

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA MECÂNICA

DESENVOLVIMENTO DE UM
SISTEMA ESPECIALISTA DE USINAGEM

Maria Cristina Aranda Batocchio

Orientador: Prof. Dr. Nivaldo Lemos Cupini

Nº de Publicação FEM 053/93

Tese apresentada à Faculdade de
Engenharia Mecânica da Universidade
Estadual de Campinas, como parte dos
requisitos para obtenção do Título de
Mestre em Engenharia Mecânica.

Dezembro - 1993.

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL
DA TESE DEFENDIDA POR Maria Cristina
Aranda Batocchio E APROVADA PELA
COMISSÃO JULGADORA EM 26/09/94.

ORIENTADOR

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

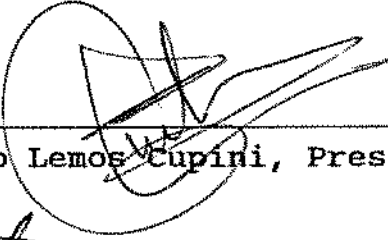
Tese de Mestrado

Título da Tese: Desenvolvimento de um Sistema
Especialista de Usinagem

Autor: Maria Cristina Aranda Batocchio

Orientador: Prof. Dr. Nivaldo Lemos Cupini

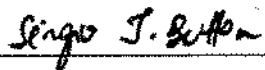
Aprovado por



Prof. Dr. Nivaldo Lemos Cupini, Presidente



Prof. Dr. Lourival Boehs



Prof. Dr. Sérgio Tonini Button

Campinas, 20 de dezembro de 1993.

Dedicatória

A meus pais Maria e Antonio,

Ao meu marido Toninho,

As minhas filhas Vanessa e Danielle;

três gerações que constituem

minha razão de ser.

Agradeço :

Ao Prof. Dr. Nivaldo Lemos Cupini pela orientação, mas, sobretudo pela confiança que em mim depositou, estimulando-me sempre que necessário.

A todos meus amigos do DEF/UNICAMP e CT/UNICAMP, sem citar nomes, para não correr o risco de esquecer alguém, pela amizade que me dedicaram nestes anos de UNICAMP.

Enfim, a todos os companheiros que me ajudaram a segurar grandes barras...

RESUMO

O objetivo deste trabalho é apresentar um Sistema Especialista de Usinagem, que otimizará as condições de usinagem com base na análise das condições de máxima produção, as quais são aqui consideradas como condição suficiente para a escolha da velocidade de corte otimizada.

O sistema foi desenvolvido para o nível de semi-automação, e, permite a determinação da velocidade ótima de usinagem a partir de valores armazenados em Bancos de Dados específicos.

Este sistema, possibilita a substituição do especialista humano em processos de manufatura, na complexa tarefa de otimização das condições operacionais de torneamento.

ABSTRACT

The purpose of this work is to present a Machining Expert System that will optimize the machining process based on the analysis of the conditions of maximum production and which are considered with the sufficient conditions in order to choose the optimized speed cut.

The related system was developed in semi-automation level and it allows to determine the machining optimal speed from values loaded in specifics Data Bases.

This system permits to replace the human expert in manufacturing process in the complex work of the optimization of turning operating conditions.

ÍNDICE

	Pag.
CAPÍTULO 1 - INTRODUÇÃO	
1.1. INTRODUÇÃO	1
1.2. OBJETIVOS	2
1.3. JUSTIFICATIVAS	3
 CAPÍTULO 2 - INTELIGÊNCIA ARTIFICIAL (IA)	
2.1. CONTEXTO GERAL	8
2.2. RETROSPECTIVA	10
2.3. INTELIGÊNCIA ARTIFICIAL X COMPUTAÇÃO CONVENCIONAL	14
2.4. PRINCIPAIS TÓPICOS DA IA	16
2.4.1. Busca Heurística	17
2.4.2. Representação do Conhecimento	17
2.4.3. Senso Comum e Lógica	18
2.4.4. Processamento de Linguagem Natural	18
2.4.5. Visão Computacional	19
2.4.6. Sistemas Especialistas	20
2.4.7. Robótica	21
2.5. LINGUAGENS E FERRAMENTAS DA IA	22
2.6. IA NO CONTEXTO DA MANUFATURA	23
2.6.1. Generalidades	23
2.6.2. Áreas de Manufatura que Utilizam-se de IA	26
 CAPÍTULO 3 - SISTEMAS ESPECIALISTAS (SE)	
3.1. CONTEXTO GERAL	31
3.2. COMPOSIÇÃO DOS SEs	33
3.2.1. Base de Conhecimento	33
3.2.2. Base de Dados	36
3.2.3. Máquina de Inferência	36

3.3. ALGUMAS ÁREAS DE APLICAÇÃO DE SEs	39
3.3.1. Áreas Gerais	39
3.3.2. Em Manufatura	40
3.3.3. Alguns dos Principais SE Construídos	42
3.3.3.1. Áreas Gerais	42
3.3.3.2. Específicos para Manufatura	45
3.4. PRINCIPAIS TIPOS DE SEs	47
3.4.1. Interpretação	47
3.4.2. Previsão	48
3.4.3. Diagnose	48
3.4.4. Planejamento	49
3.4.5. Análise de Desempenho	49
3.4.6. Treinamento	49
3.5. CARACTERÍSTICAS DE UM SE	51
3.5.1. Flexibilidade	51
3.5.2. Adaptabilidade	51
3.5.3. Desempenho e Eficiência	51
3.5.4. Explicação do Raciocínio	51
3.5.5. Interação com o Usuário	52
3.5.6. Incerteza	52
3.6. FASES PRINCIPAIS DE DESENVOLVIMENTO DE UM SE	53
3.7. CONHECIMENTO HUMANO X ARTIFICIAL	54
3.8. ARQUITETURA DE UM SE	56
3.9. SISTEMAS ESPECIALISTAS NO CONTEXTO DA MANUFATURA	58

CAPÍTULO 4 - AS BASES DA TECNOLOGIA DA USINAGEM

PARA REALIZAÇÃO DO SEU

SISTEMA ESPECIALISTA DE USINAGEM

4.1. OTIMIZAÇÃO DAS CONDIÇÕES DE USINAGEM	59
4.2. DETERMINAÇÃO DAS CONDIÇÕES DE USINAGEM	60

CAPÍTULO 5 - SISTEMA ESPECIALISTA DE USINAGEM (SEU)

5.1. GENERALIDADES	67
5.2. METODOLOGIA	67
5.3. ARQUITETURA DO SEU	71
5.3.1. Automatização do Sistema	73
5.3.2. Banco de Dados	74
5.3.3. Base de Conhecimento	74
5.3.4. Máquina de Inferência	74
5.3.5. Interface com o Usuário	79

CAPÍTULO 6 - COMUNICAÇÃO ENTRE O SEU E O SBD

6.1. GENERALIDADES	90
6.2. SISTEMA DE BANCO DE DADOS (SBD)	91
6.2.1. Generalidades	91
6.2.2. Estrutura de um Banco de Dados	94
6.2.3. Fases de Projeto de um Banco de Dados	96
6.3. INTERFACEAMENTO	97
6.3.1. CLIPPER/dBASE III	97
6.3.2. Linguagem PROLOG	104
6.3.2.1. Elementos Básicos da Linguagem PROLOG	106
6.3.2.2. Tipos de Dados PROLOG	109
6.3.3. Características da Interface	110

CAPÍTULO 7 - CÁLCULOS ESTATÍSTICOS

7.4.1. Dados de Ensaio	114
7.4.2. Estimativa da Grandeza da Amostra	114

CAPÍTULO 8 - CONCLUSÕES

121

BIBLIOGRAFIA CITADA	124
ABREVIATURAS UTILIZADAS	131
RELAÇÃO DE TABELAS	133
RELAÇÃO DE FIGURAS	134
APÊNDICE	136
SIMULAÇÕES E TESTES	141

1.2. OBJETIVOS

O objetivo deste trabalho é apresentar o desenvolvimento do Sistema Especialista de Usinagem (SEU). Este sistema deverá ser utilizado para otimizar as condições de usinagem em células de manufatura, para produção em lotes, pressupondo a existência de sensores, que garantam fixar um determinado critério de fim de vida de ferramentas, que será utilizado para definir o momento da troca da ferramenta. Este objetivo será atingido através do:

1. Desenvolvimento de um sistema que calcule os coeficientes de vida de ferramentas x e K de Taylor com base no número médio de peças usinadas por vida, contadas durante uma produção normal em ambiente fabril;

2. Determinação da velocidade de máxima produção, compatível com o intervalo de velocidades empregadas durante a usinagem e, com base nessa velocidade, decidir sobre o valor da velocidade ótima de trabalho;

3. O referido sistema será construído para trabalhar em nível de semi-automação, mas mesmo assim é um sistema capaz de tomar decisões para satisfazer os objetivos 1 e 2.

1.3. JUSTIFICATIVAS

A Inteligência Artificial e suas aplicações tendem a se firmar como um instrumento poderoso para o aumento da competitividade e a busca da eficiência, isto devido a um perfil econômico mundial que prevê a globalização dos mercados. Esta tecnologia tem se mostrado um instrumento eficiente, tanto a nível prático, quanto econômico (Oliveira 1991).

Um programa especialista, tenta combinar os dados sobre os quais ele foi informado com possíveis sintomas ou condições que ele reconhece. O programa usa tal evidência para recomendar ou providenciar uma ação (Rosham e Sudesh 1990).

Os Sistemas Especialistas contém conhecimentos generalizados, como uma coleção de "know-how", experiência e método para resolução de certos problemas dentro de uma área restrita (Krause 1991).

Nos EUA foram observadas economias de até US\$ 200.000 a US\$ 1 bilhão. Nas indústrias de manufatura, a aplicação bem sucedida dos Sistemas Especialistas já é evidente (Schorr 1989). Este é um aspecto que não se observa com tanta evidência no Brasil, pois a SOBRACON (Sociedade Brasileira de Comando Numérico, Automatização Industrial e Computação Gráfica) realizou um estudo de mercado no setor de Automatização Industrial no Brasil, demonstrando o "sobe

e desce" deste segmento no Guia Brasileiro de Automação Industrial 1992.

Ano	Unidades CNC	Máquinas CNC	CP's	Robôs	CAE/CAD /CAM	Total
1986	22,2	187,0	27,8	2,9	12,1	252,1
1987	20,0	197,5	34,0	5,1	29,0	285,6
1988	13,9	223,5	51,3	2,0	36,0	326,7
1989	20,1	224,9	65,1	4,0	99,3	413,4
1990	9,8	88,8	52,4	0,5	26,5	178,0
1991	7,7	78,4	43,6	0,6	35,9	166,2
1992*	7,7	90,0	61,5	0,8	59,6	219,6

* : Previsão

Tabela 1 - Faturamento dos Setores de Automação Industrial no Brasil (em US\$ milhões)

Na Tabela 1 pode-se verificar que a área de CAE/CAD/CAM (Engenharia, Projeto e Manufatura Assistidos por Computador) teve seu pico de faturamento em 1989, mas o mercado não deixa de investir nesta área.

A aplicabilidade do computador na indústria deve ser aprimorada e desmistificada, sendo que os mesmos, tornaram-se acessíveis a um número cada vez maior de usuários, e as sofisticadas leis de controle e automação passam a estar à disposição de potenciais aplicações, para realização de estratégias cada vez mais elaboradas na solução de problemas industriais.

Ao utilizarmos os benefícios da otimização dos parâmetros de usinagem, o que é possível quando se usa um eficiente modelo do processo, podemos verificar com clareza através de cálculos a economia que se obtém.

Colding (Colding 1992) faz considerações interessantes em seu artigo, tais como:

1- Ainda não foram feitos estudos abrangentes para a avaliação do nível médio de dados ótimos. Sua experiência pessoal é de que o nível médio (produto da velocidade e avanço) é de 40% inferior ao nível ótimo, isto é, nas condições mínimas de custo de usinagem, apesar da maioria das máquinas operatrizes modernas terem potência e velocidade de corte suficientes para trabalhar nas condições ótimas de usinagem. Ele consegue resumir os resultados dos estudos comparativos sobre dados reais de produção e os parâmetros de usinagem otimizados na Tabela 2, sendo que o ganho médio é de 40%.

2- Razões pelas quais as indústrias não estão atualizadas:

. O custo do ferramental é apenas de 1% a 2% do custo total de manufatura, de modo que usinagem é assunto de menor importância. Entretanto, é nesta atividade que são agregados valores dos produtos.

. As variações de usinabilidade poderiam destruir as ferramentas em velocidades e avanços elevados.

. A questão da escolha das melhores ferramentas e dados de usinagem ótimos é extremamente complicada e pouquíssimos engenheiros aplicam este conhecimento.

	Vida min	Veloci- dade-rpm	Avanco pol/rot	Tempo corte min	Redução tempo-t
1. Torneamento lon- gitudinal, metal duro, SAE 1045					
Real	100,00	427,00	0,0157	0,5138	
Otimo	153,00	609,00	0,020	0,2847	44,59
2. Performance Diamante, AL308					
Real	1.305,00	3.777,00	0,014	0,0442	
Otimo	349,00	7.340,00	0,0133	0,0239	45,93
3. Perfuração Diamante, AL308					
Real	10.000,00	2.041,00	0,0031	0,0659	
Otimo	1.488,00	1.488,00	0,007	0,0406	38,39
4. Torneamento lon- gitudinal, cerâ- mica, ferro fun- dido					
Real	276,00	666,00	0,0075	0,4589	
Otimo	47,00	817,00	0,0109	0,2574	43,91
5. Quatro operações Metal duro SAE 9240					
Real					
Perfuração	11,00	259,00	0,004	0,4188	
Torneamento	14,00	689,00	0,008	0,1720	
Torneamento	14,00	689,00	0,008	0,1010	
Sangramento	38,00	285,00	0,008	0,0972	
Tempo Total corte				0,7890	
Otimo					
Perfuração	29,00	340,00	-0,005	0,2501	40,28
Torneamento	11,00	555,00	0,020	0,0827	51,92
Torneamento	11,00	546,00	0,018	0,0545	46,04
Sangramento	30,00	340,00	0,005	0,0791	18,62
Tempo Total corte				0,4664	40,89

Tabela 2: Comparação entre os dados de usinagem atuais com
dados ótimos (Colding 1992)

O autor alega ser a última razão, a razão chave para o mau desempenho na usinagem, e as duas primeiras não têm fundamento concreto. Ele diz ainda que o conhecimento acumulado que existe sobre usinagem é, sem dúvida, impressionantemente grande, mesmo para um especialista.

3- O conhecimento mais moderno pode ser acessado através de PCs, como por exemplo o software **EXPERT**, desenvolvido pela Colding International Corporation, e através do qual pode-se construir softwares, como por exemplo o COMP (Planejamento da Manufatura Otimizado por Computador). O acesso a um programa destes é provavelmente a única maneira de uma pessoa deste ramo poder tirar vantagem do conhecimento atualmente existente sobre usinagem. Isto vai ser válido mais especificamente num futuro próximo quando existirem menos engenheiros especializados em ferramentas de corte, devido ao pequeno interesse nas áreas básicas em nossas faculdades de engenharia. Razões básicas para a existência de dados de usinagem insuficientes:

- . Poucos engenheiros se interessam por esta área;

- . Administradores dos escalões médio e alto estão pouco conscientizados das economias que podem ser obtidas com dados de usinagem otimizados em comparação com outros meios de redução de custos ou aumento de produtividade;

- . A maioria dos engenheiros e contabilistas não conseguem calcular adequadamente os custos, principalmente porque os atuais métodos de contabilidade de custos não servem para calcular o efeito de alterações nos parâmetros de manufatura.

CAPÍTULO 2

INTELIGÊNCIA ARTIFICIAL (IA)

2.1. CONTEXTO GERAL

A Inteligência Artificial (IA) é simplesmente uma maneira de fazer o computador pensar inteligentemente (Arariboia 1988). Isto é conseguido, estudando como as pessoas pensam quando estão tentando tomar decisões e resolver problemas, dividindo esses processos de pensamentos em etapas básicas e projetando um programa de computador que solucione problemas usando essas mesmas etapas (Levine et al. 1988).

A IA é então, um método estruturado de projetar programas complexos de tomada de decisões. Consiste num dos temas mais importantes desta área desde o surgimento do transistor (Schildt 1989). Seu sucesso hoje está baseado principalmente nos Sistemas Especialistas (SE), que foram os produtos financeiramente mais bem sucedidos em IA; na boa receptividade da IA entre os japoneses e na lenta, mas forte integração das técnicas de IA nas aplicações já existentes.

Ela visa basicamente (Kowalski 1983/84), projetar máquinas inteligentes (hardware) e processos computacionais (software) mais úteis e com entendimentos inteligentes.

As ferramentas e técnicas desenvolvidas em IA solucionam importantes problemas e criam novas oportunidades (Smart 1987).

Os programas computacionais com os quais a IA está relacionada são primariamente processos simbólicos envolvendo complexidade, incerteza e ambiguidade. Esses processos são usualmente aqueles para os quais não existem soluções algorítmicas e, é necessário pesquisar a solução (Lucena 1987) em um universo bastante grande de alternativas, ou seja, problemas típicos do ser humano no dia-a-dia (Cunha e Ribeiro 1987). Portanto, IA trata com tipos de problemas (soluções) que o ser humano encontra continuamente no mundo. Essa forma de solucionar problemas difere daquela utilizada nos métodos tradicionais de engenharia e métodos científicos, que são essencialmente de natureza numérica e, para os quais as soluções são conhecidas e produzem respostas satisfatórias. No entanto, como programas de IA tratam com palavras e conceitos, nem sempre uma solução correta é garantida. Algumas respostas erradas são toleradas, assim como ocorre quando um ser humano resolve um problema (Passos 1985) (Lucena 1987).

Automação Industrial é um processo, que se inicia com a substituição de esforço muscular ou animal por maquinário de algum tipo. Esse processo existe desde o início do século vinte como esforço organizado, e até antes, ainda que de forma mais experimental. Por exemplo, pode-se automatizar através do uso de multiplicadores mecânicos de força humana, sem o uso de qualquer outra fonte de energia do que o próprio braço do operário. Outra forma de automatização ocorre através de uso de ar comprimido, e uma terceira alternativa é o uso de elementos de máquina acionados por eletro-magnetismo. Todas as alternativas contêm

elementos de "inteligência" pré-programados de forma a obter respostas desejadas a partir de estímulos físico-químicos pré-definidos. Com o desenvolvimento da informática, tem sido possível substituir a dependência à confiabilidade comportamental dos materiais através de lógica desenvolvida por humanos. Em outras palavras, a "inteligência" do processo deixa de depender diretamente de estímulos físicos-químicos (Carvalho 1993).

A indústria tem um potencial muito grande de inovação com o uso integrado da Informática e da Automação, o que dá à indústria grande flexibilidade (Oliveira 1990).

Para que a automação de uma determinada tarefa seja bem sucedida, é necessário que a máquina que passará a realizá-la seja capaz de desempenhar cada uma das etapas constituintes do processo a ser automatizado com eficiência, de modo a garantir a repetibilidade do mesmo (Saliba 1992).

2.2. RETROSPECTIVA

A Informática possui sua origem que pode ser traçada aos primeiros passos da humanidade, mencionando-se, por exemplo, os ábacos; sendo que ela cresceu vertiginosamente a partir da década de sessenta, e com o advento dos microprocessadores, na década de setenta, a Automação e a Informática foram aglutinadas propulsionando o fenomenal progresso que se observa nos dias de hoje (Koo e Yoneyama 1989).

Nos anos 50, os pesquisadores já haviam estabelecido os fundamentos da IA, incluindo lógica matemática e teoria das funções recursivas, guiando a formulação de processamento de listas e da própria linguagem LISP - 1ª linguagem desenvolvida para pesquisa de IA, que foi desenvolvida no Massachusetts Institute of Technology (M.I.T.) por John McCarthy (Schildt 1989). O desenvolvimento da linguagem LISP, facilitou o surgimento de sistemas práticos de computação simbólica. Ao mesmo tempo emergiu uma interface mais amigável, os computadores interativos, tornando possíveis ambientes computacionais para desenvolvimento e depuração de programas incrementais. Aproximadamente nesta mesma ocasião, psicólogos cognitivos - estudantes da forma de pensar humana, criaram caminhos padrão do processo de investigação do raciocínio, modelando o aparente processo de tomada de decisão em termos de regras de produção condicionais (Schwabe e Carvalho 1987).

Os anos 60 foram um período de grande otimismo quanto à possibilidade de fazer um computador pensar (Schildt 1989) e, essa década caracteriza-se pela ênfase na heurística (Chorafas 1988). Os pesquisadores de IA tentaram simular o complexo processo de pensamento, procurando métodos gerais para resolver uma ampla classe de problemas. No entanto, apesar de progressos interessantes, as dificuldades eram enormes e não frutificaram.

Durante a década de 70 concentraram os esforços em técnicas, como representação, isto é, modo de formular os problemas de maneira a tornar a solução mais fácil, como controlá-la inteligentemente dentro da capacidade de memória do computador. Esta estratégia produziu algum sucesso, mas ainda não foi decisiva. Somente no final da década, fizeram a descoberta mais importante: o poder de resolver problemas do programa

depende mais do conhecimento que possui, do que o formalismo ou esquema de inferência empregado. Esta realização levou ao desenvolvimento de programas de computador de propósito particular, sistemas que são peritos em alguma área limitada. Estes programas são chamados de Sistemas Especialistas, e uma nova área dentro da informática se iniciou.

No campo de IA aplicada, após uma década de trabalho, emergiram 3 subcampos principais: Sistemas Especialistas, linguagem natural e robótica; incluindo visão, fala e locomoção.

Os SEs começaram a surgir comercialmente entre 1980 e 1981. Embora SEs e peritos reais possam em alguns casos desempenhar tarefas idênticas, as características de ambos são criticamente diversas. Mesmo havendo algumas vantagens evidentes dos SEs, eles não poderão substituir os peritos em todas as situações, devido a algumas limitações inerentes.

Antunes (Antunes 1993) em um de seus artigos, alega que IA até meados de 1983 nos EUA, e até 1988 no Brasil era uma área quase que exclusivamente de pesquisa. Comparados a outros ramos da Ciência da Computação, os resultados práticos alcançados eram escassos, e muitos deles, quando efetivamente se mostravam práticos, acabavam deixando de ser objeto de pesquisa da área. Na segunda metade da década de 80, este panorama sofreu uma alteração radical. O sucesso obtido por empresas como American Express e Digital (DEC) no emprego de SEs em aplicações práticas (e lucrativas) mostrou que técnicas de IA, poderiam realmente trazer retorno prático. A tendência também se refletiu no Brasil, onde grandes empresas passaram a analisar o assunto. Hoje, principalmente em instituições financeiras, nosso país tem diversos exemplares de SEs de uso prático. Tais sistemas são a

parte mais visível dessa evolução, mas se inserem em contexto maior: o processamento do conhecimento. Uma das grandes tendências da IA nas últimas décadas tem sido explorar modos de capturar a utilização do conhecimento pelo computador, como forma de fazer programas mais "inteligentes", tanto explicitamente, representando-se esse conhecimento através de símbolos, como incorporando-o em estruturas que mimetizam os neurônios do cérebro humano, os chamados modelos conexistas ou redes neurais.

Embora algumas empresas privadas tenham começado com sua própria pesquisa no setor, a maior parte do trabalho inicial foi feita em grandes universidades como Stanford, Massachusetts Institute of Technology e Carnegie-Mellon University (Hilster 1987a).

No Brasil, a primeira fase, formação de massa crítica, começou no Instituto Militar de Engenharia (IME) em abril de 1984, com a realização do 1º Encontro de Pesquisadores em Inteligência Artificial. Em julho deste mesmo ano, foram oferecidos pela SBC (Sociedade Brasileira de Computação) cursos introdutórios para mais de duzentos estudantes de graduação em computação. A SBC também patrocinou em Porto Alegre, em outubro de 1984, o 1º Congresso de Inteligência Artificial nas universidades.

As primeiras empresas a desenvolverem softwares em IA foram: Biodata, uma software-house carioca que criou uma subsidiária especializada para esta tarefa; a Embratel e o SERPRO.

Hoje, as aplicações das técnicas de IA são consideradas fundamentais devido a:

- . Aplicação de Sistemas Baseados em Conhecimento, por exemplo, os SE em diversas áreas;
- . Desenvolvimento de hardware específicos para aplicações de IA, tais como: arquitetura paralela e redes neurais;
- . Desenvolvimento da 5ª geração de processadores no Japão;
- . Desenvolvimento do projeto ESPRIT (European Strategic Program for Research and Development in Information Technology);
- . Aplicações em robótica,
- . Desenvolvimento de linguagens naturais;
- . Pesquisas em aprendizado de máquina (auto-aprendizado).

2.3. INTELIGÊNCIA ARTIFICIAL X COMPUTAÇÃO CONVENCIONAL

Uma comparação entre a IA e a Computação Convencional, e seus principais tópicos, são apresentados a seguir (Passos 1985):

Programação em
Inteligência Artificial

- Processamento Simbólico
- Busca Heurística
(Passos da solução implícita)
- Estrutura de controle normalmente separada do domínio do conhecimento
- Fácil de modificar, atualizar e ampliar
- Algumas respostas incorretas são toleradas
- Respostas satisfatórias são aceitas

Programação Convencional

- Processamento Numérico
- Busca Algorítmica
(Passos da solução explícita)
- Estrutura de controle e informação integrados
- Difícil de modificar
- Apenas respostas corretas são aceitas
- É procurada somente a melhor solução

Tabela 3 - Comparação entre IA e Computação Convencional

Uma das características centrais da programação em IA é a busca heurística e não algorítmica, e as linguagens tradicionais como BASIC ou PASCAL, vêem o programa como uma receita a ser

seguida cegamente. Já as linguagens utilizadas para programação em IA vêem o programa como sendo a descrição das condições sob as quais a solução do problema deve ser procurada. E ela encontra a referida solução sem auxílio do programador, confiando apenas na lógica. Em suma, linguagens convencionais só fazem aquilo que o programador manda, enquanto que programação que utiliza os princípios de IA faz o que as leis da lógica determinam (Costa e Gurgel 1985).

Os programas de IA possuem estruturas de controle separadas do domínio de conhecimento, e esse conhecimento estará disponível para ser usado quando necessário na solução de um problema, podendo dessa forma, executar somente a troca de base de dados quando necessário. Em programas convencionais, as trocas de bases de dados geralmente acarretam grande impacto para o programa, sendo normalmente imprescindível grandes modificações em sua estrutura básica, ou até mesmo necessidade de construção de um programa diferente para acessar essa nova base de dados.

2.4. PRINCIPAIS TÓPICOS DA IA

Pode-se, através de um diagrama (Figura 1) visualizar algumas das aplicações de IA.

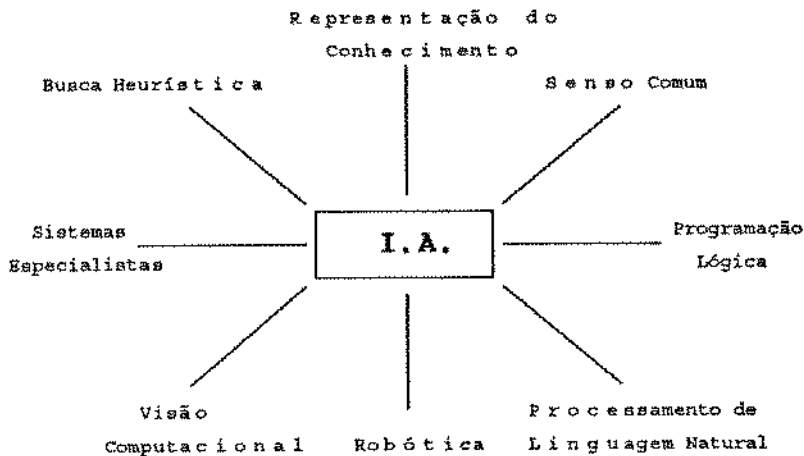


Figura 1 - Principais Aplicações de IA

Esses tópicos foram muito bem definidos por Nilsson (Nilsson 1980):

2.4.1. Busca Heurística

É um método de resolução de problemas em que se usa a tentativa e erro, executada em estrutura de árvore, começando no topo com uma condição inicial e bifurcando cada vez que se toma uma decisão. A busca heurística é uma estratégia para se tentar achar mais rapidamente a solução de um determinado problema.

2.4.2. Representação do Conhecimento

Procedimentos inteligentes não são baseados somente em métodos de raciocínio, mas principalmente no conhecimento

armazenado (devido ao vasto conhecimento que um ser humano adquire durante a vida, tornando-o um especialista em determinada área). Para que a exploração do conhecimento possa ser feita, é vital que este seja representado de forma idônea. O formalismo lógico é requisitado, pois sugere uma forma poderosa de se derivar novos conhecimentos a partir dos velhos.

2.4.3. Senso Comum e Lógica

O raciocínio de senso comum é uma das coisas mais difíceis para se modelar em um computador. Senso comum é raciocínio de baixo nível, pois está diretamente ligado a aspectos subjetivos como vivência e senso prático. É necessário repassar à máquina regras, para que ela possa executar a programação com entendimento satisfatório. A representação do senso comum, em sistemas de programação, é considerada um tópico chave em IA, com poucas chances de solução satisfatória em pouco tempo. Uma outra área muito importante em IA é a Programação Lógica, através da qual pode-se realizar deduções através de fatos e regras.

2.4.4. Processamento de Linguagem Natural

Esta área se preocupa em desenvolver programas que entendam a linguagem falada e escrita, visando a interação com máquinas por meio desta mesma linguagem. O processo empregado para se fazer um computador entender sentenças é composto de programas que juntos formam o "analisador de linguagem natural". As três divisões básicas da análise da linguagem são (Levine et al. 1988):

- . Análise léxica: análise das palavras;
- . Análise sintática: análise da colocação das palavras em uma sentença, incluindo as regras de gramática;
- . Análise semântica: análise do significado de uma sentença em si e em relação a outras sentenças.

O Processamento de Linguagem Natural tem o objetivo de tornar possível a comunicação entre a máquina e usuários não especializados. Esse processamento varia de máquina para máquina, restringindo assim seu uso para pessoas especializadas na maioria das vezes.

2.4.5. Visão Computacional

A visão processada por computadores possibilita-lhe ver, identificar e entender o que estiver "procurando" (Caulliraux e Alvarenga 1989). Esse processo pode ser subdividido em várias partes: sensoreamento, pré-processamento, segmentação, descrição, reconhecimento e interpretação.

- . Sensoreamento: o processo de aquisição de imagem;
- . Pré-processamento: serve para a redução de ruído e realce de detalhes;
- . Segmentação: é o processo de partir a imagem em objetos de interesse;

- . Descrição: serve para identificar características que são usadas para diferenciar um objeto de outro;
- . Reconhecimento: processo de identificação de um objeto específico;
- . Interpretação: atribui um significado especial a um conjunto de objetos que foram reconhecidos.

Cada uma destas partes tem uma complexidade específica, e para distinguir níveis de complexidade devemos dividi-las em três:

- . baixa: sensoreamento e pré-processamento;
- . média: segmentação, descrição e reconhecimento;
- . alta: interpretação.

No sensoreamento encontramos problemas de resolução da imagem, técnicas de iluminação, movimentos de câmera e visão digital. Detecção de contornos, de bordas e limiar, descritores geométricos, de Fourier, estatísticos, funções discriminantes estatísticas e manipulação simbólica são problemas encontrados em visão de nível médio e superior.

2.4.6. Sistemas Especialistas

São sistemas projetados para emitir uma decisão ou parecer sobre uma determinada área do conhecimento humano.

2.4.7. Robótica

Controle de robôs (não relacionado com a construção do robô), explora aspectos que envolvem flexibilidade e adaptabilidade ao ambiente, e capacidade de planejar; portanto, os robôs tornam-se mais interativos com o ambiente.

Os robôs podem ser classificados em função de seu grau de inteligência. Uma máquina é considerada inteligente se possui qualquer uma das seguintes habilidades:

- . capacidade de raciocinar e de realizar inferências;
- . capacidade de resolver problemas;
- . capacidade de acumular e de usar conhecimentos;
- . capacidade de planejar as próprias ações e de prever o resultado delas;
- . capacidade de aprender, com a experiência, livros ou professores;
- . capacidade de enxergar, ouvir e interpretar corretamente estímulos sensoriais.

2.5. LINGUAGENS E FERRAMENTAS DA IA

As linguagens mais usadas neste campo são as linguagens orientadas a manipulação simbólicas, o LISP (LIST Processing Language), que foi criada em 1957 pela comunidade científica americana, e o PROLOG (PROgramming in LOGic), que foi criada em 1972 pela comunidade científica européia, e amplamente utilizada pelos japoneses .

Na batalha entre as linguagens de IA (Hilster 1987b), LISP tem a vantagem de ser a primeira. Criada em 1957 por John McCarthy, ela cresceu e se desenvolveu com a IA desde seus primórdios. Sua estrutura simples, baseada em listas de Lambda Calculus, forneceu a liberdade e flexibilidade necessárias para criar muitas das noções primeiras sobre a inteligência humana. Embora essa linguagem tivesse muitas características desejáveis para os pesquisadores, o seu domínio a princípio aconteceu, em parte devido à falta de competição, e como resultado, tornou-se o padrão de programação em IA nos 12 anos seguintes. A mesma liberdade que deu à LISP sua flexibilidade, permitiu que os programas nela baseados se tornassem rapidamente grandes, complexos, e por isso difíceis de controlar. O conceito de algoritmos heurísticos ou "domain specific" para controlar os programas em LISP se tornou um problema para os pesquisadores em IA que tiveram de explicar como algoritmos tão específicos faziam parte da inteligência humana geral. Muitos cientistas europeus que tinham seus estudos profundamente ligados à lógica clássica não se sentiram muito confortáveis com a variedade de algoritmos diferentes que seus colegas americanos estavam produzindo e desenvolveram uma linguagem de programação própria; o PROLOG.

A linguagem PROLOG foi desenvolvida mais de uma década após a LISP. Seu inventor, Alain Colmerauer, queria desenvolver uma linguagem de computador que permitiria aos programadores especificar tarefas em lógica ao invés de usar as funções tradicionais de outras linguagens (incluindo LISP).

A maioria das ferramentas que utilizam técnicas para expressar conhecimentos, formular SEs e ajudar na programação básica, foram desenvolvidas em PROLOG.

As ferramentas mais conhecidas para construção de SEs são: OPS5, ACLS, Expert-Ease, Rulemaster, T-1, EMYCIN, XCON.

2.6. IA NO CONTEXTO DA MANUFATURA

2.6.1. Generalidades

Desenvolvimentos em IA tem tido um grande impacto em sistemas de manufatura e todas as áreas do CIM (Computer Integrated Manufacturing) tem sido afetadas pela IA (Kusiak 1990).

O sistema CIM é conceituado (Eversheim et al. 1986) como um sistema produtivo gerenciado por um sistema computacional em todas as fases, desde a concepção à fabricação do produto, vendas, compras, clientes e administração.

A IA representa uma ferramenta poderosa que muito já contribuiu e contribuirá para a realização do CIM. A razão desta afirmação ser verdadeira, é que as técnicas de IA podem ser aplicadas em todas as áreas de manufatura. Essa técnica assegura um padrão de compartilhamento de informação ou conhecimento entre os vários meios de um ambiente CIM. A maior atividade em IA aplicada à manufatura são os SEs, que podem ser usados em quase que todos os ciclos de manufatura, incluindo desenho, projeto, planejamento da produção, gerenciamento de produção e sequenciamento. Um SE tem também, a capacidade de executar diagnósticos de máquinas, processos, e podem ainda ser aplicados para monitoramento e controle de processos (Schaffer 1986). Para isto há integração das seguintes áreas através de computadores:

- . Projeto de peças e produtos;
- . Projeto de ferramentas e fixações;
- . Planejamento do processo;
- . Máquinas de Controle Numérico (CN);
- . Planejamento da produção;
- . Usinagem;
- . Montagem;
- . Manutenção;
- . Controle de qualidade;

- . Inspeção;
- . Sistemas de manuseio;
- . Sistemas de transporte e armazenamento automatizados.

Ainda mais importante que o CIM, tem-se as bases de conhecimento especialistas que tornaram-se entidades independentes e portáteis que podem ser acessadas por qualquer outro tipo de SE. Ao contrário dos Sistemas de Banco de Dados (SBD) convencionais, nos quais as informações (procedurais) sobre os dados estão contidas em um particular programa, o conhecimento de um SE será acessado, atualizado e movimentado com mais facilidade entre os SEs que servem para diferentes funções. Desse modo ocorre o compartilhamento de informação, que é essencial para o CIM. A aplicação de SE na área de fabricação leva à tomada de decisão rápida no gerenciamento de uma indústria.

O sucesso da aplicação da IA em fábricas está baseada na substituição da base de conhecimento humana especializada da melhor forma possível. Os SEs são os melhores aliados para essa substituição e são incorporados em grande quantidade de sistemas, incluindo máquinas individuais e células (Bellinger 1988).

Um FMS (Flexible Manufacturing Systems) (Shaw e Whinston 1989) é construído como um Sistema Baseado em Conhecimento, o qual consiste de três componentes básicos: uma Base de Dados, uma Base de Conhecimento e uma Máquina de Inferência, sendo que a base de dados armazena conhecimentos declarativos encontrados em um universo generalizado e a base de conhecimento armazena o domínio específico através de conhecimentos procedurais. Este conhecimento modela as ações e é frequentemente representado por

regras de produção e operadores lógicos. A Máquina de Inferência armazena o controle do conhecimento sobre a estratégia de construção do projeto, indicando como selecionar operadores e como aplicá-los.

O termo FMS (Batocchio 1991), é empregado para designar a fabricação em pequenos/médios lotes de fabricação através de uma ou mais máquinas interligadas em fluxo de material e informação, podendo automática e simultaneamente fabricar diferentes peças com diferentes sequências de operações em um sistema.

2.6.2. Áreas de Manufatura que Utilizam-se de IA

IA é um dos ingredientes-chave em interação de dados em automação. Existem processos que se não forem totalmente automatizados podem ser melhorados através da construção de SEs, esses processos podem ser: planejamento, métodos de processo, padronizações, estimativa de custo e sequenciamento (Liu 1988).

Schaffer (Schaffer 1986) cita as seguintes áreas dentro da manufatura como prováveis usuárias de IA:

- Engenharia de projeto:

- . para projetar usando modelamento de sólidos interativos com identificação de características de formas e definições de peças;
- . para projeto e simulação, usando análise de elementos finitos;

- . documentação do projeto e análise de configuração
- Planejamento da manufatura:
 - . planejamento de processo generativo usando SE, com as seguintes características básicas:
 - .. usar como entrada as características das peças;
 - .. características semelhantes para processos de manufatura;
 - .. operações de manufatura semelhantes para sequência de operação da peça;
 - .. gerar o plano total do processo;
- Planejamento da qualidade:
 - . CAI - Computer Aided Inspection
- Facilidades quanto ao planejamento:
 - . Layout da planta otimizada
- Controle da produção:
 - . Sequenciamento
 - . MRP - Planejamento de Necessidade do Material
 - . Controle do chão de fábrica

- Automação de Fábrica:

- . Automação;
- . Comando Numérico;
- . Robótica.

O mesmo autor cita algumas aplicações genéricas de IA voltadas para a área de manufatura:

- Interpretação e integração de sensores:

- . Desenvolvimento significativo de sensores de dados;
- . Fusão de sensores (integração múltipla de sensores de entrada para desenvolver interpretações de alto nível);

- Percepção e orientação visual:

- . Inspeção;
- . Identificação;
- . Verificação;
- . Direção;
- . Obstáculos;
- . Monitoramento;

- Operação de máquinas e sistemas complexos;
- Gerenciamento:
 - . Planejamento;
 - . Sequenciamento;
 - . Monitoramento;
 - . CAO - Controle Adaptativo Otimizado;
- Gerenciamento do Conhecimento:
 - . Acesso à base de dados inteligentes;
 - . Aquisição de conhecimento;
 - . Entendimento de texto;
 - . Geração de texto;
 - . Máquina de tradução;
 - . Explanação (explicação);
 - . Operações lógicas em base de dados;

- Interação com o ser humano:
 - . Entendimento de fala;
 - . Geração de fala;
- Diagnósticos de falhas e reparos:
 - . Humanas;
 - . Máquinas;
 - . Sistemas.

CAPÍTULO 3

SISTEMAS ESPECIALISTAS (SE)

3.1. CONTEXTO GERAL

A IA e o seu segmento de aplicação denominado "Sistema Especialista", constituem-se em instrumentos adequados para o tratamento de problemas que são decorrentes da experiência e das preferências pessoais do especialista (Teixeira Jr et al. 1989).

SEs são programas que imitam o comportamento de especialistas humanos. Os SEs usam informações que o usuário fornece para emitir uma opinião sobre determinado assunto (Schildt 1989).

A tarefa de construir um SE se resume a fazer com que os computadores resolvam problemas em que é necessário a presença de um especialista em determinado assunto (Campos 1988).

SE (Passos 1987) é um programa de computador destinado a solucionar problemas em um campo específico do conhecimento, que tem para isso uma base de conhecimento desse domínio restrito. Usa um "raciocínio inferencial" para executar tarefas, e tem desempenho comparável aos especialistas humanos. Na prática, uma das mais importantes características de um SE é a capacidade de explanação. Eles surgiram porque as estratégias de resolução de

propósitos gerais da IA eram muito ineficientes na solução de problemas complexos. Na década de 70, ficou claro que para os pesquisadores de IA conseguirem que seus sistemas resolvessem satisfatoriamente problemas reais, era necessário incorporar-lhes grandes quantidades de conhecimentos sobre o problema. Este fato demandou criação do campo da "Engenharia do Conhecimento" que procura formas de usar conhecimentos de especialistas na solução de problemas complexos.

A verdadeira definição de SE (Barborak et al. 1991) vive constantemente em debate, mas pode ser genericamente descrito como um sistema projetado para simular o conhecimento e raciocínio de um especialista humano, e faz com que o conhecimento fique disponível para outras pessoas de forma útil. Os termos Sistemas de Conhecimento e Sistemas Baseados em Conhecimento são frequentemente usados com sinônimos de SEs.

Os benefícios que um SE podem trazer são basicamente (Smart 1987):

- . Sintetizar e conservar conhecimento coletivo;
- . Treinamento;
- . Determinação de especificações funcionais;
- . Assegurar consistência e perfeição em decisões repetitivas;
- . Execução de tarefas por leigos;
- . Realizar as tomadas de decisão em profundidade;

. Implementação de sistemas fáceis de atualização.

Os SEs utilizam-se de fatos, informações específicas sobre a área de interesse, e heurísticas, que são representadas através de regras práticas que o especialista humano utiliza para chegar a conclusões relevantes em sua área de atuação.

3.2. COMPOSIÇÃO DOS SEs

Os SEs são compostos por três partes fundamentais: Base de Conhecimento, Base de Dados e Máquina de Inferência.

3.2.1. Base de Conhecimento

A base de conhecimento ("*Knowledge base*") é que contém toda a sabedoria de um especialista, isto é, a base de conhecimento é formada pelas regras que descrevem o conhecimento de um especialista. Essa base consiste de descrições, relacionamentos e procedimentos pertinentes a um particular domínio. As descrições identificam e diferenciam objetos e classes, e geralmente incluem regras com respeito às aplicações das descrições. Os relacionamentos descrevem as dependências e associações entre a informação na base de conhecimento. Os procedimentos especificam o raciocínio das operações que devem ser executadas.

Representação do conhecimento (Schwabe e Carvalho 1987) é a estrutura formalizada e o conjunto de operações que envolve a

descrição, relacionamento e procedimento de um determinado domínio. O conhecimento representado em um programa particular chama-se base de conhecimento.

Para o devido armazenamento do conhecimento (Krause 1991) existem diferentes representações, entre elas, regras de produção, frames, redes semânticas e expressões lógicas.

. Regras de produção são mais adequadas para representação de conhecimento dedutivo, como situação/ação, premissa/conclusão, antecedente/consequente e causa/efeito.

As regras de produção são regras do tipo "if then" compostas de condições e ações:

"if" a (s) seguinte (s) condição (ões) existe (m)
"then" execute a (s) seguinte (s) ação (ões).

Em lógica:

- . a porção "if" da declaração é chamada de antecedente, que exprime qual condição está sendo provada;
- . a porção "then" é chamada de conclusão.

Em PROLOG:

- . a porção "if" é chamada de corpo ("body");
- . a porção "then" é chamada de cabeça ("head").

Exemplo:

```
if
    127 <= Diametro <= 150,
    30 <= Comprimento <= 500,
    Quantidade >= 300
then
    write('utilizar torno copiador').
```

. **Frames** são os mais indicados para representar conhecimento descritivo e relacional que reúne ou se ajusta a algum protótipo, tal como a descrição de um ambiente hostil ou processo de contabilização. É uma estrutura de dados que contém conhecimento sobre um conceito. Ligados a cada frame existem vários tipos de informação, sendo que algumas delas são sobre o uso dos frames.

As **redes semânticas** tem capacidade de representar o conhecimento de forma genérica e flexível, com enfoque para as ligações (relações) entre os conceitos que não seguem uma organização hierárquica (Campos 1989). Sua estrutura é composta por nós conectados por arcos que representam o relacionamento entre esses nós. Os nós podem representar objetos, conceitos ou situações.

As **representações orientadas para lógica de predicados** descrevem o conhecimento através de expressões lógicas, que permitem deduzir alguma coisa a partir de um conjunto de fatos (premissas) e axiomas (regras gerais de dedução). É utilizada pela linguagem PROLOG.

De todas estas formas, a primeira (regras de produção) é a mais utilizada em aplicações comerciais atualmente.

Quanto maior for a quantidade de conhecimento contida na base de conhecimento, melhor será o desempenho do SE.

3.2.2. Base de Dados

A base de dados ("*data base*") contém os fatos e informações relativas ao domínio que está sendo explorado. É uma coleção simples de símbolos usados para refletir um fato, um estado ou uma hipótese, sendo que a interpretação desses símbolos dependem basicamente da natureza da aplicação.

A organização da base de dados pode ser tanto linear como hierárquica e seu tamanho é arbitrário, ela não exige construções complexas. Podem ser feitas atualizações (alterações, inclusões ou exclusões) em seu conteúdo quando necessário.

3.2.3. Máquina de Inferência

A Máquina de Inferência ("*inference system*") é o sistema de controle que dirige a implementação do conhecimento. Ela decide quais técnicas de busca heurística são usadas para determinar como as regras na base de conhecimento são aplicadas para o problema. A rotina de busca analisa através de soluções alternativas para proceder de um ponto de partida, ou estado inicial, para um lema (objetivo) ou vice-versa.

Uma Máquina de Inferência é um protocolo de programa para navegar através das regras e dados da representação do conhecimento para resolver o problema, ela concretiza a dedução. Sua tarefa é selecionar, e então aplicar a regra mais apropriada em cada passo da execução do SE, o que contrasta com as técnicas de programação convencional, onde o programador seleciona a ordem na qual o programa deverá executar os passos, ainda em tempo de programação. Há sistemas que operam a inferência por encaminhamento para a frente, outros utilizam caminhamento retrospectivo e ainda outros empregam ambas as direções. Em muitos casos a escolha de qual direção o sistema deverá executar é feito reproduzindo a maneira utilizada por uma pessoa para resolver o problema.

As Máquinas de Inferência possuem três funções básicas:

- . **Identificação das regras de aplicação:** todas as regras devem estar "alertas" para ver se alguma delas pode ser aplicada para o problema corrente e manuseá-la.

- . **Resolução de conflitos:** a Máquina de Inferência deve decidir qual regra deve ser executada quando estiver ocorrendo um conflito.

- . **Execução da regra:** a regra selecionada é avaliada e sua ação executada.

Estas três situações são continuamente aplicadas pelas regras disponíveis até que não haja mais regras "candidatas" para serem executadas. Basicamente as Máquinas de Inferência imitam os tipos de pensamento que empregamos quando tentamos resolver algum problema.

"Rule Chaining", refere-se como a Máquina de Inferência determina quais regras serão aplicáveis (não é uma resolução de conflito) baseando-se em quais regras já foram aplicadas. O termo "chaining" refere-se a juntar as regras como engenho de busca para inferir uma solução.

As técnicas de busca para construir uma Máquina de Inferência aqui citadas, são "forward chaining" e "backward chaining".

"Forward Chaining" (ou encadeamento para a frente ou encadeamento direto) é uma técnica de busca onde a Máquina de Inferência parte de um estado inicial (informações que o usuário fornece) e trabalha em direção ao lema (objetivo), utilizando-se para isso de mudanças de estado, isto é, movimentos por meio de uma rede de e/ou lógicos. Suas características básicas são:

- . Tentar descobrir novos fatos baseados nos fatos conhecidos;
- . Receber todas as informações (dados) antes do início do processamento;
- . Ser dirigida por esses dados;
- . Apresentar pouca interação com o usuário.

"Backward chaining" (ou encadeamento reverso) é um procedimento de inferência que trabalha a partir do lema ou conclusão para ver se as condições que o tornariam verdadeiras são satisfatórias, isto, é procura provar a meta pela prova de

submetas (estratégia usada pelo próprio PROLOG). Suas características básicas são:

- . Tentar descobrir um objetivo específico;
- . Interagir com o usuário durante o processamento;
- . Ser dirigida por objetivos.

3.3. ALGUMAS ÁREAS DE APLICAÇÃO DE SES

3.3.1. Áreas Gerais

- . Diagnoses médicas;
- . Reparo de equipamentos;
- . Configuração de computadores;
- . Tomada de decisões financeiras;
- . Interpretação de sinais;
- . Exploração de minerais;
- . Interpretação de dados químicos e elucidação de estruturas;
- . Compreensão da voz e da imagem;

- . Estratégias militares;
- . Recomendação sobre o uso de sistemas de computadores;
- . Engenharia de software.

3.3.2. Em Manufatura (Smart 1987)

(Schaffer 1986)

- . Engenharia de Projeto;
- . Planejamento do Processo;
- . Ordem de Entrada;
- . Configuração do Produto;
- . Compartilhamento de Chão de Fábrica;
- . Monitoramento;
- . Controle de Qualidade;
- . Análise e Produtividade de Fábrica;
- . Interpretação;
- . Diagnósticos;
- . Desenho, Predição, Depuração, Reparos;

. Sequenciamento;

. Aprendizado.

Schaffer diz que as principais áreas de aplicação de SE em manufatura são planejamento de processos, diagnóstico e sequenciamento; sendo que um dos fatores que mais influenciam a utilização de técnicas de IA em manufatura é o rápido aumento de complexidade, isto é, a complexidade da área aumenta mais rapidamente que o desenvolvimento e treinamento de experientes engenheiros mecânicos e projetistas.

Sistemas CAPP (Computer Aided Process Planning) tem conseguido capturar com grande facilidade o gerenciamento do conhecimento de planejamento de processos, e como exemplo disto, pode-se citar o SE desenvolvido pela Avionics Division of Honeywell Corporation, que trabalha com planejamento de processo generativo (Marks 1988).

A utilização de SE em manufatura (Pham e Oztemel 1992) podem reduzir custos de treinamento, erros de produção, taxa de refugo e a necessidade de peritos humanos.

Com relação à manufatura auxiliada por computador, o processamento de conhecimento é pré-requisito necessário para maiores aperfeiçoamentos. Os sistemas baseados em conhecimento podem cooperar com base em domínios de conhecimentos múltiplos e experiências (Krause 1991).

3.3.3. Alguns dos Principais SE Construídos

3.3.3.1. Áreas Gerais

- . Em geologia: PROSPECTOR, desenvolvido pelo SRI, tem como função a análise e interpretação de dados. Reúne conhecimentos de 20 geólogos, detecta minerais subterrâneos através de características da superfície (Fez uma descoberta de uma mina no valor de US\$ 100 milhões). Foi o *primeiro SE comercial*.
- . Em matemática: Sistemas para ensino de álgebra, geometria e programação, e sistemas adequados à manipulação simbólica de fórmulas matemáticas: SAINT, SIN, MATHLAB, MACSYMA.
- . Em química: DENDRAL, simula reações químicas e biológicas impossíveis de serem feitas em laboratórios. Foi o *primeiro sistema a usar IA* desenvolvido por Stanford University. SEQ, auxilia um biólogo molecular na realização de análises de sequências moleculares. SPEX, planeja experimentos de laboratório.
- . Em medicina: MYCIN, desenvolvido pela Stanford University, diagnostica doenças infecciosas e receita medicamentos, foi o primeiro projeto voltado para a área médica, foi o *primeiro a usar fator de certeza*. EMYCIN, tem a mesma função do MYCIN, e foi o *primeiro SE a separar conhecimento das estratégias de conhecimento*. ONCOCIN, recomenda dosagens de radioterapia e analisa efeitos tóxicos

acumulativos. PUFF, faz diagnose de doenças pulmonares, foi o primeiro SE a utilizar uma ferramenta (SHELL), também desenvolvido pela Stanford University. ANNA, monitora e administra tratamento a pacientes cardíacos.

- . Em meteorologia: sistema para previsão do tempo, da empresa Teknowledge.
- . Em aviação: Gate-assignment and Display System é da United Airlines e evita congestionamento de avião quando há atrasos por causa de mau tempo. O sistema também sabe, que portões anexos não acomodam dois DC-10 de uma só vez por questões de espaço.
- . Nas forças armadas: Pilots Associated, avisa pilotos em combate de tudo o que estiver acontecendo, desde o tempo até ataques em ar ou solo. AirLand Battle Management, é específico para traçar estratégias de combate aéreo ou em terra, e serve para simular planos táticos, tendo sob controle centenas de aviões e soldados. ASTA, identifica imagens de radar pela análise dos sinais recebidos. I&W, processa informação sobre os eventos mundiais e prevê a ocorrência de conflitos armados.
- . Para exploração espacial: Existem sistemas para monitorar trajetórias de naves espaciais, manter a vida humana no espaço e coordenar trabalhos de terra durante uma missão.

- . Em **computação**: **DART**, faz diagnósticos de falhas de hardware, desenvolvido por Stanford/IBM. **XCON**, configura sistemas de processamento de dados. **YES/MVS**, monitora e controla sistemas operacionais. **PTRANS**, gerencia a fabricação e distribuição de sistemas de computadores.
- . Em **telefonía**: **ACE**, identifica problemas em redes telefônicas e recomenda uma manutenção adequada.
- . Para **jornalismo**: Da Composition Systems Inc., de Nova York, um sistema para diagramação reúne dez programas especialistas, que trabalham em conjunto.
- . Para **finanças**: **PlanPower**, da Apex, analisa situações financeiras de um cliente, empresa ou produto e faz projeções.
- . Em **engenharia elétrica**: **REDES/GN**, auxilia no reprojeto de circuitos para atender novas especificações.
- . Para **controle de qualidade**: A IBM usa desde 1986 um sistema para teste de qualidade de winchesters.
- . Para **assistência técnica**: Os revendedores Ford podem acessar via terminal, um SE para ajudar a diagnosticar problemas mecânicos em automóveis.
- . Para **vendas**: Da Digital e Carnegie-Mellon University, um sistema que ajuda vendedores a escolher computadores e configurações adequadas para cada cliente.

- . Para construção de Sistemas Especialistas: ROSIE (Rand), RITA, NEXPERT, EXPERT (Rutgers), EDEN, CHARME, AGE (Stanford), Hearsay III (USC/ISI), EMYCIN (Stanford), OPS 5 (Carnegie-Mellon University).
- . Para geradores de bases de conhecimento: KOOL, SAFO, que foi desenvolvido pela PUC/FINEP e editorado pela Intellect.

3.3.3.2. Específicos para Manufatura

- . XPS-E, desenvolvido por membros de uma organização chamada Computer Aided Manufacturing - International (CAM-I), ele aconselha na escolha de máquinas, tipo de processos e agrupamentos de elementos.
- . DCLASS, desenvolvido por Brigham Young University, ele captura o conhecimento através de árvores de decisão cobrindo especificamente a área de usinagem.
- . ISIS, desenvolvido pela Carnegie-Mellon University juntamente com Westinghouse Electric Corporation para produção do sequenciamento.
- . PICON (Process Intelligent Control System), o qual utiliza uma máquina LISP acoplado ao sistema de um processo de produção, ele pode aceitar cerca de 20.000 pontos de medida juntamente com os controles associados.

- . GARI, seu domínio é restrito à indústria de corte de metal.
- . SIPP, para planejamento de processo generativo de peças usinadas. Sua base de conhecimento consiste de superfícies usinadas, capacidade de várias operações de máquinas e uma estrutura que manipula o conhecimento.
- . TOM, implementado pela University of Tokyo, preocupa-se com detalhamento de planos de usinagem.
- . EXCAP, para planejamento de processo generativo para usinagem de peças rotacionais.
- . CUTTECH, desenvolvido por Metcut Associates (Cincinnati), que trabalha com operações de planejamento que especificam os parâmetros detalhados de uma operação individual dentro de um plano de processos de peças. Um plano de processos típico, refere-se à coleção de todas atividades de planejamento requeridas para converter um simples desenho de peça de um produto manufaturado, e planejamento da operação é frequentemente considerado um subconjunto das atividades globais do planejamento do processo.
- . PECOS, desenvolvido em Yale para programação automática.
- . ISIS-II e KBRS , sequenciamento "job-shop", sendo que o primeiro representa o conhecimento através de regras e o segundo através de regras/frames.

- . DELTA, para diagnóstico e reparo em locomotivas construído pela General Eletric.
- . PROPLAN, para tomadas de decisão tais como: seleção de processo, seleção de máquina, seleção de ferramentas e acessórios, seleção de parâmetros de máquina.
- . GEMOS (General Eletric Machining Optimization System), SE para auxiliar na seleção de ferramentas de corte, parâmetros de máquina, caminho lógico de corte.
- . NEWCS, FILLER METAL SELECTION, são sistemas desenvolvidos para controle de soldagem.
- . ISA, é um sistema da área de fabricação, para planejar alocação de materiais contra o pedido de clientes.
- . STREAMER, atua no treinamento de operadores de caldeira.

3.4. PRINCIPAIS TIPOS DE SES (Passos 1989)

3.4.1. Interpretação

Interpretam e explicam certas situações através de observações de fatos, isto é, fazem a análise de dados e procuram determinar as relações e seus significados.

Exemplo: DENDRAL - química (Stanford), GAI - química (Stanford), PROSPECTOR - geologia (SRI).

3.4.2. Previsão

Inferir consequências prováveis de situações dadas tais como, previsões demográficas, de tempo, de tráfego, entre outras.

Basicamente, este tipo de sistema tem a capacidade de "previsão do futuro", isto a partir de uma modelagem de dados do passado e do presente. Como baseia sua solução na análise do comportamento dos dados recebidos, ele deve ter um mecanismo para criticar os vários "futuros" possíveis através de raciocínios hipotéticos.

Exemplo: I&W.

3.4.3. Diagnose

Inferir sobre o mal funcionamento de determinadas funções, irregularidades, causa e efeito.

Exemplo: INTERNIST - Medicina (University of Pittsburgh), MYCIN, PUFF - medicina (Stanford), DART - falhas computador (Stanford/IBM), PIP - medicina (M.I.T.), CASNET - medicina (Rutgers).

3.4.4. Planejamento

Programa de ações a serem tomadas para se chegar a um objetivo. São estabelecidas etapas e sub-etapas com suas prioridades definidas.

Exemplo: NOAH, ABSTRIPS - robótica (SRI).

3.4.5. Análise de Desempenho

Executa comparações entre o desempenho atual com o desempenho desejado.

Exemplos: PTRANS, YES/MVS.

3.4.6. Treinamento

Tem a habilidade de explicar sua linha de raciocínio ao usuário.

Exemplos: STREAMER, Sistemas de Treinamento para Estudantes de Medicina

Muitas áreas de engenharia podem conseguir consideráveis benefícios a partir do desenvolvimento e utilização de SEs e se aplicam em:

- . **Interpretação de dados:** Dados, como os vários parâmetros de produção, podem ser usados para apontar gargalos ou áreas problemáticas nos setores produtivos.

- . **Diagnóstico:** Podem ser desenvolvidos sistemas que auxiliem na análise de defeitos em peças, defeitos em máquinas, etc.
- . **Monitoramento:** É possível o desenvolvimento de programas que monitorem e sugiram medidas corretivas quando necessário. Esses sistemas devem interpretar as observações de sinais sobre o comportamento monitorado. Devem verificar de maneira contínua um determinado comportamento com limites pré-estabelecidos, sinalizando quando forem requeridas intervenções para o sucesso da execução. Têm mecanismos que permitem tratar dados errados, distorcidos ou até mesmo ausentes.
- . **Planejamento:** A determinação de fluxos operacionais ainda requer considerável experiência de chefes da área operacional e executivos. Com base em poucas regras estabelecidas por um especialista em fluxo, é possível facilitar o trabalho de planejamento em determinada área. Também utilizado para robôs e programação automática.
- . **Projeto:** O projeto de layout de uma determinada operação, tempos e métodos e outras coisas tornam-se um tanto trabalhosas para microcomputadores. SEs podem exercer um papel destacado nesses projetos.
- . **Previsão:** Programas de computador que podem ser escritos para se prever as tendências e estabelecer previsões das diversas necessidades.

3.5. CARACTERÍSTICAS DE UM SE

3.5.1. Flexibilidade

Um SE pode crescer ativamente pela adição de mais conhecimento na forma de regras. Além disso, estas regras podem ser adicionadas pelo especialista sem nenhum conhecimento de computação.

3.5.2. Adaptabilidade

Como as questões podem ser formuladas pela mudança do SE, novas regras devem ser adicionadas. Com esta habilidade para adaptar novas situações, a vida do SE pode ser estendida indefinidamente, sendo que os custos de desenvolvimento após o estágio inicial podem ser reduzidos.

3.5.3. Desempenho e Eficiência

Bom desempenho e eficiência para encontrar as soluções.

3.5.4. Explicação do Raciocínio

A fim de tornar o programa auto explicativo, no que concerne ao raciocínio empregado, são respondidas ao usuário algumas perguntas elucidativas. Este trabalho é executado por um

módulo chamado justificador. Na implementação do sistema essas perguntas podem ser feitas para responder perguntas fundamentais, tais como: "Como?"; "Por quê?"; "Por que não?".

Exemplo:

Como a máquina de inferência chegou a determinada conclusão?

Por quê é necessário fornecermos esse novo dado?

Por quê não foi usada determinada regra?

3.5.5. Interação com o Usuário

Os sistemas tem boa interação com o usuário, pois tem características que auxiliam o usuário amplamente. Alguns têm separadores de sílabas e outros fazem correção dos erros mais comuns da linguagem usada, outros, a facilidade de perguntar, isto é, questionar o usuário, a fim de chegar à meta mais rapidamente e com um grau de certeza maior.

3.5.6. Incerteza

Muitos SEs podem trabalhar com a incerteza, isto é, pode existir uma resposta com uma determinada chance percentual de ser verdadeira. Isto é, expressam implicitamente o grau de confiança que se tem no fato de uma resposta ser verdadeira. Para isso, cada variável de um SE deve possuir associada a ela um fator de certeza que pode variar de $[-1,1]$, portanto podemos relacionar o fator 0.8 a 80% ou mesmo 30%.

3.6. FASES PRINCIPAIS DE DESENVOLVIMENTO DE UM SE

O processo de construção de um SE é frequentemente chamado de "Engenharia do Conhecimento", tipicamente ela envolve uma forma especial de interação entre o construtor do SE, chamado Engenheiro do Conhecimento, e um ou mais especialistas humanos de alguma área.

O Engenheiro de Conhecimento "extraí" dos especialistas humanos seus procedimentos, estratégias e regras práticas para solução dos problemas "construindo" este conhecimento num SE.

O Engenheiro do Conhecimento pratica a arte de buscar os princípios e ferramentas que requerem o conhecimento de especialistas para suas soluções.

As questões técnicas na aquisição desse conhecimento, na representação, e na utilização adequada na construção e explicação das linhas de raciocínio são problemas importantes no projeto de Sistemas Baseados em Conhecimento.

1a. fase: envolve a identificação e conceituação do problema.

A identificação inclui a seleção e a aquisição de um especialista, fontes de conhecimento e outros recursos, e ainda a definição clara do problema.

A conceituação inclui a descoberta dos conceitos centrais e das relações necessárias à caracterização do problema.

2a. fase: lida com a formalização, implementação e teste de uma arquitetura apropriada para o sistema, incluindo constantes reformulações de conceito, reprojeto de representações e refinamento do sistema implementado.

A revisão resulta das críticas do especialista e sugestões para aperfeiçoamento do sistema.

3.7. CONHECIMENTO HUMANO X ARTIFICIAL

Conhecimento Humano

- Perecível
- Difícil de transferir
- Difícil de documentar
- Imprevisível
- Caro
- Discriminatório
- Individualizado
- Criativo
- Adaptável
- Enfoque Amplo
- Baseado em senso comum

Conhecimento Artificial

- Permanente
- Fácil de transferir
- Fácil de documentar
- Consistente
- Razoável
- Imparcial
- Social
- Sem inspiração
- Inflexível
- Enfoque restrito
- Técnico

Tabela 4: Comparação entre Conhecimento Humano e Artificial

Quando dizemos que um sistema é sem inspiração estamos dizendo que há falta de criatividade. Um perito pode reorganizar informações e usá-las para sintetizar novo conhecimento; pode

manusear eventos inesperados usando imaginação ou novas abordagens, inclusive raciocinando por analogia de um outro domínio completamente diferente. SEs trabalham sem inspiração, de forma rotineira.

Os seres humanos - peritos ou não - possuem o conhecimento advindo do senso comum, que se constitui de um largo espectro de conhecimento sobre o mundo, acumulado durante toda a sua vida e que permeia todas as suas decisões, enquanto que um SE não trabalha com senso comum, isto é, fatos que as pessoas conseguem discernir com facilidade, como por exemplo uma pessoa de 25 anos com 7 kg. Esse fato só seria observado por um SE que tivesse na sua base de conhecimento tal restrição.

Embora SEs e peritos reais possam em alguns casos desempenhar tarefas idênticas, as características de ambos são criticamente diversas. Mesmo havendo algumas vantagens evidentes dos SEs, eles não poderão substituir os peritos em todas as situações devido a algumas limitações.

SEs trabalham com a heurística, isto é, utilizam-se de experiências anteriores nas soluções de problemas análogos. Não são meramente aplicadores das regras de produção, onde estão representados os conhecimentos da área específica, eles tem que ter experiência anterior, intuição e criação. Às vezes esse conhecimento resolve um problema rapidamente, mas também pode falhar e logo outro conhecimento será tentado, esta é a técnica de tentativa e erro (heurística) usada na resolução de problemas. Isto não acontece quando se resolve um problema usando programação convencional, onde um único algoritmo é programado para resolver um determinado programa, não sendo necessário

pesquisar o espaço das soluções possíveis, pois os passos da solução do problema estão explícitos nos algoritmos.

SEs são dirigidos por dados e não por procedimentos, ou seja, usam programação lógica que enfatizam a estrutura lógica de um problema, isto é, os dados do problema geram um conjunto finito de sentenças que descrevem tal problema ou situação (programação declarativa); é portanto enfatizada a estrutura lógica de um problema e não a construção de um procedimento para resolvê-lo (programação procedural) como na programação convencional.

3.8. ARQUITETURA DE UM SISTEMA ESPECIALISTA

Com base nas definições anteriores, Schaffer (Schaffer 1986) propôs a arquitetura da Figura 2 para um SE, que é fundamentalmente composto por:

- Uma **Base de Conhecimento**, que contém a modelagem do conhecimento e, em alguns casos, heurísticas para manipular este conhecimento.

- Uma **Base de Dados**, que constitui a área de trabalho do sistema em questão, isto é, contém os dados referentes à manipulação do sistema.

- Uma **Máquina de Inferência**, que processa a Base de Conhecimento e a Base de Dados, usando uma linha de raciocínio, a fim de propor alguma solução para o problema que está sendo analisado.

- Um Subsistema de Explicação ("Explanation Facility"), que tem como responsabilidade explicar ao usuário como a Máquina de Inferência chega a certas conclusões, ou por que faz certas perguntas.

- Uma Interface de Linguagem Natural, que é responsável pela comunicação usuário/máquina. Esta interface fará as perguntas ao usuário, o qual receberá as respostas por ele fornecidas, sendo que estas respostas serão utilizadas pela Máquina de Inferência para inferir novas asserções.

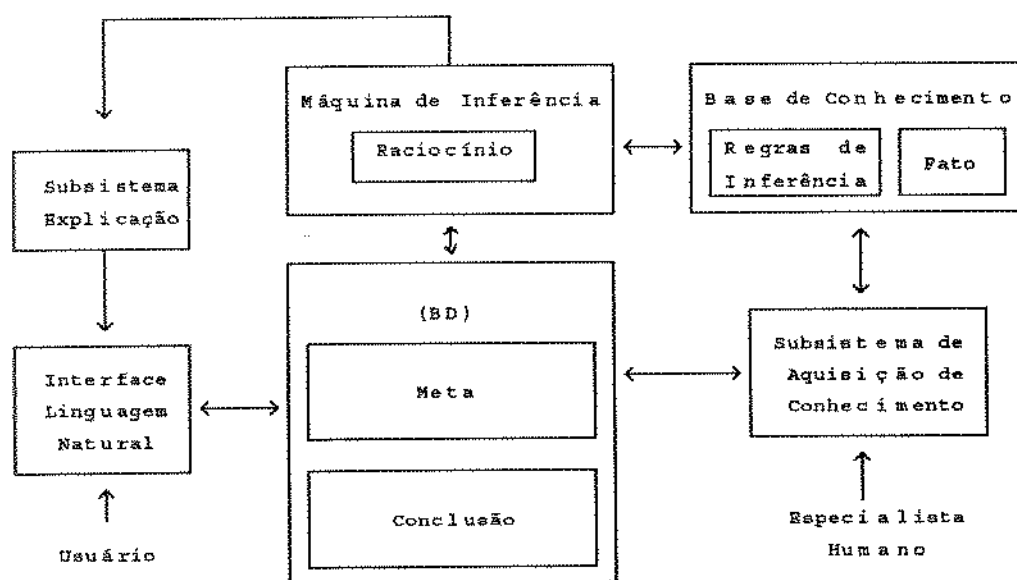


Figura 2: Arquitetura de um SE
(Schaffer 1986)

- Um Subsistema de Aquisição de Conhecimento ("Knowledge Acquisition Facility"), utilizado pelo especialista para desenvolvimento do sistema.

Esse subsistema tem que ser capaz de interagir com o especialista não apenas para receber o conhecimento, mas também fornecer/solicitar informações sobre os conceitos básicos que estão sendo usados ou que são novos para o desenvolvimento do sistema.

3.9. SISTEMAS ESPECIALISTAS NO CONTEXTO DA MANUFATURA

Na manufatura, sistemas que utilizam a técnica "*forward chaining*" trabalham com pedaços de informações elementares, tais como uma lista de inventário, que tenta combinar os elementos, até alcançar a meta, que pode ser, por exemplo, um modelo de produto que satisfizesse as especificações dadas. Esse tipo de técnica é particularmente usual em sistemas construtivos que visam a síntese, como é comum em manufatura.

E, no "*backward chaining*", ocorre a hipotetização de uma conclusão; é entretanto, o mais apropriado para diagnosticar tarefas, nos quais a meta é representada pelas condições falhas e o sistema deduz as razões básicas para as condições falhas.

Quanto aos "*rule-based system*", que compreendem um conjunto de "regras de produção" que podem representar a natureza dinâmica da aplicação, se encaixam perfeitamente a muitos problemas de manufatura.

CAPÍTULO 4
AS BASES DA TECNOLOGIA DA USINAGEM
PARA REALIZAÇÃO DO
SEU - SISTEMA ESPECIALISTA DE USINAGEM

4.1. OTIMIZAÇÃO DAS CONDIÇÕES DE USINAGEM

Um dos maiores precursores da análise econômica da usinagem dos metais foi Frederick Winslow Taylor nos EUA em 1907, chegando inclusive a uma equação que relaciona o desgaste da ferramenta com a velocidade de corte e, desta para a conhecida equação de vida de ferramentas, a "Equação de Taylor". A partir dessa equação os pesquisadores passaram a preocupar-se com a determinação das velocidades de mínimo custo, máxima produção e máximo lucro.

A indústria metal-mecânica sempre teve como um de seus princípios básicos a otimização das condições de produção, sendo este um dos primordiais fatos da evolução das máquinas-ferramenta e das ferramentas de corte, induzindo desta forma à redução dos tempos de usinagem. Para tal redução, é preciso preocupar-se principalmente com:

- . ferramenta adequada ao processo;
- . avanços e profundidades de corte compatíveis com a capacidade da máquina-ferramenta;
- . utilizar adequadamente o fluido de corte;

- . aumento da velocidade de corte, levando a uma maior taxa de remoção de cavaco (o que levaria a um decréscimo na vida da ferramenta, podendo dessa forma, levar a um maior tempo de fabricação devido ao aumento de trocas de ferramentas).

Com o aumento da velocidade de corte existem efeitos positivos, maior taxa de remoção de cavaco; e negativos, maior frequência de troca de ferramenta, mas existe uma velocidade na qual ocorre o equilíbrio entre esses efeitos, a velocidade de máxima produção (V_{mxp}).

4.2. DETERMINAÇÃO DAS CONDIÇÕES DE USINAGEM

Para se determinar as condições de usinagem de uma determinada peça, deve-se considerar (Diniz 1989):

- a- avanço e profundidade de corte baseados em considerações restritivas do sistema (geometria da ferramenta, geometria da peça, tolerâncias, rigidez do sistema máquina-dispositivo-ferramenta-peça, etc). Obedecidas as restrições, os valores de avanço e profundidade de corte devem ser máximos.
- b- uma velocidade de corte otimizada onde estejam equilibrados custo e produção. Nestes casos costuma-se dizer que a velocidade de corte deve estar dentro do Intervalo de Máxima Eficiência (I.M.E.), que é limitado pela Velocidade de Mínimo Custo (V_0) e pela Velocidade de Máxima Produção (V_{mxp}) - (Figura 3).

A V_{mxp} é a velocidade de corte para a qual a soma dos tempos de troca de ferramenta e de usinagem são mínimos.

Como mostra a Figura 3 (Ferraresi 1970), é necessário calcular-se V_o (Velocidade de Mínimo Custo) e V_{mxp} (Velocidade de Máxima Produção), o que pode ser feito segundo as fórmulas :

$$V_{\text{mxp}} = \sqrt[x]{\frac{K}{(x-1) \cdot t_{ft}}}$$

onde:

- x e K são ditos constantes de Taylor, sendo que o expoente x representa a sensibilidade de vida da ferramenta para a variação de velocidade de corte, e K representa a influência do material, isto é, pode ser interpretado como a vida da ferramenta para uma velocidade de corte de 1 m/min.

- t_{ft} é o tempo de troca de ferramentas (min).

É possível mostrar (Pallerosi e Cupini 1975) que:

$$x = \frac{\log(Z_i / Z_{i+1})}{\log(V_{i+1}/V_i)} + 1 ;$$

$$K = Z_i \cdot (V_i)^x \cdot tc_i ;$$

onde:

- Z_i é o número médio de peças usinadas por vida na condição (i);
- V_i é a velocidade de corte na condição (i);
- t_{ci} é o tempo de corte na condição (i).

Analogamente, tem-se a velocidade de mínimo custo, dada por:

$$V_0 = \sqrt[x]{\frac{C_2 \cdot K}{60 \cdot (x-1) \cdot C_3}}$$

onde:

- $C_2 = S_h + S_m ;$
- $C_3 = K_{ft} + t_{ft} / 60 \cdot (S_h + S_m) ;$
- S_h : custo do operador (CR\$/hora);

- S_m : custo total da máquina (CR\$/hora);
- K_{ft} : custo da ferramenta por vida (CR\$)

A velocidade de mínimo custo é a velocidade para a qual existe um equilíbrio entre o custo referente ao salário/homem e salário/máquina de um lado e o custo do número de ferramentas utilizadas de outro.

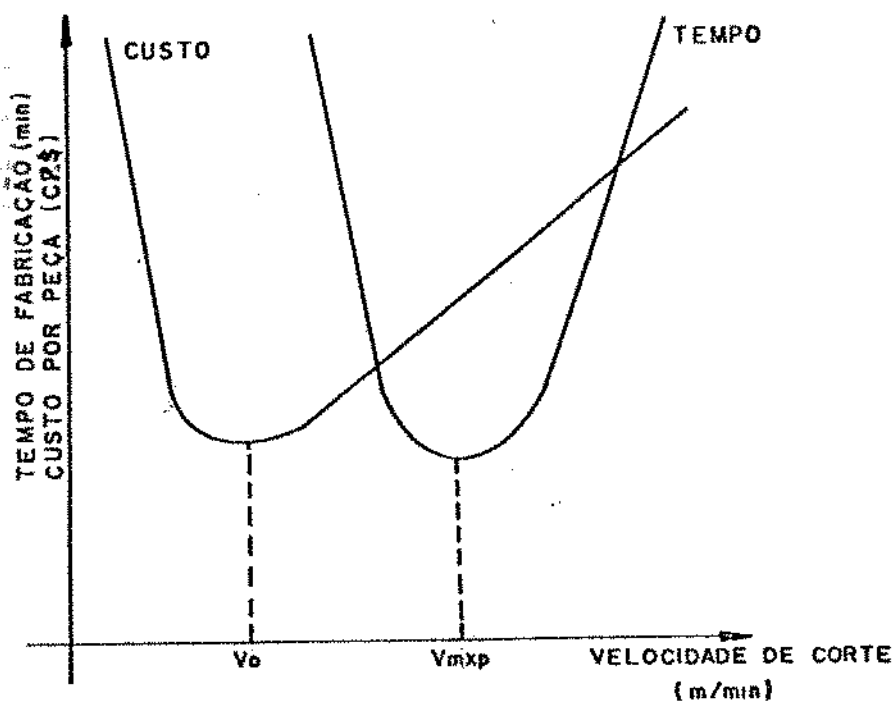


Figura 3: Intervalo de Máxima Eficiência (I.M.E.)
(Ferraresi 1970)

A Figura 3, mostra o chamado I.M.E. - Intervalo de Máxima Eficiência. Nele, encontram-se as melhores velocidades de corte que devem ser utilizadas, pois estas estão limitadas pelas velocidades de mínimo custo e máxima produção.

A teoria legada por Taylor indica que é uma condição necessária trabalhar-se no Intervalo de Máxima Eficiência.

Vilella e Cupini (Vilella 1988) concluíram que a determinação do Intervalo de Máxima Eficiência não é condição necessária para otimização da velocidade de corte, sendo a determinação da velocidade de máxima produção condição suficiente para isso. Estes autores, calcularam o índice D, sempre maior que 1, dado por:

$$D = \frac{V_{\text{exp}}}{V_0} = \left[\frac{60 \cdot K_{ft}}{(S_h + S_m) t_{ft}} + 1 \right]^{1/x}$$

No mesmo trabalho, foi ainda analisada a variação do custo total de fabricação, quando $D \rightarrow 1$ e $D \rightarrow \infty$.

$D \rightarrow 1$ quando ocorrem as seguintes situações, isoladas ou simultaneamente:

- . O custo da ferramenta (K_{ft}) tende a valores muito baixos;
- . O salário-homem (S_h) tende a valores muito altos;
- . O salário-máquina (S_m) tende a valores muito altos;
- . O tempo de troca da ferramenta (t_{ft}) tende a valores muito altos.

Portanto, sempre que $D \rightarrow 1$ ou V_{exp} tende à coincidir com V_c , fica evidenciado que esta é uma situação em que se configura uma condição suficiente para determinar a velocidade ótima de corte, escolhendo-a igual ou imediatamente inferior ao valor determinado pelo modelo no cálculo da velocidade de máxima produção.

$D \rightarrow \infty$, quando ocorrem as seguintes situações, isoladas ou simultaneamente:

. O tempo de troca da ferramenta (t_{ft}) tende a valores muito baixos, o que leva ao aumento do custo de ferramental, e neste caso, deve-se ter cuidado ao afirmar que a obtenção de V_{exp} é condição suficiente para a definição da velocidade de corte mais adequada. Em situações práticas no entanto, pode-se considerar essa situação pouco comum, porque a redução de t_{ft} leva a valores muito altos de V_{exp} , dificilmente utilizados na prática. Assim, velocidades de corte menores do que V_{exp} serão adotadas. A velocidade que deverá ser escolhida será portanto, a máxima disponível na máquina ferramenta. Neste caso, existe a possibilidade da velocidade máxima da máquina ser inferior à velocidade de mínimo custo para o par ferramenta peça. Assim, este fato estará evidenciando que a ferramenta possui qualidades muito superiores às exigidas pela máquina para a usinagem daquela peça em questão.

. O custo da ferramenta K_{ft} tende a valores muito altos, esta é uma situação crítica, pois observa-se que tanto a variação do custo total de fabricação como o custo por peça para a velocidade de máxima produção tem tendência crescente; e, numa

situação desse tipo a escolha de V_{mxp} como velocidade de trabalho pode elevar os custos a valores muito altos. Portanto, especialmente em situações onde o custo da ferramenta é muito alto, não se deve generalizar a escolha da V_{mxp} como velocidade ótima. O mais adequado seria nesses casos, fazer uma avaliação individualizada da situação. Para trabalhar próximo à velocidade de máxima produção e evitar custos elevados, e ainda evitar que a velocidade ótima seja escolhida abaixo da velocidade de mínimo custo, aqueles autores sugerem adotar valores entre 10% a 20% inferiores à V_{mxp} . Tal recomendação não é absolutamente fundamentada em cálculos, mas única e exclusivamente em análise das grandezas envolvidas na simulação de comportamento do índice D.

Finalizando, os autores ainda discutem a influência de S_h e de S_m , concluindo que estes fatores não são relevantes para as análises em questão. Ou seja, a velocidade de máxima produção, para os casos em que estes fatores crescem, ainda permite ser considerada como condição suficiente para a determinação da velocidade ótima com base nela.

CAPÍTULO 5

SISTEMA ESPECIALISTA DE USINAGEM (SEU)

5.1. GENERALIDADES

O domínio de sistemas de produção e manufatura (Eloranta 1990) é obviamente vasto, além de que os problemas a serem solucionados tem um certo nível de abstração, portanto é conveniente a utilização de sistemas inteligentes. O SEU pode ser considerado de alta complexidade, o que é visto como (Forbellone e Eberspuchen 1993) sinônimo de variedade, que por sua vez representa a quantidade de situações diferentes que um problema pode apresentar, e deve ter previsão de solução.

Neste capítulo será proposto um SE, que tem por meta resolver o problema da otimização das condições de usinagem.

5.2. METODOLOGIA

A abordagem adotada para atingir os objetivos mencionados anteriormente, foram baseadas em técnicas de Engenharia do Conhecimento, pautada em discussões com especialistas e análise de documentos por eles gerados (Vilella 1988, Pallerosi e Cupini 1975, Rodrigues et al. 1987) para a solução do problema em questão.

A Figura 4, esquematiza as técnicas de Engenharia de Conhecimento empregadas, revelando a metodologia empregada.

A fase inicial (1), a definição do problema, envolveu o entendimento do modelo de otimização das condições de usinagem, através de entrevistas e estudos detalhados do trabalho de Vilella (Vilella 1988), a identificação de suas características, esboço dos objetivos requeridos e definição da metodologia requerida para solucionar esse problema. Nesta fase, procurou-se uma linguagem comum entre os envolvidos na construção do sistema. Foi também definido que o sistema seria desenvolvido para ser executado em microcomputadores da linha IBM-PC XT/AT, pois é este o tipo que mais se encontra em ambiente fabril.

Em (2), estudo da viabilidade, foi verificado que o conhecimento do especialista em usinagem tinha um grau de complexidade relativamente alto. Neste trabalho, é considerado como especialista o técnico capaz de tratar os modelos de cálculo de tempos totais de fabricação e de custos totais de fabricação por peça, dentro de um procedimento de determinação dos coeficientes x e K da equação de Taylor, em ambiente fabril. É ainda parte do conhecimento do especialista, técnicas de levantamento e tratamento de dados oriundos de uma experimentação, ainda em (2), verificou-se a necessidade de procedimentos de busca heurística; dificuldade em formalizar as regras, e portanto concluiu-se que realmente seria necessário empregar técnicas de IA para o desenvolvimento do sistema.

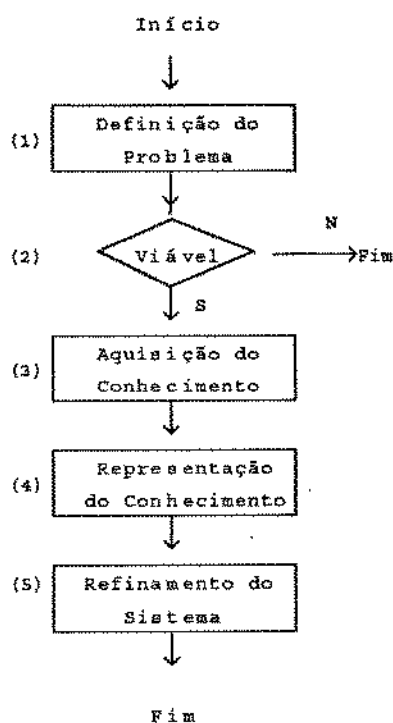


Figura 4 - Etapas de Desenvolvimento do SEU

Na fase de aquisição de conhecimento (3), o especialista em usinagem repassou fatos e idéias adquiridos durante sua vida, seus estudos, suas observações e ensaios, o qual foi implementado na fase (4) através de uma Máquina de Inferência, juntamente com o que foi absorvido dos documentos analisados em (1). Sendo que o conhecimento de usinagem adquirido do especialista em usinagem foi implementado na linguagem PROLOG, pois Vita (Vita 1987) afirma que Prolog é uma ferramenta ideal principalmente para duas classes de problemas:

- . aqueles que são difíceis de formular proceduralmente; e
- . aqueles que apresentam amplas quantidades de soluções alternativas .

E, com isto justifica-se o uso da linguagem PROLOG, pois o presente problema pertence a estas duas classes.

Tal implementação foi realizada através de:

- . **Fatos relatados**, que fazem parte da Base de Conhecimento Declarativa. Como exemplo, pode-se citar que para construir o módulo Cálculos Estatísticos , há necessidade de utilização da tabela de t-student para os devidos cálculos, sendo que a partir desta tabela verifica-se que para uma dada confiança e um determinado grau de liberdade obtém-se um único valor, o qual é considerado fato.
- . **Regras ou procedimentos**, que geram caminhos alternativos de raciocínio no SEU, que fazem parte da Base de Conhecimento Procedural. Por exemplo, uma das maneiras de se calcular a V_{clim} seria através da regra:

Se $(Ne > Nm * Neta)$

Então $V_{clim} = (Nm * Neta * 60 * 75 / P_c)$

Para as consultas e testes (5) foi necessário utilizar-se de um Banco de Dados genérico construído em CLIPPER/dBASE III e acessado através de uma interface de comunicação construída em PROLOG. A necessidade da interface deve-se ao fato da distinção entre as linguagens do Banco de Dados e o programa propriamente dito.

O sistema é considerado um sistema de tamanho médio, como a maioria dos SEs desenvolvidos para PC. Barborak (Barborak 1991) define o tamanho de um SE que trabalha com regras de produção, através do seguinte critério:

<u>Tamanho do Sistema</u>	<u>Número de Regras</u>
Pequeno	10 - 200
Médio	200 - 500
Grande	acima de 500

Tabela 5: Definição do Tamanho de um SE

5.3. ARQUITETURA DO SEU

As figuras abaixo, Figura 5.1 e Figura 5.2, mostram a arquitetura do SEU, para sistemas automáticos e semi-automáticos respectivamente.

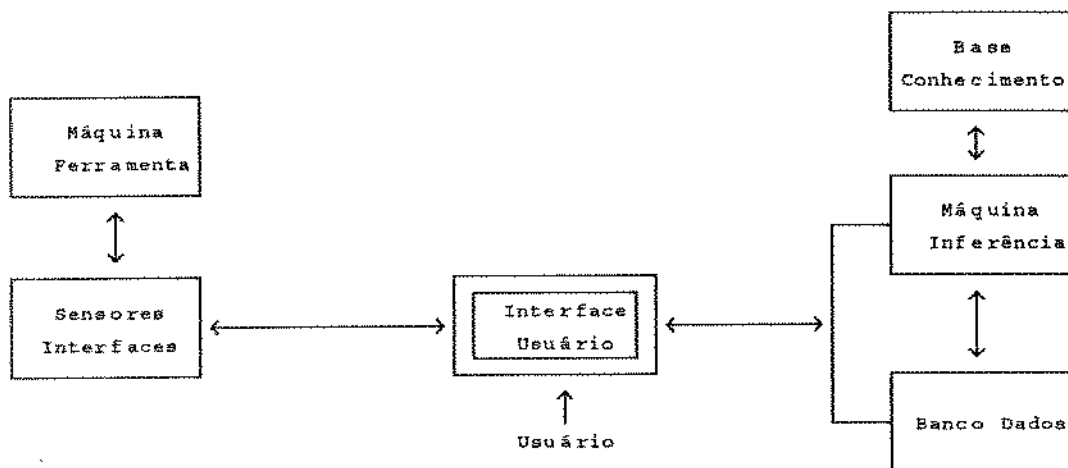


Figura 5.1: Arquitetura do SEU para Sistemas Automáticos

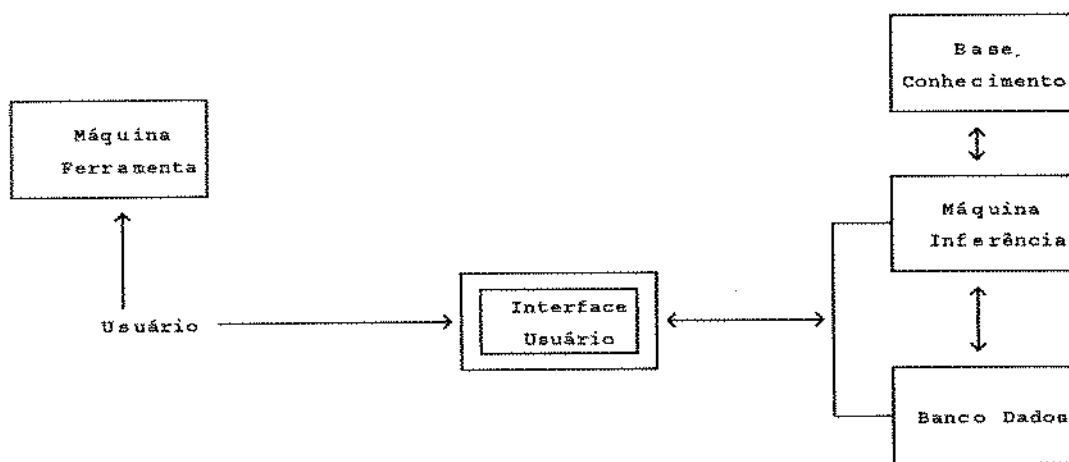


Figura 5.2: Arquitetura do SEU para Sistemas Semi-Automáticos

5.3.1. Automatização do Sistema

Com a finalidade de otimizar as condições operacionais, o SEU pode ser aplicado tanto a nível de semi-automação como para automação plena, mas ele foi testado exclusivamente para sistemas semi-automáticos. Nos dois esquemas o usuário acessará uma interface de comunicação, a qual acionará o Banco de Dados e a Máquina de Inferência. A Máquina de Inferência acessará a Base de Conhecimento conforme suas necessidades, e executará o que lhe for solicitado.

Para sistemas automáticos, o usuário, provavelmente o operador da máquina, será poupado de operar a máquina, da contagem do número de peças, das mudanças de rotação, da determinação do critério de fim de vida das ferramentas, pois para tais sistemas ocorrerá o monitoramento do processo através de sensores que meçam as condições limítrofes do processo.

Para sistemas semi-automáticos a operação da máquina é feita através de programação CNC, havendo necessidade da contagem do número de peças, mudanças de rotação e critério de vida executados pelo usuário. A introdução dos dados coletados e a retirada dos valores otimizados são feitos pelo usuário através do computador. Portanto, esse sistema interage com o usuário sempre que necessitar da alimentação de novas informações (dados) para prosseguir com sua tarefa, ou da tomada de decisão por parte do usuário.

5.3.2. Banco de Dados

O Banco de Dados contém dados de catálogo referentes a peça, ferramenta e máquina-ferramenta, foi construído em Clipper/dBASE III.

5.3.3. Base de Conhecimento

A Base de Conhecimento foi construída a partir de informações fornecidas pelo especialista e transformadas em fatos e regras. O conhecimento foi representado (codificado) através de regras de produção, um sistema que é baseado em tais regras é chamado de sistema de produção.

Na Base de Conhecimento está contido o conhecimento do especialista através de descrições (regras e fatos), relacionamentos (dependências e associações) e procedimentos (raciocínio das operações que devem ser executadas).

5.3.4. Máquina de Inferência

A Máquina de Inferência é a principal componente do SEU e ela é do tipo "forward chaining", isto é, partiu de um conjunto de dados iniciais de usinagem, os quais descrevem o estado inicial do problema e estão armazenados no banco de dados criado em dBASE III. Se as condições de uma determinada regra forem totalmente satisfeitas para esse conjunto inicial de dados, as ações pendentes serão executadas. A execução das ações tem como consequência acrescentar fatos (dados) no Banco de Dados, enriquecendo-o.

A Base de Dados modificada levará outras regras ao sucesso, e consequentemente provocará a execução das ações correspondentes. O controle operou em ciclos, sendo que, em cada ciclo as regras são examinadas pela máquina de inferência, e é ela quem determina quais são as regras aplicáveis e quais ações devem ser disparadas. Se várias regras são aplicáveis a um certo ciclo, somente uma delas será escolhida para ser executada. A esta escolha dá-se o nome de resolução de conflitos. Esse controle tem como princípio a metodologia alternativa baseada na equação de Taylor apresentada por Vilella (Vilella 1988), em sua tese de mestrado.

A aplicação desta metodologia para otimização da produção com base na velocidade de corte de máxima produção foi feita em fabricação por lotes, onde estava sendo utilizado o princípio da tecnologia de grupo, sendo a aplicabilidade desta metodologia avaliada em condições de produção.

Essa metodologia tem por base:

a - adoção de uma velocidade de corte inicial (o autor do modelo sugere que esta velocidade seja a indicada pelo processista com base em sua experiência ou escolhida no catálogo da ferramenta em uso);

b - adoção de um critério de fim de vida (o importante do critério é a sua manutenção e repetibilidade durante a evolução da aplicação do modelo);

c - cálculo estatístico do número médio de peças usinadas por vida da ferramenta para a velocidade de corte adotada;

d - cálculo e/ou medição do tempo de corte (no caso de medição, esta deverá levar a um valor médio estatístico);

e - adoção de nova velocidade de corte (20% maior ou 20% menor do que a adotada) e cálculo estatístico do número médio de peças correspondentes. O valor de 20%, é sugerido pelo autor. Todavia, qualquer outro valor pode ser adotado. Porém, o importante é que a nova velocidade seja diferente e não muito próxima do valor adotado para garantir maior precisão na determinação do número médio de peças para cada velocidade ensaiada;

f - caracterizando-se as situações acima como (i) e (i+1), pode-se associar às mesmas, as velocidades vc_i e vc_{i+1} ; o número médio de peças por vida correspondente Z_i e Z_{i+1} e os tempos de corte tc_i e tc_{i+1} ;

g - cálculo das constantes de Taylor:

$$x = \frac{\log (Z_i / Z_{i+1})}{\log (vc_i / vc_{i+1})} + 1$$

$$K = Z_i \cdot tc_i \cdot vc_i^x$$

h - paralelamente a este processo, medir com confiança estatística o tempo de troca t_{ft} , obtendo-se um tempo médio de troca. Este procedimento é um procedimento usual utilizado por Barnes (Barnes 1977).

i - calcular a velocidade de máxima produção:

$$V_{\text{mxp}} = \sqrt{x \frac{K}{(x-1) \cdot (t_{ft})}}$$

j - comparar V_{mxp} com vc_i e vc_{i+1}

. se $V_{\text{mxp}} > vc_{i+1}$, repita a operação acrescentando vc_i de 20%, isto é: $vc_{i+2} = 1.2 \times (vc_{i+1})$

. se $vc_i < V_{\text{mxp}} < vc_{i+1}$, assuma V_{mxp} como velocidade de referência

. se $V_{\text{mxp}} < vc_i$, repita a operação reduzindo-se vc_i de 20%, isto é: $vc_{i-1} = vc_i / 1.2$

k - determinação da "velocidade ótima" como um percentual da V_{mxp} definida com referência à dados sobre custos do SMFD (Sistema Máquina Ferramenta e Dispositivos) existentes na Base de Dados e Base de Conhecimento.

As regras do SEU são formadas por cláusulas que consistem de uma combinação de atributos, valores, sendo estes chamados de elementos da base.

Os atributos são os conceitos fundamentais do conhecimento que se está formalizando e correspondem aos campos do Banco de Dados.

Exemplo: CODIGO, NM, NETA, POTENCUTIL que são os campos referenetes a DADOM AQ.

Os valores são as ocorrências desses conceitos, equivalendo aos valores associados a cada campo do Banco de Dados.

Exemplo: 55, 100.0, 0.5, 50.000 referentes a CODIGO, NM, NETA, POTENCUTIL do arquivo DADOM AQ.

Cada cláusula pode definir um fato ou uma função. Os fatos se subdividem em:

. expressões de conhecimento

Ex: $P_a = P_{amax}$; e

. expressões aritméticas

Ex: $P_{autil} \text{ is } 0.9 * P_{amax}$.

Onde P_a , P_{amax} , P_{autil} são variáveis retiradas do programa.

As funções são os comandos que permitem que as tarefas que não possam ser executadas diretamente no SEU, como por exemplo busca em um dos Bancos de Dados, sejam processadas por um programa externo escrito em qualquer linguagem; desde que seja construída uma interface de comunicação; que pode receber parâmetros e também devolver valores para o SEU.

As funções implementadas no SEU, tem como princípio básico a busca de dados nos Bancos de Dados em Clipper/DBASE III através de interfaces que tornem possível a comunicação entre as duas linguagens (PROLOG e Clipper/DBASE III).

5.3.5. Interface com o Usuário

Foi implementada em PROLOG tentando-se torná-la o mais amigável possível, já que deseja-se ter um sistema utilizável em qualquer nível do sistema produtivo.

O programa final consiste da integração do Sistema Especialista de Usinagem (Máquina de Inferência), módulo de cálculos estatísticos e do módulo de interface de comunicação, apresentando uma interface gráfica que auxilia o usuário na escolha da tarefa a ser realizada pelo computador. Assim conseguiu-se tornar mais eficiente o funcionamento global do programa, através da redução do tempo gasto entre a execução de cada módulo. Os processamentos computacionais mais complicados, como a transmissão e recepção de dados realizada pelo conversor, permanecem convenientemente "escondidas". Dessa maneira, o usuário tem a sensação de que o SEU acessa diretamente o banco de dados dBASE, tornando a manipulação da interface gráfica o mais simples e natural para um usuário leigo em computação.

Para tal implementação, foi utilizado o conceito de "dialog box", que é um tipo especial de janela, a qual contém controles para executar diferentes funções. Como o próprio nome sugere, um

"dialog box" permite diálogo entre o sistema e o usuário, sendo este induzido a fazer certas escolhas dentro de uma lista de opções ou digitar a informação que lhe convier.

No Arity/Prolog existem oito padrões de controles diferentes para executarem alguma função. São eles: "choice button", "radio button", "edit field", "push button", "list box", "edit box", "edit region" e "text".

Um "dialog box" processa a informação digitada pelo usuário através da transferência de mensagens. Toda vez que um usuário pressionar uma tecla dentro de um "dialog box", o sistema terá a função de transferir essa mensagem ao sistema.

A Figura 6 mostra uma versão simplificada da arquitetura de transferência de mensagens utilizando-se de um "dialog box".

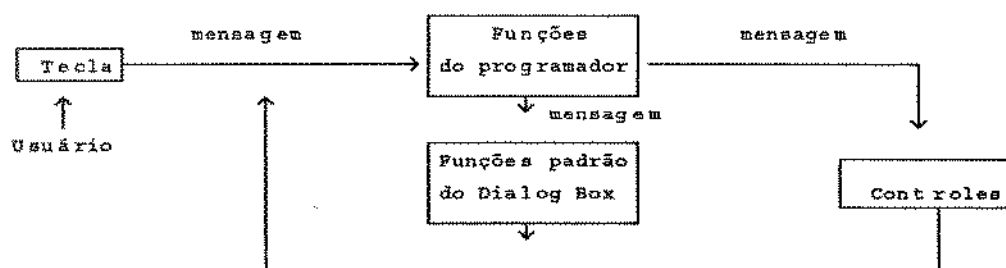


Figura 6: Arquitetura Simplificada de Transferência de Mensagens

Na implementação da interface com o usuário do SEU, não foram usadas somente os controladores "choice button" e o "radio button", dentre os seguintes controladores:

. "choice button", que tem como função listar um item singular. Um usuário pode escolher este item movendo o cursor para o "choice button", pressionando a barra de espaço sobre o item desejado. Isto acarretará no aparecimento de um caracter especificando que aquele item foi selecionado. Se a barra de espaço for pressionada novamente sobre o mesmo item, a seleção será cancelada.

. "radio button", são utilizados quando se tem duas ou mais opções, mas o usuário poderá selecionar somente uma delas.

. "edit field", permite que o usuário entre com uma linha texto em um determinado campo.

. "push button", são frequentemente utilizados para confirmar uma ação, tal como, apertar um botão de "Ok" para indicar que seja executada alguma mudança desejada. Ele executa somente ações singulares.

. "list box", apresenta uma lista de itens que navegam (rolam) verticalmente na tela. É uma forma conveniente para apresentar longas listas de itens ao usuário. Há dois tipos de "list box": "radio list box" e "choice list box". O primeiro tipo, permite a escolha de um único item enquanto que o segundo tipo, o usuário pode escolher vários itens.

. "edit box", é uma caixa que contém um editor funcional. Inclui uma borda e um "cursor location", indicando a linha e coluna em que o cursor está.

. "edit region", é idêntica ao "edit box" sem a borda e o "cursor location".

. "text", qualquer texto num "dialog box" não pertencente a outro controle, deve ser definido como um controle separado. São usados para mostrar informações, ao invés de oferecer escolha.

Portanto, a interface com o usuário desenvolvida para o SEU foi feita através de janelas, que acessam o raciocínio e cálculos numéricos a fim de fornecer ao usuário o valor da velocidade ótima de trabalho.

Ao rodar o programa, o usuário deparará com uma janela (Figura 7) que contém três itens:

. Processos: fornece alguns nomes de processos mecânicos de usinagem;

. Cálculos: Fornecerá a média e o desvio padrão para uma determinada população;

. Help: Auxiliará o usuário como utilizar os "dialog boxes", através de informações simples e básicas.

A escolha da opção processos mecânicos de usinagem se dará clicando <ENTER> sobre o item Processos, ou simplesmente teclando a letra P (Figura 8). Apesar de aparecerem listados vários processos (torneamento, fresamento, rosqueamento, mandrilamento,

retificação, alargamento, brochamento, aplainamento), o referido sistema está implementado somente para o processo de torneamento.

Juntamente com a lista dos processos estará disponível a interrupção da execução do programa através das opções "Dos Shell" ou "Halt" (Figura 9).

Após a escolha da opção torneamento, o Banco de Dados será acionado retornando com as informações em uma nova janela (Figura 10), composta pela lista de material e pelos códigos adotados para as máquinas e as ferramentas. A Máquina de Inferência será acionada após o usuário clicar OK sobre o "push button" e, dessa forma será fornecida uma nova tela (Figura 11), a qual contém dados de ensaio relativos ao material em questão, sendo que o usuário terá possibilidade de manipulação desses dados, isto é, acrescentar mais dados ou modificar os dados já existentes.

As manipulações dos dados de ensaio podem ser feitas via teclado ou através do módulo Cálculos Estatísticos. Caso o usuário queira fazer a modificação via Cálculos Estatísticos ele acionará um dos "push buttons": Estatística de Zt..., Estatística de tc... ou Estatística de tft..., retornando assim à tela do módulo Cálculos Estatísticos (Figura 15), a partir de onde dará continuidade ao sistema.

Se o usuário não optar pela utilização do módulo Cálculos Estatísticos, e após feita as necessárias alterações na tela "dados de ensaio" e optar por clicar Ok, lhe será fornecida a tela com as metas do sistema executadas (Figura 13).

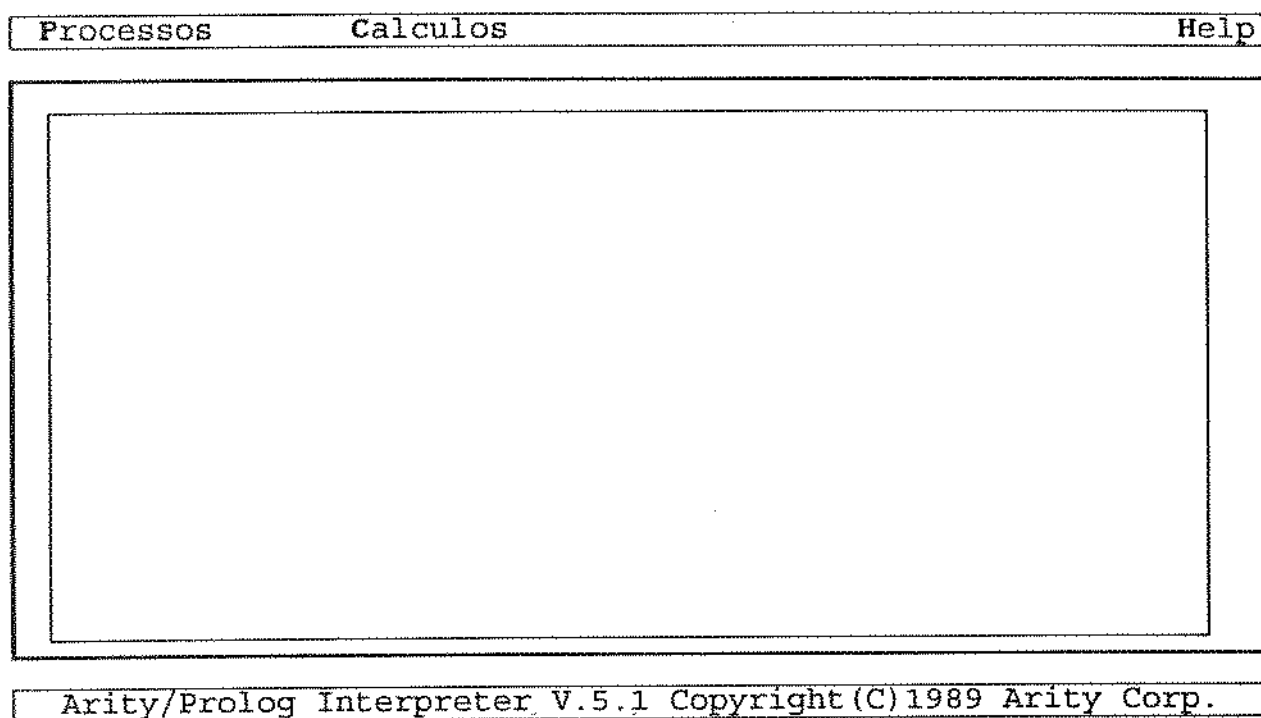


Figura 7: Interface com o usuário - Tela 1

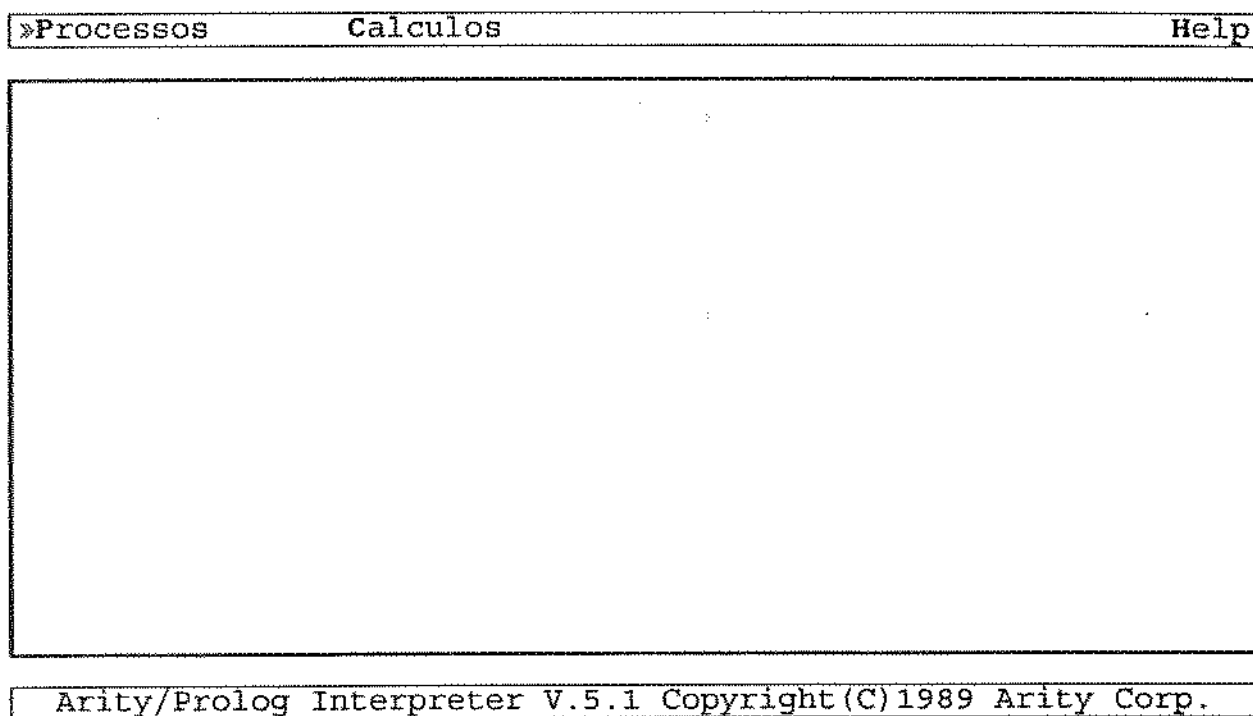


Figura 8: Interface com o Usuário - Tela 1.1

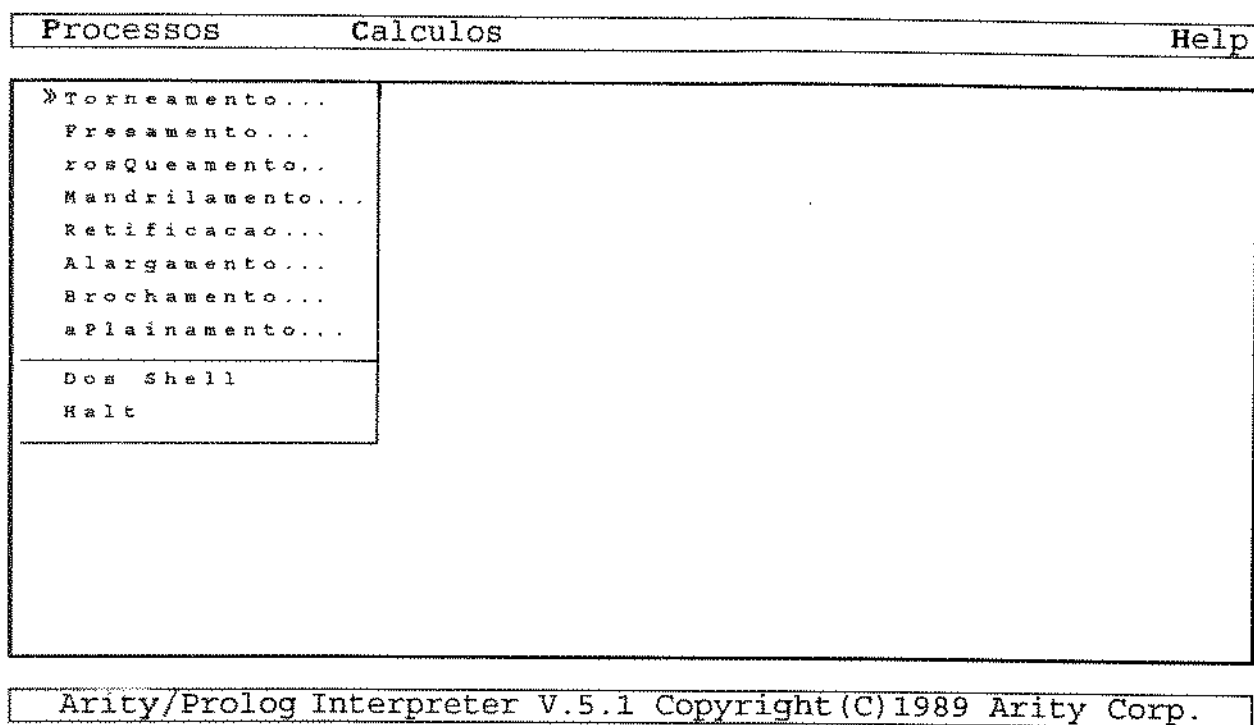


Figura 9 : Interface com o Usuário - Tela 1.2

Processos	Calculos	Help															
<table border="1"><tr><td>Nome do Material:</td><td>Codigos da Maquina:</td><td>Codigos da Ferramenta:</td></tr><tr><td> a3 </td><td> 60 </td><td> P 10 </td></tr><tr><td>Lista dos Materiais:</td><td>Lista de Codigos:</td><td>Lista de Codigos:</td></tr><tr><td> c22 c23 » a3 a31 a32 </td><td> 58 59 » 60 </td><td>» P 10 P 15 P 16 </td></tr><tr><td> OK </td><td colspan="2"> CANCEL </td></tr></table>			Nome do Material:	Codigos da Maquina:	Codigos da Ferramenta:	a3	60	P 10	Lista dos Materiais:	Lista de Codigos:	Lista de Codigos:	c22 c23 » a3 a31 a32	58 59 » 60	» P 10 P 15 P 16	OK	CANCEL	
Nome do Material:	Codigos da Maquina:	Codigos da Ferramenta:															
a3	60	P 10															
Lista dos Materiais:	Lista de Codigos:	Lista de Codigos:															
c22 c23 » a3 a31 a32	58 59 » 60	» P 10 P 15 P 16															
OK	CANCEL																
Arity/Prolog Interpreter V.5.1 Copyright (C)1989 Arity Corp.																	

Figura 10: Interface com o Usuário - Tela 1.3

Processos		Calculos		Help	
Dados Ensaio					
Lista de Codigos: Ensaio 1 2 Mensagem:		Zt Medio: 37.0 tc Medio: 2.2 tft Medio: 5.0		Estatistica Zt Estatistica tc Estatistica tft	
OK		CANCEL		ALTERA	

Arity/Prolog Interpreter V.5.1 Copyright (C)1989 Arity Corp.

Figura 11: Interface com o Usuário - Tela 1.4

Processos	Calculos	Help
Dados: Material = a3 Angulo de Posicao = 60 Avanco = 0.5 Profundidade = 2.5 Velocidade Corte Inicial = 120 Velocidade de Avanco = 600 Ks1 = 180 l-z = 0.78 Potencia da Maquina = 20 Rendimento = 20 Forca Maxima de Corte = 1000 Forca Maxima de Avanco = 250 Calculado: Vclim = 266.2 Vmxp = 225.98		
Arity/Prolog Interpreter V.5.1 Copyright (C)1989 Arity Corp.		

Figura 12: Interface com o Usuário - Tela 1.5

CAPÍTULO 6

COMUNICAÇÃO ENTRE O SEU E O SBD

6.1. GENERALIDADES

Como foi visto anteriormente, o SEU foi totalmente desenvolvido em PROLOG, e como o intuito do sistema é a utilização do mesmo em indústrias do ramo metal-mecânico chegou-se à conclusão de que o Banco de Dados que esse sistema deveria acessar deve ser em dBASE III, pois a maioria das indústrias deste ramo trabalham com Banco de Dados construídos nestas linguagens. Uma pesquisa sobre a utilização de Banco de Dados no Brasil, realizada pela Folha de Informática da Folha de São Paulo mostra, que a utilização do dBASE no mercado nacional atinge 70%, restando somente 30% para os demais bancos de dados. A seguinte tabela foi publicada no referido jornal em 5 de maio de 1993, com os seguintes dados sobre a utilização de Banco de Dados no Brasil:

dBASE	70%
FoxBase	7%
Dataflex	7%
Outros	16%

Tabela 6: Utilização de BD no Brasil

O dBASE é uma linguagem de alto nível com a facilidade de manuseio de dados, e basicamente manipula arquivos, apresentando um ambiente específico para esta tarefa, por serem mais apropriados para o usuário (no nosso caso, indústrias do ramo metal-mecânico).

Para poder então trabalhar com linguagens de características tão distintas, foi necessário a implementação de uma interface de comunicação, que permite transferências de dados entre o Banco de Dados, cujo formato de apresentação é diferente da implementação do SEU, que foi construído em PROLOG. Ainda há a possibilidade de se escolher outro SGBD quando necessário, tendo somente que implementar nova interface de comunicação.

6.2. SISTEMA DE BANCO DE DADOS (SBD)

6.2.1. Generalidades

Um Sistema de Banco de Dados (SBD) corresponde ao conjunto composto de:

- . Banco de Dados
- . Sistema Gerenciador de Banco de Dados

Não existe uma definição rígida para Banco de Dados (BD), mas pode ser considerado como um conjunto de arquivos relacionados entre si, de forma que a organização desses arquivos e os relacionamentos são feitos adequadamente, a fim de se atender, com eficiência, todas as aplicações a que ele se dispõe.

Sistema Gerenciador de Banco de Dados (SGBD) é um conjunto de programas que administram o Banco de Dados. Assim, os programas de aplicação não tem acesso ao BD diretamente, todas as chamadas passam pelo SGBD, ele é um módulo que proporciona a interface entre dados de baixo nível armazenados no Banco de Dados e os programas de aplicação e consultas submetidas ao sistema.

Existem algumas definições básicas que devem ser citadas:

- **Entidade:** um objeto de interesse para um Sistema de Informação.
- **Atributos:** são fatos ou dados relativos às entidades.
- **Campo:** unidade básica que representa a informação, é possível definir seu tamanho e formato.
- **Registro:** conjunto de campos; não possui um tamanho necessariamente fixo, pela simples existência de grupos repetitivos.
- **Arquivo:** é, normalmente, uma relação de registros do mesmo tipo, ou seja é uma estrutura de dados na qual cada registro não ocupa uma posição fixa dentro da estrutura, não possuindo portanto tamanho pré-estabelecido. Os registros são formados por unidades de informação denominados campos (Forbellone e Eberspücher 1993).

O objetivo de um Sistema de Banco de Dados (SBD) é simplificar e facilitar o acesso aos dados (Date 1988), (Korth e Silberschatz 1989); o desempenho do sistema depende da eficiência das estruturas de dados usadas para representá-lo no Banco de Dados e de quão eficientemente o sistema será capaz de operar nestas estruturas.

O SGBD é responsável pelas seguintes tarefas:

- **Interação com o Gerenciador de Arquivos:** os dados básicos estão armazenados em disco, utilizando o sistema de arquivos que normalmente é parte integrante de um sistema operacional convencional. O Gerenciador de Banco de Dados traduz as várias instruções, em linguagem de manipulação de dados, em comandos de baixo nível do sistema de arquivos. Assim, o Gerenciador de Banco de Dados é responsável pelo real armazenamento, consulta e atualização dos dados no Banco de Dados.

- **Garantia de integridade:** os valores de dados armazenados no Banco de Dados precisam satisfazer certos tipos de restrições de consistência, ou seja, os limites necessitam ser especificados para que o Gerenciador de Banco de Dados possa verificar se atualizações no Banco de Dados resultarão na violação de quaisquer dessas restrições/limitações e, se assim for, uma ação apropriada precisa ser tomada.

- **Garantia de segurança:** nem todo usuário do Banco de Dados necessita ter acesso a todo conteúdo do Banco de Dados, pois é trabalho do Gerenciador de Banco de Dados assegurar esses requisitos de segurança.

- **Recuperação/Backup:** todo sistema de computador está sujeito a falhas. Há uma grande variedade de causas para tais falhas, incluindo, quebra de discos, falhas na alimentação elétrica e erros de software. Em cada um desses casos, a informação concernente ao Banco de Dados é perdida, sendo de responsabilidade do Gerenciador do Banco de Dados detectar tais falhas e restaurar o Banco de Dados ao estado anterior.

6.2.2. Estrutura de um Banco de Dados (Korth e Silberschatz 1989)

Para descrever a estrutura de um Banco de Dados, necessitamos definir o conceito de modelo de dados. Modelo de dados é uma coleção de ferramentas conceituais para descrição dos dados, relacionamento entre os dados, semântica e restrições dos dados.

Diversos modelos de dados foram propostos e estão divididos em três diferentes grupos:

- **Modelos Lógicos Baseados em Objetos:** são usados na descrição de dados nos níveis conceitual e visão. Caracterizam-se por proporcionar ampla e flexível capacidade de estruturação e por permitir a especificação de restrições de dados de forma explícita. Os mais conhecidos são:

- . modelo entidade-relacionamento (E-R);
- . modelo binário;

. modelo de semântica de dados;

- **Modelos de Dados Físicos:** são usados para descrever os dados em seu nível mais baixo. Os mais conhecidos são:

. modelo unificador;

. memória em frames.

- **Modelos Lógicos Baseados em Registros:** são usados para especificar tanto a estrutura lógica global do Banco de Dados como uma descrição em alto nível da implementação. Eles não proporcionam, no entanto, facilidades para especificar explicitamente restrições de dados. Os três modelos lógicos baseados em registros, mais utilizados, são:

. **modelo relacional:** os dados e os relacionamentos entre os dados são representados por uma coleção de tabelas, cada qual com um número de colunas e nomes únicos;

. **modelo de rede:** os dados são representados por coleções de registros e os relacionamentos entre os dados são representados por ligações que podem ser vistas como ponteiros. Os registros no Banco de Dados são organizados como coleções arbitrárias de grafos;

. **modelo hierárquico:** é similar ao modelo de rede no sentido em que dados e relacionamentos entre si são representados por registros e ligações, respectivamente. O modelo hierárquico

difere do modelo de rede porque os registros são organizados como coleções arbitrárias de árvores em vez de grafos.

Uma tarefa importante na modelagem do Banco de Dados é a especificação da distinção entre entidades (registros) e relacionamentos. Conceitualmente, entidades individuais e relacionamentos são distintas, mas na perspectiva do Banco de Dados a diferença entre eles precisa ser expressa em termos de seus atributos (campos). Para fazer tais distinções, uma superchave é assinalada para cada conjunto de entidades. Esta superchave é um conjunto de um ou mais atributos que, tomados em conjunto, permitem identificar unicamente uma entidade no conjunto de entidades.

6.2.3. Fases de Projeto de um Banco de Dados

A execução de um Banco de Dados pode ser desenvolvida em três etapas básicas:

- **Projeto conceitual:** esta fase independe do SGBD escolhido, pois antes de fazer o cadastramento das informações no sistema, necessita-se definir com cuidado quais informações ficarão em quais arquivos. Para ter-se um Banco de Dados eficiente, deve-se evitar redundância de dados; isto é o armazenamento de um mesmo dado duas ou mais vezes; e possibilitar que as consultas sejam rápidas.

- **Projeto lógico:** para esta representação é necessário a utilização de uma linguagem para especificar os esquemas lógicos.

- **Projeto físico:** é a descrição da implementação da base de dados em memória secundária. Descreve estruturas de armazenamento e métodos de acesso, tendo grande influência do Gerenciador de Banco de Dados escolhido.

6.3. INTERFACEAMENTO

Por interfaceamento pode-se entender uma união física e, normalmente, lógica entre dois sistemas que não poderiam se conectar diretamente. A interface de comunicação em questão permite a transferência de dados entre o Banco de Dados criado em dBASE III e o Sistema Especialista construído em Prolog.

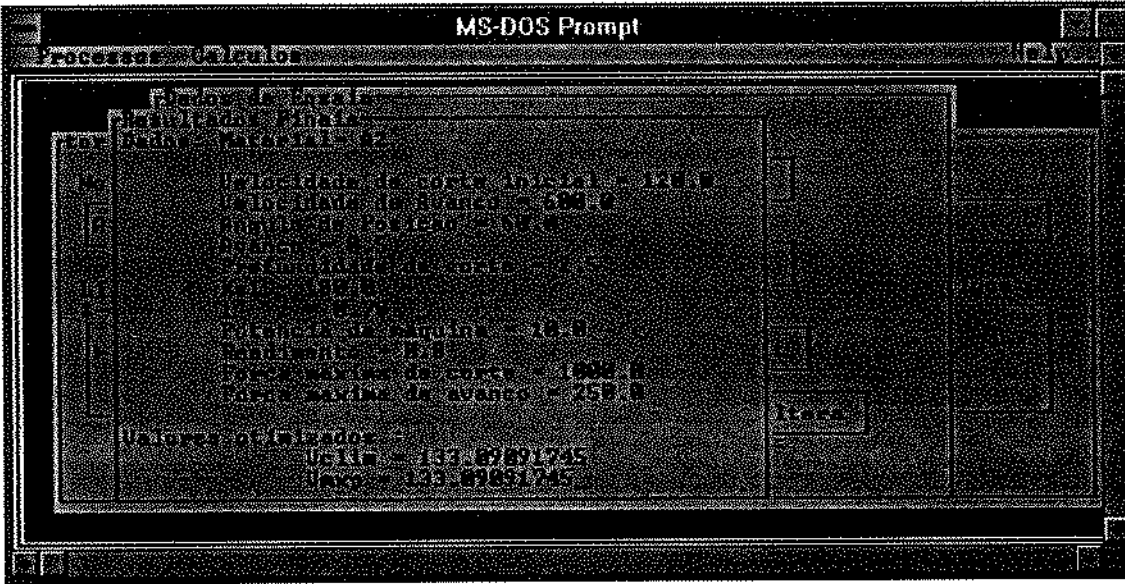
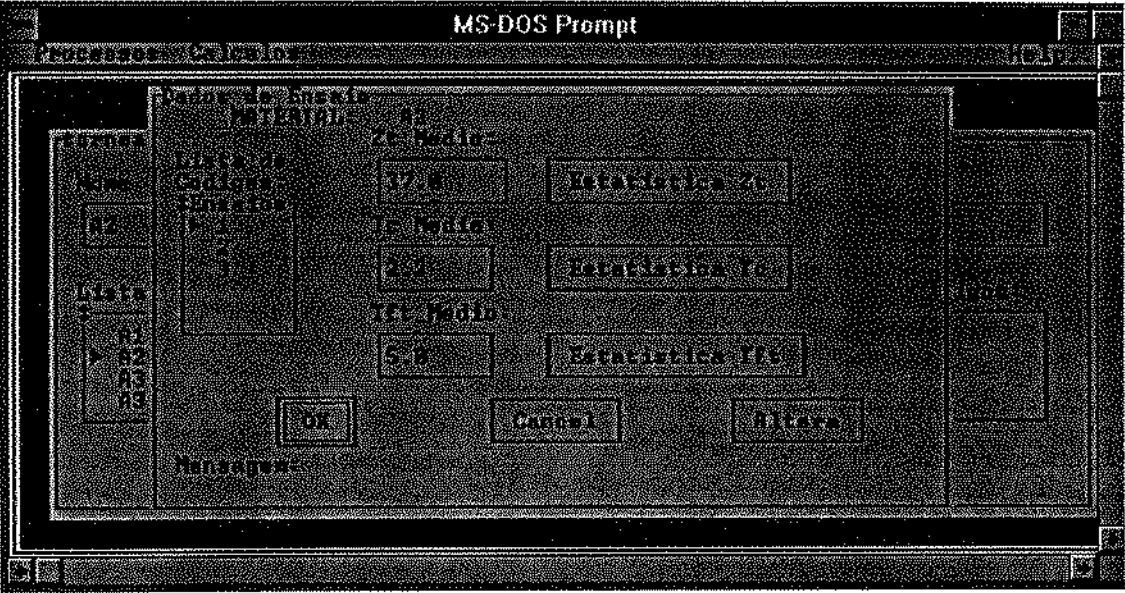
6.3.1. CLIPPER/dBASE III (Vidal 1989)

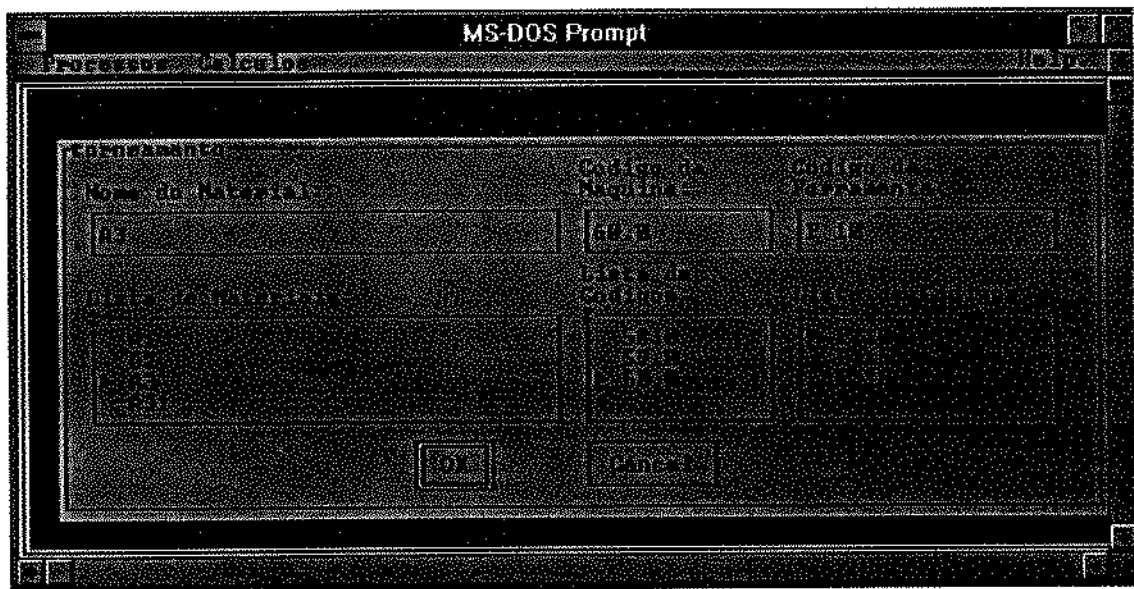
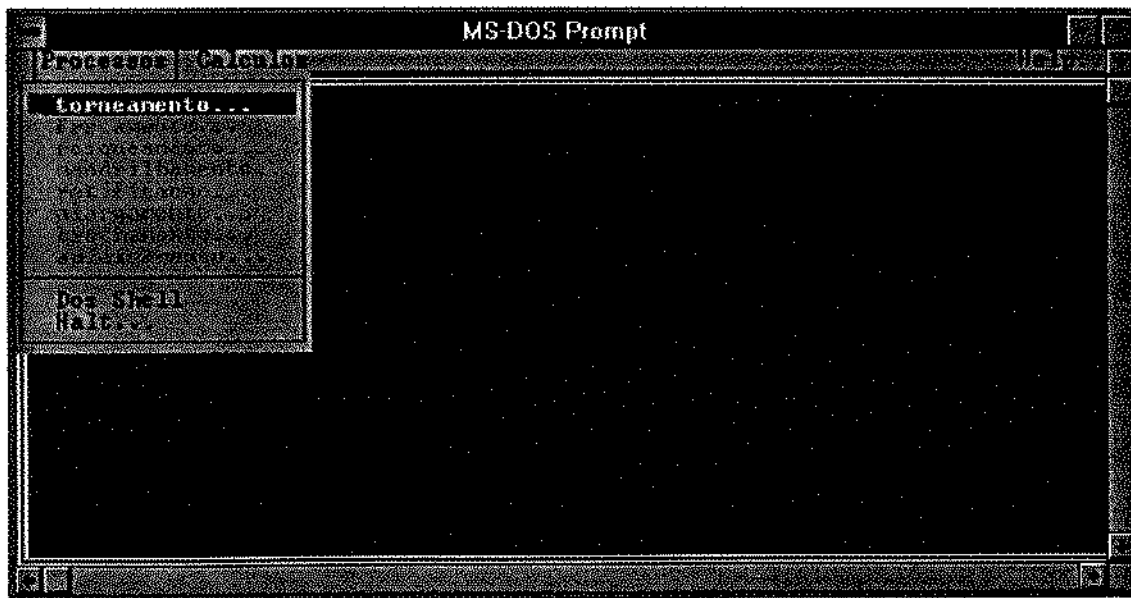
(Ramalho 1988)

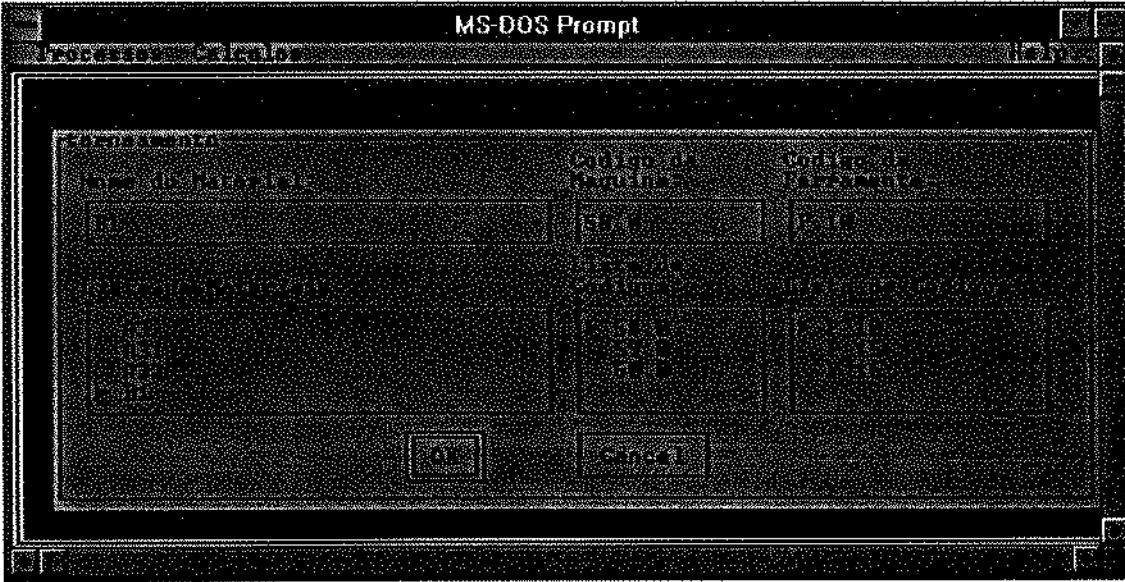
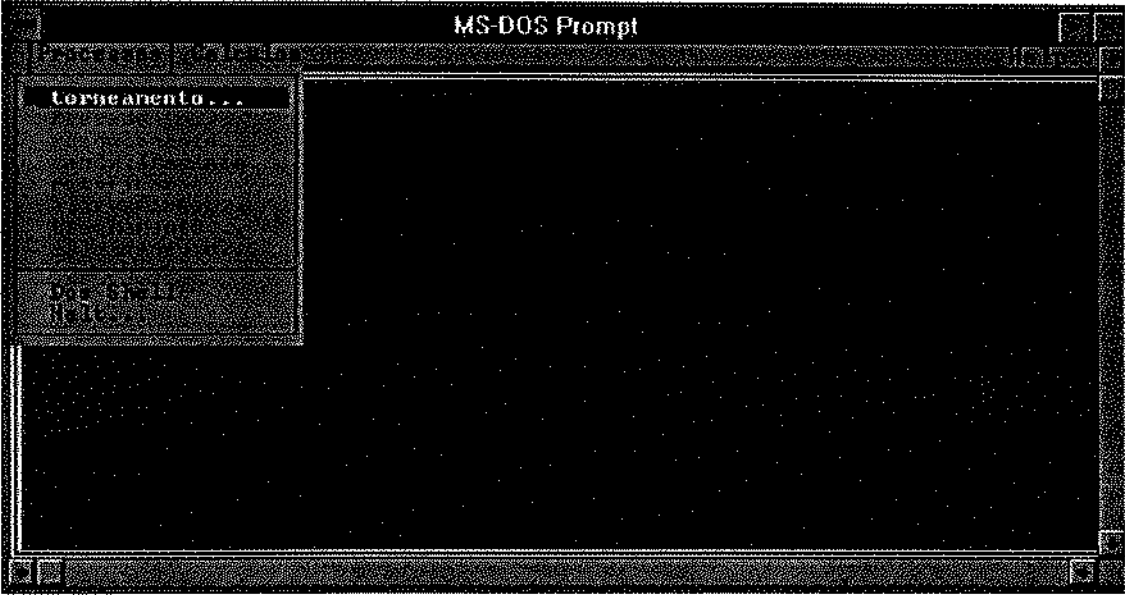
(Consentino 1987)

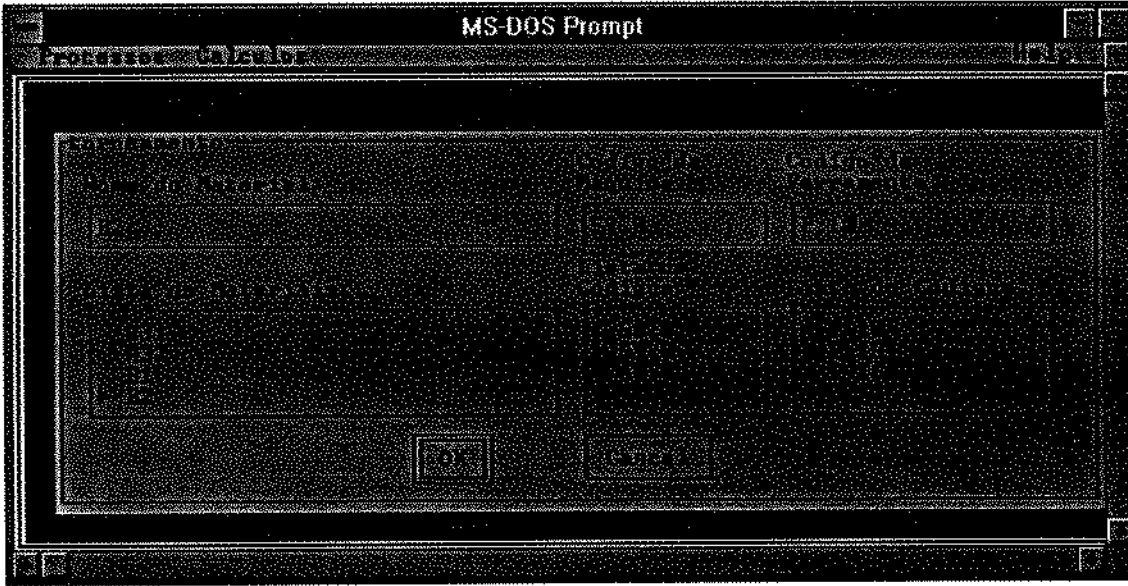
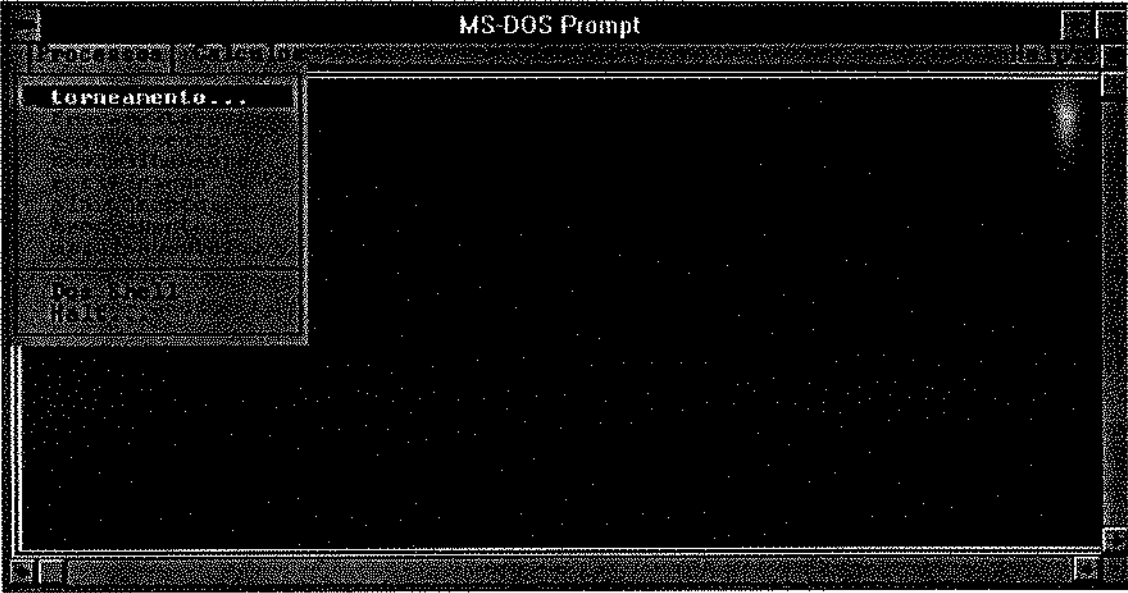
O dBASE III, cuja denominação significa "data base" ou base de dados, permite criar, manipular e gerenciar Bases de Dados de pequeno e médio porte utilizando microcomputadores da linha PC XT/AT, baseados em discos magnéticos. Através desses softwares pode-se criar e utilizar rapidamente arquivos de dados das mais variadas naturezas, conforme sua necessidade de gerar informações.

Utilizando o CLIPPER/dBASE III é possível:

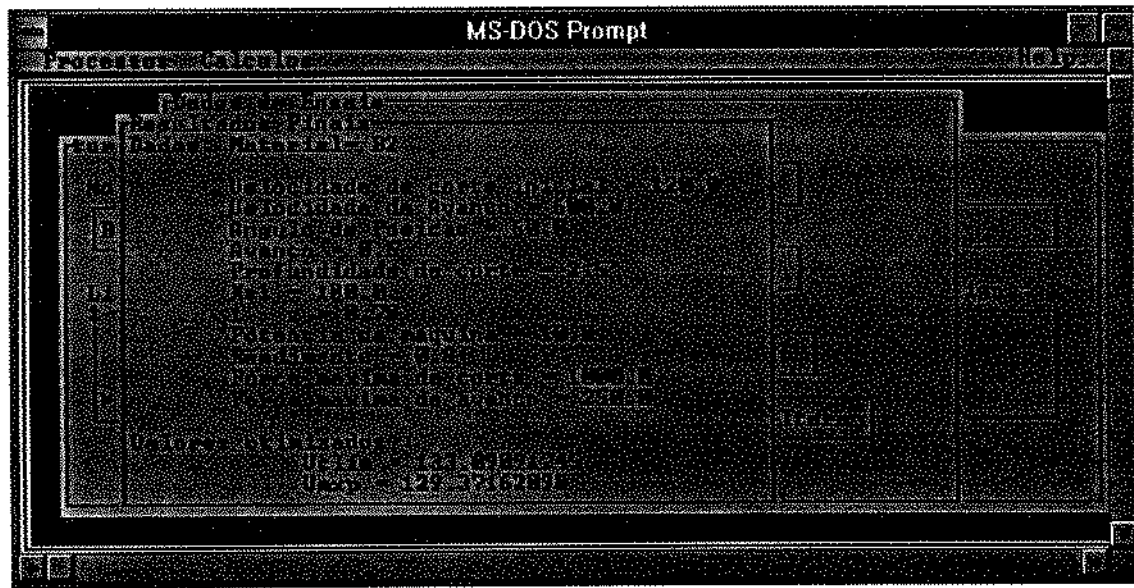
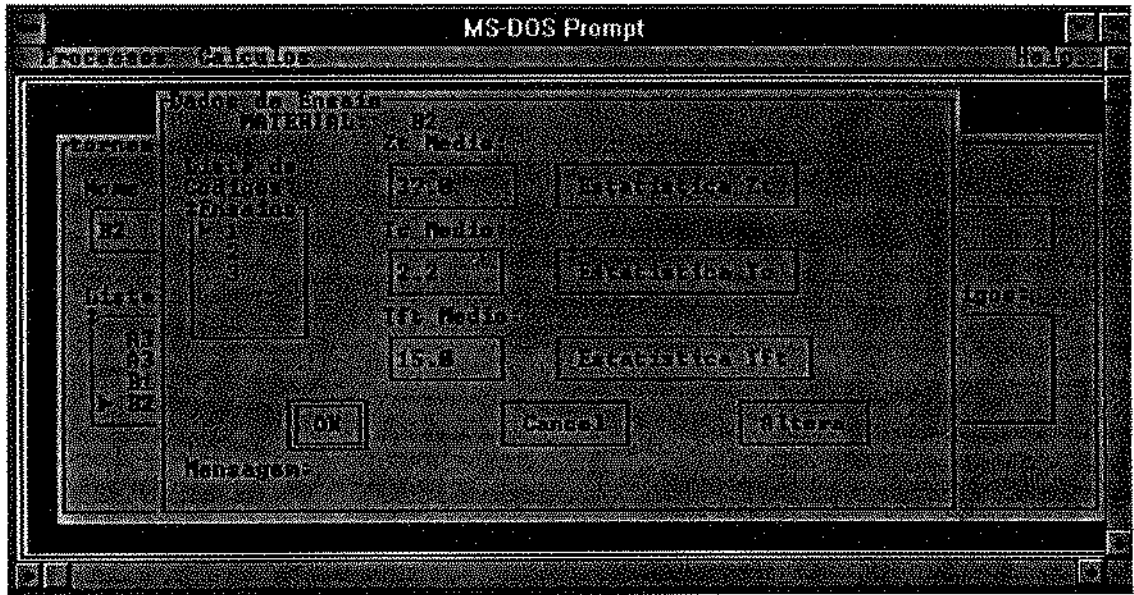


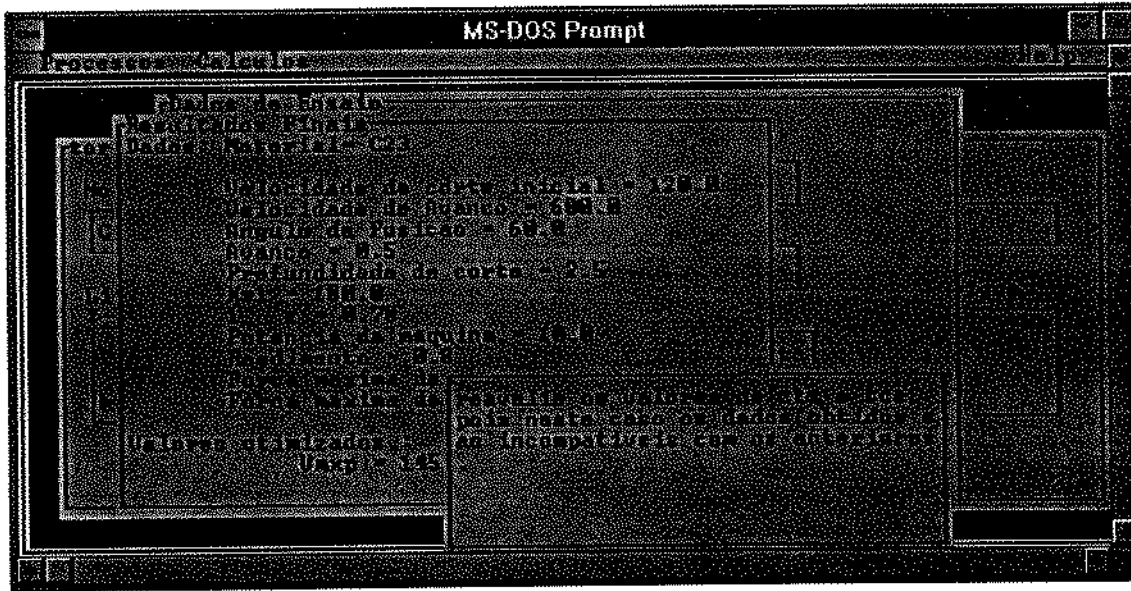
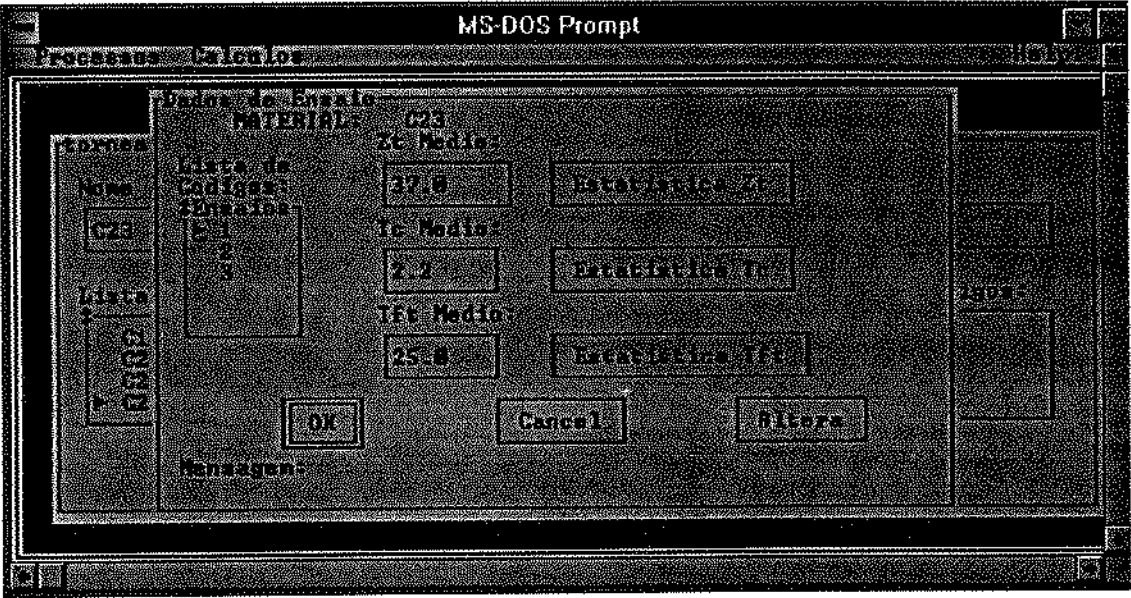






+





- . criar, organizar, classificar, copiar, selecionar e relacionar conjuntos de arquivos que formam a Base de Dados;
- . adicionar, alterar, eliminar, exibir e listar global ou seletivamente as informações contidas nos arquivos de dados;
- . gerar relatórios padronizados, efetuar automaticamente somas, agregações, contagens e operações aritméticas sobre os valores dos dados armazenados nos arquivos;
- . formatar telas na entrada de dados no vídeo e gerar relatórios, tabelas e listagens complexas na impressora, de acordo com as necessidades do usuário;

Seus arquivos são conjuntos de registros altamente estruturais, podendo ser recuperados rapidamente por "campos-chave" que os identifiquem. Os registros de um mesmo arquivo possuem sempre o mesmo formato e são, em geral, armazenados em ordem cronológica de inclusão no arquivo. As técnicas de recuperação e classificação de registros baseiam-se na utilização de campos especiais contidos no registro, referidos como campos-chave. Os campos armazenam os dados ou atributos descritores de eventos ou entidades.

Os arquivos de dados de extensão .DBF, gerados pelo CLIPPER/dBASE III, contém a estrutura dos registros e dos campos, e os dados de cada registro. Podem ser criados através dos utilitários CREATE.exe ou DBU.exe e dos comandos CREATE e CREATE FROM.

Partes dos arquivos .DBF do SEU fazem parte do Apêndice, sendo que:

. O Apêndice 1 é relativo ao arquivo PEÇA, sendo que o arquivo peça é composto pelos campos:

Codigo: campo "chave" que identifica a peça,
Material: material da peça,
Dureza_hb: dureza da peça,
Rtracaompa: resistência à tração,
Vc: velocidade de corte inicial,
Quant_Peca: quantidade de peças usinadas,
Naresta_Vid: número de arestas por vida,
Vf: velocidade de avanço.

O arquivo PEÇA é acessado pela máquina de inferência através do seguinte comando:

```
retrieveb(peca,Material,peca(Codigo,Material,Dureza_hb,  
Rtracaompa,Vc,Quant_Peca,Naresta_Vid,Vf)).
```

. O Apêndice 2 é relativo ao arquivo DADOMAQ, sendo que o arquivo dadomaq é composto pelos campos:

Codigo_Maq: campo "chave" que identifica a máquina,
Nm: potência nominal,
Neta: rendimento do sistema,
Potencutil: potência de corte útil.

O arquivo DADOMAQ é acessado pela máquina de inferência através do seguinte comando:

```
retrieveb(dadomaq,Codigo_Maq,dadomaq(Codigo_Maq,Nm,Neta,
Potencutil)).
```

. O Apêndice 3 é relativo ao arquivo DADOENSA, sendo que o arquivo dadoensa é composto pelos campos:

Codensaio: campo "chave" que identifica o ensaio,
CodPeca: código que identifica a peça,
ZtMedio: número médio de peças usinadas por vida
em uma determinada condição (I),
VcMedio: velocidade de corte na condição (I),
TcMedio: tempo de corte na condição (I),
TftMedio: tempo médio de troca de ferramentas.

O arquivo DADOENSA é acessado pela máquina de inferência através do seguinte comando:

```
retrieveb(dadoensa,CodEnsaio,dadoensa(CodEnsaio,CodPeca,
ZtMedio,TcMedio,TftMedio)).
```

. O Apêndice 4 é relativo ao arquivo FERRAM, sendo que o arquivo ferram é composto pelos campos:

Codigo_Fer: campo "chave" que identifica a ferramenta,
Xr: ângulo de posição,
Cvidaresta: vida da aresta

O arquivo FERRAM é acessado pela máquina de inferência através do seguinte comando:

```
retrieveb(ferram,Codigo_Fer,ferram(Codigo_Fer,Xr,_,  
Cvidaresta)).
```

. O Apêndice 5 é relativo ao arquivo DADOENTR, sendo que o arquivo dadoentr é composto pelos campos:

Cod_Peca: campo "chave" que identifica a peça,
I_Z e Ks1: coeficientes de Kienzle,
F: avanço,
Vc_Limite: velocidade de corte máxima para a potência
de corte útil e para avanço e profundidade
de usinagem escolhidas,
Forcacorte: força de corte,
Ap: profundidade de usinagem,
Pcmax: força de corte máxima,
Pamax: força de avanço máxima.

O arquivo DADOENTR é acessado pela máquina de inferência através do seguinte comando:

```
retrieveb(dadoentr,Cod_Peca,dadoentr(Codigo_Peca,I_Z,Ks1,  
F,Ap,Pcmax,Pamax)).
```

. Existe ainda o arquivo DADOCALC, no qual ocorre o armazenamento dos valores otimizados pelo SEU, isto é x e K de Taylor ótimos com sua respectiva V_{max} .

O dBASE III pode ser classificado como um Sistema de Gerenciamento de Bancos de Dados do tipo relacional. A organização relacional é uma das mais naturais e flexíveis do ponto de vista do usuário, pois trata os dados como se estivessem

organizados em relações ou tabelas. Nestas tabelas, as colunas são identificadas por nomes descritivos (os campos) e as linhas são os registros, onde são armazenados os dados, desta forma elas equivalem às fichas de um arquivo tradicional.

Nesta estrutura de arquivos, um arquivo é, portanto, um conjunto de registros do mesmo tipo e, por sua vez, cada registro é formado por um conjunto de campos onde estão armazenados os dados que constituem a informação desejada.

Os registros de um arquivo são numerados sequencialmente, de acordo com a ordem cronológica de sua inclusão no arquivo.

Os nomes dos campos dos arquivos podem conter letras, números ou o símbolo sublinhado "_". A primeira posição do nome de um campo deve ser obrigatoriamente uma letra. Os nomes dos campos podem possuir até 10 caracteres, mas não podem possuir espaços em branco entre eles.

Para se formar os registros dos arquivos de dados podem ser definidos os seguintes tipos de campos:

- **Caractere:** quando os dados a serem armazenados possuírem, na sua composição, caracteres numéricos (0-9), caracteres alfabéticos (a-z ou A-Z) ou especiais (+, *, %, \$).
- **Numérico:** quando os dados a serem nele armazenados possuírem, na sua composição, somente caracteres numéricos, os sinais + ou - e o ponto decimal. Estes campos destinam-se ao armazenamento de dados com os quais serão realizados cálculos matemáticos.

- **Lógico:** quando os dados a serem armazenados representarem apenas dois estados possíveis, refletindo as condições de sim ou não; falso ou verdadeiro.
- **Data:** quando os dados armazenados possuírem, na sua composição, 8 caracteres com números e a barra "/" no formato 99/99/99. Este campo armazena apenas datas; permite a realização de uma série de operações com elas.
- **Memo:** um campo memo pode conter textos longos, como por exemplo observações ou comentários, específicos para cada registro do arquivo de dados. Estes textos serão armazenados em um arquivo separado identificado pelo mesmo nome do arquivo de dados a ele associado e pela extensão (.DBT).

Basicamente, um SGBD deve conter as seguintes rotinas para manutenção de arquivos de dados: inclusão, alteração, exclusão e busca de registros. O dBASE III nos auxilia na implementação dessas rotinas através dos seguintes comandos:

inclusão: APPEND BLANK, APPEND FROM;

alteração: REPLACE;

exclusão: DELETE, PACK, RECALL, ZAP;

posicionamento nos registros: GO/GOTO, GO BOTTOM, GO TOP, SKIP;

pesquisa em arquivo: LOCATE, FIND, SEEK

6.3.2. Linguagem Prolog (Bratko 1986)
(Clocksin e Mellish 1987)
(Marcus 1986)

Em 1970, pesquisadores da França (Roussel e Colmerauer) e da Inglaterra (Edinburgh Kowalski) confeccionaram grande parte de tudo o que, atualmente, se conhece da linguagem PROLOG.

Em 1977, John Backus (Lazarev 1987), o principal desenvolvedor das linguagens FORTRAN e ALGOL 60, apresentou à comunidade de informática uma análise crítica das linguagens de programação convencionais em seu Turing Award Lecture: "Can Programming be Liberated from the Von Newman Style?". Backus caracterizou as linguagens convencionais como "amplas, complexas e inflexíveis", e com isso obteve-se o incentivo à utilização de linguagens de programação não convencionais, as linguagens inteligentes.

Através desta linguagem, podemos implementar programas na forma de lógica simbólica, utilizando um processo de inferência para produzir resultados. Na programação em lógica, os programas não estabelecem como um resultado é computado, descrevem a forma do resultado. Um programa feito nesta linguagem deve possuir uma base, onde se realiza o cálculo de predicados, e proposições, que são afirmações sobre um objeto/conceito e seus relacionamentos.

Os programas PROLOG são escritos em sentenças de Horn, um jargão criado pelo matemático Alfred Horn para exprimir com precisão axiomas em lógica. As sentenças de Horn são constituídas por relações entre os objetos e divididas em afirmações e regras (Costa e Gurgel 1985).

Basicamente, PROLOG chega à dedução das soluções dos problemas através do raciocínio. Programas Prolog são compostos de um conjunto de cláusulas, as quais podem ser fatos ou regras.

Outra característica importante do PROLOG é o seu processo de inferência, que consiste de um "match" (atribuição) de uma dada meta com a base de dados: o sistema parte da meta ("backward chaining") com busca em profundidade (procura e encontra um caminho completo de "matches" para a 1ª submeta, antes de trabalhar as outras).

Os três conceitos fundamentais de programação em PROLOG são:

- . Satisfação: sucesso;
- . Falha: para forçar a ocorrência de um retrocesso na busca de solução, e
- . Unificação, com os seguintes critérios:
 - .. duas constantes são iguais se são escritas do mesmo jeito;
 - .. dois números são iguais se representam o mesmo valor;

- .. quando uma variável é comparada com um objeto qualquer, ela se torna igual a este objeto, diz-se neste caso, que a variável está representando o objeto;
- .. duas estruturas são iguais se possuem o mesmo funtor (nome do predicado) e a mesma aridade (número de argumentos).

6.3.2.1. Elementos Básicos da Linguagem PROLOG

- **átomo:** cadeia de caracteres começando com letra minúscula, é o tipo de palavra (objeto) mais comum.

Por exemplo: vmxp, v1, dados.

- **variável:** cadeia de caracteres começando com letra maiúscula, usadas para representar elementos desconhecidos ou genéricos.

Por exemplo: Vmxp, V1, X_Taylor, K_Taylor.

- **predicado:** é uma construção usada para declarar algo a respeito dos objetos. Em PROLOG, a declaração é representada por um átomo e é seguida pelos objetos (argumentos) que devem ser colocados entre parenteses e estar separados uns dos outros por vírgulas.

Por exemplo: valores_Taylor(X,K).

- **estrutura:** termo composto que representa as proposições atômicas.

Esquemáticamente, temos:

<nome da função>(<lista de argumentos>), onde:

nome da função ou funtor é invariavelmente um átomo e serve para identificar a estrutura;

lista de argumentos são usadas para nomear as partes do objeto composto. Eles podem ser átomos, variáveis ou, mesmo outras estruturas.

Por exemplo:

calculo_vmxp(valores_Taylor(X,K),Tft,Vmxp).

- **fato:** expressa, às vezes, a veracidade sobre a informação dada, é considerada uma cláusula unitária.

Por exemplo:

atribuir_a(Fnovo,Nap/4).

Formalmente as estruturas são idênticas aos predicados. Os predicados são estruturas declarando sentenças que podem ser Verdadeiras(V) ou Falsas(F). Todo predicado é uma estrutura, mas a volta não é verdadeira, isto é, toda estrutura é um predicado porque há estruturas que estão apenas nomeando objetos e, portanto não fazem nenhuma declaração que possa ser considerada V ou F.

- **regra:** expressa como novos fatos devem ser inferenciados (deduzidos), uma regra expressa o relacionamento entre os fatos, é considerada uma cláusula não-unitária, isto é:

conclusão $\Leftarrow$ condições.

Por exemplo:

```
possibilidades_vmxp(Vmxp1,Vmxp2,V1,V2,Vclim):-
    (Vmxp1 < V2;Vmxp1 = V2),
    (Vmxp1 > V1;Vmxp1 = V1),
    write('Vmxp1 = '),
    write(Vmxp1),
    write(' é a solução'),nl.
```

Regras em Prolog são baseadas nas cláusulas de Horn, que consistem de um lado esquerdo, um separador ":-" e um lado direito. As cláusulas de Horn simplesmente afirmam que o lado esquerdo é verdadeiro se todos os fatos do lado direito forem verdadeiros.

- **metas:** proposições que o sistema deve provar.

- **listas:** (Boises 1987) tem uma estrutura uniforme que pode ser facilmente decomposta através de técnicas recursivas básicas e seu tamanho pode ser arbitrário.

Listas são entidades de dados específicas; uma lista é uma sequência de elementos ordenada em uma ordem significativa. Átomos, inteiros ou outras listas podem ser elementos de uma lista, sendo que devem ser colocados entre colchetes e estarem separados uns dos outros por vírgulas.

Representação genérica de listas:

$[X / Y]$, onde:

X é a cabeça da lista, isto é, o primeiro elemento da sequência;

Y é a cauda da lista (lista menos a cabeça).

6.3.2.2. Tipos de Dados Prolog

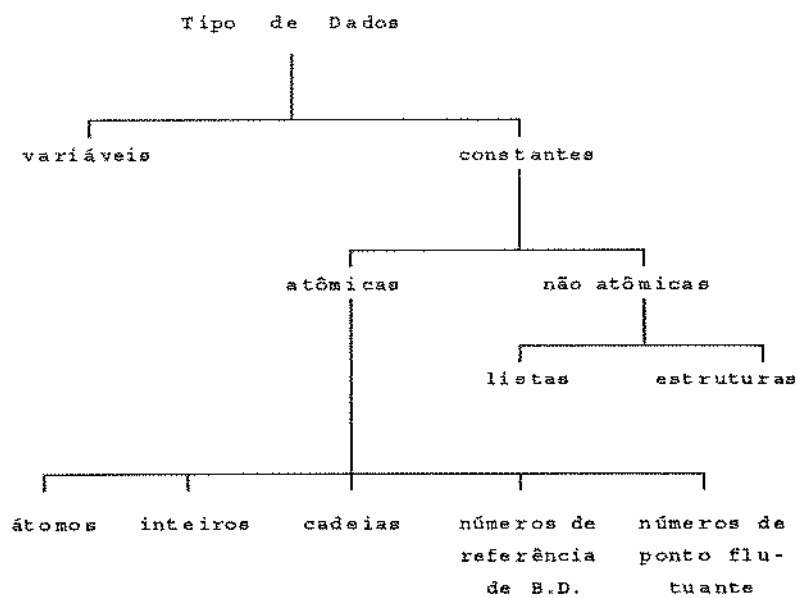


Figura 13: Esquema de Tipos de Dados Prolog

6.3.3. Características da Interface

O interfaceamento entre arquivos criados em dBASE III (de extensão .DBF) e a máquina de inferência, codificada em PROLOG, foi realizado por duas rotinas, implementadas em PROLOG: `build_tree` e `store_dbf`. A finalidade da interface consiste em solucionar o problema da comunicação (transmissão e recepção de dados) entre as duas entidades mencionadas (máquina de inferência e arquivos CLIPPER/dBASE III).

Num arquivo criado em dBASE III, o acesso a um determinado registro é feito através do algoritmo de busca binária, implementado no respectivo arquivo de índice (criado pelo próprio dBASE para aumentar a rapidez de acesso aos registros), sendo este ordenado por um ou mais campos-chave (Date 1988) (Korth e Silberschatz 1989).

O predicado "`build_tree`" é responsável pelo protocolo de recepção de dados, transforma cada registro de um arquivo CLIPPER/dBASE III em uma estrutura (cada elemento desta estrutura é um campo), armazenando-a numa estrutura de dados denominada árvore-B; para tal foram utilizados os predicados "`get_header`" e "`get_db_record`".

A utilização de árvore-B possibilita a indexação de dados numa base de dados. Esse tipo de estrutura de dados é chamada de árvore-B, devido ao fato da estrutura dos índices poder ser visualizada como "galhos numa árvore", sendo que os seus elementos são chamados de nós. Toda árvore-B é balanceada, isto é, cada ponta de galho (folha) da árvore está tão longe do elemento do topo quanto qualquer outra folha. A busca inicia-se a

partir do nó do topo (raiz). No Arity/Prolog, cada nó pode ter no máximo 127 galhos, e cada nó contém a informação necessária para que a busca prossiga no caminho correto. O último nível alcançado durante a busca é formado exclusivamente por folhas, as quais são os termos do Arity/Prolog que são armazenados na árvore-B.

Buscas envolvendo termos armazenados em árvore-B geralmente são mais rápidas do que buscas sequenciais, por fazerem menos comparações.

Assim, parte da comunicação entre o Banco de Dados e a Máquina de Inferência foi estabelecida aplicando-se o módulo de "conversão direta" sobre os arquivos PEÇA.DBF, DADOMAQ.DBF, FERRAM.DBF, DADOENTR.DBF, DADOENSA.DBF e DADOCALC.DBF, ocorrendo a transferência de dados destes arquivos para diferentes árvores-B, sendo que cada árvore-B corresponde a um arquivo diferente.

O problema do protocolo de comunicação (transmissão e recepção) foi resolvido através da utilização dos seguintes predicados: `get_header`, `get_db_record`, `put_header` e `put_db_record`.

A estrutura de arquivos dBASE inicia-se com um cabeçalho (header) e os registros são posicionados imediatamente depois. O cabeçalho fornece informações que serão úteis no processo de leitura dos registros: número da versão, data, número de registros, comprimento do cabeçalho (em bytes), comprimento de cada registro (em bytes), bytes reservados e as definições de cada campo. Os registros são em formato fixo, sem separadores de campo, nem terminadores de registro.

O predicado `get_header(Handle,Header,Format)` lê o cabeçalho do arquivo .DBF cujo apontador é o "Handle"; "Format" é uma lista com o tipo de dado de cada campo; e "Header" conterá as informações do cabeçalho do arquivo. O predicado "get-header" será executado após a abertura de um arquivo para leitura e antes de que qualquer arquivo seja lido.

O predicado `get_db_record(Handle,Format,Record)` acessa o próximo registro do arquivo dBASE cujo apontador é "Handle"; "Format" é uma lista dos tipos de dados que descrevem cada campo de um registro e "Record" é o registro retornado.

O predicado `put_header(Handle,header(Version,Date,NumRec,Hlen,Rlen,Reserve,Fields),Format)` escreve um cabeçalho de arquivo dBASE no arquivo especificado pelo apontador "Handle"; "Format" é uma lista do tipo de dado para cada campo no registro. esta lista é gerada automaticamente pelo sistema e é utilizada por "put_db_record". Neste cabeçalho, "Version" é a especificação do número da versão, isto é, 03H sem um arquivo .DBT e 83H com um arquivo .DBT; "Date" é a data da última atualização em formato Prolog (`date(Year,Month,Day)`). "NumRec" é o número de registros presentes no arquivo. Este número pode ser um inteiro ou um float (ponto flutuante); "Hlen" é o comprimento do cabeçalho, e pode estar ou não instanciado (caso não esteja instanciado, Hlen será calculado, e se estiver instanciado ele será checado); "Rlen" é o comprimento do registro; "Reserve" é uma cadeia com 20 caracteres reservados e "Fields" é uma lista de descritores de campo. O formato de um descritor de campo é apresentado a seguir:

`field(Name,Type,Address,Length,DecimalCount,Reserved)`, onde:

Name: cadeia de no máximo 11 caracteres;

Type: tipo do dado do campo, que pode ser n (numérico), c (cadeia de caracteres), l (lógico booleano: true/false), m (memo) e (d) data;

Address: é um inteiro que é armazenado como ponto flutuante;

Length: comprimento do campo em bytes;

DecimalCount: número de dígitos decimais;

Reserved: cadeia de 14 caracteres reservados.

O predicado `put_db_record (Handle, Format, record (Deleteq, Data))` coloca um registro num arquivo DBASE cujo apontador é "Handle"; "Deleteq" pode ser `deleted` se o registro está correntemente eliminado, ou pode ser `not_deleted` se o registro está correntemente ativo. "Data" é uma lista dos dados num registro, campo por campo. Registros são armazenados sequencialmente após o armazenamento do cabeçalho, e após o último registro ser armazenado, o byte 1AH (EOF) deve ser escrito no arquivo fechando-se então o arquivo.

A outra parte da comunicação foi realizada através do módulo de "conversão inversa". Este módulo constitui-se do predicado "store_dbf", sendo responsável pelo protocolo de transmissão de dados, isto é, converte estruturas mantidas em árvore-B em registros com formatos estruturados em CLIPPER/dBASE III. Este módulo foi aplicado após o processamento da máquina de inferência, constituindo a última etapa do programa. Neste módulo, utilizou-se os predicados "get_header", "put_header" e "put_db_record". O predicado "store_dbf" foi implementado, principalmente com o objetivo de armazenar os valores otimizados produzidos pelo SEU, num arquivo em formato .DBF.

CAPÍTULO 7

CÁLCULOS ESTATÍSTICOS

7.1. Dados de Ensaio

Todas as vezes que recorreremos aos ensaios experimentais para analisar uma característica física ou tecnológica de uma população, ou ainda quando comparamos essa população com outras para quantificar uma determinada dependência fenomenológica, deparemos com as seguintes perguntas:

- Quantas vezes devemos repetir o experimento para garantir que a amostra é representativa da população?

- Qual a confiança que temos na comparação de diferentes populações através de suas amostras?

O objetivo é buscar na estatística as ferramentas necessárias para responder tais perguntas com o auxílio do computador.

7.2. Estimativa da Grandeza da Amostra

(Box et al. 1987),

(Franco et al. 1983)

Deseja-se analisar uma população, quanto a uma determinada característica com distribuição normal nessa população. Se a população for pequena, é possível medir essa variável em cada

membro da população, obtendo-se assim a média e a dispersão (Desvio Padrão) dessa característica na população. Porém, para grandes populações e mesmo populações infinitas, fica impraticável tal procedimento e, nesses casos, analisa-se a população através de uma amostra. Assim sendo, a média e o desvio padrão da população são estimados pela média e pelo desvio padrão da amostra.

A amostra deve ser representativa dentro da população e seu tamanho deve ser o menor possível, sem que comprometa a representatividade. Para calcular-se esse tamanho mínimo de amostra, é necessário antes definir qual o erro máximo que é tolerado e qual o risco que pode ser assumido.

O erro máximo tolerável, é o erro que aceita-se cometer quando estima-se a média da população igual à média da amostra; ou seja, se a média da amostra for X , aceita-se que a média da população pode ser $(X \pm \text{erro tolerável})$. O risco assumido é a probabilidade de que a média da população esteja fora do intervalo tolerado de $(X \pm \text{erro})$.

Intuitivamente, percebe-se que:

. quanto menor for o erro tolerado e menor for o risco assumido, maior será o tamanho da amostra para assegurar uma melhor representatividade. Outro fator que influencia o tamanho mínimo de amostra é a dispersão da população ou seja, o desvio padrão da amostra.

. quanto menor for o desvio padrão, mais "comportada" é a população e, portanto, menor será a amostra necessária para garantir a representatividade exigida.

Assim sendo, para encontrar o tamanho mínimo da amostra é necessário antes obter uma amostra inicial de tamanho N_0 , julgada conveniente para estimar a dispersão da população. Em seguida, calcula-se o valor do tamanho mínimo de amostra N .

Se $N_0 > N$ então a amostra inicial é representativa da população;

Senão é necessário tomar uma amostra maior N_0 , tal que, $N_0 > N$, que garanta a representatividade exigida.

Para populações contínuas, o tamanho mínimo da amostra é definido por:

$$N = \left(\frac{S \cdot t}{d} \right)^2$$

onde:

S = estimativa do desvio padrão da população ou desvio padrão da amostra inicial;

d = erro tolerável;

t = valor obtido na tabela de distribuição "t" de Student para γ graus de liberdade ($\gamma = N_0 - 1$) e confiança P ($P = 1 - \alpha$), e α é o risco assumido.

A tabela "t" de Student é uma tabela de distribuição normal modificada que permite trabalhar com as estimativas da média e do desvio padrão da população.

Essa tabela fornece os valores de "t" para níveis de confiança indo de 55 a 99,5% e diversos graus de liberdade (1 a 900).

A partir dos dados da amostra mínima, calcula-se a média e o desvio padrão da população, dentro da margem de segurança pré-estabelecida.

Com a finalidade de tornar a análise de dados de usinagem do SEU mais completa e confiável, foi implementado em PROLOG um módulo que manipula informações contidas num arquivo de dados de ensaio, produzindo resultados, como a estimativa das médias de populações e o desvio padrão, sendo que esses valores podem ser utilizados "on line" ou armazenados em um arquivo para posterior manipulação do SEU.

O arquivo de dados de ensaio é constituído por conjuntos de vários campos, cada conjunto representando um tipo diferente de população:

- . população 1 - é composta por um número limitado de campos, campos estes que contém o número de peças usinadas por vida em uma determinada máquina;
- . população 2 - é composta por um número limitado de campos, que contém o intervalo de tempo que a máquina demorou para usinar o número de peças especificado

anteriormente, no mesmo registro, pelo respectivo campo da população 1.

Este módulo é acessado pelo usuário via interface gráfica através da janela inicial (Figura 7) fornecida pelo sistema, onde o usuário poderá optar pela opção Cálculos clicando <ENTER> sobre ele ou digitando somente a palavra C (Figura 14). Após este procedimento ser executado, será fornecido ao usuário uma nova janela, onde pode-se optar pela realização de cálculos estatísticos para Z_t , t_c ou para t_{ft} , e após esta escolha será fornecida nova tela (Figura 15), onde:

. O usuário deverá digitar os valores de Z_t , t_c ou t_{ft} sendo esses valores mostrados ao usuário através de uma "list-box" com capacidade para cinco informações, sendo que quando houver mais do que cinco itens ocorrerá uma "navegação" vertical nos itens. Será fornecido no rodapé da janela um alerta que poderá ser quaisquer uma dessas duas mensagens:

"Entre com os valores de Z_t , t_c ou t_{ft} !" , ou

"Adicione mais dados de Z_t , t_c ou t_{ft} !" ;

. Será apresentado ao usuário uma tabela fixa de confianças onde ele deverá escolher somente uma delas, sendo que sua escolha será mostrada acima da lista mencionada através de uma "radio list box";

. O usuário terá três "push buttons", sendo que através deles ele terá as possibilidades de execução dos cálculos estatísticos clicando sobre o "button" OK, cancelar a execução desses cálculos, ou ainda deletar algum valor digitado;

. Os resultados da Média e do Desvio Padrão serão fornecidos na própria janela;

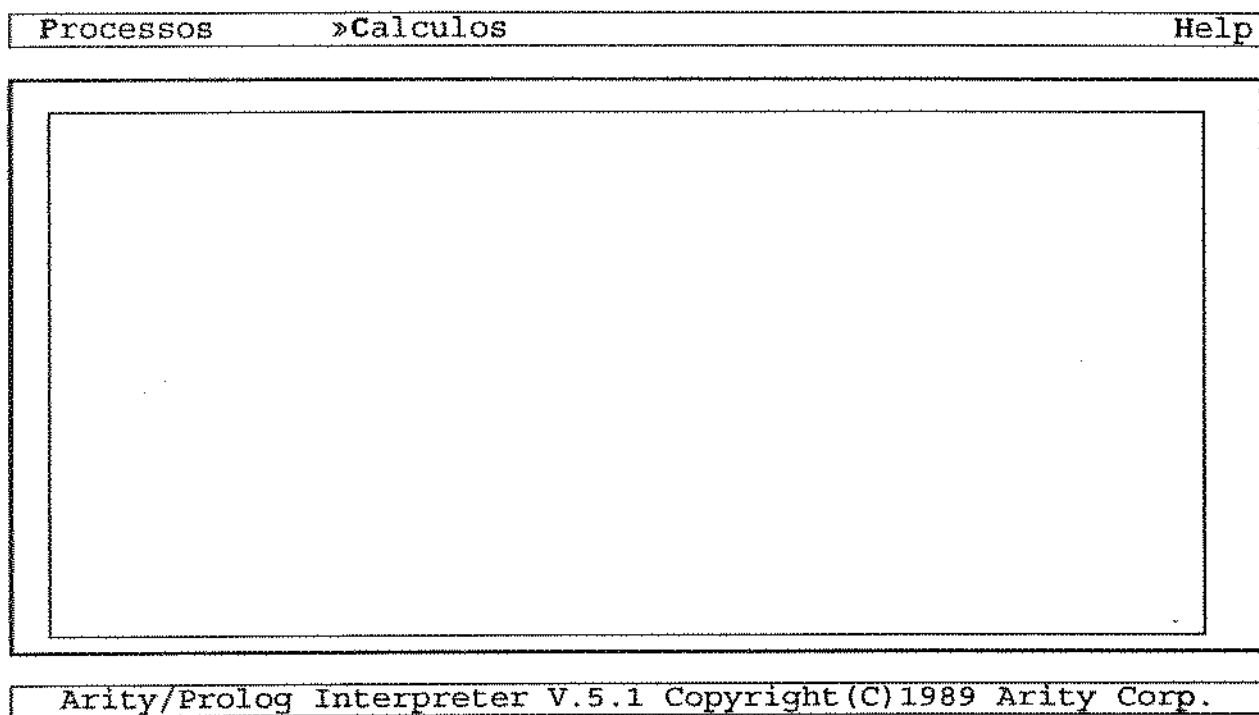


Figura 14: Interface com o Usuário - Tela 2.1

Valores de Zt/tc/tft:		Valores da Confianca:	99.95
Lista Zts/tcs/tfts:		Confianças:	»99.95
			99.90
			99.75
		Erro Tolerado(%):	1
	OK	Cancela	Deleta
Media:		Desvio Padrao:	
Mensagem: Entre com os valores de Zt ou tc!			

Figura 15: Interface com o usuário - Tela 2.2

CAPÍTULO 8

CONCLUSÕES

Com o desenvolvimento deste trabalho, verificou-se que:

1. A possibilidade de otimização das condições operacionais de usinagem, em termos de utilização mais eficiente de máquina-ferramenta e ferramenta.

2. O SEU permite dispensar o especialista em usinagem de cálculos e procedimentos complexos contidos no algoritmo matemático e computacional utilizados, liberando-o para usar seus conhecimentos com outros procedimentos.

3. A linguagem PROLOG mostrou-se adequada para o desenvolvimento do SEU, inclusive no momento do interfaceamento entre a Máquina de Inferência e o Banco de Dados.

4. O SEU pode ser aplicado para sistemas semi-automáticos.

5. No caso de automação plena (CAO - Controle Adaptativo Otimizado), algumas dificuldades deverão ser resolvidas, não referentes ao modelo proposto, mas sim relacionadas com problemas de sensores para o monitoramento e de comunicação entre SEU e o sistema de programação da máquina ferramenta. A comunicação mencionada concentra-se na troca da velocidade de corte toda vez que o modelo exigir.

6. Consegue-se tirar vantagens do conhecimento atualmente existente sobre usinagem através da utilização de softwares.

7. A utilização de regras de produção no sistema possibilita:

7.1. Uma maior modularidade, isto é, as produções são individualizadas e podem ser adicionadas, retiradas ou alteradas independentemente uma das outras.

7.2. Uma maior uniformidade na representação do conhecimento devido à rigidez de codificação das informações em regras de produção, tornando o sistema de fácil entendimento por outras pessoas com conhecimentos em PROLOG.

8. O sistema reduz significativamente o tempo requerido para obter as informações desejadas. A redução percebida é aquela existente entre o tempo transcorrido para que o especialista execute os cálculos e a execução do programa após simples digitação de dados, a qual ocorre "on line".

9. O sistema poderá ser usado com sucesso por uma grande variedade de usuários finais, independente de serem especialistas em usinagem, ocorrendo dessa forma uma maior disponibilidade do conhecimento especializado.

10. A praticidade do SEU é evidenciada ao longo das conclusões anteriores às quais basta ser aditada a realidade de que microcomputadores são utilizados sem problemas em ambientes fabris.

Trabalhos futuros:

- . Automação do sistema;
- . Inclusão do cálculo de custo;
- . Implementação do sistema para outras operações;
- . Desenvolver SE acoplado ao CAPP.

BIBLIOGRAFIA CITADA

- (Antunes 1993) Antunes, A., "Inteligência Artificial: Sistemas Especialistas, Conhecimento, e mais... ", Makron Books Informa, Makron Books, Fev/1993, n° 56, pag. 3.
- (Arariboia 1988) Arariboia, G., Inteligência Artificial: Um Curso Prático, LTC, 1988.
- (Barborak et al. 1991) Barborak, D.M., "PC - Based Expert System and their Application to Welding", Welding, pags 29-38, Jan. 1991.
- (Barnes 1977) Barnes, R.M., Estudo de Movimentos e de Tempos: projeto e medida do trabalho, Editora Edgard Blucher Ltda, 1977.
- (Batocchio 1991) Batocchio, A., "Um Modelo de Índice de Automação Relacionado à Flexibilidade e à Produtividade dos Sistemas de Manufatura", Tese de Doutorado, UNICAMP, Agosto 1991.
- (Bellinger 1988) Bellinger, J.G., "The Factory of the Future: Technological Aspects", Annals of CIRP, Vol. 37, pags 551-552, Feb 1988.
- (Boises 1987) Boises, S., "Language Processing - Using Definite Clause Grammars", AI Expert, Vol.2, N° 6, Pags 46-56, June 1987.
- (Box et al. 1987) Box, G.E.P. et al., Statistics for Experimenters, John Willey & Sons, 1987.

(Bratko 1986) Bratko, I., Prolog Programming for Artificial Intelligence, 1986.

(Campos 1988) Campos, M.B., "Um molde para Sistemas Especialistas", PC-Mundo, dez 1988, pags 40-42.

(Campos 1989) Campos, S., "IBM Promove Discussão sobre o Futuro de Banco de Dados", Datanews, 12/1989.

(Carvalho 1993) Carvalho, L.R.M., "Uma estratégia para automação", Automação&Indústria, Ano V, N° 45, 1993.

(Caulliraux 1989) Caulliraux, H.B.; Alvarenga, M.A.B., "Robótica: Opção Nacional", Boletim Sobracon n° 45, pags 41-58, 1989.

(Chorafas 1988) Chorafas, D.N., Sistemas Especialistas: Aplicações Comerciais, McGraw Hill, 1988.

(Clocksin 1987) Clocksin, W.F.; Mellish, C.S., Programming in Prolog, 3ª ed., Springer-Verlag, 1987.

(Colding 1992) Colding, B.N., "A boa escolha de parâmetros de Usinagem leva a muitos ganhos", Máquinas e Metais, Aranda Editora Técnica Ltda, pags 86-94, Fev 1992.

(Consentino 1987) Consentino, L.J.L., dBASEIII Interativo, Editora Atlas S.A. - 1987.

(Costa e Gurgel 1985) Costa, A.; Gurgel, C., "Inteligência Artificial, quinta geração a caminho", Micro Sistemas, pags 8-10, maio 1985.

- (Cunha e Ribeiro 1987) Cunha, H.; Ribeiro, S., Introdução aos Sistemas Especialistas, Livros Técnicos e Científicos Editora S/A, 1987.
- (Date 1988) Date, C.J., Bancos de Dados: Tópicos Avançados, Ed. Campus, 1988.
- (Diniz et al. 1989) Diniz, A.E. et al., "Otimização das condições de usinagem em células", Máquinas e Metais, Aranda Editora Técnica Ltda, Junho 1989, pags 48-54.
- (Eloranta et al. 1990) Eloranta, E. et al., "Knowledge-based tool for manufacturing system design", Computer-Integrated Manufacturing Systems, Vol.3, N° 3, pags 163-170, 1990.
- (Eversheim et al. 1986) Eversheim, W. et al., "Changes in the Role of Production Management in the CIM-Era", Annals of the CIRP, vol.35, pags 505-512, 1986.
- (Ferraresi 1970) Ferraresi, D., Fundamentos da Usinagem dos Metais, 1ª ed., São Paulo, Edgard Blucker, 1970.
- (Forbellone e Eberspuchen 1993) Forbellone, A.L.V.; Eberspuchen, H.F., Lógica de Programação, McGraw Hill, 1993.
- (Franco et al. 1983) "Estatística Aplicada à estimativa da Grandeza de Amostras e Testes Comparativos da Amostra", UNICAMP, 1983.
- (Hilster 1987a) Hilster, D., "Um sonho antigo", PC-Mundo, junho 1987.

- (Hilster 1987b) Hilster, D., "Prolog versus LISP", PC-Mundo, Set 1987, pags 64-69.
- (Koo e Yoneyama 1989) "Monitoração, Supervisão e Controle com Microcomputadores", Revista IF, seção Controle de Processos, Fev/Mar 1989, pags 16-20.
- (Korth e Silberschatz 1989) Korth, H.F.; Silberschatz, A., Sistemas de Bancos de Dados, McGraw-Hill, 1989.
- (Kowalski 1983/1984) Kowalski, R., Software Engineering and Artificial Intelligence, The SPL - Insight 1983/84, Award Lecture.
- (Krause 1991) Krause, F.L., "Configuração de Produtos com SE", Máquinas Metais - Aranda Editora Técnica Ltda, n°307, Ag. 1991 - pags 50-65.
- (Kusiak 1990) Kusiak, A., Intelligent Manufacturing Systems, Prentice Hall, 1990.
- (Lazarev 1987) Lazarev, G.L., "Solving Problems with PROLOG", AI Expert, pags 59-68, July 1987.
- (Levine 1988) Levine, C. et al., Inteligência Artificial e Sistemas Especialistas, McGraw Hill, 1988.
- (Liu 1988) Liu, D., "Utilization of Artificial Intelligence in Manufacturing", CAPP: From Design to Production, First Edition, Society of Manufacturing Engineers, pags 184-200, 1988.

(Lucena 1987) Lucena, C., Inteligência Artificial e Engenharia de Software, Publicações Acadêmicas Científicas - PUC/RJ e IBM Brasil, 1987.

(Marcus 1986) Marcus, C., Prolog programming, 1986.

(Marks 1988) Marks, P., "expert System Based Generative Process Planing", CAPP: from Design to Production, First Edition, Society of Manufacturing Engineers, Dearborn, Michigan, pags 157-173, 1988.

(Nilsson 1980) Nilsson, N.J., Principles of Artificial Intelligence, Tioga Publishing Company, Palo Alto, California, 1980.

(Oliveira 1990) Oliveira, C.M., "Informática e Automação: instrumentos de competitividade", Datanews, nov 1990.

(Oliveira 1991) Oliveira, J.L., "Mercado Mundial de IA Cresce 50% em 10 anos", Datanews, abril 1991.

(Pallerosi e Cupini 1975) Pallerosi, C.A.; Cupini, N.L., "Durabilidade de Ferramentas de Corte na Usinagem dos Metais", Metalurgia, Vol. 31, nº 215, Out. 1975.

(Passos 1985) Passos, E.L., "Sistemas Especialistas", Micro Sistemas, Dez/85, pags 58-61.

(Passos 1987) Passos, E.L., Inteligência Artificial e Sistemas Especialistas, LTC Editora Ltda, 1987.

(Pham e Oztemel 1992) Pham, D.T.; Oztemel, E., "XPC: an on-line expert system for statistical process control", Int. J. Prod. Res., Vol. 30, N° 12, pags 2857-2872, 1992.

(Ramalho 1988) Ramalho, J.A., dBaseIII Plus: Guia de Referência Básica, McGraw-Hill, 1988.

(Rich 1988) Rich, E., Inteligência Artificial, McGraw Hill, 1988.

(Rodrigues et al. 1987) Rodrigues, A.C.S. et al., " Análise d condições operacionais visando a obtenção de condições de usinagem otimizadas. 7° Seminário de Comando Numérico do Brasil, São Paulo 1987.

(Rosham e Sudesh 1990) Rosham, H.Md ; Sudesh, H., "Sistemas Especialistas para Análise de Defeitos de Fundição", Máquinas e Metais, Aranda Editora Técnica Ltda, n° 295, pags 56-59, Ag. 1990.

(Saliba 1992) Saliba, W.L.C., Técnicas de Programação, McGraw Hill, 1992.

(Schaffer 1986) Schaffer, G.H.; "Artificial Intelligence: a tool for smart manufacturing", American Machinist & Automated Manufacturing, pags 84-94, Ag. 1986.

(Schildt 1989) Schildt, H., Inteligência Artificial Utilizando Linguagem C, McGraw Hill, 1989.

(Schorr 1989) Schorr, H., Conference on Innovation Application on Artificial. Proceedings, March 28-30, 1989, Stanford University, Callifornia/USA.

(Schwabe e Carvalho 1987) Schwabe, D.; Carvalho, R.L., "Engenharia de Conhecimento e Sistemas Especialistas", Editorial Kapelusz S.A., Fev. 1987.

(Shaw e Whinston 1989) Shaw, M.J.; Whinston, A.B., "An Artificial Intelligence Approach to the Scheduling of Flexible Manufacturing Systems", IIE Transactions, Industrial Engineering Research & Development, Vol. 21, N° 2, pags 170-183 , June 1989.

(Smart 1987) Smart, S.B., "Planejamento e Arquitetura do "CIM" (Computer Integrated Manufacturing)", 7° SCNB - Seminário de Comando Numérico no Brasil e 3ª Jornada Internacional de Automação Industrial), São Paulo, 1987.

(Teixeira Jr et al. 1989) Teixeira Jr, A. et al., "inteligência Artificial aplicada ao Planejamento de Processos de Usinagem", 1° Encontro Regional de Automação e Instrumentação, pags IA14-IA19, 1989.

(Vidal 1989) Vidal, A.G.R., CLIPPER-Versão Summer 87, Vol.1. - Livros Técnicos e Científicos Editora Ltda, 1989.

(Vilella 1988) Vilella, R.C., "Metodologia prática visando a otimização das condições de usinagem em células de fabricação", Tese de mestrado, UNICAMP, Campinas, 1988.

(Vita 1987) Vita, J.P. - "The Knight's Tour", AI Expert, June 1987, Vol.2, N°6.

ABREVIATURAS UTILIZADAS

BD: Banco de Dados

CAE/CAD/CAM: Engenharia, Projeto e Manufatura Assistidos por Computador

CAM/I: Computer Aided Manufacturing/International

CAO: Controle Adaptativo Otimizado

CAPP: Computer Aided Process Planning

CIM: Computer Integrated Manufacturing

CN: Comando Numérico

COMP: Planejamento da Manufatura Otimizado por Computador

ESPIRIT: European Strategic Program for Research and Development in Information Technology

FMS: Flexible Manufacturing System

IA: Inteligência Artificial

IME: Instituto Militar de Engenharia

I.M.E.: Intervalo de Máxima Eficiência

M.I.T.: Massachusetts Institute of Technology

SBC: Sociedade Brasileira de Computação

SBD: Sistema de Banco de Dados

SEU: Sistema Especialista de Usinagem

SGBD: Sistema Gerenciador de Banco de Dados

SMFD: Sistema Máquina-Ferramenta e Dispositivos

SOBRACON: Sociedade Brasileira de Comando Numérico, Automatização
Industrial e Computação Gráfica

RELAÇÃO DE TABELAS

	Pag.
Tabela 1: Faturamento dos Setores de Automaatização Industrial no Brasil	4
Tabela 2: Comparação entre IA e Computação Convencional	6
Tabela 3: Comparação entre Conhecimento Humano e Artificial	15
Tabela 4: Definição de um SE quanto ao seu tamanho	54
Tabela 5: Utilização de BD no Brasil	71
Tabela 6: Comparação entre os dados de usinagem atuais com os dados ótimos	90

RELAÇÃO DE FIGURAS

	Pag.
Figura 1: Principais Aplicações de IA	17
Figura 2: Arquitetura de um SE	57
Figura 3: Intervalo de Máxima Eficiência (I.M.E.)	63
Figura 4: Etapas de Desenvolvimento do SEU	69
Figura 5.1: Arquitetura do SEU Automático	72
Figura 5.2: Arquitetura do SEU Semi-automático	72
Figura 6: Arquitetura Simplificada de Transferência de Mensagens	80
Figura 7: Interface com o usuário - Tela 1	84
Figura 8: Interface com o usuário - Tela 1.1	85
Figura 9: Interface com o usuário - Tela 1.2	86
Figura 10: Interface com o usuário - Tela 1.3	87
Figura 11: Interface com o Usuário - Tela 1.4	88
Figura 12: Interface com o usuário - Tela 1.5	89

Figura 13: Esquema de Tipos de Dados Prolog	109
Figura 14: Interface com o usuário - Tela 2.1	119
Figura 15: Interface com o usuário - Tela 2.2	120

Field Name	Type	Width	Dec	Bytes remaining:	3933
------------	------	-------	-----	------------------	------

1	CODIGO	Numeric	4		
2	MATERIAL	Character	30		
3	DUREZA_HB	Numeric	3		
4	RTRACAOMPA	Numeric	4		
5	VC	Numeric	5	1	
6	QUANT_PECA	Numeric	5		
7	NAREST_VID	Numeric	1		
8	VP	Numeric	5	1	

```

=====
*GDS STRUCTURE <c> PECA      Field: 1/8      Ins
=====

```

```

=====
*GDS names begin with a letter and may contain letters,
                        digits and underscores
=====

```

Apêndice 1: Arquivo PECA

Field Name Type Width Dec Bytes remaining: 3969

	Field Name	Type	Width	Dec
1	CODIGO_MAQ	Numeric	4	
2	NW	Numeric	4	1
3	NSTA	Numeric	3	1
4	POTENCUTIL	Numeric	11	4

MODIFY STRUCTURE <> DADOMAO Field: 1/4 Ins

Field names begin with a letter and may contain letters,
digits and underscores

Apêndice 2: Arquivo DADOMAO

Field Name Type Width Dec Bytes remaining: 3969

	Field Name	Type	Width	Dec
1	COD_ENSA	Numeric	4	
2	COD_PPCA	Numeric	4	
3	ZT_MEDIO	Numeric	6	2
4	TC_MEDIO	Numeric	4	1
5	TFT_MEDIO	Numeric	5	1

MODIFY STRUCTURE <> DADOENSA Field: 1/5 Ins

Field names begin with a letter and may contain letters,
digits and underscores

Apêndice 3: Arquivo DADOENSA

Field	Name	Type	Width	Dec
1	CODIGO_FBR	Character	15	
2	XR	Numeric	4	1
3	TTROCAPER	Numeric	4	1
4	CVIDARESTA	Numeric	7	2

MODIFY STRUCTURE <c:> FERRAM Field: 1/4 Ins

Field names begin with a letter and may contain letters,
digits and underscores

Apêndice 4: Arquivo FERRAM

Field Name	Type	Width	Dec
1 COD_PCCA	Numeric	4	
2 I_Z	Numeric	5	3
3 KSI	Numeric	3	0
4 F	Numeric	4	2
5 VC_LIMITE	Numeric	12	8
6 FORCACORTE	Numeric	12	8
7 AP	Numeric	4	2
8 PC_MAX	Numeric	4	0
9 PA_MAX	Numeric	4	0

MODIFY STRUCTURE <c:> DADOENTR Field: 1/9 Ins

Field names begin with a letter and may contain letters,
digits and underscores

Apêndice 5: Arquivo DADOENTR

SIMULAÇÕES E TESTES

