibilidade do engenheiro de processo, a partir da implementação do sistema, permitiu com que as especificações de engenharia fossem revisadas e o plano de manutenção preventiva fosse melhorado. O que refletiu no aumento do tempo médio entre falhas.
5. A redução do tempo médio de reparo e o aumento do tempo médio entre falhas aumentou a disponibilização do equipamento a produção.

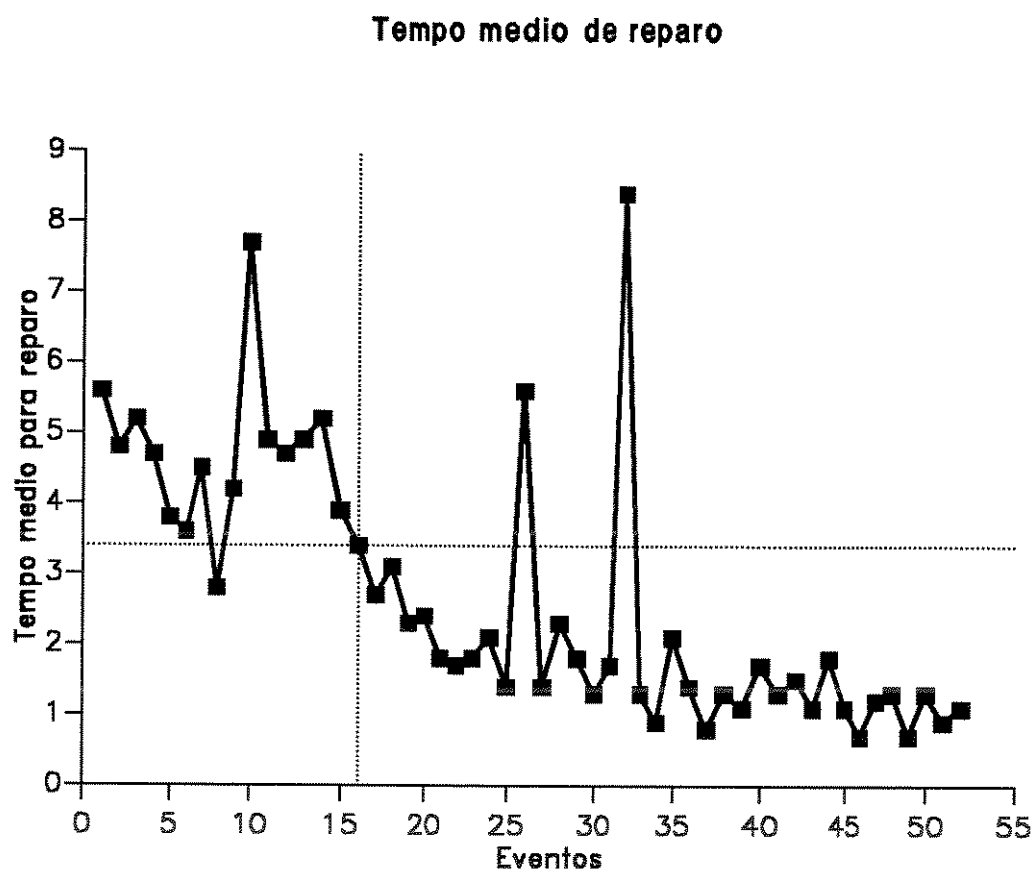


Figura 22. Tempo médio para reparo.

4.6 Resumo

Esse capítulo discorreu sobre a aplicação do modelo proposto por esse trabalho. Essa aplicação deu-se nas instalações da IBM Brasil, no processo de limpeza de componentes metálicos da fabricação de unidades de disco magnético 3380 modelos J e K.

Sendo que o equipamento escolhido para a aplicação do modelo justificou-se pelo grau de dificuldade da manutenção corretiva e da alta dependência dos conhecimentos do engenheiro de processo.

Os aspectos fundamentais abordados nesse capítulo concentram-se no processo de aplicação do modelo, que contou respectivamente com as etapas de: aquisição do conhecimento, definição da estratégia de diagnóstico, seleção da ferramenta de implementação e a implementação do Sistema Especialista de diagnóstico.

Sendo que no processo de avaliação do Sistema Especialista foram adotadas as técnicas de verificação, validação e de aceitação do usuário.

Finalmente, foram apresentados os resultados da avaliação do sistema que comprovaram a validade do modelo proposto, fundamentados nos resultados dos processos de avaliação e da qualidade dos resultados obtidos da implantação do sistema no nível de chão de fábrica.

5.0 Conclusões

O presente trabalho atingiu o objetivo proposto, comprovando a validade e a eficácia na utilização de um sistema de diagnóstico automático com uma abordagem integrada baseada em modelo causal e em classificação heurística, através da integração dos conhecimentos experimental e fundamental, visando o auxílio a manutenção de máquinas e equipamentos.

Essa conclusão fundamenta-se nos consistentes resultados obtidos da aplicação do modelo proposto, através da avaliação a níveis de: verificação, validação e de aceitação do usuário; do sistema de diagnose implementado. E comprovados no nível prático, pelos bons resultados obtidos pela implantação do sistema especialista no nível de chão de fábrica.

No empenho dessa tarefa as seguintes conclusões ainda foram tiradas:

1. Quanto a estratégia de diagnóstico baseada na abordagem integrada do conhecimento:

- O sistema baseado na abordagem do conhecimento integrado permite diagnosticar problemas mais rapidamente que o especialista humano, porque a parte experimental prove a estrutura do processo de diagnóstico de uma forma consistente enquanto que a parte fundamental provê a estrutura do dispositivo que contém as especificações técnicas codificadas no modelo causal funcional.
- A abordagem baseada em classificação demanda a eliciação do conhecimento especializado em diagnóstico, sua interpretação e a formulação desse conhecimento em associações empíricas. Esse processo depende da presença do especialista em simular o diagnóstico e portanto requer tempo; o tempo necessário para acumular as regras apropriadas ou base de conhecimento experimental. Além disso o equipamento deve contar de um mínimo histórico de problemas, ou seja, ter quebrado inúmeras vezes. E também, de preferência, o(s) mesmo(s) profissional(is) devem participar dessa história de forma se tornarem o(s) especialista(s). Por outro lado a abordagem baseada em modelo causal tem um acesso mais direto quando se trata de aquisição de conhecimento, porque demanda informações tipicamente disponíveis quando da construção do equipamento, como:- especificações técnicas de projeto e do processo de fabricação, porém restringe-se aos mesmos.
- O processo de aquisição do conhecimento pode ser considerado como uma das etapas mais críticas porque constatou-se que ao passar de longo tempo de vivência com a máquina, o conhecimento torna-se muito detalhado e interconectado. Sendo que métodos utilizados na resolução não são totalmente claros e são utilizados automaticamente pelo especialista, o que torna difícil a expressão e a organização desse conhecimento e da lógica utilizada. É crítica também, porque da qualidade do conhecimento extraído e representado depende a precisão e a qualidade do sistema especialista.
- A natureza mais complexa dos equipamentos demanda um processo de diagnóstico que aborde diferentes estratégias que se cooperem, de forma a se obter um processo mais eficiente.
- O fato do especialista e do engenheiro do conhecimento serem interpretados pela mesma pessoa por vezes dificultou o processo de análise, e aquisição do conhecimento.

2. Quanto ao uso de sistemas especialistas:

- Dentro do contexto do rápido aumento da complexidade das máquinas e equipamentos; e da crescente dificuldade do entendimento e da manutenção de tais sistemas, em função do elevado e diversificado conhecimento necessário, conclui-se que a importância dos Sistemas Especialistas no auxílio ao diagnóstico de máquinas e equipamentos aumenta na proporção direta da:
 - necessidade de se acelerar o processo de diagnóstico, ou de se reduzir o tempo gasto na identificação de um problema.
 - aumento da complexidade dos sistemas diagnosticados.
 - necessidade de consistência do diagnóstico.

- necessidade de se capturar o conhecimento humano especializado para divulgação.
- Dado que o sistema também pode simular qualitativamente como o equipamento funciona, o mesmo oferece condições não somente para o modelamento do mau funcionamento mas também para operações normais no sentido de um estudo ou de uma investigação.
- A execução contínua do sistema de diagnóstico agrega ao usuário, um maior grau de educação sobre o equipamento que opera.
- Uma consequência decorrente do aprendizado e do contato dos operadores com um sistema de diagnóstico automatizado, foi a redução do nível de “mistificação” com relação aos sistemas computacionais, porque permitiu o entendimento dos propósitos, das limitações e das contribuições de tais sistemas.
- Um programa do tipo Sistema Especialista necessita de contínua manutenção de sua base de conhecimento, de forma a torná-lo interativamente mais inteligente. Essa característica exige uma metodologia que deve ser disciplinadamente seguida, de maneira a não tornar o programa desatualizado e conseqüentemente cair em desuso.

3. Quanto a representação do conhecimento através de regras de produção:

- Uma lição encorajadora da presente implementação é o bom desempenho que pode ser obtido através do uso de regras de produção, principalmente porque essa forma de representação é comumente caracterizada para representação de raciocínio superficial. Para a grande maioria dos problemas de implementação através da abordagem baseada em classificação e baseada em modelos, a simples sintaxe da cláusula condicional relacionada a um fato e associada a uma cláusula de ação, foi muito apropriada. Entretanto observou-se que há uma relação direta entre a simplicidade e poder de expressão. Quanto mais simples a representação, maior a facilidade de usar, comprometendo porém a capacidade do desenvolvimento e da representação de objetos e de conhecimentos mais complexos.
- Os pontos negativos experimentados, quanto a essa forma de representação foram:
 - baixa eficiência:- a execução de uma regra de produção impõe uma grande sobrecarga nos recursos do sistema, obstruindo respostas a situações de seqüências pré-determinadas, e restringe sua habilidade de executar grandes passos de raciocínio em situações que assim o demandam.
 - baixa clareza:- é comum a dificuldade de se acompanhar o fluxo do controle. O significado dos algoritmos é menos aparente que o significado de expressões, correspondentes em linguagens de programação mais tradicionais.

4. Quanto a utilização de ferramentas do tipo shell para a implementação:

- Com base na experiência do autor, tem-se que a utilização de uma ferramenta do tipo “shell”, permite um ciclo de desenvolvimento e implementação menor, comparado com o processo de desenvolvimento e implementação através da codificação utilizando-se linguagens de programação como LISP, PROLOG e outras. A utilização do “shell” no presente trabalho, permitiu o estabelecimento de uma análise comparativa entre alguns parâmetros úteis a serem considerados quando da escolha da forma de desenvolvimento e implementação de um Sistema Especialista, bem como da escolha do “shell” adequado a aplicação:
 - flexibilidade:- Dado o caráter genérico de um programa do tipo “shell”, é inevitável que o seu grau de “customização” seja limitado e portanto, menos capaz de se moldar e de atender a algumas necessidades específicas. Dessa forma, no desenvolvimento de sistemas baseados em conhecimento que requeiram alto nível de flexibilidade e de “customização”, recomenda-se o desenvolvimento de sistemas específicos a necessidade, ou seja, a não utilização de ferramentas “shell”. Dessa forma, pode-se concluir que a utilização de uma ferramenta desse tipo fornece um alto grau de produtividade para a implementação de um sistema especialista. Em

contrapartida tem-se uma redução do grau de flexibilidade em casos onde há a necessidade de uma maior particularização, em função do alto grau de generalidade.

- dificuldade de implementação:- os sistemas especialistas desenvolvidos para finalidade específica utilizam-se de linguagens básicas de programação, usualmente as linguagens: LISP, PROLOG ou "C". Assim sendo, o grau de dificuldade do desenvolvimento e implementação, é diretamente proporcional ao nível de conhecimento e ao nível de experiência de codificação utilizando-se técnicas de Inteligência Artificial. Enquanto que os sistemas baseados em "shell" são desenvolvidos em linguagem de alto nível, o que significa que os conhecimentos específicos de programação e de sistemas computacionais requeridos, são menores quando comparados aos sistemas codificados.
- tempo de implementação:- esse fator também é função da experiência e do conhecimento técnico do desenvolvedor do sistema. Porém se considerarmos um mesmo profissional dedicado ao desenvolvimento de dois sistemas destinados a resolver o mesmo problema. Sendo uma das implementações através de um "shell" e a outra através da codificação em uma linguagem de programação. O sistema baseado em "shell" terá um menor ciclo de implementação porque o mesmo carece fundamentalmente da base de conhecimento, sendo que o sistema codificado além da base de conhecimento necessita da construção da máquina de inferência, do módulo de explicação e de outros elementos da arquitetura funcional de um Sistema Especialista.
- capacidade de representação do conhecimento:- as várias as técnicas de representação de conhecimento, em tese, tem a mesma capacidade representar o conhecimento. Entretanto, dado que a maioria do "shells" disponíveis utilizam-se de regras de produção, tem-se que nem todos os tipos de conhecimento tenham a sua melhor representação através desse método. Assim sendo, o uso do "shell", pode não ser a melhor solução.
- custo de implementação:- esse fator está fortemente associado ao tempo dispendido no desenvolvimento, na implementação e na utilização ou finalidade associada do Sistema Especialista. Por exemplo, comumente um programa do tipo "shell" conta com um módulo de desenvolvimento que permite a implementação de múltiplas aplicações de Sistemas Especialistas. Por outro lado um sistema codificado, via de regra, tem uma baixa capacidade de reutilização e atende satisfatoriamente a função para o qual foi concebido. Portanto, se o objetivo é o desenvolvimento de mais de um Sistema Especialista então a escolha de um "shell" pode ser a mais indicada, e vice-versa.
- grau de facilidade de manutenção do sistema:- os sistemas especialistas codificados exigem que o seu desenvolvedor ou o mantenedor do sistema tenham conhecimentos profundos do sistema implementado e de técnicas de programação (linguagem utilizada, sistema operacional, comunicação entre processos, entre outros), elevando o grau de dificuldade de manutenção do sistema. Enquanto que os sistemas do tipo "shell", que em sua grande maioria utilizam linguagem de alto nível, permitem uma boa manutenção com um mínimo de treinamento quando comparado ao treinamento e experiência necessários dos programas ditos convencionais. Tem-se dessa forma que a manutenção do Sistema Especialista restringe-se a manutenção da base de conhecimento.
- interface homem/máquina:- uma interface homem/máquina com alta capacidade de resolução gráfica, por exemplo através fotos, diagramas digitalizados ou de recursos de multimídia em muito auxiliariam na qualidade e no desempenho da comunicação entre o usuário e o sistema. Também é recomendável que a interface homem/máquina seja adequada ao nível de escolaridade e ao nível técnico do grupo de usuários majoritário, de forma a não se criar constrangimento e dificuldade de comunicação, o que trará como consequência direta um desestímulo na utilização do sistema.
- plataforma:- é de extrema utilidade que o sistema desenvolvido opere em múltiplas plataformas computacionais. E também permita a sua utilização compartilhada através da rede de computadores.

5. Quanto a ferramenta escolhida:

- Permitiu a implementação do modelo de forma satisfatória.
- A inexistência de um interface gráfica na ferramenta escolhida é uma clara desvantagem para o processo de implementação, e principalmente para o processo de interação dos usuários com o programa.
- A ferramenta ESE opera unicamente em plataformas de computadores de grande porte, restringindo assim sua utilização em outros ambientes; e particularmente sua utilização de forma compartilhada em redes de computadores.

5.1 Sugestões para Próximos Trabalhos

Com base na experiência e conhecimentos adquiridos com o presente trabalho, tem-se como sugestões de pesquisas e implementações para trabalhos futuros, na área de sistemas baseados em conhecimento voltados a diagnóstico para suporte a equipamentos e máquinas de produção, os seguintes tópicos:

1. Dadas as adequadas características dos componentes mecânicos, elétricos e eletromecânicos, desenvolver um Sistema Especialista orientado a objetos de caráter genérico, voltado a manutenção de máquinas e equipamentos, através da construção de uma biblioteca de classes, desenvolvidos em linguagem orientada a objetos - por exemplo C++ , que contemple alguns dos componentes mais comumente utilizados em máquinas e equipamentos. Por exemplo:

```
classe_motor:  
  variação:- (passo-a-passo, corrente_alternada)  
  atributos:- torque, tensão, rotação, rele térmico,...
```

```
classe_válvula_hidráulica:  
  variação:- (retenção, de fluxo, de pressão)  
  atributos:- pressão de trabalho, velocidade,...
```

Esse sistema permitiria o rápido desenvolvimento de sistemas especialistas voltados ao diagnóstico ou a simulação como uma alternativa mais flexível aos sistemas desenvolvidos em shell. Utilizando-se ainda da abordagem integrada da estratégia de diagnóstico.

2. Desenvolver um Sistema Especialista baseado no modelo proposto, dotado de interface baseada em multimídia ou de alta definição gráfica, com o objetivo de maximizar o potencial desse sistema.
3. Selecionar um equipamento de produção de natureza de projeto e operação complexos. Contactar o fabricante para o desenvolvimento de um Sistema Especialista baseado no modelo funcional do equipamento, bem como prover o sistema com funções que permitam a entrada do conhecimento experimental. Embarcar o sistema desenvolvido junto com o equipamento como parte integrante da tecnologia do produto. Esse trabalho tem por objetivo disseminar a tecnologia de Inteligência Artificial e prover um fator diferenciador do produto a ser comercializado.

6.0 Bibliografia Utilizada

6.1 Referências Bibliográficas

1. ADRION,W.R. et al *Validation,verification and testing of computer software* ACM Computing Surveys,vol.14, n.2, pp.159-192, 1992.
2. AGOSTINHO,O.L. *Manufatura Integrada por Computador - notas de aulas* UNICAMP-FEM, 1989.
3. APPLICATION PROGRAMMING GUIDE *Expert System Environment* nr. SC38-7020-02 IBM, 1990.
4. ARCISZEWSKI,T. et all *A Method for Evaluation of Learning Systems* Heuristics, vol.5, n.4, 22-31, 1992.
5. BARR,A. & FEINGENBAUM,E.A. *The handbook of Artificial Intellingence* William Kaufmann,Inc., 1981.
6. BERGEL,H. & ROACH,D. *Representing Government Travel Regulation Expert Systems as Rules* Heuristics, vol.5, n.4, pp.62-71, 1992.
7. BERRY,D.C. & BROADBENT,D.E. *Expert systems and the man machine interface* The International Journal of Knowledge Engineering, vol.4, pp.18-28, 1987.
8. BERRY,D.C. & HART,A.E. *Evaluating expert systems* Expert Systems, vol.7, n.4, Nov.1990, pp.199-208, Nov.1990.
9. BERRY,D.C. & HART,A.E. *Expert Systems: human issues* Chapman and Hall, 1989.
10. BOLEY,H. *Expert systems shells: very-high-level languages for Artificial Intelligence* Expert Systems, vol.7, n.1, pp.2-8, Feb.1994.
11. BOLASNA,S.A. et al *Air-bearing support of a magnetic recording head* IBM Disk Storage Technology, pp.61-63, Feb.1992.
12. BOOSE,J.H. *Personal construct theory and the transfer of human expertise* Advances in Artificial Intelligence, T.O'Shea, Elsevier Science Publisher, pp.51-59, ECCAI 1985,
13. BOUCHE,M.; PLAUCHU,U.; RETOUR,D. *Troubleshooting Expert System for Industrial Maintenance: pre-development guidelines* Expert Systems, vol.7, no.3, August 1990.
14. BOWEN,J. & BOWEN,D. *Develloping Intelligent Systems* AI Expert Magazine, vol.9, n.1, pp.34-37, Jan.1994.
15. BRADAC,M. et all *Prototyping a Process Monitoring Experiment* IEEE Transaction on Software Engineering, vol.20, n.10, pp.774-784, Oct.1994.
16. BRADLEY,J. *The Role of General Cognitive Abitivity in Expertise: A Mode for Expert Selection* Heuristics, vol.6, n.3, pp.01-09, Fall 1993.
17. BREUKER,J. & WIELINGA,B. *Models of expertise in knowledge acquisition* ESPRIT Program of the European Community, 1988.
18. BROEDNER,P. *The Shape of Future Technology: the anthropocentric alternative* Springer-Verlag, 1990.
19. BUCHANAN,B.G. & SHORTLIFFE,E.H. *Rule-based expert systems* Addison-Wesley Publishing Company,Inc.1985
20. BURTON,S. et all *Quality and Knowledge in Software Engineering* AI Magazine, vol.14, n.4, pp.43-50, 1993.

21. CARVALHO,R.Q. *Tecnologia e trabalho industrial* LPM Editores, 1987.
22. CHANDERASEKARAN,B. et al *Explaining control strategies in problem solving* IEEE Expert, pp.9-24, Spring 1989.
23. CHANDRASEKARAN,B. & MITTAL,S. *Deep versus compiled knowledge approaches to diagnostic problem-solving* Developments in Expert Systems - Academic Press,Inc., pp.23-34, 1989.
24. CHANDRASEKARAN,B. et al *Task-structure analysis for knowledge modeling* Communications of the ACM, vol.35, n.9, pp.124-136,Sept.1992.
25. CHANDRASEKARAN,B. et all *Reasoning with Diagrammatic Representatives* AI Magazine, vol.14, n.2, pp.49-56, 1993.
26. CLANCEY,W.J. *Model construction operators* Artificial Intelligence, vol.53, n.1, pp.1-111, Jan.1992.
27. CLANCEY,W.J. *The epistemology of a rule-based expert system - a framework for explanations* Artificial Intelligence Journal, vol.20, n.3, pp.215-251, May 1983.
28. CLANCEY,W.J. *Heuristic classification* Artificial Intelligence Journal, vol.20, n.3, pp.289-350, May 1983.
29. CLANCEY,W.J. *Viewing knowledge bases as qualitative models* IEEE Expert, pp.9-22,Summer 1989.
30. CLEGG,C. et al *People and Technology: How to evaluate your company's new technology* Ellis Horwood, 1988.
31. COHEN,P.R. & FEIGENBAUM,E.A. *The Handbook of Artificial Intelligence* vol.3, Morgan Kaufmann, 1982.
32. CORBERTT,J.M.; RASMUSSEN,L.B.; RAUNER,F. *Crossing the Border: the social and engineering design of computer integrated manufacturing* Springer-Verlag, 1991.
33. CORIAT,B. *A Revolução dos Robôs: o impacto sócio-econômico da automação* Editora Busca Vida, 1988.
34. CORIAT,B. *Automação Programável: novas formas e conceitos de organização da produção* Editora Hucitec, 1988.
35. D'AMBROSIO,A. *Modeling real-world processes: deep and shallow knowledge integrtded with approximate reasoning in a diagnostic expert system* Artificial Intelligence in Process Engineering, Academic-Press,Inc., 1990.
36. DAVIS,R. *Diagnostic reasoning based on structure and behavior* Artificial Intelligence Journal, vol.24, n.1-3, pp.347-410,Dec 1984.
37. DAVIS,R. *Applications of meta level knowledge to the construction, maintenance and use of large knowledge bases* Report STAN-CS-76-552, Stanford Artificial Intelligence Laboratory, Stanford University, Jul.1976.
38. DAVIS,R. & HAMSCHER,W. *Model-based troubleshooting* Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence, Morgan Kaufmann, cap.8, pp.297-346,1988.
39. DERTOUZOS,M.L.; LESTER,R.K.; SOLOW,R.M. *Made in America: regaining the productivity edge* MIT Press, 1989.
40. DEUTCH-MCLEISH,M. et all *Knowledge Acquisition Methods for Extracting Belief Function from Continuous Data with and Application to Medical Diagnosis* Heuristics, vol.6, n.2, 38-47, 1993.
41. DUBAS,M. *Expert Systems in Industrial Practice: Advantages and drawbacks* Expert Systems, vol.7, no.3, pp.47-53, August 1990.
42. DRUCKER,P. *Sociedade Pós-Capitalista* Livraria Pioneira Editora, 1993.
43. ESTEVANIK,S. *Artificial Intelligence in Wargames* AI Magazine, vol.9,. n.5, pp.22-31, May 1994.

44. EVANS, T.G. *A heuristic program to solve geometric-analogy problems* Artificial Intelligence, AFIPS Press, pp.5-16, 1984.
45. EXPERT SYSTEM DEVELOPMENT ENVIRONMENT USER'S GUIDE *Expert System Environment* nr.SC38-7006-03 IBM, 1989.
46. EXPERT SYSTEM CONSULTATION ENVIRONMENT USER'S GUIDE *Expert System Environment* nr.SC38-7005-04 IBM, 1990.
47. FAVILLA, J.R. *Sistema especialista para diagnóstico de unidades de disco* Tese de mestrado apresentada a FEE-UNICAMP, São Paulo, 1988.
48. FELDMANN, P.R. *Robo, Ruim com Ele, Pior sem Ele* Trajetória Cultural, 1988.
49. FERREIRA, M.V. *Uma abordagem baseada em modelos na construção de sistemas baseados em conhecimento para diagnose de equipamentos* Tese de mestrado apresentada a FEE-UNICAMP, São Paulo, 1991.
50. FINK, P.K. et al *A general expert system design for diagnosis problem solving* IEEE Transactions on Pattern and Machine Intelligence, vol.PAMI-7, n.5, pp.553-560, Sep.1985.
51. FINK, P.K. & LUSTH, J.C. *Expert systems and diagnostic expertise in the mechanical and electrical domains* IEEE Transactions on Systems, Man and Cybernetics, vol.SMC-17, n.3, pp.340-348, May/Jun-1987.
52. FIOL, G. & NICOLAU, J. *A Diagnosis Problem Approach Based on Inductive Acquisition of Knowledge from Examples* Heuristics, vol6, n.3, pp.54-65, 1993.
53. FULDA, J. *Multiple Experts and Enthymemes in Expert Systems: Illustrations from the Medical Domain* Heuristics, vol.6, n.2, pp73-77, 1993.
54. GAMMACK, J.G. *Different techniques an different aspects on declarative knowledge* Knowledge Acquisition for Expert Systems - A practical handbook, Kidd, A.L.(Ed.), 1987
55. GARDNER, H. *The mind's new science* Basic Books, 1985.
56. GAREGNANI, P. et al. *Progresso Técnico e Teoria Econômica* Editora Hucitec, 1980.
57. GARG-JANARDAN, G. & SALVENDY, G. *A structured knowledge elicitation methodology for building expert systems* Int.J.Man-Machine Studies, vol.29, pp.377-406, 1988.
58. GASCHNIG, J. et al *Evaluation of expert systems: issues and case studies*, in F.Hayes-Roth et al., Building Expert Systems, Addison-Wesley, 1983.
59. GENESERETH, M.R. *The user of design descriptions in automated diagnosis* Artificial Intelligence Journal, vol.24, n.1-3, pp.411-436, Dec 1984.
60. GERMOND, A. & NIEBUR, D. *Survey of Knowledge-Based Systems in Power Systems: Europe* IEEE Proceedings, vol.80, n.5, pp.732-744, May 1992.
61. GIORNO, F. *Equipment diagnosis through knowledge-based systems* IBM Rio Scientific Center, Nov 1990.
62. GIORNO, F.A.C. et al *Methods and techniques for knowledge elicitation* IBM Rio Scientific Center, Technical Report CCR 065, Dec 1988.
63. GISOLFI, A. & BALZANO, W. *Constructing and consulting knowledge base of an expert system shell* Expert System, vol.10, n.1, pp.29-34, Feb 1993.
64. GONÇALVES, C.A. *Aquisição e representação de conhecimento para sistemas especialistas* Tese de doutorado apresentado a FEA/USP, São Paulo, 1986.
65. GRUBER, R.T. *Acquiring strategic knowledge from experts* Int.J.Man-Machine Studies, vol.29, pp.579-597, 1988.
66. HARMOND, P. & KING, D. *Expert systems* John Willey & Sons, Inc - 1985.

67. HART,A. *Knowledge elicitation: issues and methods* Computer-aided Design, vol.17, n.9, pp.455-462, Nov.1985.
68. HAYES-ROTH,F. et al *Building Expert Systems* Addison-Wesley Publishing Company,Inc.1983.
69. HIEBB,M. et all *Rule Acquisition for Dynamic Engineering Domains* Heuristics, vol5, n.4, pp.72-82, 1992.
70. HOFFMAN,R.R. *The problem of extracting the knowledge of experts from the perspective of experimental psychology* AI Magazine, pp.53-67, Summer 1987.
71. KATTAN,M. *Inductive Expert Systems vs. Human Experts* vol.9, n.4, pp.32-38, Apr.1994.
72. KEUNEKE,A.M. *Device representation, the significance of function knowledge* IEEE Expert, pp.22-25, Apr 1991.
73. KIRSH,D. *Artificial Intelligence* vol47, 1991, Esenvier Amsterdam.
74. KOBAYASHI,S. & NAKAMURA,K. *Knowledge compilation and refinement for fault diagnosis* IEEE Expert, vol.6, n.9, pp.39-46, Oct.1991.
75. KOWALSKIA. et all *PITCH Expert - A Problem-Solving System for Kraft Mills* AI Magazine, vol.14, n.3, pp.81-99, fall 1993.
76. KUIPERS,S.J. *A frame for frames* Representantion and Understanding, Academic Press, 1975.
77. KUSIAK,A. *Artificial Intelligence Implications for CIM* Springer-Verlag, 1988.
78. KRISHNAMURTH,M. & PHILLIPS,D.T. *An expert system framework for machine fault diagnosis* Computers Industrial Engineering, vol.22, n.1, pp.67-84, 1992.
79. LAUFMANN,S.C. et al *A methodology for evaluating potencial KBS applications* IEEE Expert, vol.5, n.6, pp.43-60, Dec 1990,.
80. LEÃO,B.F. & ROCHA,A.F. *Proposed methodology for knowledge acquisition: A study on congenital heart disease diagnosis* Meth.Inform.Med., vol.29, n.1, pp.30-40, 1990.
81. LENZ,J.E. *Flexible Manufacturing - benefits for the low-inventory factory* Marcell Dekker,Inc., 1988.
82. LIEBOWITZ,J. *Useful approaches for evaluating expert systems* Expert Systems,3, pp.86-96, 1986.
83. LIROV,Y. *Performance Analysis of Real-Time Knowledge-Based Control System* Heuristics, vol.6, n.2, pp.849-907, 1993.
84. McCORDUCK,P. *Machine Who Think* W.H.Freeman and Company, 1979.
85. MICHALSKI,R.S. et al *Machine learning - volume II* Morgan Kaufmann, 1986.
86. MILNE,R. *Strategies for diagnosis* IEEE Transactions on Systems, Man, and Cybernetics, vol.SMC-17, n.3, pp.333-339, May/Jun-1987.
87. MINSKY,M. *A framework for representing knowledge* Readings in Cognitive Science, Morgan Kaufmann Publishers,Inc., pp.156-188, 1988.
88. MITCHEL,T.M., MAHADEVAN,S. & STEINBERG,L.I. *LEARP: a learning apprentice system for VLSI design* LSCR Technical Report 64, Department of Computer Science Rutgers University, New Brunswick-NJ, 1985.
89. MORALES,E. & GARCIA,H. *A modular approach to multiple fault diagnosis* Artificial Intelligence in Process Engineering, Academid Press, 1990.
90. MUSSI,S. & MORPURGO,R. *Acquiring and representing strategic knowledge in the diagnosis domaing* Expert System, vol.7, n.3, pp.157-165, Aug 1990.
91. NEWEL,A. & SIMON,H.A. *Human problem solving* Prentice -Hall, Englewood Cliffs-NJ, 1972.

92. NEWELL,A. & SIMON,H.A. *GPS, a program that simulates human thought* Readings in Cognitive Science, Morgan Kaufmann Publishers,Inc., pp.453-460, 1988.
93. NEWELL,A. & SIMON,H.A. *The theory of human problem solving* Readings in Cognitive Science, Morgan Kaufmann Publishers,Inc., pp.33-51, 1988.
94. NORMAN,D.A. *Categorization of action slips* Psychological Review, n.88, pp.1-15, 1981.
95. QUADROS,R.C. *Tecnologia e Trabalho Industrial: as implicações sociais da automação microeletrônica na indústria automobilística* L&PM Editores S/A, 1987.
96. QUILLIAN,R. *Semantic memory* Semantic Information Processing, M.Minsky(Ed.), MIT Press, Cambridge,Mass.,1968.
97. RAVDEN,S. & JOHNSON,G. *Evaluating Usability of the human-computer interface* Ellis Horwood, 1989.
98. RICH,E. *Inteligência Artificial* McGraw-Hill, 1988.
99. ROTH,R. & WOOD II,W *Knowledge Acquisition from Several Versus Multiple Experts: A Filed Study Comparison Using the Delphi Technique* Heuristics, vol.6, n.3, pp.11-20, Fall 1993.
100. SEKINE,Y et all *Fault Diagnosis of Power Systems* IEEE Proceedings,vol.90, n.9, pp.673-683, May 1992.
101. SCHANK,R. *What is AI, anyway ?* AI Magazine, pp.59-65, Winter 1987.
102. RUBINSTEIN,R. & HERSH,H. *The Human Factors - Designing Computer Systems for People* Digital Equipament Corporation, 1984.
103. SCHANK,R. *What is AI, anyway ?* AI Magazine, pp.59-65, Winter 1987.
104. SCHIAVINI,M.M. *Aquisição de conhecimento* DCA/FEE UNICAMP, Relatório Técnico, Dez.1989.
105. SCHUTZER,D. *Artificial Intelligence* Van Nostrand Reinhold Company, 1987.
106. SCHMITZ,H & CARVALHO,R.Q. *Automação, Competitividade e Trabalho: A experiência internacional* Editora Hucitec,1988.
107. SUEN,C. et al *Verifying, validating and measuring performance of expert systems* Expert System Application,2, pp.93-102, 1990.
108. STEELS,L. *Components of expertise* AAAI Magazine, pp.29-49 Summer, 1990.
109. STEELS,L. *Diagnosis with a function-fault model* Applied Artificial Intelligence, v.3, n.2-3, 1989.
110. STEVENS,L.D. *The evolution of magnetic storage* IBM Journal of Research and Development, vol.25, n.5., pp.663-674, Sept.1981.
111. THAKORE,A. *Parallel Rule Processing Against Large Object-Oriented Knowledge Bases* Heuristics, vol.6, n.4, pp.1-15, 1993.
112. TAUILE,J.R. *Automação Microeletrônica e Competitividade: tendencias no cenário internacional* Editora Hucitec, 1988.
113. TOFFFLER,A. *A Terceira Onda* Editora Record, 1980.
114. TOLLIVER,D.L. *Handbook of Contamination Control in Microelectronics* Noyes Publication, 1988.
115. TORASSO,P. & CONSOLE,L. *Diagnostic problem solving - Heuristic, Approximate and Causal Reasoning* Van Nostrand Reihold, 1989.
116. TURNBAN,E. *Review of Expert System Technology* IEEE Transactions on Engineering Managemente, vol.35, no.2, May 1988.
117. TURING,A.M. *Computing Machinery and Intelligence* Readings in Cognitive Science, Morgan Kaufmann Publishers,Inc.,pp.6-19, 1988.

118. TRINZIC *Developing Applications with the Aion Development System, Version 1.0 DC0319* Trinzic Corporation, 1994.
119. TSUTSUI,S. *Constructing a Class Scheduling Systems Using Rule-Based and Neurocomputing-Based Approaches* Heuristics, vol.6, n.1, pp.12-23, 1993.
120. VINSON,J.M. et al *Automatic rebuilding of qualitative models for diagnosis* IEEE Expert, vol.7, n.4, pp.23-30, Aug.1992.
121. WARKENTIN,M. *Verbal Domain Knowledge Acquisition: Unplugging the Bottleneck* Heuristics, vol.6, n.3, pp 21-26 1993.
122. WATERMAN,D.A. *A guide to expert systems* Addison-Wesley Publishing Company,Inc., 1985
123. WEERAHANDIS & HAUSMAN,R. *Software Quality Measurement on Fault-Detecion Data* IEEE Transaction on Software Engineering, vol.20, n.9, pp.665-676, Sept.1994.
124. WHITE,M. & GOLDSMITH,J. *Standards and Review Manual for Certification in Knowledge Engineering - Handbook of Theory and Practice* Sponsored by the International Association of Knowledge Engineers. Published by Susemware Corporation, 1990.
125. WNEK,J. & MICHALSKI,R. *Experimental Comparison of Symbolic and Subsymbolic Learning* Heuristics, vol.5, n.4, pp.2-21, 1992.
126. WOMACK,J.P.; JONES,D.T.; ROOS,D. *A Máquina que Mudou o Mundo* Editora Campus Ltda, 1992.
127. WRIGHT,P.K. & BOURNE,D.A. *Manufacturing Intelligence* Addison-Wesley Publishing Company,Inc., 1988.
128. YASI,R. & ZIARKO,W. *An expert system for conceptual schema design: A machine learning approach* Int.J.Man- Machine Studies, vol.29, pp.351-376, 1988.
129. YOON,W.C. & HAMMER,M.J. *Deep-Reasoning Fault Diagnosis: An Aid and a Model* IEEE Transactions on Systems, Man and Cybernetics, vol.18,no.4, July/August 1988.

6.2 Bibliografia Adicional Consultada

1. ABU-HANNA,A. et al *Device understanding and modeling for diagnosis* IEEE Expert, pp.26-31, 1991.
2. AHAR,V. & POPLE,H. *Rule-based versus structure-based models for explaining and generating expert behavior* Communications of ACM, vol30, n.6, pp.542-555, Jun 1987.
3. ALLEMAG,D. *Using functional models in automatic debugging* IEEE Expert, pp.13-18, Dec 1991.
4. AREZZO,F. *Microcontamination in processing and cleaning precision parts* Microcontamination, pp.36-43, Nov 1986.
5. BARTLETT,F.C. *Remembering: a study in experimental and social psychology* Cambridge University Press, 1932.
6. BARGHOUTI,N.S. & KAISER,G.E. *Modeling concurrency in rule-based development environments* IEEE Expert, pp.15-27, Dec 1990.
7. BUBLIN,S.C. & KASHYAP,R.L. *Merging heuristic classification with causal reasoning in machine fault diagnosis* Proceedings Fourth Conference in Artificial Intelligence Applications, IEEEE, San Diego-CA, March,1988.
8. BOUCHE,M. et al *Troubleshooting expert systems for industrial maintenance: pre-development guidelines* Expert Systems, vol.7, n.3, pp.142-148, Aug.1990.

9. BUCK,L. *Human operators and real-time expert systems* Expert Systems, vol.6, n.4, pp.227-236, Nov 1989.
10. BURBIDGE,J.L. *The Introduction of Group Technology* Willian Heinemann Ltd., 1985.
11. BUTLER,C.W. & HODIL,E.D. *Building knowledge-based systems with procedural languages* IEEE Expert, pp.47-59, Summer 1988.
12. CHORAFAS,D.N. *Sistemas Especialistas: aplicações comerciais* McGraw-Hill, 1988.
13. DENNING,P.J. *Towards a science of expert systems* The Science of Computing, vol.74, n.1, pp.18-20, Jan-Feb,1986.
14. ERMAN,L. *The HEARSAY-II speech-understanding system: integrating knowledge to resolve uncertainties* Computing Surveys, vol.12, n.2, pp.213-253, Jun.1980.
15. GALBRAITH,L. et al *Expert systems in engineering* IEEE AES Magazine, pp.12-13, Feb.1988.
16. GALLANTI.M. et al *Integrating shallow and deep knowledge in the design of an on-line process monitoring system* Internation Journal Man-Machine Studies, n.27, pp.641-664, 1987.
17. GLASS,A.L. et al *Cognition* Addisson-Wesley, 1988.
18. GRAY,M. *Generalization as a Mechanism for Automatic Knowledge Acquisition* Heuristic, vol.6. n.3, pp.27-38, 1993.
19. HASLING,D.W. et al *Strategic explanations for a diagnostic consultation system* Developments in Expert Systems - Academic Press,Inc., pp.117-133, 1989.
20. HEDBERG,S. *Emerging Genetic Algotihms* AI Expert Magazine, vol.9, n.9, pp.25-29, Sep.1994.
21. KLEER,J. & WILLIAMS,B.C. *Diagnosing multiple faults* Artificial Intelligence Journal, vol.20, n.3, pp.97-130, May 1983.
22. KERSHNER,J.D. & LORENZ,R.D. *Construction techniques for factory floor expert system* Journal of Engineering for Industry, vol.110, pp.395-397, Nov 1988.
23. LEVINE,R.I. et al *Inteligência Artificial e Sistemas Especialistas* McGraw-Hill, 1988.
24. LIEBOWITZ,J. *Beyond decision support systems: the role of operations research in expert systems* Computers in Industry Engineering, vol.14, n.4, pp.415-418, 1988.
25. MACHADO,R.J. *Sistemas especialistas conexionistas* Centro Cientifico Rio - IBM Brasil, 1988.
26. MALETZ,M.C. *An architecture for consideration of multiple faults* Proceedings of the Second Conference in Artificial Intelligence Applications, Miami Beach-FL, pp.60-67, Dec 1985.
27. MINSKY,M. *A Sociedade da Mente* Livraria Francisco Alves Editora S.A., 1989.
28. MARSH,R.C. *Historical Development of U.S. Fed.Std.209 for air cleanliness classification* 9th Internation Sumposium on Contamination Control Proceedings, Los Angeles-CA,IES, Sep 1988.
29. MATSUDA,Y. *Human-Centred Systems in the Global Economy* Springer-Verlag, 1992.
30. MICHALSKI,R.S. & CHILAUSSKY,R.L. *Learning by being told and learning from examples: an experimental comparissom* International Journal of Policy Analysis and Information Systems, vol.4, n.2, pp.125-160, 1980.
31. MITTAL,K.L. *Treatise on clean surface technology* vol.1, Plenun Press, 1987.
32. MORRIS,L. *Maintenance-oriented design of expert systems* ISA Transactions, vol.28, n.1, pp.23-26, 1989.
33. MORRISON,P.W. *Environmental control in electronic manufacturing* Van Nostrand Reinhold Company, 1973.

34. MULVANY,R.B. & THOMPSON,L.H. *Innovations in disk file manufacturing* IBM Journal of Research and Development, vol.25, n.5, pp.711-722, Sep.1981.
35. MYERS,W. *Introduction to expert systems* IEEE Expert, pp.100-109, 1986.
36. NG,H.T. *Model-based, multiple-fault, diagnosis of dynamic, continuous physical devices* IEEE Expert, pp.38-41, Dec 1991.
37. OSTROWSKY,M.F. & SWEZEY,R.C. *An expert system shell: expert environment/VM* ISA Transactions, vol28, n.1, pp.9-14, 1989.
38. PAZZANI,M.J. *Failure-driven learning of fault diagnosis heuristics* IEEE Transactions on Systems, Man and Cybernetics, vol.SMC-17, n.3, pp.380-393, May/Jun-1987.
39. PERKINS,W.A. et al *Knowledge base verification* Topics in Expert System Design, Elsevier North-Holland, pp.353-376, 1989.
40. PREECE,A. *Towards a methodology for evaluating expert systems* Expert Systems, vol.7, n.4, pp.215-223, Nov 1990.
41. RAUCH-HINDIN,W. *Artificial Intelligence in business, science and industry* Prentice Hall,Inc., 1985.
42. REITER,R. *A theory of diagnosis from first principles* Artificial Intelligence Journal, vol.32, n.1, pp.57-95, April 1987.
43. SHACKEL,B. *Ergonomics in design for usability*, People and Computers: Design for Usability, Cambridge University Press, 1986.
44. SHALIN,V.L. et al *A formal analysis of machine learning systems for knowledge acquisition* Academic Press Limited, pp.429-449, 1988.
45. SCHOEN,E. et al *Design of knowledge-based system with a knowledge-based assistant.* IEEE Transaction on Software Engineering vol.14, n.12, pp.1771-1790, Dec 1988.
46. SLAGLE,J.R. et al *Knowledge specification of an expert system* IEEE Expert, pp.29-37, Aug 1990.
47. STOTTLER,R. *CBR for Cost and Sales Prediction* AI Expert Magazine, vol.9,n.8, pp.24-33, Aug.1994.
48. STYLIANOU,A.C. et al *Selection criteria for expert system shells: A socio-technical framework* Communications of the ACM, vol.35, n.10, pp.32-47, Oct 1992.
49. TOLLIVER,D.L. *Handbook of contamination control in microelectronics* Noyes Publication, 1988.
50. TURBAN,E. *Review of expert systems technology* IEEE Transactions on Engineering Management, Vol.35, n.2, pp.71-81, May 1988.
51. VONDRAK,I. *Object-Oriented Neural Network* AI Magazine, vol.9, n.6, pp.20-25, June 1994.
52. YOON,W.C. & HAMMER,J.M. *Deep-reasoning fault diagnosis: an aid and a ...* IEEE Transaction on Systems, Man and Cybernetics, vol.18, n.4, pp.659-675, Jul/Aug-1988.
53. WATERMAN,D.A. & NEWELL,A. *Protocol analysis as a task for Artificial Intelligence* Artificial Intelligence, n.2, pp.285-318, 1971.
54. WATSON,I. et al *The client-centred approach: expert system maintenance* Expert Systems, vol.9, n.4, pp.189-195, Nov.1992.
55. WILLIAMS,C. *Expert Systems, Knowledge Engineering, and AI Tools - and overview* IEEE Expert, pp.66-70, Winter 1986.
56. WISNIEWSKI,E.J. et al *A conceptual clustering program for rule generation* International Journal of Man-Machine Studies, vol.27, pp.295-313, 1987.

Apêndice A. Teste de casos

Abaixo encontram-se os Testes de Caso elaborados pelos usuários do sistema:

1. Desconecte eletricamente o sensor 24S0.13 Carregue todas as posições de alimentação sobre os transportadores de entrada da máquina. O elevador número 1, não será informado da presença do cesto na “posição de pegada”, e portanto ocorrerá um congestionamento dos segmentos transportadores de rolo, que deverá ser informado pelo alarme do painel. Consulte o sistema especialista.
 - diagnóstico esperado: problema no sensor 24S0.13.
 - resultado obtido: diagnóstico correto.

2. Desconecte a chave 51Q6, e veja o elevador fixo de número 3 perder o movimento lento. Consulte o sistema especialista.
 - diagnóstico esperado: problema na chave 51Q6.
 - resultado obtido: diagnóstico correto.

3. Desconecte eletricamente o componente 58Q4, e veja o elevador móvel número 4 perder o movimento rápido. Consulte o sistema especialista.
 - diagnóstico esperado: problema com a chave 58Q4.
 - resultado obtido: diagnóstico correto.

4. Desconecte o fusível F32, e veja a parada do sistema de refrigeração. Consulte o sistema especialista.
 - diagnóstico esperado: queima do fusível F32.
 - resultado obtido: diagnóstico correto.

5. Desconecte um dos três conjuntos de elementos resistivos (R41, ou R42 ou R43) que geram a carga térmica inicial para zona de vapor dos tanques 7, 8 e 9. O sistema 3 da bomba de calor levará muito tempo para vencer a baixa carga térmica inicial. Consulte o sistema especialista para verificar as causas desse problema.
 - diagnóstico esperado: De acordo com a documentação de projeto da máquina e com a experiência, um dos seguintes elementos pode estar apresentando o problema:
 - queima de um dos fusíveis: F71, F72 ou F73
 - problema na chave 19K2, ou
 - problema de um dos elementos de um dos conjuntos resistivos: R41, R42 ou R43.
 - resultado obtido: diagnóstico correto.

6. Drene manualmente a maior parte da água do separador gravitacional de solvente. Você observará que o solvente passará a circular diretamente para o tanque de dreno, diminuindo o volume dos tanques de processo e acionando o alarme no quadro cinóptico de processo. Consulte o sistema especialista.
 - diagnóstico esperado: De acordo com a documentação de projeto da máquina e com a experiência, um dos seguintes elementos pode estar apresentando o problema:
 - baixo nível de água no separador gravitacional, ou

- problema no sistema de resfriamento após o desligamento da máquina - “shut down cooling system”
 - resultado obtido: diagnóstico correto.
7. Abra a porta superior número 4, e deposite um cesto na estação de enxague de spray de alta pressão. Cuidadosamente coloque um segundo cesto na guancheira do elevador 6 e com o controle no modo manual, movimente o elevador número 6 vagarosamente sobre o cesto depositado sobre o tanque. Você verá que todo o carro do sistema de transporte fletirá levemente no sentido anti-horário acionando um sensor infravermelho de segurança. Consulte o sistema especialista sobre o possível motivo de acionamento desse alarme.
- diagnóstico esperado: acionamento do sensor infravermelho por encavalamento de cestos - acidente dentro dos tanques, ou por um outro motivo acidental - por exemplo, interrupção do feixe com a haste da bomba de imersão dos tanques durante o abastecimento manual de um dos tanques.
 - resultado obtido: diagnóstico correto.
8. Abra a porta da ante-câmera de saída máquina e segure firmemente o primeiro cesto que passar pela ante-câmera não permitindo que a mesma saia dessa posição. Você verá o alarme: “Basket Check Unload Station” ser acionado. Consulte o sistema especialista para saber os motivos de acionamento desse alarme.
- diagnóstico esperado: De acordo com o histórico desse problema os seguintes fatos poderão acionar esse alarme, por ordem de frequência:
 - túnel de peças congestionado pela não retirada dos cestos na sala limpa, ou
 - cesto enroscado em um dos segmentos do transportador de rolos:- segmentos 4,5 ou 6. Ou cestos com problema de tráfego e/ou problemas no transportadores do túnel de comunicação com a sala limpa, ou
 - problema específico no segmento número 5 do transportador de rolos.
 - resultado obtido: diagnóstico correto.
9. Desconecte o fusível F124 e veja que a máquina se desligará abruptamente - todas as funções serão imediatamente desativadas, porque o microprocessador de programa de funcionamento da máquina foi desligado. Consulte o sistema especialista.
- diagnóstico esperado: queima do fusível F124 ou problema na unidade de programação de funcionamento.
 - resultado obtido: diagnóstico correto.
10. Suponha que o alarme: “Check Solvent Regeneration Still” foi acionado. Consulte o sistema especialista para saber quais os principais fatores que causam o acionamento desse alarme.
- diagnóstico esperado: de acordo com a documentação de projeto da máquina, esse alarme pode ter as seguintes causas:
 - queima de um dos fusíveis: F205, F206 ou F207,
 - problema na chave K300,
 - problema em uma das resistências: 2R1, 2R2 ou 2R3
 - diagnóstico obtido: diagnóstico correto.

11. Desconecte o fusível F31, aguarde a parada do sistema de todo o sistema de refrigeração e consulte o sistema especialista para diagnosticar a origem da parada do sistema de refrigeração.
 - diagnóstico esperado: queima de um dos fusíveis: F31, F32 ou F33.
 - resultado obtido: diagnóstico correto.

12. Desligue um dos conjuntos de ventiladores situados sobre as placas transdutoras do tanque de número 1 do processo, aguarde a manifestação do alarme e consulte o sistema especialista.
 - diagnóstico esperado: De acordo com os documentos de projeto, um dos elementos abaixo deverá ser identificado como a causa do problema.
 - queima do fusível F103 (2.0 amperes),
 - problema na chave contatora 12K7
 - problema em um dos ventiladores: 18M3, 18M3.1, 18M3.2, 18M3.3, 18M3.4, 18M3.5
 - resultado obtido: problema não identificado pelo sistema.
 - ação: incluir na base de conhecimento.

13. Abra um dos painéis atrás da máquina e acione manualmente o sensor infravermelho, aguarde a manifestação do problema e consulte o sistema especialista.
 - diagnóstico esperado: identificar um possível acionamento acidental (que não por encavalamento de cestos dentro de um dos tanques) do sensor e reinício do processo de limpeza de acordo com o Manual Geral do Processo.
 - resultado obtido: diagnóstico correto.

14. Desconecte a bomba de recirculação e filtragem do tanque de número 7, aguarde a manifestação do problema e consulte o sistema especialista ?
 - diagnóstico esperado: De acordo com os documentos de projeto, um dos elementos abaixo deverá ser identificado como a causa do problema.
 - queima do fusível F118 (4.0 amperes),
 - problema na chave contatora 19K7
 - problema na bomba 19M7 - tipo 1042 (120 watts).
 - resultado obtido: problema não identificado pelo sistema.
 - ação: incluir na base de conhecimento.

15. Desconecte o fusível F50 ou F51 ou F52, aguarde até que o conjunto de aquecimento elétrico do tanque 3 seja desligado. A temperatura de processo vai cair para abaixo do especificado e acionar o alarme de processo. Consulte o sistema especialista para identificar o problema de queda de temperatura do tanque 3.
 - diagnóstico esperado: Queima de um dos fusíveis: F50, F51 ou F52.
 - resultado obtido: diagnóstico correto.

16. Dado um problema na correia sincronizada do carro de movimento vertical, consulte o sistema especialista sobre quais os possíveis elementos de segurança a serem acionados ?
 - diagnóstico esperado: chaves 49Q5 e 49Q7.

- resultado obtido: diagnóstico correto.
17. Desconecte o exaustor sobre os tanques de número 3 e 4 processo, aguarde o acionamento do alarme de segurança no painel de controle e consulte o sistema especialista para a identificação das possíveis causas.
- diagnóstico esperado: De acordo com os documentos de projeto, um dos elementos abaixo deverá ser identificado como a causa do problema.
 - queima dos fusíveis F109, F110 ou F111 (10.0 amperes),
 - problema na chave contatora 16K1
 - problema no exaustor 16M1 - tipo TEA 01-250-4, 1200 watts.
 - resultado obtido: problema não identificado pelo sistema.
 - ação: incluir na base de conhecimento.
18. Durante o processo de abastecimento dos tanques 1 e 2, através da bomba manual de imersão dos tanques, a bomba parou de repente. Quais os possíveis diagnósticos para esse problema ?
- diagnóstico esperado: De acordo com os documentos de projeto, um dos elementos abaixo deverá ser identificado como a causa do problema.
 - queima do fusível F104 (10.0 amperes),
 - problema na chave contatora 14K0
 - problema no soquete elétrico número X1.
 - resultado obtido: problema não identificado pelo sistema.
 - ação: incluir na base de conhecimento.
19. O alarme identificado como: “Control Voltage In”, localizado no painel de controle, está indicando um problema. Quais os possíveis diagnósticos para esse problema ?
- diagnóstico esperado: De acordo com os documentos de projeto, trata-se de um problema no sistema central de recebimento e distribuição de energia de alguns sistemas vitais da máquina. E os passos para se identificar as possíveis causas são: Vá até o módulo mais a esquerda do painel de controle (olhando-se o painel de frente) e tome as seguintes ações:
 - pare todo o processo de limpeza e retire todas as peças que estão sendo processadas,
 - certifique-se de que a chave geral está ligada,
 - certifique-se de que não há nenhuma lâmpada queimada no painel de controle, através de um teste de todas as lâmpadas, apertando o botão identificado como: “Lamp Test”
 - meça a presença de tensão em cada uma das três fases de entrada, que deve resultar em 440 volts,
 - verifique a integridade dos fusíveis principais: F1, F2 e F3 - 63 amperes cada,
 - verifique a integridade dos fusíveis do sistema de retificação: F11, F12 e F13 - 20 amperes cada,
 - verifique se disjuntor térmico Q2 (4.0 - 6.0 amperes) foi acionado,
 - verifique a integridade dos fusíveis do sistema de transporte: F21, F22 e F23 - 20 amperes cada,
 - verifique a integridade dos fusíveis do sistema de regeneração de solvente: F31, F32 e F33 - 20 amperes cada,
 - caso não seja nenhuma das causas acima, chame o engenheiro responsável pelo processo.

- resultado obtido: diagnóstico correto.
20. Desconete o fusível F11, aguarde a parada do conjunto retificador de corrente e o posterior alarme no quadro de controle da máquina. Consulte o sistema especialista.
- diagnóstico esperado: queima de um dos fusíveis: F10, F11 ou F12.
 - resultado obtido: diagnóstico correto.
21. Desconete a chave K301, aguarde a parada do sistema elétrico da bomba de calor do sistema de regeneração de solvente com o posterior alarme no quadro de comando. Consulte o sistema especialista para diagnóstico.
- diagnóstico esperado: problema na chave contatora K301.
 - resultado obtido: diagnóstico correto.
22. Uma possível causa de problema no módulo gerador de ultrassom do tanque 1 poderia ser os fusíveis F101 e F102. Quais outras possíveis causas de problemas no módulo gerador de ultrassom para o tanque 1 poderiam ser identificados ?
- diagnóstico esperado: identificação de um problema no módulo gerador de ultrassom do tanque 1.
 - resultado obtido: não é possível diagnosticar um problema dessa natureza, porque decidiu-se que dada a natureza do sistema de geração de ultrassom, o diagnóstico e reparo dos mesmos deverá ser feita somente pelo engenheiro eletrônico responsável.
23. Afrouxe no mínimo dois parafusos de fixação dos elementos transdutores de ultrassom. O som emitido pelos elementos se alterará significativamente, porém nenhum alarme será acionado pela máquina. Consulte o sistema especialista.
- diagnóstico esperado: afrouxamento dos parafusos de fixação dos transdutores, em função da vibração inerente ao transdutor.
 - resultado obtido: diagnóstico correto.
24. Alimente a máquina com no mínimo 20 cestos de peças vazios e peça aos operadores da sala limpa não retirarem os cestos dos transportadores que provem da máquina de limpeza KLN. Aguarde a manifestação de um alarme consulte o sistema
- diagnóstico esperado: congestionamento do processo devido a não retirada dos cestos do transportador dentro da sala limpa.
 - resultado obtido: diagnóstico correto.
25. Vá até a área de “kitting” e pegue um cesto de peças com as alças desalinhadas e coloque no transportador de entrada da máquina. Alimente a máquina até que o cesto desalinhado entre na antecâmara número 2. A partir desse instante a máquina não se alimentará mais de cestos. Consulte o sistema para saber o motivo desse problema.
- diagnóstico esperado: O elevador número 1 não está pegando o cesto “da vez”, provavelmente devido a um desalinhamento das alças do cesto.
 - resultado obtido: diagnóstico correto.
26. O alarme identificado como: “Check Roller-Track”, localizado no painel de controle, está indicando um problema. Consulte o sistema para saber quais os possíveis diagnósticos para esse problema.

- diagnóstico esperado: De acordo com a documento de projeto da máquina, esse alarme é acionado quando há um um problema de sobrecarga em um dos seis segmentos de rolo da máquina - três segmentos na alimentação de peças e três segmentos na saída de peças. E o alarme é disparado pelo acionamento de um dos seguintes disjuntores térmicos:
 - a. 49Q1 - motor do segmento de rolos 1 - entrada, externa à máquina.
 - b. 49Q2 - motor do segmento da ante-camera de entrada.
 - c. 49Q3 - motor do segmento sob o elevador de alimentação, elevador número 1.
 - d. 59Q4 - motor do segmento sob o elevador de descarga, elevador número 10.
 - e. 59Q5 - motor do segmento da ante-camera de saída.
 - f. 59Q6 - motor do segmento de rolos 6 - saída, externa à máquina (túnel sala limpa).

Os disjuntores acima poderão ter sido acionados por um dos seguintes motivos:

- acidente com o cesto de peças e/ou peças que caíram no transportador,
- problema no motor de acionamento dos rolos,
- problema na corrente de transmissão,
- falha do disjuntor térmico,
- problema no rolamento dos rolos,

- resultado obtido: diagnóstico correto.

27. Desconecte a válvula solenóide 12Y2 e aguarde que o alarme do sistema 1 da bomba de calor seja acionado. Consulte o sistema sobre a possível causa do acionamento do alarme do sistema 1 da bomba de calor.

- diagnóstico esperado: problema na válvula 12Y2 responsável pelo circuito direto de expansão do trocador de placas do do sistema 1 da bomba de calor.
- resultado obtido: diagnóstico correto.

28. Acione manualmente o disjuntor térmico 12Q7, aguarde o desligamento do compressor TYP4 e o acionamento do alarme no painel de controle. Consulte o sistema sistema especialista.

- diagnóstico esperado: problema no disjuntor térmico 12Q7.
- resultado obtido: diagnóstico correto.

29. Se a válvula que controla o circuito de resfriamento do tanque 1 - “shut down cooling” estiver fechada qual o diagnóstico esperado ?

- diagnóstico esperado: o tanque 2 encontra-se em processo de resfriamento.
- resultado obtido: diagnóstico correto.

30. Desligue o compressor TYP17, aguarde o alarme do painel de controle, e consulte o sistema especialista sobre as possíveis causas.

- diagnóstico esperado: de acordo com o histórico e com a documentação de projeto a parada desse compressor pode ter origem em um dos seguintes fatores:
 - queima de um dos fusíveis: F60, F61 ou F62 (25 Amp),
 - desarme do disjuntor térmico 18Q0,

- problema na contatora 18K0,
 - problema no pressostato de alta pressão,
 - resultado obtido: diagnóstico correto, porém incompleto pois existem outros elementos do circuito termodinâmico que poderiam causar um problema no compressor TYP17.
 - ação: é fato que o conhecimento termodinâmico poderia ter sido inserido na base de conhecimentos, porém como a medição dos parâmetros do circuito exigem equipamentos e técnicas específicas de um especialista em refrigeração, decidiu-se que os circuitos seriam parcialmente diagnosticados. Concentrando-se nos módulos de acionamento elétrico e pressostatos de acionamento elementar.
31. Desligue a resistência elétrica R2 e aguarde aproximadamente 20 minutos para o acionamento do alarme de problema na bomba de calor do sistema 1. Consulte o sistema especialista para identificar possível razão desse fato ?
- diagnóstico esperado: esse problema tem origem no trocador de placas da bomba de calor do sistema 1 e pode ter uma das seguintes causas:
 - o trocador de placa está recebendo água muito gelada - abaixo de 6 oC, ou a vazão de água está muito baixa - menor que a metade da vazão permitida pela válvula,
 - problema no sistema de aquecimento elétrico - ver resistência de número R2 (2.8 kW),
 - problema no compressor TYP11 (4 kW / 7 amp)
 - resultado obtido: diagnóstico correto.
32. Ligue a chave 50Y1, certifique-se de que o freio motor do carro de movimentação horizontal foi acionado. Aguarde o acionamento do rele térmico e do alarme do sistema de transporte.
- diagnóstico esperado: problema na chave 50Y1.
 - resultado obtido: diagnóstico correto.
33. Desconecte o único elemento resistivo da unidade de regeneração de solventes. Essa ação provocará a interrupção do processo de destilação do solvente. Aguarde o alarme e consulte o sistema
- diagnóstico esperado: De acordo com a documentação de projeto da máquina e com a experiência, um dos seguintes elementos pode estar apresentando o problema:
 - queima de um dos fusíveis: F205, F206 ou F207
 - problema na chave K300, ou
 - problema de continuidade na resistência elétrica da unidade de regeneração de solvente.
 - resultado obtido: diagnóstico correto.
34. Desligue manualmente a chave 49K5, veja o carro de transporte horizontal parar perder o movimento rápido para a direita. Consulte o sistema especialista.
- diagnóstico esperado: problema na chave contatora 49K5
 - resultado obtido: diagnóstico correto.
35. Acione manualmente o rele térmico 52Q3 e veja elevador móvel número 4 parar. Consulte o sistema especialista.
- diagnóstico esperado: problema no rele térmico 52Q3.
 - resultado obtido: diagnóstico correto.

36. Desligue manualmente a chave 53K7, veja o elevador móvel número 6 perder o movimento lento para cima. Consulte sistema especialista.
- diagnóstico esperado: problema na chave contatora 53K7.
 - resultado obtido: diagnóstico correto.
37. Desligue manualmente a chave 54K1, veja o elevador móvel número 7, perder o movimento rápido para baixo. Consulte o sistema especialista.
- diagnóstico esperado: problema na chave contatora 54K1.
 - resultado obtido: diagnóstico correto.
38. Acione manualmente o freio motor do elevador fixo número 12, através da chave 57Y6 e veja o elevador parar. Consulte o sistema especialista.
- diagnóstico esperado: problema na chave 57Y6.
 - resultado obtido: diagnóstico correto.
39. Em caso de problema na válvula solenóide 12Y3 do circuito indireto de troca de calor da bomba de calor a chave contatora 12K4 pode ser responsabilizada pelo problema ?
- diagnóstico esperado: sim, a chave 12K4 é a chave responsável pela alimentação da solenóide 12Y3 do circuito indireto.
 - resultado obtido: diagnóstico correto.
40. Acione manualmente a chave 57K7 e veja o elevador fixo número 3 perder o movimento rápido para cima. Consulte o sistema especialista.
- diagnóstico esperado: problema na chave 57K7.
 - resultado obtido: diagnóstico correto.
41. Acione o rele de segurança 59K0 do elevador fixo número 4, ou elevador de número 14. Veja o elevador parar e consulte o sistema especialista.
- diagnóstico esperado: problema na chave 59K0.
 - resultado obtido: diagnóstico correto.
42. Desconecte manualmente o fusível F64 que desligará o compressor TYP10. Veja o alarme no painel de controle e consulte o sistema especialista.
- diagnóstico esperado: queima do fusível F64.
 - resultado obtido: diagnóstico correto.
43. Desligue a chave 19K2.1 e aguarde a queda da temperatura dos tanques de pré-aquecimento 7,8 e 9, devido ao desligamento da resistência elétrica número 4.
- diagnóstico esperado: problema na chave 19K2.1
 - resultado obtido: diagnóstico correto.

44. Reduza a pressão de ar comprimido da máquina para 3.0 bar e veja que todo o sistema de transporte da máquina será paralisado pelo pressostato de segurança, sem fornecer qualquer alarme ao painel de controle. Consulte o sistema especialista para saber o motivo da parada “silenciosa” do sistema de transporte. Obs.: O pressostato tem por objetivo evitar qualquer acidente com as guilhotinas das antecâmaras de entrada e de saída da máquina de limpeza.
- diagnóstico esperado: pressão do ar comprimido abaixo de 5.0 bar.
 - resultado obtido: diagnóstico correto.
45. Qual a causa mais provável para o acionamento de alarme do sensor de infravermelho ?
- diagnóstico esperado: se o acionamento não for acidental pelo operador, a causa mais provável é um acidente interno na máquina por uma falha dos sensores indutivos - de posição que acarretaram na sobreposição dos cestos de peças. E consequentemente causam um desalinhamento do conjunto transportador.
 - resultado obtido: diagnóstico correto.
46. Desligue qualquer um dos elementos de proteção do elevador móvel de número 3: 51K4, 51K5, 51K6, 51K7, 52K0, 52Y0, 51Q4 ou 51Q6. Desligue a máquina e religue logo a seguir, certificando-se de que todo o sistema de transporte não está conseguindo completar o ciclo de inicialização e reconhecimento das coordenadas do sistema. Consulte o sistema especialista.
- diagnóstico esperado: problema em alguns dos seguintes componentes de acionamento e/ou proteção desse elevador: 51K4, 51K5, 51K6, 51K7, 52K0, 52Y0, 51Q4 ou 51Q6.
 - resultado obtido: diagnóstico correto.
47. Desconecte a válvula solenóide 1Y6, causando a falta de água de circulação no circuito 2 da bomba de calor. Aguarde o alarme das bombas de calor e consulte o sistema especialista.
- diagnóstico esperado: problema na válvula solenóide 1Y6.
 - resultado obtido: diagnóstico correto.
48. Acione manualmente o rele térmico Q301 e cause um problema no circuito de refrigeração 2, devido a parada do compressor TYP13. Aguarde o alarme desse sistema e consulte o sistema de diagnóstico.
- diagnóstico esperado: problema no rele térmico Q301.
 - resultado obtido: diagnóstico correto.
49. Desconecte o fusível F123 ou a chave K2 e veja o Controlador Lógico Programável se desligar. Consulte o sistema especialista.
- diagnóstico esperado: queima do fusível F123 ou problema na chave contatora K2.
 - resultado obtido: diagnóstico correto.
50. Qual a causa mais provável para altas perdas de solvente das câmaras 1 e 2 de solvente ?
- diagnóstico esperado: o termostato do sistema de resfriamento provavelmente está ajustado para uma temperatura superior a 15 oC.
 - resultado obtido: diagnóstico correto.
51. Qual a causa mais provável para um transbordamento de água desmineralizada do tanque número 5 ?

- diagnóstico esperado: problema no sensor de nível 4S0.
 - resultado obtido: diagnóstico correto.
52. Desarme manualmente a chave 49Q5 localizado no painel de controle e veja o carro horizontal do sistema de transporte perder o movimento rápido. Consulte o sistema especialista.
- diagnóstico esperado: problema na chave 49Q5.
 - resultado obtido: diagnóstico correto.
53. Desconecte eletricamente o motor 49M2, e veja o transportador de rolos da ante-câmara de entrada (segmento 2) parar de funcionar. Consulte o sistema especialista.
- diagnóstico esperado: problema no motor 49M2.
 - resultado obtido: diagnóstico correto.

Apêndice B. Estruturas Funcionais do ESE

A base de conhecimento é composta por:

- **Objetos:** objetos da base de conhecimento são os elementos que definem a base de conhecimento. Esses elementos são: parâmetros, regras, FCBs (do inglês “Focus Control Block”), grupos e telas.
- **Método de inferência:** Método de inferência é a lógica que define como o problema é desenvolvido na base de conhecimento. As duas técnicas de lógica que podem ser utilizados no método de inferência são conhecidos por: encadeamento para trás e encadeamento para frente.

B.1 Objetos da base de conhecimento

O domínio do conhecimento é composto pelos seguintes objetos:

- fatos do domínio, chamados de parâmetros.
- relacionamento entre os fatos do domínio, chamado de regras.
- unidades de trabalho, chamados de FCBs - blocos de controle.
- coleção de parâmetros, regras ou FCBs, chamado de grupos.
- apresentador, chamado de telas.

B.1.1 Parâmetros

Um parâmetro é um fato no domínio da base de conhecimento. Um parâmetros tem um nome, como por exemplo “forma_do_objeto”, ou “tamanho_do_objeto”, sendo que existem cinco tipos de parâmetros. Um parâmetro pode assumir um ou mais valores, e ao seu valor pode ser especificado um grau de certeza. Um parâmetro também é portador de um número de características - chamadas propriedades. Define-se essas propriedades para dar ao sistema a informação que ele necessita para usar o valor do parâmetro. **Tipos de parâmetros**

Os cinco tipos de parâmetros são definidos pelo tipo de informação que o usuário final pode entrar como valor do parâmetro. E podem ser um dos valores a seguir:

1. parâmetro numérico:- é atribuído a um valor numérico,
2. parâmetro sequência de bits:- é atribuído a um valor binário,
3. parâmetro sequência de hexadecimais - é atribuídos a um valor hexadecimal,
4. parâmetro sequência de caracteres:- é atribuído a uma sequência de caracteres como o valor do parâmetro,
5. parâmetro booleano:- é atribuído a um valor verdadeiro ou falso.

Valores de parâmetros

Um parâmetro pode ter um ou múltiplos valores. Quando da definição de um parâmetro pode-se associar uma das seguintes características:

- univalorado:- onde pode por exemplo assumir um valor numérico 127 do parâmetro “valor_de_torque”

- multivalorado:- onde pode por exemplo assumir valores como: “níquel”, “cromo”, “vanádio” do parâmetro “elementos_da_liga_metálica”
- ordenado:- onde pode ter vários valores que devem aparecer em uma ordem particular como: “engrenagem_da_1a_marcha”, “engrenagem_da_2a_marcha”, “engrenagem_da_3a_marcha” do parâmetro “sequência_de_engrenagens”

Certeza dos parâmetros

O sistema pode solicitar ao usuário final um valor de certeza do parâmetro, e se apropriado, o usuário pode designar um fator de certeza ao valor. A certeza corresponde ao grau de confiança no valor do parâmetro. Pode acontecer do usuário final nunca ter 100% de certeza de uma resposta e o sistema ESE é capaz de processar de processar informações que não são certas. Quando fornecido um valor de parâmetro, o usuário final pode designar a certeza a sua resposta com um número entre -1.0 e +1.0, em frente ao valor mostrado pela tela do programa. Entretanto, entrando-se com valores de certeza entre -0.2 e +0.2 tem-se um grau de certeza tão baixo que o valor é considerado como desconhecido, pelo sistema.

Propriedades dos parâmetros

Consiste na identificação de características ou de outras informações utilizadas pelo sistema na determinação ou na demonstração do valor do parâmetro. Pode-se definir e/ou modificar as propriedades de uma parâmetro para um dos abaixo relacionados:

- restrição (“constraint”):- restrição inicial ao valor do parâmetro,
- sequência de fornecimento (“sourcing sequence”):- ordem em que os parâmetros são fornecidos e considerados para o processo de encadeamento para trás,
- indicador (“prompt”):- pergunta, mostrada pelo sistema ao usuário final, quando necessita de uma entrada relativa a um dado parâmetro.
- indicador longo (“long prompt”):- é o texto mostrado pelo sistema em resposta do comando “WHAT”.
- formato da máscara (“format mask”):- propriedade que controla a aparência de um número quando o mesmo é mostrado ao usuário.
- restrição padrão (“default constraint”):- valor do parâmetro pré-definido, caso nenhum outro seja definido.
- valor esperado (“expect value”):- um valor ou uma faixa de valores esperado pelo sistema proveniente do usuário final. Valores fora da faixa não são considerados inválidos.
- tela (“screen”):- nome da tela ou relação dos nomes das telas para uso do sistema quando solicitado ao usuário final para uma dada entrada.
- nome da fonte do dado (“data source name”):- nome descritivo de uma rotina de origem de dados, utilizado no modo embutido.
- nome do procedimento (“procedure name”):- nome do procedimento do dado externo.
- argumentos do procedimento (“procedure args”):- argumentos a serem passados para a rotina de dados externos.
- indicador de mudança do valor (“value can change flag” - val can chg flg):- indicador que mostra se o valor do parâmetro pode ou não ser modificado após a resolução.
- comentário (“comment”):- comentário sobre o parâmetro.
- nome (“name”):- nome dado ao parâmetro.
- nome de apresentação (“print name”):- nome que o sistema mostra ou imprime para esse parâmetro.
- autor (“author”):- autor do parâmetro.

restrição (“constraint”)

Por definição um parâmetro pode ter os seus valores restritos. A essa restrição dá-se o nome de: “constraint”. Cada parâmetro pode ter uma restrição definida. E as restrições devem ser definidas como uma das abaixo relacionadas:

- seleção de uma lista,
- lista,
- intervalo inclusive/exclusive,
- igualdade.

Por exemplo, a restrição de um parâmetro chamado “fonte_de_energia” pode ser definida como:

```
TAKEN FROM("solar", "hidrica", "nuclear"); MULTIVALUED
```

Essa restrição para o parâmetro “fonte_de_energia” significa que o usuário final pode selecionar somente os valores listados: “solar”, “hidroelétrica”, “nuclear”. Porque o parâmetro tem a restrição - MULTIVALUED (multivalorado), o usuário final pode selecionar mais de um valor da lista do parâmetro “fonte_de_energia”

B.1.2 Regras

Uma regra define a relação de dois ou mais parâmetros em uma base de conhecimento. Uma regra consiste de duas partes: uma cláusula SE (“IF”) -chamada de premissa, e uma cláusula ENTÃO (“THEN”) - chamada de ação. O sistema prova se a premissa é verdadeira ou falsa examinando os valores do parâmetro. Se a premissa é verdadeira, o sistema tomará a ação especificada na cláusula de ação da regra.

A premissa examina os valores dos parâmetros e a ação confirma os valores dos parâmetros.

Propriedades das regras

As seguintes propriedades de uma regra podem ser editadas:

- texto da regra (“rule”):- cláusulas de premissa e ação que compõe o texto da regra.
- FCBs possuídos (“owning FCBs”):-relação de todos os FCBs que usam essa regra.
- tipo de regra (“rule type”):- de inferência,ou monitor de disparo único, ou monitor de múltiplos disparos.
- comentário (“comment”):- comentário sobre a regra.
- justificativa (“justification”):- informação de referência para a regra.
- nome (“name”):- nome da regra.
- nome de apresentação (“print name”):- nome da regra mostrada ao usuário final.
- autor (“author”):- identificação do autor da regra.

Cláusula de premissa da regra

Uma declaração de premissa de uma regra inicia-se com a cláusula “IF” seguida por uma ou mais cláusulas de premissa separadas por “AND” ou “OR”. Sendo que o operador “AND” tem uma precedência maior que o operador “OR”. Precedências adicionais podem ser arranjadas pelo uso de parêntesis. Existem sete tipos de cláusulas de premissas de regra que se seguem como abaixo relacionadas:

- cláusula de relação:- Esse tipo de premissa indica uma relação entre valores ou conjunto de valores. Exemplo: SE $x > y$ ENTÃO...
- cláusula de existência:- Esse tipo de premissa é provada por verdadeiro ou falsa através da avaliação da certeza do valor do parâmetro. Exemplo: SE existe UM parafuso_sextavado ENTÃO...
- cláusula de certeza:- Esse tipo de premissa é provada por verdadeiro ou falsa relacionando-se a certeza da premissa a uma expressão. Exemplo: SE A CERTEZA QUE (cor = "preta") ESTÁ NO INTERVALO $> = -1 < -0.1$ ENTÃO...
- cláusula de evidência:- Esse tipo de premissa é provada por verdadeiro ou falso pela comparação da certeza na premissa ao limiar da certeza. Exemplo: SE HÁ FORTE EVIDÊNCIA QUE material É "cerâmico" ENTÃO...
- cláusula de confiança:- Esse tipo de premissa é provada verdadeira ou falsa testando, se qualquer valor de uma parâmetro nominado encontra o teste de certeza especificado por uma expressão de confiança. Exemplo: SE material É CONHECIDO ENTÃO...
- cláusula de processamento:- Esse tipo de premissa testa se um parâmetro tem sido processado pelo sistema. Exemplo: SE cor TEM SIDO PROCESSADO ENTÃO...
- cláusula de seleção:- Esse tipo de premissa testa se um número mínimo de premissas de um dado grupo são verdadeiras. Exemplo: SE 2 DOS SEGUINTEs (comprimento = 3, largura > 2, altura < = 10) ENTÃO...

Cláusula de ação da regra

Uma declaração de ação de uma regra inicia-se com "THEN", seguido por uma ou mais cláusulas de ação separadas por "AND" ou vírgulas. As cláusulas de ação de regras podem:

- atribuir um valor para um parâmetro usando expressão como: "IS" ou =
- restringir o valor de um parâmetro usando uma expressão como: "TAKEN FROM "
- mostrar informações sobre o valor de um parâmetro ou uma relação de valores do parâmetro ao usuário final através da expressão "SHOW"
- eliminar um parâmetro, FCB ou grupo de consideração utilizando a expressão "DONT CONSIDER"
- processar diretamente para ou além de um FCB ou grupo utilizando a expressão "(DONT) PURSUE ANOTHER"
- iniciar a execução de um FCB ou uma lista de FCBs utilizando a expressão "ESTABLISH"

A seguir é apresentado um conjunto de exemplos de cláusulas de premissas e ação de regras:

1. IF concentração_soluto > 50.000 THEN solução IS 0.35

Se o valor do parâmetro numérico "concentração_soluto" for maior que o valor 50.000 então atribua o valor de 0.35 ao parâmetro número "solução"

2. IF tipo_de_metal IS "ferroso" e aparência_do_metal IS "brilhante" THEN característica IS "INOXIDÁVEL"

Se o valor do parâmetro do tipo sequência de caracteres "tipo_de_metal" é igual a sequência de caracteres "ferroso", e o valor do parâmetro do tipo sequência de caracteres "aparência_do_metal" é igual a sequência de caracteres "brilhante", então atribua a sequência de caracteres "inoxidável" para a sequência de caracteres "característica"

3. IF aquecimento IS TRUE OR resistência_elétrica IS "aquecida" OR resistência_elétrica IS "OK" THEN atividade IS "ferramenta_pronta" AND processo is "coloque_material"

Se o valor do parâmetro booleano "aquecimento" é igual a TRUE ou o valor do parâmetro do tipo sequência de caracteres "resistência_elétrica" é igual a sequência de caracteres "aquecida" ou "OK" então

atribua a sequência de caracteres “ferramenta_pronta” ao parâmetro do tipo sequência de caracteres “atividade” e atribua a sequência de caracteres “coloque_material” ao parâmetro do tipo sequência de caracteres “processo”

4. IF tempo_de_deslocamento IS NOT KNOW AND porosidade_tipo_xy IS KNOW THEN THERE IS 0.7 EVIDENCE THAT porosidade = porosidade_tipo_xy

Se o valor do parâmetro número “tempo_de_deslocamento” é desconhecido e um valor para o parâmetro “porosidade_tipo_xy” é conhecido, então faça o valor do parâmetro “porosidade” igual ao valor do parâmetro numérico “porosidade_tipo_xy” e atribua o grau de certeza igual a 0.7 para o valor de “porosidade”

5. IF 2 OF THE FOLLOWING (tem_pneu IS TRUE, transporte = “motor”, transporte = “câmbio”, transporte = “faról”, rodovia IS TRUE) THEN impressão = “automóvel”

Se quaisquer duas das condições premissas (listadas entre parêntesis) forem satisfeitas então atribua ao parâmetro do tipo sequência de caracteres “impressão” a sequência de caracteres “automóvel”

6. IF (parm_1_bool, parm_2_bool, parm_3_bool) IS TRUE THEN R = SQROOT(TOTAL)

Se pelo menos um dos valores dos parâmetros booleanos: “parm_1_bool”, “parm_2_bool”, “parm_3_bool” for igual a TRUE, então atribua ao parâmetro numérico “R” o valor igual a raiz quadrada do valor do parâmetro numérico “TOTAL”

Tipos de regra

Pode-se definir três tipos de regras:

1. inferência - “default”,
2. monitor de disparo único,
3. monitor de múltiplos disparos.

Regras de inferência são processadas como parte do processo de inferência do encadeamento para trás ou encadeamento para frente. Regras monitor de disparo único operam independentemente do processo de inferência normal e são processadas imediatamente após a atribuição de um valor a um parâmetro em uma das cláusulas de premissa da regra. Monitor de disparos múltiplos são similares ao monitor de disparo único exceto que podem ser executadas mais de uma vez.

B.1.3 FCB

“Focus Control Block ”

Um FCB diz ao ESE - Expert System Environment como levar a cabo uma única unidade de trabalho - ou *focus* durante o processo de consulta. Bases de conhecimento muito simples contam com um único FCB, entretanto bases de conhecimento mais complexas possuem múltiplos FCBs que são arranjados hierarquicamente, de forma a orientar o sistema durante o processo de consulta.

Primeiramente o sistema base de conhecimento processa o FCB principal denominado de raiz ou - “root”. O FCB raiz diz ao sistema qual o FCB na hierarquia deve ser processado a seguir dependendo do resultado obtido. O sistema segue a rota mapeada pela hierarquia dos FCBs até que o problema seja resolvido.

Por exemplo, assuma que um sistema baseado em conhecimento para a diagnose de falha de computadores tenha uma estrutura hierárquica de FCB semelhante a seguir.

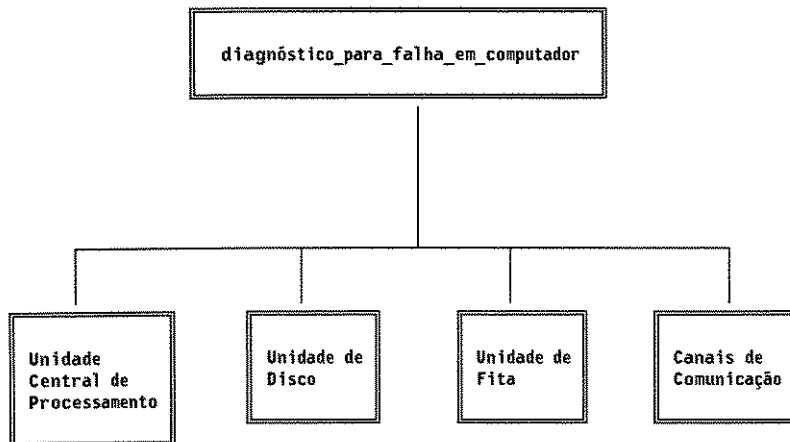


Figura 23. Exemplo de estrutura hierárquica FCB.

O FCB raiz diz ao sistema qual o dado inicial deve ser solicitado ao usuário final, e baseado nos resultados obtidos pelo processamento do dado inicial, o FCB raiz indica qual os FCBs da hierarquia inferior (chamados de filhos ou decedentes) deverão ser processados a seguir. Um FCB que decende um FCB filho é chamado FCB pai.

Propriedades do FCB

Cada FCB é definido por propriedades que especificam o desejado fluxo de controle para aquela unidade particular de trabalho. As propriedades de um FCB que podem ser editáveis incluem:

- gol ("goal"):- lista de parâmetros gol para esse FCB,
- dado inicial ("initial data"):-lista de parâmetros dos dados iniciais para esse FCB,
- resultados ("results"):- lista de parâmetros resultantes para esse FCB,
- dado externo ("external data"):- lista de parâmetros dos dados externos para esse FCB,
- parâmetros ("parameters"):- lista dos parâmetros utilizados por esse FCB,
- regras ("rules"):- lista das regras utilizadas por esse FCB,
- texto de controle ("control text"):- comandos de controle e opções de estratégia para esse FCB,
- anúncio ("announce"):- texto exibido ao usuário final no início do processamento da instanciação do FCB,
- solicitação inicial ("initial query"):- questão formulada ao usuário final antes do processamento da primeira instanciação de um FCB,
- solicitação adicional ("Addl Inst Query - additional instances query"):- questões formuladas ao usuário final antes do processamento das instâncias subsequentes de um FCB,
- distinguindo características ("Dist Features - distinguishing features"):- lista dos parâmetros que diferenciam instanciações de FCBs,
- número máximo de instâncias ("Max Instances - maximum number of instances"):- numero máximo de instanciações permitidas a esse FCB.
- tela de apresentação ("display screen"):- nome padrão de tela ou relação dos nomes padrões de telas.

- tela de múltipla escolha (“Mult Choice Scr - multiple choice screen”):- nome padrão de tela ou relação dos nomes padrões de telas.
- nome de apresentação (“print name”):- nome do FCB apresentado ao usuário final.
- comentário (“comment”):- comentário sobre FCB.
- ordenação dinâmica da regra (“DynRuleOrder”):- indica se o processo de ordenamento dinâmica das regras está ativo ou não.
- eliminar quando concluído (“DisposeWhenDone”):- indicação se a data e hora da execução do FCB deve ser apagada ou não da memória virtual.

Declarações

No ESDE - ambiente de desenvolvimento do ESE, a linguagem de controle consiste de nove declarações utilizadas para se dizer ao sistema como processar o conhecimento da base de conhecimento. Essas declarações são:

1. ACCESS :- estabelece relação com o SQL/DL e DB2 - banco de dados.
2. ACQUIRE :- obtém valores de parâmetros de fontes de dados externos.
3. ASK :- solicita dados ao usuário final a fim de se obter valores aos parâmetros.
4. DETERMINE :- invoca o módulo de encadeamento para trás com informações sobre as fontes dos novos valores (provenientes do usuário final, regras, “defaults”, ou fontes externas de dados) a fim de encontrar os valores dos parâmetros.
5. DISCOVER :- invoca o módulo de encadeamento para frente sob um conjunto de regras para encontrar os valores dos parâmetros.
6. DISPLAY :- mostra os parâmetros e e seus valores ao usuário final.
7. ESTABLISH :- inicia o processamento de outros FCBs.
8. EXECUTE :- permite uma base de conhecimento invocar outra.
9. PROCESS :- passa o controle para rotinas externas.

Estratégias de FCB

O ESDE também permite as estratégias abaixo listadas que sejam utilizadas para modificar as declarações nas linguagens de controle:

- ACQUIRE ALL :- adquire todos os objetos em uma chamada ACQUIRE.
- DONT ACQUIRE :- exclue objetos de ACQUIRE e DETERMINE.
- DONT ASK :- exclue objetos de ASK e DETERMINE.
- DONT INFER :- exclue objetos de DETERMINE.
- FOCUS ON :- considere objetos primeiro em: ACQUIRE,ASK,DETERMINE, DISPLAY e ESTABLISH.
- IGNORE :- ignore todos objetos durante ações de controle em: ASK, DETERMINE,DISPLAY e ESTABLISH.
- ORDER RULES BY :- ordene regras por esse critério em DETERMINE e DISCOVER.
- STOP ON :- pare nesse critério em DISCOVER.
- USE :- use objetos durante ação de controle em ASK,DISCOVER e DISPLAY.

Por exemplo, na seguinte declaração de controle e estratégia o sistema ordena o grupo de regras usadas para encontrar um valor para o parâmetro “falha”, colocando aqueles com menor número de parâmetros desconhecidos em suas cláusulas de premissa em primeiro.

```
DETERMINE falha -- ORDER RULES BY menor_numero_de_premissas_desconhecidas
    _primeiro
```

Também pode-se usar outras estratégias com DONT ACQUIRE, DONT ASK e DONT INFER, para alterar uma específica fonte de um objeto durante a execução de uma declaração do tipo DETERMINE. Por exemplo, para se otimizar e prevenir o sistema de perguntar ao usuário final sobre a microestrutura de um material, a base de conhecimento poderia ter a estratégia DONT ASK. Se a tensão de cisalhamento for importante o suficiente para ser determinado em primeiro quando houver uma escolha de parâmetros para se determinar, a estratégia FOCUS ON poderia ser adicionada. Assim as instruções ao FCB se pareceria com:

```
DETERMINE identidade_do_material
    -- DONT ASK microestrutura
    -- FOCUS ON tensão_de_cisalhamento
```

B.1.4 Grupos

São coleções de objetos similares da base de conhecimento. A definição de grupos torna mais fácil a referência de objetos similares. Por exemplo, pode-se definir um grupo de vários parâmetros e nomeá-lo de forma a referenciá-los como um único grupo ao invés de uma lista de nomes.

Parâmetros de Grupos

Pode-se editar os seguintes parâmetros para um definido grupo:

- autor (“author”):- o autor do grupo.
- lista de elementos (“member list”):- relação dos objetos desse grupo.
- comentário (“comment ”):- comentário sobre o grupo.
- nome (“name”):- nome do grupo.
- nome de apresentação (“print name”):- nome do grupo apresentado ao usuário final.

B.1.5 Telas

Pode-se criar telas de apresentação que sejam diferentes das telas de apresentação criadas automaticamente pelo sistema - telas “default”, utilizando-se o “screen layout facility”

Telas “customizadas” podem apresentar uma ou mais questões e/ou campos de respostas em uma tela. Também pode-se atribuir cores, negritos, intensidade, e molduras para qualquer campo. Pode-se também mudar a sequência de aparecimento dos campos nas telas, fornecer campos maiores para perguntas ou textos de respostas, ou ambos e designar campos específicos para os comandos: “HOW, WHY e WHAT ”.

Parâmetros de Telas

As propriedades de uma tela que podem ser editáveis incluem:

- autor (“author”):- o autor da tela.

- comentário (“comment”):- comentário sobre a tela.
- nome (“name”):- nome da tela.
- nome de apresentação (“print name”):- nome da tela apresentada ao usuário final.
- nome da fonte do dado (“Data Source Name”):- nome descritivo de uma rotina utilizada no modo embutido.

Não é requerido que se defina nenhuma propriedade para uma tela.

B.2 Métodos de inferência

Após a definição dos objetos da base de conhecimento, deve-se definir o método de inferência que solucionará as consultas. Durante o processo de consulta, o sistema tenta encontrar valores para um ou mais parâmetros inferindo-os das regras e dos valores conhecidos de outros parâmetros.

Existem dois métodos de raciocínio inferencial que podem ser utilizados pelo ESE. Esses métodos são: *encadeamento para trás*, que é direcionado a um objetivo e trabalha para trás - a partir de um dado objetivo, até encontrar os dados que o suportem. E o *encadeamento para frente*, que é direcionado a dados e trabalha para frente - a partir um dado conhecido até encontrar uma nova informação.

B.2.1 Encadeamento para trás

Na técnica de encadeamento para trás, seleciona-se parâmetros para os quais desejam-se valores, o sistema seleciona esses parâmetros de forma indireta durante consulta. Esses parâmetros são chamados de: objetivos da base de conhecimento.

O sistema procura por regras com cláusulas de ações que atribuam valores para os parâmetros selecionados, e executa as regras uma de cada vez. Em se processando as cláusulas da regra alguns parâmetros podem não ser conhecidos. Esses parâmetros desconhecidos são ajustados pelo sistema como: sub-gols, e o sistema passa a procurar por regras que possam utilizar para concluir sobre os novos sub-gols. Durante a execução dessas regras, o usuário final pode ser solicitado a fornecer valores para alguns dos parâmetros.

Uma das propriedades que se pode definir para parâmetros individuais é a sequência de fornecimento - “sourcing sequence”. Essa propriedade diz ao sistema onde encontrar informações para se determinar o valor do parâmetro. O sistema pode determinar o valor pelo processamento de regras, solicitando que o usuário final forneça os valores, obtendo um valor “default”, ou obtendo o valor de dados externos. A propriedade de sequência de fornecimento, relaciona a ordem em que o sistema considerará essas fontes. O encadeamento para trás usa essa propriedade para determinar as fontes para um valor de parâmetro.

B.2.2 Encadeamento para frente

Na técnica de encadeamento para frente o sistema coleta parâmetros com valores conhecidos na FCB sob considerações da lista de parâmetros. E também coleta regras de inferência aplicáveis a esse FCB de uma lista de regras. O sistema executa as ações declaradas para cada regra cuja premissa seja verdadeira, baseada nos valores dos parâmetros relacionados na lista de parâmetros. Da determinação de novos valores de parâmetros pela execução dessas regras, o sistema atualiza a lista de parâmetros de forma incluir os nomes desses parâmetros.

Os problemas, em sua maior parte, podem ser solucionados através do encadeamento para frente ou para trás das regras de produção. Na maioria das vezes, entretanto, um dos métodos é sempre preferível ao outro.

Uma regra geral para a escolha da técnica de inferência a ser utilizada é que se um houver a alteração de um item e deseja-se determinar a suas várias implicações, ou se deseja que a aplicação tenha uma entrada do tipo formulário onde tem-se várias informações conhecidas para uma análise, então utiliza-se o *encadeamento para frente*. Se há suspeita que algo ocorreu e deseja-se encontrar as causas de um conjunto de causas possíveis, ou prefere-se que a aplicação tenha uma interação baseada em diálogo, então utiliza-se o *encadeamento para trás*.

Exemplos de situações em que o encadeamento para frente é preferível incluem:

- um sensor indica uma falha de máquina e deseja-se descobrir quais as consequências que se seguirão.
- um usuário dá entrada de um valor errado em um resgate de seguro e deseja-se alertar o mesmo sobre o erro.
- um cliente fornece uma série de informações e deseja-se fazer uma recomendação para o mesmo sobre a aquisição de uma apólice ou investimento.
- o valor das ações caem repentinamente, e deseja-se prever a resposta do mercado.

E exemplos de situações em que o encadeamento para trás é preferível incluem:

- dado um defeito em um sistema, deseja-se identificar a possível causa do problema.
- observa-se um defeito em um dado produto e deseja-se encontrar a máquina/processo responsável.
- suspeita-se de um sobre-faturamento e necessita-se verificar as entradas erradas.
- o índice da bolsa de valores cai e deseja-se conhecer quais são as ações afetadas.