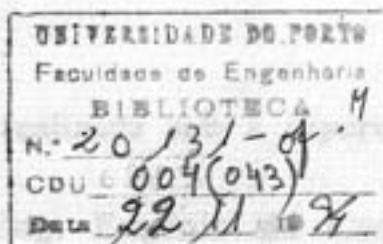


Alcínio Soeiro Miguel

Licenciado em Engenharia Electrotécnica
pelo
Instituto Superior Técnico
Universidade Técnica de Lisboa

**Sistema Gerador de Sistemas Periciais
com
Bases de Conhecimento Cooperantes**

Tese submetida para satisfação parcial dos requisitos do
programa de mestrado em
Engenharia Electrotécnica e de Computadores
(Informática Industrial)



N.º 34735

Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Electrotécnica e de Computadores

Porto • Janeiro de 1991

043 M

M 577 si

ex. 2

tese realizada sob a supervisão do

Doutor Eng. Eugénio da Costa Oliveira

Professor Auxiliar do Departamento de Engenharia Electrotécnica
FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Agradecimentos

Quero expressar a minha gratidão a todos os que, por várias formas, contribuíram para a realização deste trabalho. De entre eles é justo destacar:

O Prof. Eugénio Oliveira, pelos conselhos e colaboração que me deu como orientador deste projecto, assim como pela disponibilidade com que sempre me recebeu.

Ao Instituto Politécnico de Bragança - pela disponibilização de tempo e recursos computacionais que incluíram a aquisição prepositada do compilador de Prolog.

Os meus colegas do Núcleo de I.A., pelo apoio e ambiente de trabalho que me proporcionaram.

O Eng. Rui Camacho pela paciência com que respondeu às minhas constantes solicitações.

O Dr. Mário Rocha, que embora sendo de uma área não ligada à informática teve disponibilidade para me ler o texto original.

O Sr. Atilano Suarez, pela prestímosa colaboração na elaboração dos esquemas.

A minha família, por toda a ajuda e paciência demonstrada

ÍNDICE

RESUMO	1
ORGANIZAÇÃO	2
1 - INTRODUÇÃO	4
1.1 MOTIVAÇÃO	4
1.2 O QUE É A INTELIGENCIA ARTIFICIAL	7
1.2.1 Caracterização extensiva da Inteligência Artificial	9
1.2.1 Diferenças entre inteligência natural e artificial.....	10
1.3 LINGUAGENS DA INTELIGENCIA ARTIFICIAL	13
1.3.1 • Programação tradicional versus programação lógica....	13
2 - INTELIGÊNCIA ARTIFICIAL	15
2.1 REPRESNTAÇÃO E INTEGRAÇÃO DE CONHECIMENTOS	17
2.1.1 Representação de conhecimentos	17
2.1.2 Integração de conhecimentos.....	21
2.2 MECANISMOS DE INFERÊNCIA	23
2.3 MATCHING.....	24
2.4 PESQUISA	24

2.4	RACIOCÍNIO INEXACTO	27
2.4.1	Enquadramento teórico.....	28
3 -	IA DISTRIBUIDA E COOPERAÇÃO	34
3.1	INFORMATICA DISTRIBUIDA.....	35
3.2	INTELIGENCIA ARTIFICIAL DISTRIBUIDA.....	37
3.3	COOPERAÇÃO	39
3.3.1	Abordagens da Cooperação	41
3.4	ARQUITECTURAS BLACKBOARD.....	44
3.4.1	Alternativas quanto ao blackboard.....	48
3.4.2	Quanto às fontes de conhecimento.....	48
3.4.3	Alternativas quanto aos comportamentos.....	49
3.4.4		50
4 -	S. G. DE SISTEMAS PERICIAIS	54
4.1	INTRODUÇÃO	54
4.2	ÁREAS DE APLICAÇÃO DA IA	56
4.3	SISTEMAS PERICIAIS	59
4.3.1	Sistemas Periciais formados por.....	59
4.4	S. GERADORES DE SISTEMAS PERICIAIS.....	60
4.4.1	primeira geração.....	60
4.4.1.1)	Emycin.....	60

4.4.1.II) Expert	62
4.4.1.III) Kas.....	62
4.4.1.IV) Ops	63
4.4.1.V) Rosie.....	64
4.4.2 Segunda geração.....	65
4.4.2.I) Kee	65
4.4.2.II) Art.....	66
4.4.2.III) Personal consultant plus(PC).....	67
4.4.3 Sistemas com capacidade de indução	68
4.4.3.I) ExpertEase.....	68
.....	69
4.4.4 Sistemas com arquitecturas "blackboard"	70
4.4.4.I)Hearsay-III.....	70
4.4.4.II)Age	70
4.4.5 terceira geração.....	72
4.4.5.I)- Controlo de tráfego aéreo.....	72
4.4.5.II)- Negoaciação.....	73
4.4.5.III)- Agentes heterogéneos	73
4.4.5.IV)- Hecodes.....	74

5 - O S.G.S.P. COM BASES DE CONHECIMENTO COOPERANTES.....	79
5.1 INTRODUÇÃO	79
5.2 POSSIBILIDADES E CARACTERISTICAS	80
5.2.1 algumas características importantes.....	81
5.3 DESENVOLVIMENTO DAS CARACTERISTICAS DO SISTEMA	82
5.3.1 Teoria de modelização do domínio.....	82
5.3.2 Aquisição do conhecimento.....	84
5.3.3 Conhecimento Estrutural.....	85
5.3.4 Conhecimento Extensivo	85
5.3.5 Conhecimento Compreensivo	85
5.3.6 Metaconhecimento.....	86
5.4 REPRESENTAÇÃO DO CONHECIMENTO.....	87
5.5 MOTOR DE INFERENCIA	90
5.6 RACIOCÍNIO INCERTO	91
5.7 O ACESSO AOS DIFERENTES AMBIENTES	95
5.8 GERAÇÃO DE SISTEMAS PERICIAIS	95
5.9 CONSULTA DE SISTEMAS PERICIAIS.....	97
5.9.1 Geração do conhecimento da comunidade.....	98
5.9.2 Fases da resposta.....	100

5.9.3	Mecanismos de Cooperação.....	102
5.9.4	Comunicação.....	103
5.10	PROJECTO E IMPLEMENTAÇÃO	104
5.11	CONCLUSÕES	120
5.10.1	Desenvolvimento.....	120
5.8.2	Desenvolvimentos futuros	121
5 - BIBLIOGRAFIA.....		122

ANEXO - Manual de Utilização

ÍNDICE DE FIGURAS

FIGURA 1.1 - limitações das técnicas de IA	5
FIGURA 1.2 - Comparação entre os métodos de programação convencional e os de I.A.	8
FIGURA 1.3 - Sistema inteligente.....	11
FIGURA 1.4 - O projecto da quinta geração.....	12
FIGURA 2.1 - Os elementos básicos da Inteligência Artificial	16
FIGURA 2.2 - Conjunto de regras.....	25
FIGURA 2.3 - Métodos de pesquisa.....	26
FIGURA 2.4 - Mecanismos de inferência.....	26
FIGURA 2.5 - Comparação entre o sistema de incerteza adoptado por Mycin e Personal Consultant	30
FIGURA 3.1 - Fases de Resolução de um problema de modo distribuído	36
FIGURA 3.2 - Partilha de tarefas	41
FIGURA 3.3 - Partilha de resultados.....	42

FIGURA 3.4 - Architecturas Blackboard	44
FIGURA 3.5 - Puzzle	46
FIGURA 3.6 - Arquitectura Funcional do Sistema Hearsay	47
FIGURA 4.1 - Sistema pericial	55
FIGURA 4.2 - Engenharia do Conhecimento e Sistemas Periciais.....	56
FIGURA 4.3 - Senso comum versus perícia	57
FIGURA 4.4 - Análise comparativa entre as shells.....	71
FIGURA 5.1 - Rede semântica.....	83
FIGURA 5.2 - base de conhecimentos fauna_flora	84
FIGURA 5.3 - conhecimento factual.....	87
FIGURA 5.4 - regras de produção.....	87
FIGURA 5.5 - clausulas unitárias <u>prolog</u>	88
FIGURA 5.6 - enquadramentos	88
FIGURA 5.7 - conjunção de evidências.....	92
FIGURA 5.8 - disjunção de evidências.....	92
FIGURA 5.9 - Força evidencial	93
FIGURA 5.10 - conclusão por mais de uma regra	94
FIGURA 5.11 - Navegabilidade no sistema	95
FIGURA 5.12 - Exemplo	97

FIGURA 5.13 - fauna_floramcc.....	98
FIGURA 5.14 - eco_humanamcc	99
FIGURA 5.15 - esquema de funcionamiento interno.....	101
FIGURA 5.16 - Diagrama de contexto	107
FIGURA 5.17 - DFI - nivel 1 -SGSP	109
FIGURA 5.18 - DFI - nivel 2 -1.1	111
FIGURA 5.19 - DFI - nivel 2 -1.2	113
FIGURA 5.20 - DFI - nivel 2 -1.3.....	115
FIGURA 5.21 - DFI - nivel 2 -1.4.....	118
FIGURA 5.22 - Arquitectura de bases cooperantes.....	119

RESUMO

"O GERADOR DE SISTEMAS PERICIAIS COM BASES DE CONHECIMENTO COOPERANTES" **SGSP** é uma Shell integrando um módulo de Geração e outro de Consulta, tendo como tônica fundamental a possibilidade da Cooperação entre as Bases de Conhecimento, quando durante uma sessão de consulta, cada uma das bases não possui só por si todo o conhecimento necessário para responder.

Sendo a cooperação entre peritos humanos importante quando da resolução de problemas complexos, e a comunicação entre os peritos essencial para essa comunicação, lógica será a necessidade de desenvolvimento de sistemas periciais que cooperem.

Um objectivo essencial do presente trabalho foi desenvolver um programa que ilustrasse a cooperação entre sistemas periciais computadorizados. Essa cooperação é indispensável quando se trata de resolver problemas complexos que exigem a utilização de conhecimentos de áreas diferenciadas em simultaneidade.

Recorreu-se às técnicas utilizadas habitualmente, nas áreas de Engenharia do Conhecimento, nomeadamente a toda a experiência anteriormente acumulada no Núcleo de Inteligência Artificial(IA) da F.E.U.P.

O peso significativo da importância atribuída actualmente à I A, o reconhecimento da importância da programação em lógica e, conseqüentemente, da linguagem de programação PROLOG, levaram-nos a optar por este domínio científico e esta ferramenta para trabalho de tese.

ORGANIZAÇÃO

Esta tese está no fundamental dividida em duas partes. A primeira consta dos quatro primeiros capítulos, constituindo o 5º capítulo, dedicado ao trabalho de tese, a segunda parte.

Tentaremos na primeira parte caracterizar de modo sucinto o campo científico da I A, as suas técnicas, aplicações e perspectivas, focando em especial os "SISTEMAS PERICIAIS", e os "GERADORES DE SISTEMAS PERICIAIS", campo ao qual dedicamos o trabalho de tese, e as realizações mais paradigmáticas neste domínio científico.

Neste primeiro capítulo, como motivação, vamos abordar os primeiros conceitos sobre o que é a Inteligência Artificial, e quais as suas linguagens.

No segundo e terceiro capítulos serão tratados os temas fundamentais do domínio científico em que a tese se insere:

segundo capítulo:

- Inteligencia Artificial(breve caracterização das suas técnicas).
- A utilização do raciocínio inexacto em sistemas inteligentes.

terceiro capítulo:

- Inteligência Artificial Distribuida(IAD) e cooperação na Inteligência Artificial.
- Sistemas de Inteligência Artificial com arquitecturas Blackboard.

No quarto capítulo tentaremos fazer um levantamento do estado da arte no que diz respeito a:

1) Sistemas Periciais.

2) Sistemas Geradores de Sistemas Periciais nomeadamente os que:

-Utilizam técnicas de IAD

-Cooperam na busca de soluções.

O quinto capítulo será todo ele dedicado ao trabalho que serviu de suporte à tese (Sistema Gerador de Sistemas Periciais com Bases de Conhecimento Cooperantes). Far-se-á uma análise estruturada de todo o trabalho numa tentativa de tornar clara a explicação da arquitectura funcional do Sistema.

São referidas algumas limitações do Sistema, algumas perspectivas de evolução e finalmente são apresentadas as conclusões.

A parte final corresponde à compilação das referências bibliográficas feitas ao longo da Tese

Em anexo:

- Manual de funcionamento do Sistema Gerador de Sistemas Periciais com Bases de Conhecimento Cooperantes.

1 - INTRODUÇÃO

1.1 MOTIVAÇÃO

*"O que olha para o futuro pode
ver melhor o presente"*

Provérbio Mongol.

Vivemos numa época em que a informática já penetrou e mudou o mundo e podemos constatar que estamos efectivamente na idade dos computadores .

Eles penetram massivamente nos lares, aliciam adultos e crianças e mesmo instituições com grande inércia, como as estruturas do ensino secundário oficial e da administração pública, já se preparam para a sua introdução generalizada.

Fala-se de casas e carros inteligentes, os meios de comunicação social mostram-nos os progressos na tentativa de substituir a actividade inteligente do homem por robots e neste ambiente de esperança temos de reconhecer que este tema é de uma grande actualidade.

Os temas sobre informática e em especial os denominados sistemas inteligentes, ainda recentemente soavam como ficção científica mesmo no mundo ocidental. No entanto em poucos anos, o "milagre" tecnológico verificou-se e hoje sabemos que as suas realizações são da maior importância sendo a única dúvida a incerteza sobre qual será, no futuro, o ritmo do seu crescimento.

As técnicas da IA significaram um grande passo em frente nas ciências da Computação já que permitiram desbloquear barreiras e limitações impostas pelas técnicas tradicionais e assim resolver problemas muito mais complexos. A figura que segue ilustra a ideia de que essas barreiras também são inerentes às técnicas de IA.

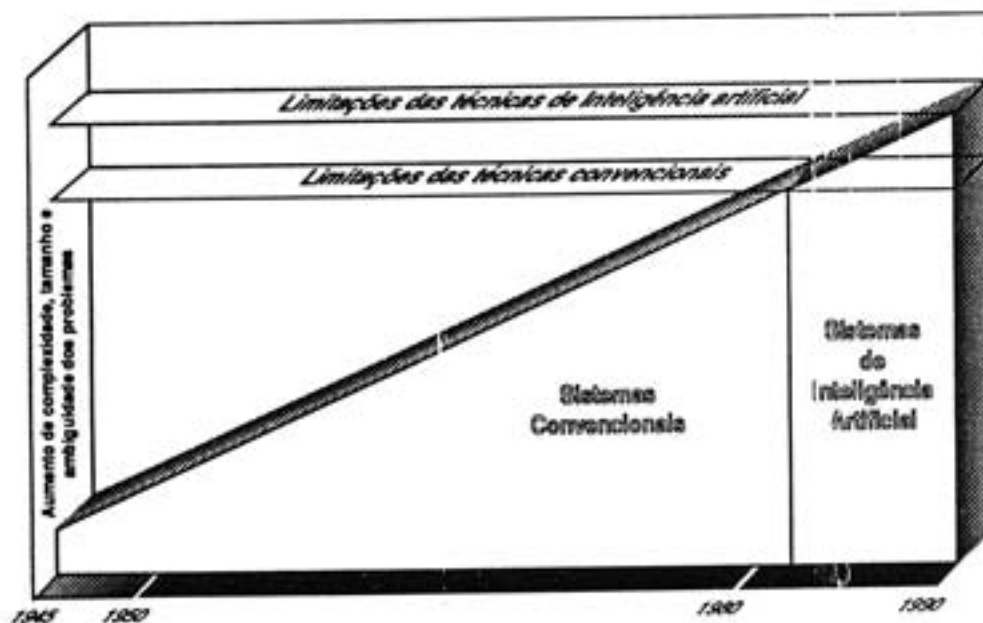


figura 1.1 -limitações das técnicas de IA

Por isso será necessário traçar novos rumos para fazer face à crescente complexidade, tamanho e ambiguidade dos problemas. Este trabalho ao abordar as fronteiras da Inteligência Artificial Distribuída(IAD) e ao implementar a cooperação entre Bases de Conhecimento(BC) dá também o seu contributo nesse campo.

Inteligência Artificial e projecto da 5ª Geração de Computadores são temas que toda a imprensa, mesmo a não especializada, aborda frequentemente.

A maneira como o faz muitas vezes pode levar a uma ideia errada do que realmente já se realizou de facto e do que é o campo de investigação no momento destes domínios.

Apresentarei de seguida alguns tópicos sobre os temas que me motivaram, assim como abordarei certas questões que estão presentes em todos aqueles que trabalham nestes domínios.

1.2 O QUE É A INTELIGENCIA ARTIFICIAL ?

A Inteligência Artificial (IA) como domínio científico e tecnológico em franca expansão tem recentemente atraído a atenção de diversos sectores aos mais variados níveis. Muitas aplicações encontram-se agora em desenvolvimento. Uma visão simplista da IA é a de que se debruça sobre a feitura de programas de computadores destinados a torná-los mais semelhantes ao comportamento humano. Assim, a investigação em IA visa o desenvolvimento de aproximações computacionais ao comportamento inteligente. Esta investigação tem dois objectivos:

- tornar as máquinas mais úteis
- compreender a inteligência.

Os programas de computadores com que a IA se preocupa são antes de tudo processos simbólicos envolvendo complexidade, incerteza e ambiguidade. Estes processos são habitualmente aqueles para os quais não existem soluções algorítmicas. Assim, a IA trata dos tipos de resolução de problemas e tomada de decisões tais como os que se nos deparam constantemente no dia a dia e para os quais os métodos convencionais de programação não têm soluções adequadas.

Esta forma de resolução de problemas diferencia-se nitidamente dos cálculos científicos e de engenharia que são primariamente de natureza numérica e para os quais se conhecem soluções que produzem respostas exactas.

Em contraste, os programas de IA tratam com palavras e conceitos e muitas vezes não garantem um resultado totalmente exacto. Algumas respostas prováveis são toleráveis, ou melhor, preferíveis à ausência de resposta, tal como acontece para a resolução humana de problemas.

Comparação entre os métodos de programação convencionais e os de IA

Programação Convencional	Programação em Inteligência Artificial
Passos de solução explícitos (algorítmica)	Passos de solução implícitos (heurísticas)
Informação e controle integrados	Separação entre controlo e conhecimento
Modificação difícil	Fácil de modificar e actualizar
Respostas exactas	Respostas prováveis podem ocorrer
Só uma resposta é possível	Respostas satisfatórias são aceitáveis

figura 1.2

As características centrais da programação em IA, são:

- Busca heurística
- Conhecimento sobre o domínio da aplicação.

O quadro anterior fornece uma possível comparação entre a IA e os programas de computadores convencionais.

Uma característica chave dos programas IA é a "pesquisa heurística". Cita-se Minsky[Gevarter, 84] como tendo dito " Se não consegues dizer a um computador a melhor maneira de fazer uma coisa, programa-o para fazer muitas tentativas." No entanto, para problemas complexos, o número de possíveis caminhos para soluções pode ser enorme. Assim, a resolução de problemas de IA é geralmente guiada por linhas empíricas - regras que são fruto da experiência e referidas como heurísticas - que ajudam a restringir o espaço de procura das soluções.

1.2.1 Caracterização extensiva da Inteligência Artificial

Algumas definições habitualmente citadas apresentam-se a seguir:

- Para uns ainda uma arte, para outros já ciência, estuda as metodologias e as técnicas, que permitem implementar sistemas computacionais para resolver problemas que normalmente requerem inteligência.
- Parte das ciências da Computação que visa aproximar tanto quanto possível a máquina do homem e, por isso mesmo tentar construir sistemas de computação que evidenciem algumas características que se associam ao comportamento humano inteligente.
- Área das Ciências da Computação interessada no uso de computadores em tarefas que, normalmente, exigem conhecimento, percepção, raciocínio, aprendizagem, entendimento e habilidades cognitivas semelhantes.
- Estuda processos de Representação e Manipulação Simbólicas do Conhecimento diferindo, portanto, quer da computação numérica quer do armazenamento e retribuição da informação organizada.
- Tende a apoiar as suas Metodologias de raciocínio mais em heurísticas do que em fórmulas ou unicamente algoritmos sistemáticos.
- Tem como objectivo conceber sistemas cujo comportamento interpretativo e activo tenha consequências semelhantes ao humano, independentemente dos processos utilizados[OLI., 87].

A Inteligência Artificial não é somente a segunda revolução dos sistemas de informação, mas é sobretudo a essencial.

O conhecimento, a representação do conhecimento e a sua disponibilização são vistas como a ponta de lança para as grandes desafios.

O emprego de processos baseados em computadores e de processos baseados em conhecimento intensivo melhora, sem dúvida, a performance do gestor, do administrador e do utilizador, e aumenta o seu nível de inteligência.

Há diferenças, mas também há semelhanças entre a inteligência natural e a artificial.

1.2.2 Diferenças entre inteligência natural e inteligência artificial

Segundo **Luis Moniz Pereira** são:

- O cérebro é susceptível de regimes de trabalho que não são do tipo "tudo ou nada" como o computador.
- O computador tem que funcionar num único estado bem caracterizado.
- O cérebro tem grandes capacidades de paralelismo, como é sabido, e só hoje em dia se começam a explorar computadores com capacidades de paralelismo.
- O cérebro tem a característica de ser autoprogramável, isto é, tem um sistema motivacional e uma consciência reflexiva que tem uma capacidade de, em grande medida, controlar todo o funcionamento do cérebro e mesmo de superar todos os mecanismos de nível mais baixo.

Na computação, tal como nós a entendemos usualmente, o programa é fixo, embora em certos casos as técnicas de IA possibilitem alteração dinâmica desses programas.

Até hoje pouco se explorou a possibilidade de o programa se modificar a si próprio.

Um sistema inteligente terá que ser capaz de reformular os seus objectivos de modo dinâmico, refazendo o seu plano de actuação sempre que as condições iniciais se alterarem. Em suma, como o mundo exterior está em constante mudança, o mundo interior do sistema deverá também adaptar-se para melhor gerir a sua actuação. Na figura seguinte ilustramos esse facto, mostrando que o efeito da actuação, é avaliado e influencia a actuação seguinte.

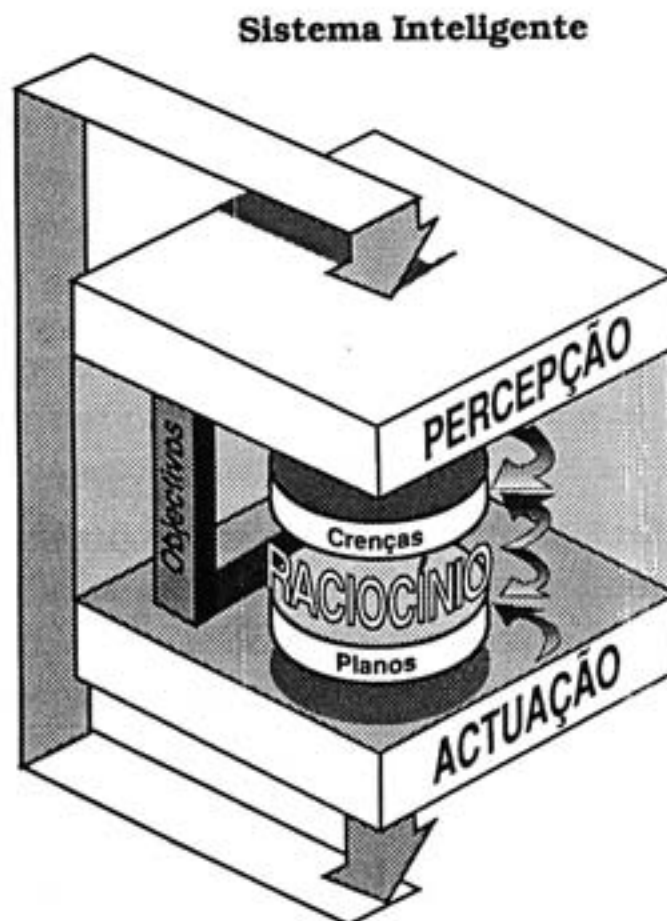


figura 1.3

Quais as linhas de investigação fundamentais no sentido de implementação de sistemas "inteligentes"?

Vejamos a opinião de Norbert Wiener, "Prof. do MIT e pai da Cibernética":

"Cibernética" - assimilada ao princípio de "feedback", ou seja de adaptação inteligente ao meio envolvente, enfim a capacidade biológica do ser humano em aprender, governar a sua acção, formulando objectivos, implementando-os, modificando-os.

Qualquer sistema com pretensões de inteligência deverá ter uma tal capacidade".

A actividade de um sistema inteligente consiste fundamentalmente em aprender através dos seus órgãos de percepção as coisas que o rodeiam, em raciocinar sobre essas coisas e ter uma actuação em conformidade. (Ver Figura 1.3)

Esta actuação é determinada por um conjunto de crenças e objectivos internos e de acordo com aquilo que está a ser apreendido do exterior. [Morgado, 85].

Para que a actividade seja efectiva é necessário que o agente inteligente adquira um modelo do mundo que o rodeia e o mantenha actualizado.

As aplicações mais conseguidas da IA são os Sistemas Periciais(SPs). Eles permitiram o salto qualitativo que a IA precisava, devido aos seus sucessos a nível comercial.

Os sistemas periciais fornecem aos seus utilizadores conclusões sobre áreas de conhecimento específico. Este conhecimento está representado essencialmente na forma de um conjunto de regras e essas regras são utilizadas para orientar o diálogo entre o sistema e o utilizador e daí retirar as conclusões.

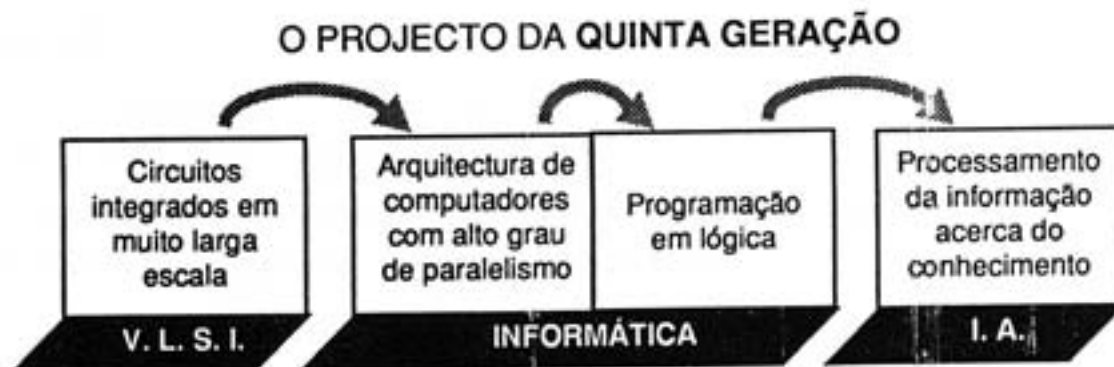


figura 1.4

A quinta geração procurará tirar o maior partido possível das técnicas da Inteligência Artificial pois os projectos das novas gerações de computadores tentam associar os métodos actuais de programação às técnicas específicas da Inteligência Artificial.

Uma das respostas para esta questão(métodos actuais de programação e técnicas de IA) parece ser "programação em lógica" (**Prolog**) . Alguns protótipos preliminares de "máquinas de dedução" ou "máquinas de inferência" já são construídas com base neste princípio.

Esses sistemas deverão permitir a implementação mais eficiente de sistemas periciais, de ambientes para construção de sistemas periciais e de sistemas para reconhecimento da voz e da imagem...

Os "**Geradores de Sistemas Periciais**", tentam criar o ambiente necessário para um utilizador, sem qualquer tipo de formação específica aprofundada, poder gerar e posteriormente consultar os seus próprios Sistemas Periciais.

1.3 LINGUAGENS DA INTELIGÊNCIA ARTIFICIAL

1.3.1 • Programação tradicional versus programação lógica

Na elaboração de um programa tradicional temos de nos adaptar à máquina e ao seu modo de funcionamento elaborando o programa em função desta.

A "programação em lógica" está mais próxima do esquema de raciocínio humano já que o que fazemos é fornecer à máquina informação (conhecimento e esquema do raciocínio) para ela depois resolver os problemas e encontrar as soluções possíveis.

Segundo Eugénio Oliveira, "a programação em lógica, como formalismo computável que é, permite dar o salto qualitativo de que a IA sentia necessidade.

Passa assim a ter-se uma linguagem mais orientada no sentido do homem, em comparação com as linguagens tradicionais, onde cabem a programação, a especificação de problemas, a descrição de bases de dados, as linguagens de interrogação e a definição de restrições de integridade de informação". [Oli., 84].

Prolog(Programação em lógica) é a ferramenta por excelência para implementar a programação em lógica e foi a escolhida para o trabalho. Por isso em anexo lhe serão feitas com mais promenor algumas referências.

Nota: Quer este capítulo quer os que se lhe seguem não pretendem de modo algum ser o resultado de um estudo aturado sobre os temas a tratar. A perspectiva adoptada foi a de olhar os vários assuntos aflorados sob o ângulo de quem vai construir um Gerador de Sistemas Periciais utilizando técnicas da IA e por isso mesmo fazer uma revisão da literatura que aborda o assunto e os temas mais importantes a ele associados.

2- INTELIGÊNCIA ARTIFICIAL

"Os seres humanos são animais(Darwin).

Os animais são máquinas(Descartes).

Os seres humanos são máquinas

Neste capítulo vamos abordar de modo sucinto quatro técnicas fundamentais para a Inteligência Artificial: Matching, Pesquisa, Representação do Conhecimento e Inferência .

De seguida faremos uma abordagem teórica sobre o raciocínio Inexacto em Sistemas Periciais

A abordagem aos temas é influenciada pelo opções tomadas no trabalho de tese. Como exemplo quer os métodos de representação do conhecimento, quer o estudo do raciocínio impreciso está feito levando em linha de conta a implementação prática utilizada. Existem diversos métodos de lidar com raciocínio impreciso em Sistemas Periciais. Por exemplo Mycin e Personal Consultant utilizam variantes. Vamos debruçar-nos sobre as opções que nos nortearam.

OS ELEMENTOS BÁSICOS DA INTELIGÊNCIA ARTIFICIAL



figura 2.1 (Adaptado de Eugénio Oliveira [OLI., 87])

A Inteligência Artificial é algumas vezes identificada e mesmo confundida com a nova disciplina denominada Engenharia do Conhecimento, que estuda sistemas baseados no Conhecimento e que tem sido objecto de grandes avanços na última década.

Sendo um campo de grandes potencialidades, são já muitas as suas aplicações. É assim essencialmente a aquisição, transferência, representação e aplicação do conhecimento que certas áreas da Inteligência Artificial se esforçam por formalizar.

A Engenharia do Conhecimento é o processo de obtenção, representação e manipulação do conhecimento num sistema computadorizado.

Inclui todas as metodologias de projecto de Bases de Conhecimento, a aquisição dos conhecimentos, a aprendizagem pelo computador, a especificação da representação do conhecimento nas Bases de Conhecimento e os processos de consulta incluindo métodos de raciocínio.

Algumas considerações acerca das quatro técnicas essenciais utilizadas pela Inteligência Artificial

2.1 REPRESENTAÇÃO E INTEGRAÇÃO DE CONHECIMENTOS

2.1.1 Representação de Conhecimentos

Nos programas de IA é essencial a utilização de "conhecimento do domínio". A inteligência depende do conhecimento. O conhecimento para ser usado durante a busca deve estar disponível. Nos programas de IA, separa-se este conhecimento dos processos de pesquisa. Assim as modificações no conhecimento só requerem mudanças na base de conhecimentos. Em programas convencionais de computador, conhecimento do domínio e controle são inseparáveis. Como resultado, os programas convencionais de computador são difíceis de modificar.

Desde cedo, que a investigação em IA descobriu que o comportamento inteligente não se deve só aos métodos de raciocínio mas também ao conhecimento de que se dispõe para raciocinar (A experiência da vida proporciona-nos quantidades de conhecimento enormes). Assim quando algum conhecimento substancial tem que ser chamado para interferir num problema, são necessários métodos para modelar eficientemente este conhecimento de forma a que esteja rapidamente acessível. O resultado desta ênfase no conhecimento é que a representação do conhecimento é uma das áreas actuais de investigação mais activas na IA.

De acordo com Helder Coelho podemos arrumar os esquemas de representação do conhecimento em três grandes paradigmas, que são o lógico, o estrutural (formas de representação declarativa) e o procedimental [Camacho 88]:

- **o lógico** descrevendo o conhecimento através de asserções postuladas como verdadeiras (um conjunto de axiomas) acerca do que é modelado;
- **o estrutural** utilizando objectos e relações sobre as entidades a serem modeladas (conceitos, hierarquias de estruturas, descrições de classes de elementos, e instâncias individuais ou componentes de objectos).

- o **procedimental** descrevendo o conhecimento mais operacional através de procedimentos (como o próprio nome sugere).

Ernesto Morgado defende que "a lógica é o método clássico de representação do conhecimento. Ela corresponde a uma representação declarativa[Morgado 88].

A lógica não usa qualquer tipo de estruturação para as suas fórmulas. A sua principal vantagem reside na existência de um conjunto de regras (as regras de inferência) que permitem obter nova informação com base na informação já existente

O PROLOG, que é a linguagem por excelência da programação em lógica também não usa qualquer tipo de estruturação para as suas cláusulas, pelo menos ao nível em que é visto pelo utilizador".

Em anexo é feita uma caracterização mais detalhada da linguagem de programação "PROLOG".

Métodos para Representar Conhecimento

Entre os métodos mais utilizados podemos chamar a atenção para:

- Representações baseadas em lógica
- Representação factual
- Redes Associativas
- Enquadramentos e Guiões
- Regras de Produção

Vou dar atenção especial à representação factual, redes associativas, enquadramentos e às regras de produção pois vai ser o tipo de representação do conhecimento que utilizarei no SGSP.

Representação factual

Os factos são representados normalmente por triplos <<objecto-atributo-valor>> conhecidos por O-A-V. Nesta associação os atributos são características dos objectos (entidades) e cada atributo tem valores. O conhecimento pode ser estruturado de molde a ser representado por várias associações de triplos O-A-V. Os triplos podem ser usados para representar quer aspectos estáticos quer dinâmicos do conhecimento.

- as entidades podem ser físicas ou conceptuais
- os atributos são propriedades associadas a essas entidades

A equação semântica para estes triplos é:

Atributo do objecto = valor

Redes associativas

Uma rede associativa ou rede semântica pode ser vista como uma colecção de objectos chamados nós ligados por arcos. Os nós representam entidades (objectos) e os arcos relações entre eles. A flexibilidade é uma das principais vantagens das redes semânticas. Isto porque novos nós e novas relações podem ser definidos sempre que se tornar necessário.

A herança é outra característica destas redes e refere-se à capacidade de um nó herdar características de outros que com ele estão relacionados.

O conceito de redes semânticas, nós e hierarquias de heranças estão todos relacionados tal como os processos humanos de armazenamento de informação[Neves 88].

Enquadramentos

Os enquadramentos foram pela primeira vez propostos por Marvin Minsky e permitem um tipo especial de aglomeração de conhecimento em blocos. Um enquadramento é a descrição de um objecto ou classe de objectos contendo gavetas(slots) para toda a informação relacionada com esse objecto ou classe. As gavetas podem guardar atributos, valores, apontadores para outros enquadramentos etc.

A ideia fundamental dos enquadramentos é simplesmente a intenção de agrupar o conhecimento nos ditos blocos que podem ser uteis para compreender situações particulares. Uma vez encontrado o bloco particular toda a informação relevante com ele relacionada é imediatamente acedida.

Regras de produção

Devido à representação modular do conhecimento e à sua facilidade de expansão e modificabilidade, as regras de produção são provavelmente a representação mais usada em I A, sendo escolhidas para muitos Sistemas Periciais.

As regras têm a forma $A \Rightarrow C$ podendo ter diferentes leituras (semânticas) em função do domínio de aplicação.

Elas compõem-se, normalmente, por certo número de antecedentes(ou premissas) e por um conseqüente: um conseqüente é verdadeiro quando o é a equação booleana dos antecedentes.

Existem duas aplicações típicas da regra de produção:

1-A partir de uma regra de produção é possível deduzir outro conhecimento, além do que já está explicitamente disponível;

2-A partir de uma regra de produção pode ser especificado que acção realizar perante uma determinada situação.

Representar o conhecimento como pares condição $\leftrightarrow$ acção é uma maneira natural de extrair e codificar conhecimento baseado em regras, em muitas aplicações. Os sistemas de produção são largamente utilizados para construir Sistemas Baseados em Conhecimento para propósitos específicos, designados por Sistemas Periciais.

Uma grande vantagem dum sistema deste tipo é a sua modularidade. Quando existe alteração no domínio do Conhecimento, basta acrescentar, modificar ou retirar uma regras, sem nos preocuparmos com o resto da base de conhecimentos, desde que a coerência seja mantida.

As regras de produção revelaram-se como um meio simples e eficaz de representação do nosso conhecimento e da forma como o mundo funciona.

A sua utilidade aumentou à medida que a heurística adquiriu um papel predominante: praticamente todos os sistemas inteligentes representam o conhecimento heurístico sob a forma de regras. As regras de produção continuam a ser muito utilizadas, sobretudo pela relativa facilidade da sua manipulação, e porque as regras se adaptam bem e facilitam o processo de dedução.

Um conjunto de regras de produção opera numa memória de curto prazo de conceitos relevantes para o domínio de aplicação. Uma regra de produção toma a forma geral "Se condição ENTÃO acção", podendo existir por vezes um "SENÃO alternativas". Um ciclo básico de controlo tenta uma regra de cada vez executando a metade correspondente à acção, só se a metade correspondente à condição se verificar.

Segundo o autor de **mycin** [Shortliff 76], as vantagens de sistemas baseados em regras incluem:

- a utilização de conhecimentos gerais (de manuais ou de peritos); mesmo livros bem indexados podem apresentar dificuldades à consulta ;
- a utilização de conhecimento opinativo para consideração de classes muito pequenas acerca das quais não existem boas informações específicas;
- facilidade de modificação; visto que as regras não estão explicitamente relacionadas umas com as outras e não há necessidade de uma árvore de decisões pré-estruturada, a modificação das regras e a adição de novas regras não necessitam de requerer considerações complexas no que concerne as interações com o restante conhecimento do sistema;
- pesquisa facilitada por eventuais inconsistências e contradições na base de conhecimento; os critérios, armazenados explicitamente em pacotes como regras, podem ser pesquisados e comparados sem dificuldade de maior;
- mecanismos directos para explicar decisões a um utilizador identificando e comunicando as regras relevantes.

2.1.2 Integração de Conhecimentos

Os Sistemas Periciais foram na década de 80 as aplicações por excelência da IA. Em Sistemas Periciais, a aquisição e integração do saber, adquirido ao longo dos anos pelos peritos, é tarefa fundamental para os engenheiros do conhecimento.

A construção de sistemas com capacidade de integração de novos conhecimentos tem obrigado a estudos sobre o tema que têm vindo a ser testados e confrontados com os resultados experimentais; contudo, este é ainda um campo fundamental de investigação.

Torna-se pertinente referir aqui o conceito de aprendizagem pelos computadores. Para Pavel B. Brazdil são dois os principais motivos da importância do tema da aprendizagem:

- Dotar os computadores com esta capacidade. Já existem programas com a capacidade de aprender mas estamos longe de conseguir transmitir conhecimentos ao computador em vez de o programar;

- Desenvolver modelos de aprendizagem em sistemas informáticos ajudar-nos-à a compreender melhor o próprio processo humano de aprendizagem[Brazdil 87].

Ernesto Costa lembra-nos que o maior problema encontrado no desenvolvimento de Sistemas Periciais, reside na aquisição e depuração por parte do Engenheiro do Conhecimento do saber adquirido ao longo dos anos pelos peritos. A meta é colocar o perito em contacto directo com o sistema. Este pede ao primeiro a classificação e análise de casos concretos inferindo das respostas os conceitos/regras usados pelo perito[Costa 88]. Ver no capítulo 3 **sistemas com capacidade de indução**.

Não começa por se construir um sistema acabado, mas um sistema interactivo que permita a transferência gradual de tarefas inteligentes para a máquina.

O conhecimento numa "base de conhecimentos" é uma colecção de factos relacionados com uma certa estrutura global e não uma colecção aleatória de factos. "Base de Conhecimentos" é uma colecção de factos e heurísticas(regras de julgamento), utilizadas para guardar toda a informação e os conhecimentos sobre um certo domínio de problemas[Coelho 86].

Eugénio Oliveira caracteriza no essencial as bases de conhecimento(BC) e evidencia a sua diferenciação relativamente às bases de dados e, em simultâneo, alguma aproximação às bases de dados relacionais. É também salientado o facto de serem mais úteis, mais generalizáveis e sobretudo mais inteligentes.

Duas características importantes das bases de conhecimento:

- No que respeita à sua auto-suficiência e reflexividade: O metaconhecimento;

- No que respeita ao seu dinamismo e evolução temporal: a assimilação do conhecimento[Oli 86].

Contudo, após construídas as bases de conhecimento, o objectivo fundamental a atingir é o de conseguir a exaustão dos mecanismos dedutivos.

2.2 Mecanismos de Inferência

- *Raciocínio* - Racionar consiste em utilizar o conhecimento anteriormente adquirido e disponível para criar nova informação que não estava explícita.

- . Os homens são mamíferos
- . Todos os mamíferos respiram por pulmões

O facto de que os homens respiram por pulmões não estando explícito pode no entanto ser obtido através de raciocínio.

Os raciocínios podem ser no fundamental executados de dois modos:

raciocínio dedutivo - baseia-se no facto de a conclusão se alcançar a partir das premissas, gerando conclusões do geral para o particular, e do princípio para a consequência.

raciocínio indutivo - baseia-se no facto de poder concluir asserções gerais(leis) a partir de factos particulares(exemplos), e fundamenta-se na relação causa efeito. É tema de investigação nas áreas de aprendizagem simbólica, mas que não cabe no domínio do trabalho.

Num sistema baseado em conhecimento, o raciocínio dedutivo é essencial.

Desde sempre a IA se debruçou sobre a construção de programas que procuram de modo "inteligente" soluções para problemas. Cada vez que se toma uma decisão, a situação modifica-se para as novas decisões. Um dos modos normais de representar o espaço de estados de um problema, em IA, é em termos de uma árvore, começando no topo com uma condição inicial e ramificando-se cada vez que uma decisão é tomada. À medida que se continua pela árvore abaixo abrem-se muitas possibilidades diferentes de decisão, de forma que o número de ramificações no final pode ser enorme para alguns problemas.

necessitando de muitos passos de solução. Portanto, é importante que a forma de efectuar a pesquisa seja eficiente.

Um aspecto valioso de uma linguagem de programação em lógica, como o **prolog**, é que a sua execução já fornece um método geral básico e eficiente de inferência lógica: pesquisa sequencial numa árvore, do topo para a base(da raiz para as folhas), da esquerda para a direita e com retrocesso sistemático.

Inicialmente usaram-se métodos "cegos" para percorrer as árvores em busca das soluções . Eram buscas ordenadas que garantiam que o mesmo caminho para a solução não seria experimentado mais que uma vez. No entanto, para problemas mais complexos que estes métodos eram inadequados. Assim, regras fruto da experiência (regras empíricas), referidas como "heurísticas", foram necessárias para ajudar a escolher as ramificações que mais provavelmente levam à solução, de forma a limitar a pesquisa.

Como exemplo, uma heurística simples para os marinheiros no mar alto e sem instrumentos seria orientarem-se pelas estrelas. Isto pode não resultar na opção pelo melhor caminho, mas pode ajudar à prossecução do objectivo. Regras heurísticas como estas podem ajudar a guiar a pesquisa, reduzindo-a enormemente.

2.3 matching

O processo de matching ou, traduzindo para português, "adaptação" é resumidamente o processo de seleccionar de um conjunto de entidades (predicados, termos) aqueles ou aquelas que são seleccionáveis num certo estado.

2.4 pesquisa

Quando se interroga um sistema inteligente computadorizado, partindo do princípio que ele tem o conhecimento necessário para responder, poderemos colocar-nos as questões seguintes:

- O que significa responder à pergunta formulada e o que devemos esperar de uma resposta
- Quais são os mecanismos internos do programa que permitem responder e como funcionam no fundamental.

António Porto adopta a visão de que, em geral, responder a uma pergunta através de um sistema inteligente é procurar uma descrição alternativa de algo descrito na própria pergunta[Porto 88].

Os métodos de pesquisa são os mecanismos que determinam o comportamento do sistema, seleccionando o curso de acção interna do raciocínio[Coelho 88]. Vejamos os métodos sistemáticos mais importantes:

Quanto à direcção de controlo

encadeamento para trás(inverso) - A computação parte do objectivo e encontra uma ou mais regras gerais cuja conclusão se adequa ao objectivo e formula, como novo objectivo, o problema de satisfazer todas as condições dessas regras.

encadeamento para a frente(directo) - A computação parte de factos particulares que são adequados às condições das regras e tenta provar estas continuando recursivamente até que o objectivo final seja deduzido.

Conjunto de regras

- 1.Se o tempo está de sol
e a distância menor ou igual a 20 quilómetros
então iremos de bicicleta
- 2.Se o transporte é de bicicleta
então não é necessário gasolina
- 3.Se não é necessária gasolina
então o custo do combustível é igual a zero
- 4.Se não houver gasolina nas bombas
então é impossível comprar gasolina
- 5.Se é impossível comprar gasolina
então o custo do combustível é igual a zero

Pergunta:

O custo do combustível é igual a zero?

figura 2.2

Assumimos no exemplo que se segue que

está sol e distância= 15 Km.

Encadeamento para trás (directo do objectivo; cima-baixo)

É conhecido o facto?.....Não
 Pode ser obtido de uma regra.....Sim, regra 3
 Que factos é necessário conhecer....."Não é necessária gasolina"
 Este facto é conhecido?.....Não
 Pode ser obtido de uma regra.....Sim, regra 2
 Que factos é necessário conhecer.....Transporte é de bicicleta
 Este facto é conhecido?.....Não
 Pode ser obtido de uma regra.....Sim, regra 1
 Que factos é necessário conhecer.....Tempo está de sol
 e dist. <= 20 km
 São estes factos conhecidos?.....Sim, "Tempo está de sol"
 e "dist. = 15 km".

Encadeamento para a frente (baixo-cima)

É conhecido o facto?.....Não
 Que factos conhecemos?....."Tempo está de sol" e
 "dist. <= 20 km"
 Que factos se seguem daí?....."Transporte é bicicleta"
 pela regra 1"
 É isso que queremos provar?.....Não
 Que factos daí se concluem?....."Não é necessária gasolina"
 pela regra 2"
 É isso que queremos provar?.....Não
 Que factos daí se concluem?....."Custo do combustível igual a 0
 pela regra 3"
 É isso que queremos provar?.....Sim

figura 2.3

Os mecanismos de inferência utilizam estratégias de pesquisa para localizar conhecimento pertinente em bases de conhecimento.

Exemplifiquemos com o mecanismo encadeamento inverso para

primeiro em profundidade



primeiro em largura



figura 2.4

2.3 Raciocínio Inexacto

Este é um dos principais problemas que se depara no momento da representação do conhecimento. Contudo, é deste tipo grande quantidade da nossa inferência quotidiana.

As "Ciências exactas" que sempre trataram com quantidades exactas não sabem como comportar-se com as inexactas.

Elas construíram um universo ideal, feito de regras simples, precisas e sem excepções, esperando que fosse uma aproximação aceitável do universo real, complexo, imperfeito e pleno de excepções.

Em muitas aplicações é vulgar deparar com conhecimentos cuja validade é à partida considerada incerta, podendo ser afectada de um grau de credibilidade.

Ora nenhum computador pode gabar-se de um comportamento inteligente se não for capaz de elaborar os milhões de informações incertas que caracterizam o nosso conhecimento.

Um salto em frente da IA foi enriquecer as próprias regras de produção com factores de incerteza que lhes estabeleceram credibilidade.

Introduzir tais factores significa ser-se capaz de realizar cálculos posteriores sobre entidades daquele tipo.

O tratamento dos graus de credibilidade pode ser feito basicamente de dois modos distintos[Porto 85]:

- Um deles é usar uma teoria probabilística, em que os graus são probabilidades, e o mecanismo dedutivo usa regras de composição de probabilidades durante a inferência.

O outro é usar quantificadores vagos para representar o grau, e definir as regras de decomposição para estes, segundo os princípios da "fuzzy logic".

As regras de um sistema de produção, com efeito, têm um efeito de "propagadores de incerteza":

com quanta incerteza de

"se neva é Inverno a 80%"

"se é Inverno é Janeiro a 33%"

se pode inferir que

"se neva é Janeiro"? a 80%? A 33%?

A 80 por 33 isto é, a 26,4%?

Para cada regra coloca-se sempre o mesmo problema.

Ainda não existe uma teoria completa e coerente capaz de explicar todos os fenómenos inexactos.

E poderíamos inclusive perguntar-nos o que são exactamente fenómenos inexactos.

2.3.1 Enquadramento teórico

É difícil formalizar numa teoria os processos do raciocínio humano usados na dedução de alguns factos a partir de outros que podem ser incertos .Imprecisos, não totalmente conhecidos ou compreendidos, subjectivos, etc.... Pois devido às características mencionadas dos factos ou acontecimentos acerca dos quais se pretendem tirar conclusões, torna-se muitas vezes difícil compreender de que maneira influenciam a conclusão tirada ou sequer porque é que a influenciam

Vamos no caso do presente trabalho inspirar-nos no modelo do **mycin** pois está suficientemente testado e constitui como que um paradigma neste domínio.

Os conceitos tradicionais de ciências "exactas" versus "inexactas", com as ciências sociais inseridas na segunda categoria, baseiam-se numa distinção falsa, geralmente reflectindo a presença ou ausência de notação matemática. Saliente-se que apenas uma pequena porção das ciências podem ser classificadas como exactas - áreas como as matemáticas puras e subcampos da física, nos quais alguma da exactidão "tem sido levada ao teste final da axiomatização formal"[Shortliffe 76].

Em diversas áreas das ciências da natureza aplicadas, por outro lado, as decisões, predições e explicações são feitas apenas depois de os processos exactos terem sido misturados com conhecimentos não formalizados.

Mas se por um lado há teorias matemáticas para tratamento de incerteza ou imprecisão, assentes numa base sólida, essas mesmas teorias parecem não seguir as leis do pensamento. Se hoje concluimos um dado facto, nada nos garante que amanhã ou depois, em presença dos mesmos factores, concluiríamos o mesmo facto.

O uso de probabilidades em técnicas de raciocínio inexacto, levanta alguns problemas de pontos de vista quer teóricos quer práticos, o que não impede alguns dos sistemas mais bem sucedidos em algumas áreas da inteligência artificial(por exemplo sistemas periciais), incorporem nas metodologias que disponibilizam aos utilizadores métodos de raciocínio baseados no cálculo de probabilidades ou afins.

É o caso da utilização do teorema de Bayes no "Prospector" e da teoria da confirmação no "Mycin", isto para mencionar os mais conhecidos. O primeiro usa uma aproximação "Baysiana" subjectiva e o segundo baseia-se numa interpretação diferente da probabilidade de um acontecimento, aliada à ideia de confirmação.

É importante salientar mais uma vez que estas aproximações se baseiam em métodos empíricos, não totalmente suportados por teorias firmes e universalmente aceites como válidas na manipulação de imprecisão ou incerteza em sistemas dedutivos. Têm no entanto a vantagem inegável de produzir na prática os resultados pretendidos.

O modelo **mycin** é uma aproximação à probabilidade condicional. Ele foi concebido tendo em vista a área de problemas tratados em **mycin** mas é potencialmente aplicável a qualquer domínio no qual o conhecimento real tenha de ser combinado com a experiência antes de se obter uma opinião informada para explicar observações ou sugerir um curso de acção. Embora não adoptemos à letra o modelo referido, podemos considerar que tem sido o grande inspirador de todos aqueles que trabalham no domínio.

COMPARAÇÃO ENTRE O SISTEMA DE INCERTEZA ADOPTADO POR
MYCIN E PERSONAL CONSULTANT

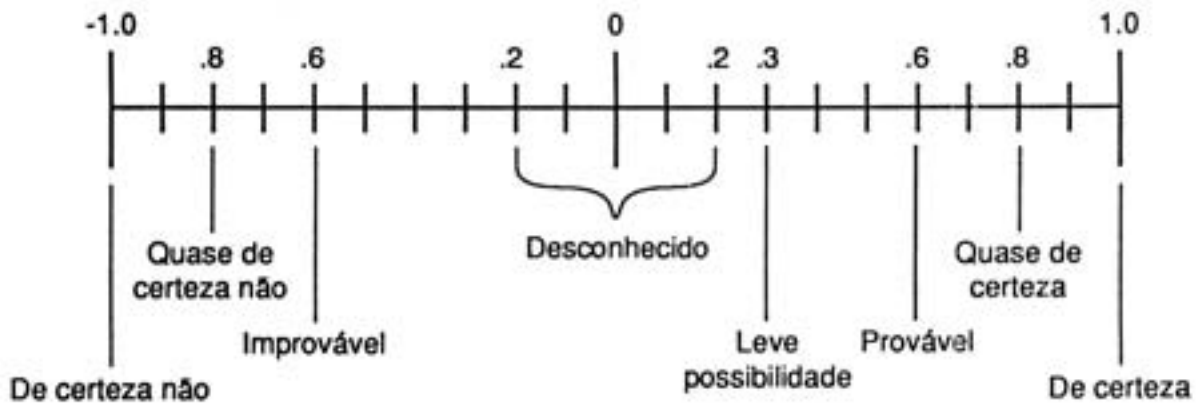


figura 2.5

Análise detalhada sobre o modelo teórico de força evidencial proposto em **mycin**.

Duas unidades de medida de Crença e Descrença são propostas. A notação é como se segue:

- (1) $MC[h|e] = X$ significa que "A medida do aumento da Crença na hipótese h , baseada na evidência e é X "
- (2) $MD[h|e] = Y$ significa que "A medida do aumento da Descrença na hipótese h , baseada na evidência, e é Y "

A evidência e não necessita ser um acontecimento observado, mas pode ser uma hipótese (em si própria sujeita a confirmação). Assim, posso escrever $MC[h_1, h_2]$ para indicar a medida de incremento da Crença na hipótese h_1 se a hipótese h_2 for verdadeira. Paralelamente $MD[h_1, h_2]$ indica a medida de incremento da Descrença na hipótese h_1 se a hipótese h_2 for verdadeira.

Para ilustrar consideremos que $MC[h|e] = 0.7$ segundo a regra de amostragem que nos foi fornecida pelo perito. A relação entre o número 0.7 e a probabilidade será explicada à medida que prossiga. Por agora direi simplesmente que a crença do perito em que h seja verdade aumenta com o conhecimento de que e é verdadeiro. Por outro lado $MD[h|e] = 0$ para o mesmo exemplo, significa que o perito não tem razão alguma para aumentar a sua descrença em h em vista de e .

De acordo com a teoria de probabilidades subjectiva, pode dizer-se que a probabilidade pessoal do perito $P(h)$ reflecte esta crença em h em qualquer momento dado. Assim $1-P(h)$ pode ser visto como uma

da descrença do perito na verdade de h . Se $P(h|e)$ for maior que $P(h)$ a observação de e aumenta a crença do perito em h , enquanto que diminui a sua descrença quanto à verdade de h .

De facto, a diminuição proporcional da descrença é dada pela razão:

$$\frac{P(h|e)-P(h)}{1-P(h)}$$

Esta razão é chamada a medida de acréscimo de crença em h resultante da observação de e , isto é, $MC[h,e]$.

Suponhamos, por outro lado, que $P(h|e)$ fosse inferior a $P(h)$. Então a observação de e faria diminuir a Crença do perito em h enquanto que faz aumentar a sua Descrença em relação à verdade de h . O decréscimo proporcional de Crença seria neste caso dado pela razão:

$$\frac{P(h)-P(h|e)}{P(h)}$$

Chamamos a esta razão a medida de acréscimo da Descrença em h resultante da observação de e , isto é $MD[h,e]$. Para resumir estes resultados em palavras, consideramos a medida de acréscimo da Crença $MC[h,e]$ como o decréscimo proporcional de Descrença em relação à hipótese h que resulta da observação e . Similarmemente, a medida de acréscimo da Descrença $MD[h,e]$ é o decréscimo proporcional de Crença em relação à hipótese h que resulta da observação e , em que a Crença se estima por $P(h)$ em qualquer momento dado e a Descrença por $1-P(h)$. Estas definições correspondem de perto a conceitos intuitivos de confirmação e infirmação. De notar que um elemento de evidência não pode ao mesmo tempo favorecer e desfavorecer uma mesma hipótese, quando $MC[h,e] > 0$, $MD[h,e] = 0$ e quando $MD[h,e] > 0$, $MC[h,e] = 0$.

Para além disto, quando $P(h|e) = P(h)$ a evidência é independente da hipótese (não confirma nem infirma) e $MC[h,e] = MD[h,e] = 0$.

As definições acima podem agora ser formalmente especificadas em termos de probabilidades condicionais e *a priori*:

$$MC[h,e] = \begin{cases} 1 & \text{se } P(h)=1 \\ \frac{\max[P(h|e), P(h)] - P(h)}{\max[1, 0] - P(h)} & \text{nos outros casos} \end{cases}$$

$$MD[h,e] = \begin{cases} 1 & \text{se } P(h)=0 \\ \frac{\min[P(h|e), P(h)] - P(h)}{\min[1,0] - P(h)} & \text{nos outros casos} \end{cases}$$

De notar que aqui $P(h)$ se utiliza para quantificar probabilidades *a priori*. Pode ser mais correctamente escrito como $P(h|0)$, isto é, a probabilidade de h sem provas. O exame destas expressões revelará que são idênticas às definições introduzidas acima. A definição formal é, no entanto, introduzida para demonstrar a simetria entre as duas medidas. Adicionalmente, é definida uma terceira medida, chamada factor de certeza (CF) que combina a MC com a MD de acordo com a definição seguinte:

$$CF[h|e] = MC[h,e] - MD[h,e]$$

O factor de certeza é assim um artefacto para combinar graus de Crença e Descrença num número único. Tal número é necessário para facilitar comparações entre a força evidenciadora de hipóteses competentes. A utilização deste número resultante de uma composição poderá ser consultada por quem desejar aprofundar o tema. As observações seguintes ajudam a clarificar as características das três medidas que foram definidas (MC, MD, CF):

Características das Medidas de Crença

(1) Amplitude de graus:

- (a) $0 \leq MC[h,e] \leq 1$.
- (b) $0 \leq MD[h,e] \leq 1$.
- (c) $-1 \leq CF[h,e] \leq +1$.

(2) Força evidencial e hipóteses mutuamente exclusivas:

Se h for demonstrado correcto $P[h|e] = 1$:

- (a) $MC[h,e] = [1 - P(h)] / [1 - P(h)] = 1$.
- (b) $MD[h,e] = 0$.
- (c) $CF[h,e] = 1$.

Se a negação de h for tida como certa $P[\text{não } h|e] = 1$:

- (a) $MC[h,e] = 0$.
- (b) $MD[h,e] = [0 - P(h)] / [0 - P(h)] = 1$.
- (c) $CF[h,e] = -1$.

De notar que isto dá $MC[não h, e] = 1$ se e só se $MD[h, e] = 1$ de acordo com as definições de MC e MD acima. Mais, o número 1 representa a Crença (ou Descrença) absoluta para MC (ou MD). Assim, se $MC[h_1, e] = 1$ e h_1 e h_2 forem mutuamente exclusivos $MD[h_2, e] = 1$.

(3) Falta de evidência:

(a) $MC[h, e] = 0$ se h não é confirmado por e (i.é., e e h são independentes ou e infirma h).

(b) $MD[h, e] = 0$ se h não é infirmado por e (i.é., e e h são independentes ou e confirma h)

(c) $CF[h, e] = -0$ se e não confirma nem infirma h (i.é., e e h são independentes)

3. INTELIGÊNCIA ARTIFICIAL DISTRIBUÍDA E COOPERAÇÃO

"O mais útil e menos avançado de todos os conhecimentos humanos parece-me ser o próprio homem."

Jean Jacques Rousseau

Neste capítulo faremos uma abordagem teórica sobre alguns dos temas que serviram de suporte para as opções adoptadas no nosso trabalho:

Inteligência Artificial Distribuída(IAD)

Cooperação e abordagens da cooperação

Arquitectura exemplo para cooperação("Blackboard")

Toda a abordagem aos temas é orientada pelas opções que posteriormente na análise do trabalho de tese iremos expor. Mesmo quando possa parecer não haver uma ligação directa ela estará implícita. Como exemplo é tratado o enquadramento "blackboard", embora não tenha sido essa a opção tomada para permitir a comunicação entre as Bases de Conhecimento. A Cooperação implica comunicação e o enquadramento "blackboard", teve o mérito de nos inspirar e ajudar a encontrar uma solução, a tal ponto que podemos considerar que temos com a solução adoptada um "blackboard" conceptual.

3.1 INFORMÁTICA DISTRIBUÍDA

A tecnologia das redes de computadores evoluiu extraordinariamente e hoje é utilizada em grande escala. Por outro lado as redes de transmissão de dados interligam o mundo. Para essa interligação os sistemas computacionais de diversas empresas e instituições formam redes, resultantes da interligação de redes mais pequenas - Internetworks. Essas redes, geograficamente distribuídas, são resultantes de interligação de hardware e de software heterogêneo, utilizando diversos tipos de protocolos de comunicação.

É lógico, quer dentro quer fora das ciências da computação, o interesse nos sistemas computacionais distribuídos.

As arquitecturas distribuídas desde que mantendo a autonomia funcional dos elementos do sistema poderão oferecer algumas vantagens sobre as arquitecturas centralizadas, nomeadamente em termos de operacionalidade e velocidade global.

Vantagens:

- Maior eficiência
- Maior disponibilidade
- Modularidade
- Flexibilidade[Giozza 86]

É evidente que o necessário "Know how" para operacionalizar todos estes sistemas complexos é fruto do interesse dos mais variados sectores e por isso é grande a necessidade da investigação neste campo.

Também, no domínio dos Sistemas Inteligentes e, nomeadamente, nos Sistemas Baseados em Conhecimento, se torna útil a investigação no domínio dos Sistemas Distribuídos.

Um sistema distribuído resulta da Partilha de um conjunto de tarefas entre um conjunto de elementos de processamento, lógica ou fisicamente disjuntos.

Sendo as tarefas cometidas a um Sistema Inteligente normalmente complexas, entende-se a utilidade de conseguir operacionalizar as técnicas de IA em ambientes distribuídos.

Fases de Resolução de um problema de modo distribuído[Smith 81].

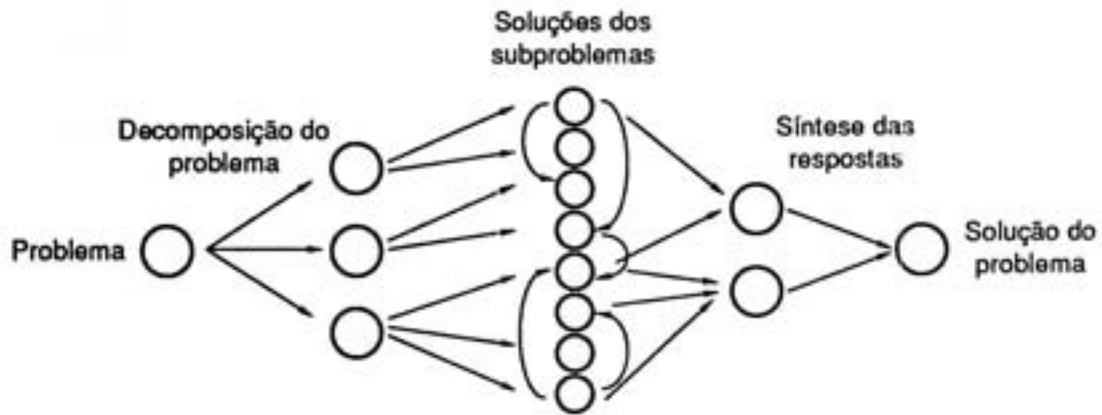


figura 3.1

J. Macedo e V. Freitas defendem a utilização das técnicas da inteligência artificial, nomeadamente sistemas periciais na gestão de ambientes distribuídos [Macedo, 90].

As características operativas de um sistema de gestão daquele tipo, segundo os autores citados deverão ser as seguintes:

- portabilidade
- tolerância a falhas
- velocidade
- simplicidade
- segurança

Alguns problemas se colocam a sistemas deste tipo, nomeadamente:

- Sincronização entre os elementos do sistema.
- Quando se deve esperar por uma resposta ou avançar com os conhecimentos disponíveis.
- Como devem ser decompostas as tarefas.
- Como devem ser as mensagens entre sistemas para serem inteligíveis pelos outros componentes.

3.2 INTELIGENCIA ARTIFICIAL DISTRIBUIDA(IAD)

Distribuição e paralelismo são dois importantes temas da IA. As bases conceituais para processos concorrentes em IA remontam aos anos 50 começando com a perspectiva de Minsky da mente como uma sociedade de Agentes cooperantes e a concepção de Arbib do cérebro como um conjunto de esquemas "concorrentes".

Poderemos dividir, com base nos interesses dos investigadores, o mundo da Inteligência Artificial Distribuída(IAD) em duas áreas base:

- A investigação em Resolução Distribuída de Problemas ("Distributed Problem Solving") cujo processo de resolução de um problema específico é dividi-lo num certo número de módulos ou "nodos" que cooperam a nível de divisão e partilha de conhecimentos sobre o problema e sobre a solução a desenvolver;

- Uma segunda área a que chamaremos SMA (sistemas multi agentes) na qual a investigação se preocupa com a coordenação de procedimento inteligente entre um grupo de "agentes" autonomamente inteligentes(possivelmente pré-existent). Como podem coordenar o seu conhecimento, fins, capacidades e planos conjuntamente para agir ou resolver problemas. Os "agentes" num SMA podem trabalhar para um fim único ou para fins individuais separados que depois interagem.

Como os módulos no Sistema Pericial Distribuído(SPD), os agentes num sistema MA devem partilhar o conhecimento sobre problemas e soluções, mas devem também raciocinar sobre os processos de coordenação entre os agentes. Nos sistemas MA a tarefa de coordenação pode ser bastante difícil visto poderem existir situações(nos chamados sistemas abertos) em que não haja possibilidade de controlo global, conhecimento globalmente consistente, fins globalmente partilhados, critérios globais de sucesso ou mesmo a representação global de um sistema[Bond 88].

A investigação numa terceira área relacionada com a IAD a que chamaremos "IAP"(Inteligência Artificial Paralela), debruça-se sobre o desenvolvimento de arquitecturas computacionais paralelas, e sobre as linguagens e algoritmos apropriados a essas arquitecturas.

A IAP preocupa-se primordialmente com a resolução de problemas de desempenhos de sistemas de IA e não para progressos conceptuais na compreensão da natureza do raciocínio e do comportamento inteligente entre agentes múltiplos.

Características dos sistemas da IAD:

- Coerência global - comporta o sistema quando visto como um conjunto de funções, do ponto de vista estratégico, e, por outro lado, o quanto são legíveis as actividades do sistema, como um todo para facilitar a explicação e "debugging" dos elementos específicos;
- Inteligência "emergente"- um sistema IAD é uma rede de sistemas inteligentes individuais projectados para cooperar de algum modo[Feeling, 89].

Para coordenar as suas acções, os "agentes inteligentes" necessitam de representar e raciocinar sobre o conhecimento, acções e planos de outros "agentes". A investigação em IAD preocupa-se também com novas técnicas de representar e utilizar o conhecimento sobre crenças, acções, planos e objectivos mais apropriadas a sistemas inteligentes distribuídos.

Existem três classes de objecto de estudo para a IAD, todas englobando grupos coordenados de "agentes inteligentes":

- Aproximações de Sistemas Naturais;
- Perspectiva de Ciência de Engenharia;
- Aproximação de Coordenação Pessoa-máquina[Bond 88]

Comparação quanto ao controlo entre sistemas IAD:

- Sistemas completamente centralizados.
- Sistemas distribuídos com um elemento central de controlo e planeamento.
- Sistemas distribuídos com um grupo hierárquico de elementos de controlo.
- Sistemas totalmente distribuídos no qual cada elemento é responsável pelo seu próprio controlo.

- Os dados acedidos pelos elementos do sistema podem ser distribuídos entre os elementos ou estarem contidos numa base de dados centralizada.
- Prever modificações no controlo, quando o sistema aumenta [Feeling, 89].

No início a IAD não era diferenciada enquanto subdisciplina separada da IA, mas a sua importância é cada vez maior e hoje a investigação em IAD promete ter um largo impacto:

- na ciência cognitiva (modelos mentais, cognição social,...);
- nos sistemas distribuídos (raciocínio sobre conhecimento e acções em sistemas distribuídos, suportes arquitecturais e linguísticos para IAD);
- na interacção computador homem (atribuição de tarefas, interfaces inteligentes, coerência de diálogo e actos de fala);
- na investigação básica em IA (representação de problemas, epistemologia, raciocínio cooperativo e métodos de resolução de problemas);
- no projecto de sistemas de IA (sistemas periciais cooperativos, robots cooperantes, concepção cooperativa de resolução de problemas).

A cooperação entre peritos humanos é importante quando da resolução de problemas complexos e a comunicação entre os peritos é essencial para essa cooperação. A cooperação entre sistemas periciais computurizados é também essencial quando se trata de problemas complexos.

Sendo a cooperação fundamental também em termos de IAD vamos dedicar-lhe uma especial atenção.

3.3 COOPERAÇÃO

A finalidade de um sistema pericial é resolver problemas significativos no " mundo real " os quais normalmente exigem a presença de um perito humano para a sua resolução . A maioria dos sistemas periciais existentes simula o que faria um único perito humano, em situações estereotipadas mas não simula o que resultaria da cooperação entre vários peritos humanos.

Pela observação do método de resolução de problemas dos peritos humanos, torna-se claro que tal cooperação é essencial. Por exemplo o diagnóstico e tratamento de um doente num hospital muitas vezes requer cooperação entre médicos de diferentes especialidades; a interpretação de dados geológicos necessita cooperação entre peritos geólogos, geofísicos e geoquímicos; a tomada de decisões de estratégia militar necessita cooperação entre peritos militares a vários níveis de comando. ...

Existem vários modos de cooperação de acordo com as suas relações de interdependência[Zhang 89]:

Cooperação horizontal - quando cada perito no grupo cooperativo pode dar soluções para os problemas sem depender dos outros. Por exemplo a cooperação entre médicos para diagnóstico de doenças por vezes conduz-nos para a cooperação horizontal ("segunda opinião"). A consulta e a comparação de opiniões melhora significativamente o valor do diagnóstico final;

Cooperação em árvore - quando um sistema pericial, ao mais elevado nível, depende de sistemas periciais a níveis mais baixos, com vista à resolução de problemas. Por exemplo, uma decisão de um engenheiro chefe depende muitas vezes dos engenheiros subordinados.

Cooperação recursiva - quando diferentes sistemas periciais dependem um do outro para encontrar soluções para os problemas.

Por exemplo, para interpretar os dados geológicos, os geólogos, os geofísicos, e os geoquímicos, muitas vezes dependem uns dos outros num caminho recursivo. Existe uma dependência mútua nas mesmas subtarefas.

Cooperação híbrida - quando diferentes peritos utilizam cooperação horizontal em alguns níveis de uma árvore ou na cooperação recursiva.

Por outro lado, eles podem de modo igualmente válido utilizar cooperação recursiva ou em árvore, em alguns passos de uma cooperação horizontal. Um exemplo disto é quando se obêm várias opiniões de modo a conseguir otimizar a qualidade do resultado final dado pelo engenheiro chefe.

Estes podem também ser os principais métodos de cooperação a utilizar, quando se trata de problemas de cooperação entre peritos não humanos.

De facto, realizar tal cooperação entre sistemas periciais é considerada como uma das mais importantes tarefas para a próxima geração de sistemas periciais.

3.3.1 Abordagens da Cooperação

Dois conceitos que surgem frequentemente na literatura são o de "**Partilha de tarefas**(Task Sharing)" e "**Partilha de resultados**(Result Sharing)", eles surgem associados ao conceito de cooperação na resolução de problemas por várias fontes de conhecimento distribuídas. Nenhuma das fontes de conhecimento é capaz, por si só, de resolver o problema.

Algumas questões que se colocam sobre a cooperação e em que moldes ela deve ser estabelecida são:

- Como devem interagir os SP's cooperantes para resolver um problema.
- Como distribuir a "carga" computacional entre eles.
- Como integrar os resultados para obtenção da solução e sua comunicação para o exterior da comunidade.

Partilha de tarefas.

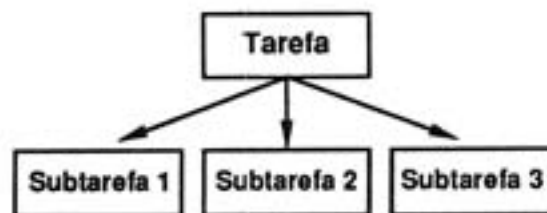


figura 3.2

A carga é dividida entre sistemas que resolvem independentemente subproblemas do problema completo[Smith, 81].

Este princípio pressupõe negociação

Quem irá fazer o quê?

O sistema SP₁ pede auxílio ou porque a tarefa é muito custosa para que ele possa resolvê-la sozinho ou porque ele não é um perito nessa tarefa. Sendo assim tomará as seguintes decisões de acordo com a situação:

tarefas custosas - dividir em subtarefas posteriormente distribuídas.

tarefas onde não é perito - negociar directamente com o perito mais apropriado ou, no caso da tarefa ser simples, apresentá-la ao grupo.

Neste último caso o sistema SP₁ coloca uma tarefa que necessita resolver aos restantes SPs. O sistema SP_n pode achar que é capaz de resolver a tarefa e informa SP₁ que opta por qual ou quais irão então efectuar a tarefa.

Partilha de resultados[Smith 81].

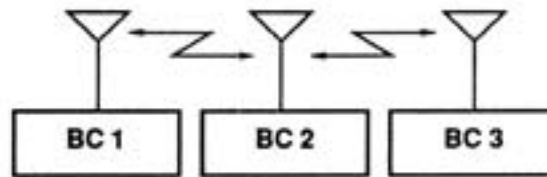


figura 3.3

Princípio utilizado quando os subproblemas não podem ser resolvidos por peritos trabalhando independentemente. É então necessária uma segunda forma de cooperação. Os SP's transferem periodicamente resultados parciais para outros peritos. Pressupõe-se que os peritos trabalham em tarefas com alguma similaridade. Neste caso verifica-se que os resultados de um perito podem interferir no raciocínio de outros peritos. Peritos representados na figura-3.3 por BCs.

Resumo

A Partilha de tarefas é mais apropriada para problemas nos quais seja apropriado definir uma hierarquia de tarefas (pesquisas heurísticas) . Esses problemas conduzem à decomposição num conjunto relativamente independente de subtarefas com pouca necessidade de informação global ou sincronização.

A Partilha de resultados é mais útil e necessária nos problemas onde os resultados atingidos por um nó influenciam ou restringem os que podem ser obtidos por outro nó.

A cooperação entre SPs é um dos aspectos particulares da Segunda Geração de Sistemas Periciais. Um Sistema distribuído amplia significativamente as capacidades dos SPs. Sistemas Inteligentes geográfica e/ou logicamente distribuídos e que possam ser partilhados por muitos utilizadores em lugares diferentes ao mesmo tempo é um dos objectivos a atingir pela investigação em IAD.

A cooperação entre SPs envolve 3 elementos essenciais:

- a) os cooperadores (agentes que cooperam);
- b) a gestão da cooperação;
- c) comunicação.

Ora tudo isto implica uma arquitectura base que permita implementar os elementos essenciais da cooperação. Iremos debruçar-nos com alguma atenção sobre as arquitecturas "blackboard". Elas são referência obrigatória para todos aqueles que de um modo ou outro se debruçam sobre estes temas.

3.4 Arquitecturas Blackboard

A arquitectura "blackboard" é uma estrutura de dados globalmente acessíveis a recursos de conhecimento independentes (designados na literatura por "fontes de conhecimento") que cooperam na formulação de hipóteses. O primeiro sistema importante utilizando esta arquitectura foi **Hearsay**, projecto iniciado em 1967(Frederick Hayes, Victor Lesser, e outros), que foi concebido como um conjunto de «especialistas» cooperantes que comunicavam através dum "blackboard", inventando assim um novo e importante paradigma de sistema inteligente.

Um sistema com uma arquitectura "Blackboard" trata a resolução de problemas como um processo incremental e por oportunidades de juntar de modo satisfatório os vários elementos necessários até obter a solução. A abordagem básica consiste em dividir o problema em sub-tarefas distintas e hierarquicamente colocadas ao mesmo nível.

Baseado nas últimas alterações à informação registadas no "blackboard" o módulo de controlo selecciona e activa a próxima fonte de conhecimento; somente as "fontes de conhecimento" estão autorizadas a fazer alterações no blackboard(fig. 3.4)

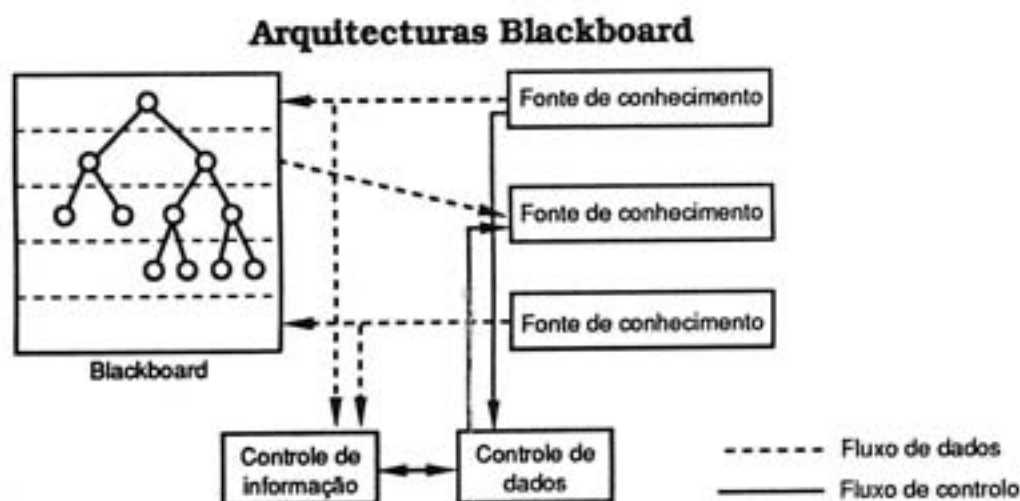


figura 3.4

As arquitecturas "blackboard" são constituídas por três componentes funcionais (figura 3.4)[Patterson 90].

1.- Existe um número de "fontes de conhecimento" que são conjuntos separados e independentes de conhecimento codificado. Cada fonte de conhecimento pode ser imaginada como um especialista numa área limitada necessária para resolver um dado subconjunto de problemas. As "fontes" podem conter conhecimento na forma de procedimentos, regras, ou outras formas.

2.- Uma estrutura de dados globalmente acessível, chamada "Blackboard", que contém o estado corrente do problema e a informação necessária pelas "fontes de conhecimento" (dados de entrada, soluções parciais, controlo de dados, alternativas, soluções finais). As "fontes de conhecimento" fazem alterações nos dados do "blackboard" que de um modo incremental, buscam uma solução. A comunicação e interacção entre as "fontes de conhecimento" tem lugar através do "blackboard".

3.- O controlo de informação pode ser incluído dentro das fontes, no "blackboard", ou possivelmente num módulo separado (é de realçar que no entanto o "blackboard" não constitui uma unidade de controlo). O controlo do conhecimento monitoriza as mudanças no "blackboard" e determina qual o passo seguinte na resolução do problema.

As regiões do "blackboard" poderão disponibilizar dados necessários às sub-tarefas ou receber dados produzidos por estas. O conhecimento é dividido em "fontes de conhecimento" especializado que executam as sub-tarefas. A informação em cada região está globalmente acessível no "blackboard" tornando-o um meio de interacção entre as "fontes de conhecimento". Genericamente, uma "fonte de conhecimento" recebe informação de uma região como seu input e envia output informativo para outra região. A decisão para utilizar uma determinada "fonte de conhecimento" é tomada dinamicamente utilizando o presente estado de solução (a informação actualizada contida na estrutura de "blackboard"). Esta abordagem particular à decomposição do problema e à aplicação do conhecimento é muito flexível e funciona bem em diversos domínios da aplicação, tais como troca de mensagens, sistemas de tempo real, robótica, etc... .

Vamos descrever a estratégia de resolução de problemas com o blackboard utilizando a seguinte analogia.

Imaginemos uma sala com um quadro negro grande no qual um grupo de especialistas estão em conjunto completando um puzzle.

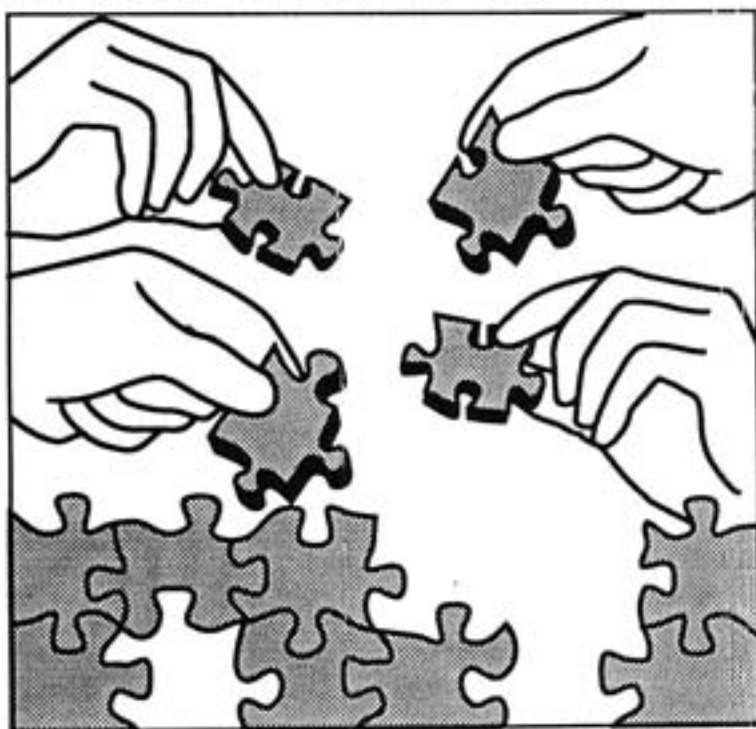


figura 3.5 (puzzle)

Cada um dos especialistas tem algum conhecimento especializado acerca do modo de resolver o puzzle (por exemplo, um especializado nos bordos, outro nos sombreados, outro nas cores, etc...). Cada membro examina a sua ou as suas peças e decide se ele deve ser encaixado no puzzle incompleto. Esses membros têm as peças apropriadas para incrementar, no quadro, o puzzle e completá-lo. O puzzle pode ser resolvido em silêncio absoluto sem qualquer tipo de comunicação entre os membros do grupo. Cada elemento é auto-suficiente, sabendo quando pode contribuir para a solução. As peças no quadro estão organizadas de modo hierárquico em níveis que facilitam a análise e a solução. A informação de um nível funciona como input para um conjunto de "fontes de conhecimento". As fontes modificam o conhecimento e colocam-no no mesmo ou em nível diferente.

Os sistemas convencionais "blackboard" exemplificados por "Hearsay-II" e os seus descendentes embora utilizando várias fontes de conhecimento e simulando paralelismo não eram

verdadeiramente paralelos. Funcionavam em máquinas série e incorporavam "schedulers" para assegurar a invocação sequencial de "fontes de conhecimento". No entanto podem ser chamados quase paralelos visto que conceptualmente o seu funcionamento é quase paralelo e visto que se baseiam em concepções semânticamente distribuídas do Conhecimento. A arquitectura do "Hearsay-II" mostra-nos claramente os princípios acabados de expor.

ARQUITECTURA FUNCIONAL DO SISTEMA HEARSAY

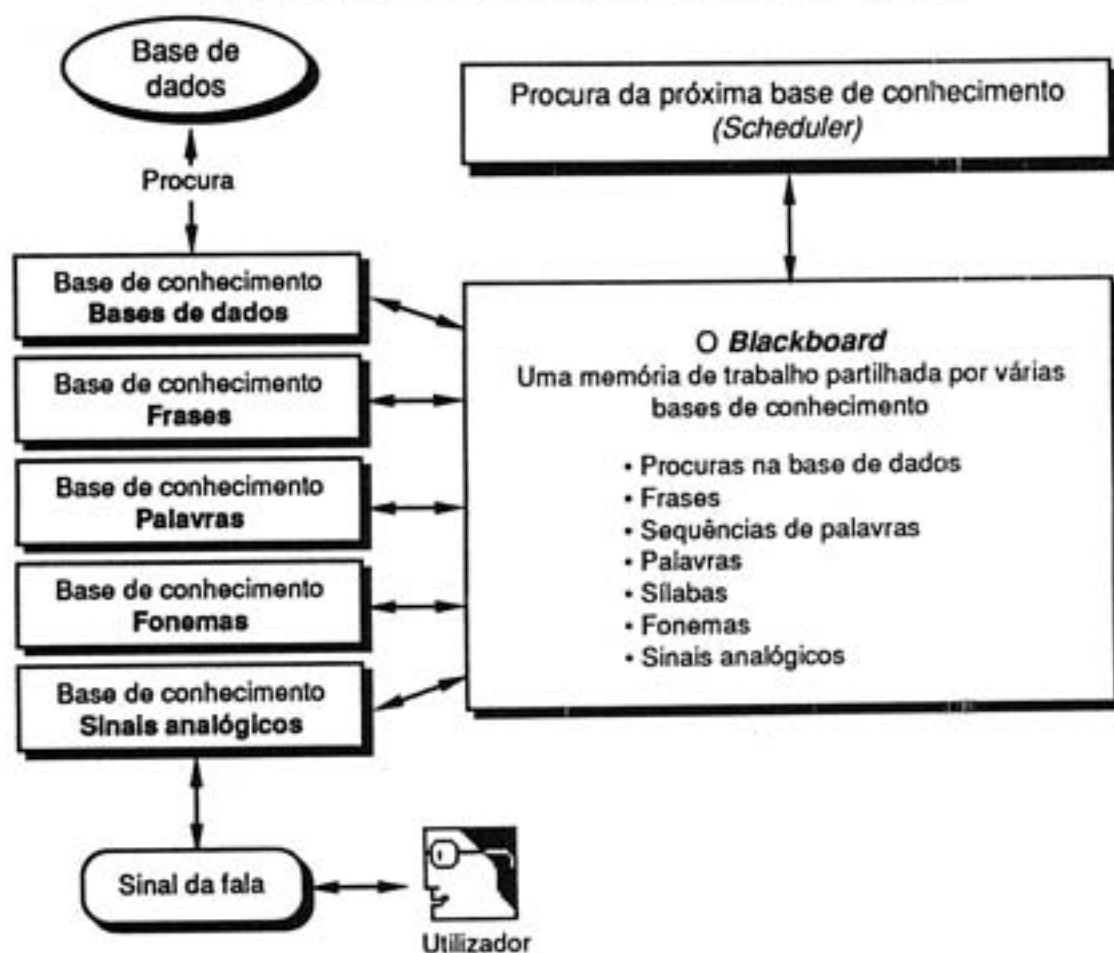


figura 3.6

Numa análise dos sistemas que trabalham utilizando o enquadramento "blackboard" apercebemo-nos que embora utilizando a mesma filosofia básica existem diferenças significativas no modo de implementação final. Vale a pena rever brevemente algumas das alternativas, ou "graus de liberdade" dos sistemas "blackboard" e vamos resumidamente fazê-lo no que diz respeito a duas das suas componentes funcionais[Engelmore 88].

3.4.1 Alternativas quanto ao blackboard

- 1) Painéis "blackboard" simples ou múltiplos
- utilização de um painel separado para controlo de "blackboard".
- 2) "Blackboard" global ou disperso
- regiões que são apenas acessíveis a um sub-conjunto de fontes de conhecimento.
- 3) Entradas de "blackboard" simples ou de competição
- entradas alternativas que são competidoras (i.e. mutuamente exclusivas).

3.4.2 Alternativas quanto às fontes de conhecimento

- 1) Estrutura homogênea ou heterogênea
cada fonte de conhecimento num sistema poderá ter uma estrutura interna totalmente diferenciada das restantes (procedimentos regras ...), desde que adira ao protocolo comum de acesso "blackboard".
- 2) Organização das regras
fontes de conhecimento baseadas em regras, as partes "condição" e "acção" da fonte de conhecimento são uma unidade única, enquanto em alguns sistemas as partes condição e acção da fonte de conhecimento são entidades separadas.
- 3) Estrutura da condição da fonte de conhecimento
as fontes de conhecimento contêm geralmente um conjunto de condições que têm de ser especificadas antes de se tornarem executáveis, mas a forma deste teste (processos, regras, ...) e as funções subsidiárias possíveis (precondições, variáveis locais de ligação, ...) variam bastante.

3.4.3 Alternativas quanto aos comportamentos

Alguns comportamentos apontados como necessários às arquitecturas Blackboard para controlo, serem consideradas inteligente [Barbara, 85]:

- 1) Tomar explicitamente decisões para resolver o problema do controlo .
- 2) Decidir:
 - que acções tomar reconciliando decisões independentes;
 - quais as acções desejáveis e quais as exequíveis.
- 3) Adopção de:
 - heurísticas de controlo que foquem todos os "atributos da acção" que sejam úteis na situação corrente da resolução do problema.
 - heurísticas de controlo de importância variada.
- 3) Adoptar, reter ou desprezar heurísticas de controlo individuais em resposta a situações dinâmicas de resolução de problemas.
- 4) Planear dinamicamente sequências estratégicas de acções.
- 5) Raciocinar acerca das prioridades relativas das acções do domínio e do controlo.

Alguns pressupostos da "filosofia" de arquitecturas blackboard para controlo:

- 1) A estrutura do "blackboard" está organizada em duas dimensões:
 - intervalos de solução - Diferentes intervalos de solução tem a ver com diferentes fases da solução em alguma dimensão dum problema específico.
 - níveis de abstracção - Diz respeito à solução em diferentes níveis de detalhe, para a mesma solução.

- 2) Os elementos da solução são gerados por processos independentes designados como "fontes de conhecimento". Cada fonte de conhecimento tem um formato condição-acção. Se a configuração do "Blackboard" satisfaz a condição de uma fonte de conhecimento forma-se uma estrutura a que chamamos KSAR (Knowledge Source Activation Record) que, quando é reconhecida pelo "scheduler", "corre" a fonte de conhecimento. É um mecanismo guiado por eventos.
- 3) Em cada ciclo de resolução do problema, o "scheduler" escolhe o próximo KSAR. O processo mais usual é usar um "scheduler" sofisticado com variados critérios.

Porquê usar uma arquitectura "blackboard"?

A arquitectura "blackboard" é apenas uma das várias alternativas possíveis para a resolução automática de problemas. Pensamos que há várias razões para que a abordagem do tipo "blackboard" seja preferido às outras alternativas[Engelmore 88]:

- 1) Modularidade - porque a arquitectura é por inerência modular (fontes de conhecimento, níveis de "blackboard", estruturas de controle, ...) a especificação, teste e manutenção do sistema podem ser facilitadas.
- 2) Controlo dinâmico - a arquitectura fornece um amplo leque de capacidades de controlar dinamicamente o comportamento solucionador de problemas do sistema.
- 3) Eficiência - uma política de controlo, que seleccione o conhecimento mais eficaz a aplicar e/ou focaliza o esforço nas regiões mais profícuas do "blackboard", pode traduzir-se em ganhos consideráveis de eficiência.
- 4) Convergência - a modularidade e a estrutura flexível de controlo podem permitir o processamento paralelo de partes separadas de um problema.

3.4.4 Concorrência (Convergência)

Uma das atracções da arquitectura "blackboard" é o aparente potencial para a concorrência. Muitos dos sistemas têm, pelo menos em conceito, fontes de conhecimento concorrentes. Na prática, o benefício desta perspectiva conceptual tem sido facilitar

o desenvolvimento do ponto de partida da modularidade do sistema.

Virtualmente todos os sistemas "blackboard" (ou pelo menos aqueles sobre os quais temos documentação pública) invocam as fontes de conhecimento em série e utilizando um único processador. Embora o modelo "blackboard" seja paralelo por inerência, de facto a tradução do potencial para um sistema prático concorrente revelou-se não ser trivial por várias razões. Uma das razões principais é que a maior parte das aplicações da IA envolvem decomposição de problemas mal estruturados. Em contraste com soluções algorítmicas bem estruturadas para problemas numéricos, não se pode prever ab initio o fluxo do controlo, ou o tamanho ou frequência da criação das estruturas de dados, por exemplo. Outra razão está relacionada com a noção de "blackboard" global. Não se pode implementar o "blackboard" como estrutura de dados que seja em simultâneo centralizada e distribuída. No caso de um "blackboard" centralizado, não se podem *ultrapassar* os custos do acesso à memória. Para um "blackboard" distribuído, a gestão de acesso aos conteúdos pode tornar-se bastante dispendiosa quando se trata de dados independentes. Problemas semelhantes levantam-se quando se pretende tornar paralela a execução de fontes de conhecimento, que muitas vezes não são independentes no que respeita ao acesso aos dados ou à ordem de execução. Assim, embora o modelo "blackboard" possa parecer idealmente apropriado à execução paralela em multiprocessadores, encontram-se basicamente as mesmas dificuldades em atingir bons desempenhos como se se tratasse de modelos de computação sequencial.

Há duas maneiras óbvias de permitir a concorrência num ambiente "blackboard":

- 1) dividir o problema em vários sub-problemas independentes e operar vários sistemas "blackboard" autónomos, com comunicação entre os sistemas.

- 2) introduzir a concorrência em operações dentro do mesmo sistema; por exemplo, correndo várias fontes de conhecimento em paralelo.

Na prática, introduzindo a concorrência num sistema "blackboard" simples parece colocar o maior desafio técnico. A chegada de sistemas multiprocessadores tem dado um novo ímpeto à investigação. Possivelmente os problemas-chave estão em encontrar processos de ligar a memória global do modelo "blackboard" a um sistema concorrente, sem introduzir problemas de inconsistência e/ou contenção de memória.

Uma abordagem a estes problemas, que tem sido objecto de investigação (por exemplo na Univ. de Stanford), é desenhar o sistema em termos de fontes de conhecimento como agentes que comunicam uns com os outros através de buffers de comunicação [Shoen, 86], resultando assim um sistema "blackboard" sem um "blackboard". O modelo "blackboard" fica apenas a um nível conceptual.

Concluindo:

As fontes de conhecimento são independentes na medida em que não se invocam umas às outras e habitualmente não possuem qualquer informação sobre o conhecimento das outras.

Elas são cooperantes na medida em que fornecem uma contribuição para uma solução partilhada.

A arquitectura "blackboard" concilia a independência e a cooperação entre fontes de conhecimento permitindo-lhes influenciarem a resposta (solução do problema) embora indirectamente, respondendo e modificando a informação gravada no "blackboard". Por outro lado a arquitectura de controlo com "blackboard" define explicitamente dois "blackboards":

- um dependente do domínio de aplicação;
- outro de controlo independente do domínio.

Tentamos ilustrar e validar o âmbito e o campo de aplicação dos sistemas "blackboard". Temos já elementos que nos permitem evidenciar o ponto de vista expresso pelo prof. Eduard Feigenbaum[Engelmore 88], em que descreve a abordagem

"blackboard" como " a arquitectura de sistemas de conhecimento mais geral e flexível para construir sistemas periciais".

Chegou o tempo de retirar proveitos e também da consolidação de toda a experiência acumulada até ao momento. Uma grande variedade de estruturas podem ser construídas dentro do modelo "blackboard" por causa das opções disponíveis para o projectista em vários estádios Vale a pena em jeito de conclusão realçar também que muitos dos objectivos e requisitos dos sistemas "blackboard" têm contrapartida em desenvolvimentos fora de AI; por exemplo os aspectos de controle de um "blackboard" têm muito em comum com desenho de sistemas operativos.

4 - S. G. DE SISTEMAS PERICIAIS

"In the future, there will be only one industry : entertainment"

Marvin Minsky

4.1 INTRODUÇÃO

Entre todos os campos de aplicação da IA, os Sistemas Periciais (SPs) constituem a sua coroa de glória pois são, como se tem vindo a referir, as mais conseguidas.

Vamos neste capítulo fazer uma análise de alguns ambientes de programação (ferramentas) destinados à construção de SPs agrupando-os em três gerações. O critério utilizado para o agrupamento em gerações foi, para além de temporal, baseado nas capacidades acrescidas, significando por isso saltos qualitativos entre elas.

Justifica-se esta análise de ambientes de programação, destinados à construção de SPs, pois foi também a partir das experiências conhecidas que foram amadurecendo as ideias para o trabalho desta tese (SGSP).

Na última década tem-se assistido ao aparecimento de aplicações de utilidade prática evidente, cada vez mais sofisticadas e eficientes utilizando já os métodos e as técnicas da Inteligência Artificial.

Esta nova geração de aplicações, designadas por Sistemas Periciais, visa alcançar um comportamento semelhante aos dos peritos no seu domínio específico.

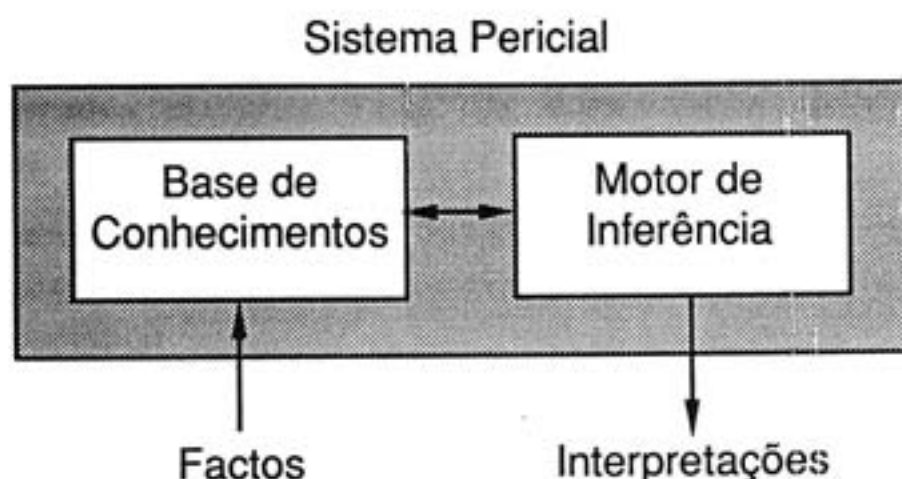


figura 4.1

Tais sistemas adquirem certos sub-conjuntos restritos do conhecimento humano e transferem-no para a máquina possuindo capacidades de raciocínio (motor de inferência) que lhe permitem inferir novos conhecimentos a partir dos conhecimentos que integram.

A construção de Sistemas Periciais, porque são sistemas baseados no Conhecimento, engloba-se na nova disciplina denominada Engenharia do Conhecimento.

4.2 ÁREAS DE APLICAÇÃO DA IA

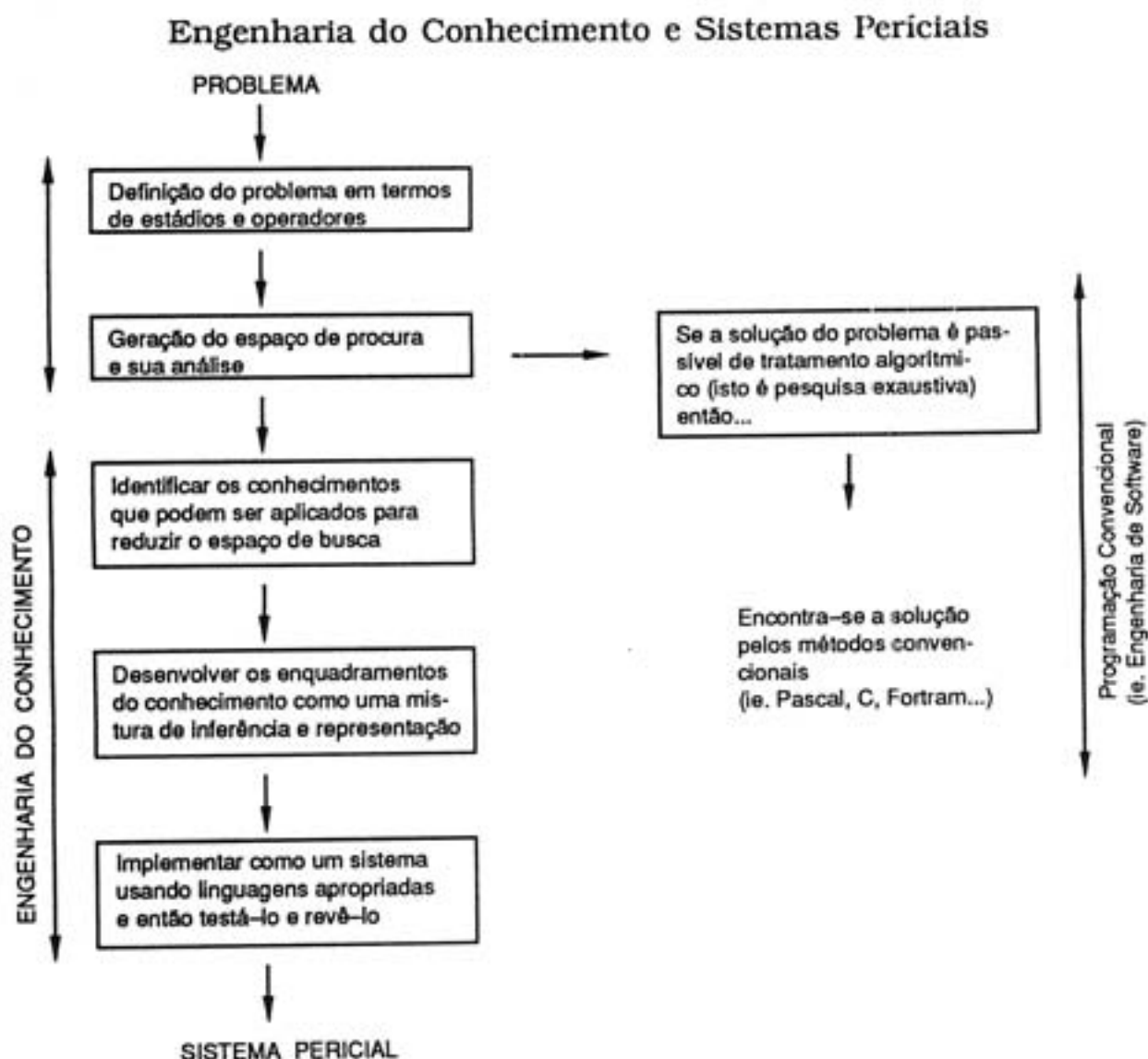


figura 4.2

A Engenharia do Conhecimento está no limiar de uma nova era. Do ponto de vista dos esforços de investigação podemos encara-la como resultado da coerência do desenvolvimento do que tem sido feito até aqui.

A Inteligência Artificial tem-se virado decididamente para as aplicações concretas. Os Sistemas Periciais e os Geradores de Sistemas Periciais representam um verdadeiro passo em frente numa evolução técnica contínua e entre outros indicadores, são prova disso os recentes sucessos comerciais (ver anexo). Assim nasceram os primeiros <<produtos IA>> comercializáveis. A IA saiu da redoma e os seus

<<sistemas periciais>> tornaram-se na coqueluche do mercado da computação. Sistemas Periciais há-os cada vez mais e para mais domínios:

Jogos de xadrez, reparação dos próprios computadores, explorações petrolíferas ou outros minerais, diagnóstico médico, apoio inteligente à decisão ...

Como já foi referido a transferência do saber adquirido ao longo de muitos anos pelos peritos tem representado um grande problema para os engenheiros do conhecimento. Um objectivo é conseguir eliminar o interface humano entre o especialista e o sistema, facilitando assim o processo de transferência e permitindo incrementar o volume de conhecimentos de modo simples.

Contudo, os progressos são significativos e neste momento podemos afirmar que a tecnologia de construção de SPs está num estágio avançado. Podemos prever sem dificuldade que sistemas periciais, nas mais diversas áreas do saber, continuarão a aparecer e não virá longe o dia em que a sua utilização será um facto normal e natural.

Senso comum versus perícia

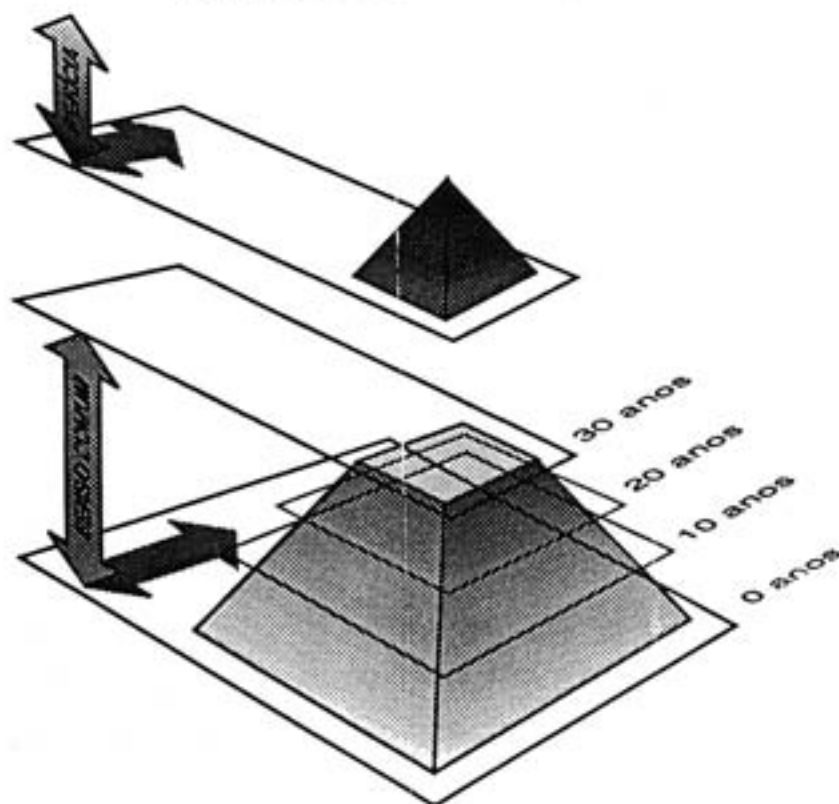


figura 4.3

"O que acontecerá quando a inteligência artificial promover a automatização massiva das tarefas dos especialistas ?"

Lembremos entretanto que o conhecimento disponível nos SPs corresponde ao representado na figura pelo topo da pirâmide. Esse é o conhecimento ralmente mais difícil de obter e por isso a vantagem de o tornar acessível através de sistemas computadorizados. Ainda está longe, contudo o momento em que será possível disponibilizar de forma automática todo o tipo de conhecimento incluindo o do senso comum, pois isso corresponde a quantidades demasiado grandes de conhecimento. Por outro lado a representação computacional do senso comum(considerado quase um predicado natural do ser humano e essencialmente baseado na experiência) é um dos problemas da IA com poucas probabilidades de ser resolvido, satisfatoriamente, no futuro próximo.

Deste modo teremos de reconhecer que os SPs continuarão a ter uma grande lacuna na sua tentativa de substituir completamente os peritos humanos.

Mas será que algum dia essa substituição será mesmo uma realidade?

(Hubert Dreyfus) – Prof. de Filosofia da Universidade da Califórnia (Berklei), – Um computador jamais actuará como nós, nunca será um ser humano... Um computador não tem emoções, não tem necessidades, não vive em sociedade.

(Alan Turing) – "**Génio Inglês dos computadores**", – Um computador merecerá ser considerado inteligente no dia em que a sua reacção(saída) em nada se distinguir do resultado de um acto inteligente humano... e isto aos olhos dos próprios humanos.

Perante as opiniões atrás expressas lembremos Galileu e pensemos "tal como a Terra, a I. A. move-se".

Os efeitos da IA na sociedade têm sido lentos, mas as previsões apontam para que, pelo início do próximo século, os resultados devem ser revolucionários.

A deslocação do emprego da manufactura pode ser tão dramática quanto o foi a da agricultura e implicará naturalmente uma modificação no trabalho de tipo administrativo - serviços, pesquisa, tempos livres.

4.3 SISTEMAS PERICIAIS

Os sistemas periciais são sistemas dedicados à resolução de problemas não triviais num dado domínio específico, mais ou menos restrito, em que o conhecimento é altamente especializado[Leitão, 88].

Segundo Waterman, o coração de um sistema pericial é o poderoso corpo de conhecimentos que acumula durante a sua construção[Waterman, 86].

4.3.1 SISTEMAS PERICIAIS essencialmente formados por:

- Base de conhecimentos de factos sobre o domínio da aplicação e heurísticas associadas ao problema.
- Um mecanismo de inferência (que é uma estrutura de controlo) para uma correcta utilização da base de conhecimentos na solução dos problemas.
- Uma memória de trabalho - uma base de dados - para os dados de um problema ou de um domínio.
- Um dispositivo de diálogo - capaz de facilitar uma comunicação homem-máquina, estimular, orientar, extrair respostas e apresentar resultados .

Um sistema pericial deverá ter a possibilidade de se autoenriquecer quando em contacto com um especialista num determinado tema.

Além de resolver problemas, deve também ser auto-explicativo podendo portanto instruir inclusivé outros peritos no desenvolvimento de conhecimentos especializados.

Em suma, deve e pode ser tutorial e convincente.

4.4 SGSP (Sistemas Geradores de Sistemas Periciais)..

Um sistema pericial é, de modo sucinto, a junção de uma base de conhecimentos a um motor de inferência.

Por **SGSP** entende-se um sistema pericial independente do domínio, podendo ser utilizado, de uma forma genérica, na resolução de qualquer problema e que actua interactivamente respondendo a questões do utilizador.

Os **SGSP** são ferramentas primordiais na construção dos sistemas periciais. Ao longo dos últimos anos tem sido desenvolvida a sua técnica e hoje já é possível dispor de um bom leque de opções em ferramentas genéricas deste tipo.

Mas uma escolha criteriosa entre elas, torna-se uma tarefa difícil e crucial reservada a peritos. O seu estudo e conhecimento detalhado é importante também para quem se propõe construir um, mesmo sendo uma actividade académica e tratando-se de um protótipo.

Na questão dos SGSP vamos diferenciar três gerações. Um SGSP da primeira geração é essencialmente um esqueleto de um sistema pericial, em que o utilizador integra a base de conhecimentos relativa ao domínio de aplicação. As estratégias de raciocínio são fixas e por via de regra bastante elementares. Os SGSP <<históricos>> desta geração, são aqueles que foram desenvolvidos nas principais universidades americanas durante os anos 70 e constituem, ainda agora, fonte de inspiração para quem se abalança a experiências nestes domínios.

4.4.1 primeira geração

4.4.1.1) emycin- [Melle 81],[Hayes-Roth 83],[Oli, 84]

emycin realizado por Melle; Shortliffe e associados na Universidade de Stanford. É um instrumento clássico para a construção de sistemas periciais de consulta. O seu motor de inferência é o do MYCIN, o famoso sistema pericial para diagnóstico médico realizado em Stanford, e a sua base de conhecimento é estruturada por regras de inferência com factores de confluência (<<se neva, inverno a 80%>>). O

editor permite introduzir tais regras na base de conhecimento de modo simples.

O interface pode descrições pormenorizadas de todas as entidades em jogo, à medida que são citadas nas regras de inferência. Os factos são representados com triplos <<objecto-atributo-valor>>, conhecidos por O-A-V.

Uma vez introduzidas as regras por meio do editor, é preciso <<compilá-las>> por meio de um compilador apropriado que as transforma para a sintaxe própria do **emycin**. Nesta altura o sistema pericial está pronto. O utilizador pode colocar perguntas ao **emycin**, cujo interpretador varrerá a base de regras para inferir a resposta. O método de controlo do **emycin** é um típico encadeamento inverso. Em cada momento o **emycin** procura avaliar o valor de um atributo: Para isso procura a regra da qual depende o seu valor e tenta avaliá-la; se lhe falta qualquer valor procura-o e assim sucessivamente. Se no final faltam regras, o **emycin** pede ao utente o último valor encontrado e não calculado. Uma vez encontrado o valor, seja através de uma regra seja através do utilizador. O **emycin** volta atrás e calcula todos os valores em suspenso.

Para decidir o valor de um atributo, o **emycin** examina todas as regras que fornecem um valor. A regra escolhida será aquela cuja conclusão é calculada com o factor de certeza mais elevado. Se nenhuma regra fornece um valor superior a 20%, este é considerado ainda desconhecido.

Este esquema de comportamento (regras de inferência, factor de confiança, triplos O-A-V, encadeamento para trás, etc.) foi copiado por muitos sistemas (independentes do domínio), da primeira geração.

4.4.1.II) expert- [Hayes-Roth 83],[Oli, 84]

expert realizado por Rutgers no Stanford Research Institute(SRI). Este sistema é tal como **emycin** apropriado ao diagnóstico e classificação e representa uma aproximação alternativa. Possui uma base de dados de hipóteses e observações e uma base de regras de <<decisão>> que relaciona estas entidades. O seu raciocínio é orientado pelo objectivo de <<classificar>> objectos. Define-se a classe das hipóteses (isto é, das conclusões admitidas) e introduzem-se as regras de decisão. Três tipos de regras são implementáveis e permitem: de factos inferir factos; de hipóteses inferir factos e de hipóteses inferir hipóteses. Durante a consulta, o sistema adquire todos os dados (<<observações>>) de que necessita e na base dos quais decide qual a hipótese mais admissível.

Todas admitem factores de confiança para exprimir o grau de certeza reposto pelo perito na inferência em questão. Contrariamente ao **emycin**, o **expert** examina as regras ordenadamente, seleccionando de cada vez a regra com o mais elevado grau de confiança.

A estratégia do **expert** revelou-se mais eficaz que a do **emycin** nos casos de diagnóstico médico, em que as <<observações>> são os sintomas e as <<hipóteses>> são as doenças.

III) kas- [Hayes-Roth 83],[Oli, 84]

kas, realizado por Reboh no Stanford Research Institute(SRI). Derivado do Prospector, um outro célebre sistema pericial, baseia-se numa rede de inferência. Os seus nós são factores de decisão derivados das regras de inferência. Uma regra estabelece como certos factores influenciam um outro factor. A cada factor fica associada uma medida da própria evidência, análoga à <<certeza>> atribuída por **emycin** às suas próprias conclusões. O **kas** começa por absorver informações do utilizador. Cada informação provoca uma propagação de probabilidade no interior da rede, de factor em factor. No final desta fase, o sistema escolhe a hipótese que parece a <<mais provável>>, na qual concentra esforços (depois de ter raciocinado em encadeamento directo), colocando-se, então, o problema de verificar a exactidão (ou antes, a <<evidência>>) por meio de perguntas dirigidas ao utilizador (isto é, raciocinando em encadeamento inverso).

O **kas** difere do **emycin** na forma como trata a falta de informações. Enquanto o **emycin**, na dúvida, pede para serem acrescentadas regras, o **kas** dota as suas regras com dois factores de confiança: um para estabelecer quando a verdade da solução depende da verdade da premissa e outra para estabelecer o inverso. Com efeito, ambas as considerações são úteis para exprimir um juízo: só que <<se neva é Inverno a 80%>>, mas também só que <<se não há folhas secas no chão a 80% não é Outono>>. Por outro lado, o **kas** mistura encadeamento directo e encadeamento inverso e estrutura as regras de inferência segundo uma rede semântica. Foi aliás inicialmente apresentado como um editor interactivo de redes semânticas (associativas), e essa é a sua principal função. Cria redes de conhecimento expressas em proposições ou regras, percorre-as, modifica-as, e acrescenta ou apaga nós.

No essencial, o **kas** é um editor de três tipos de redes: a rede de inferência (para exprimir o conhecimento activo que conduz à convicção), a rede semântica (normalmente usada para exprimir relações dinâmicas entre objectos) e a rede taxonómica (para exprimir classificações estáticas dos objectos). A representação sob forma de rede é uma extensão dos triplos <<O-A-V>> adoptada pelo **emycin**.

4.4.1.IV) **ops** - [Forgy 81], [Hayes-Roth 83], [Oli, 84].

ops-5 desenvolvido por C. Forgy na Universidade de Carnegie Mellon. É o instrumento de auxílio para a construção de sistemas periciais baseados em regras e com ele foi escrito o R1, o sistema pericial para a configuração de hardware da Digital. O **ops-5** pretende ser uma linguagem de uso geral, aplicando regras de produção que operam com as expressões existentes na sua memória de trabalho. O sistema de produção do **ops-5** consta de uma memória de produção que contém as regras de inferência e de uma memória de trabalho que contém o estado do sistema sob a forma de cópias atributo-valor.

O raciocínio do **ops** é estritamente data-driven, ou seja, guiado pelos dados presentes na sua base de dados. O antecedente de uma regra é o conjunto de valores que certos atributos devem assumir para que a regra possa ser considerada verdadeira. O consequente é a acção de interpretar quando a regra é verdadeira.

Dado um estado inicial do sistema, é seleccionada uma das regras que resulta verdadeira e segue-se a acção especificada. O estado assim produzido (pois a acção acrescentou valores ou modificou os existentes) aparece subposto ao mesmo procedimento. À medida que o ciclo funciona a memória de trabalho expande-se e o seu conteúdo representa uma solução parcial, ainda não completa, mas definitiva.

Possui, no entanto, alguns defeitos notáveis: não fornece explicações sobre o próprio funcionamento; não dispõe de métodos para representação do conhecimento incerto; não procura a melhor solução em absoluto; não permite a tradicional programação por procedimentos.

Sobre este último, a prática tornou populares algumas astúcias de programação. Por exemplo, para controlar a execução do **ops** utilizam-se "goal" artificiais, cuja finalidade consiste em dirigir a atenção do programa para aquela direcção e que são cancelados logo que desnecessários. Assim, o equivalente a reclamar um procedimento consiste em fazer verificar uma regra que prevê, entre as suas acções, a geração do "goal" que fará nascer a regra relativa a esse procedimento. Isto demonstra como e quando a falta dos modos de programação tradicionais pode tornar confortáveis os programas de IA. O **ops** é um sistema para ser usado apenas por especialistas de IA, não possuindo facilidades de interacção com um não iniciado.

4.4.1.V) rosie- [Fain 82],[Hayes-Roth 83],[Oli, 84]

A linguagem **rosie** foi desenvolvida (pela equipa de Frederick Hayes-Roth em 1981 na Rand Corporation.

Aqui as asserções são, frases em inglês, que devem pertencer a uma das cinco construções reconhecidas pelo sistema facto que o torna bastante amigável. Com a sua gramática em inglês é possível descrever o próprio modelo do mundo sob a forma de relações entre elementos.

A elaboração de tal modelo de relações é feita por meio de conjuntos de regras. Os conjuntos de regras generativos destinam-se a definir um conjunto de elementos, enquanto os conjuntos de regras de reconhecimento se destinam a reconhecer o n-uple dos elementos que satisfazem uma certa condição.

Misturando diferentes tipos de conjuntos de regras podem construir-se sofisticadas estratégias de controlo.

Apesar de ter alguns atractivos o **rosie** peca sobretudo pela impossibilidade de acesso às suas regras para modificação.

4.4.2 segunda geração

Uma segunda geração, que surgiu mais recentemente, adopta já métodos de represenção e de raciocínio muito variados. São ambientes de programação completos, expressamente feitos para o desenvolvimento de sistemas periciais. Cada um destes SGSP permite resolver problemas numa perspectiva diferente. Normalmente são optimizados em determinadas situações, as quais dependem do método de representação mais indicado e do tipo de raciocínio que se pretende realizar sobre o conhecimento. A escolha correcta de um SGSP significa uma determinada perspectiva para todo o resto de um projecto.

4.4.2.1) kee- [Martins, 87],[Martin, 88]

kee sistema desenvolvido em 1983 pela IntelliCorp de Mountain View, empresa formada em 1980 por Edward Feigenbaum e que foi a primeira empresa de IA do mundo. **kee** é derivado dum sistema utilizado em Stanford chamado **units**.

É um ambiente completo que reúne diversas metodologias usadas há anos nos laboratórios de IA:

- representação do conhecimento por frame;
- programação object-oriented (tipo Smalltalk);
- taxonomia hereditária;
- raciocínio inferencial;
- raciocínio data-driven;
- programação funcional LISP;
- interface gráfica, etc.

A unidade fundamental da representação do conhecimento é a frame e chama-se unit. Cada uma é composta por slots, os quais podem ter uma base de regras ou uma sequência de instruções, ou simplesmente um valor. As unit estão organizadas hierarquicamente e cada uma especifica quais os atributos que devem ser transmitidos por via hierárquica, comunicando por envio de mensagens. Uma mensagem, por exemplo, pode exigir que a unit calcule o valor de um slot, para o que pode ser necessário executar regras e estas podem exigir os valores de outros slots, etc.

A interface gráfica de **kee** permite ao engenheiro construir e modificar a base de conhecimento e todo o sistema pericial de forma simples e rápida. Por exemplo, um conjunto de regras é esquematizado no ecrã como uma lista de elementos que mostra todas as ligações existentes entre as várias regras; um editor gráfico adequado permite realizar modificações directamente ao nível visual.

4.4.2.II) **art**- [Williams 84],[Martins, 87],[Martin, 88]

O inventor do **art** pode ser considerado Chuck Williams, que já tinha realizado Hearsay-III, o último da dinastia dos Hearsay. A linguagem de representação do **art** permite utilizar quer esquemas quer regras de inferência. O **art** reconhece quatro tipos de regras de inferência:

- regras absolutas (produzem um novo estado na base de dados);
- regras hipotéticas (produzem somente situações possíveis, susceptíveis de rectificação);
- vínculos (impedem certas situações)
- convicções. O **art** é capaz de gerir diversas hipóteses concorrentes simultaneamente.

O **art** está fortemente orientado para encadeamento directo, uma técnica de raciocínio muito poderosa mas também muito dispendiosa, pois necessita de provar todas as vias consentidas pelos factos actuais. Para isso o **art** utiliza o algoritmo (pattern matching) mais veloz do mundo, o RETE de Charles Forgy. Este, uma vez presentes os dados e os factos na memória de trabalho, encontra todos os seus "match" nos membros esquerdos da regra.

Comparando o **art** e o **kee**, diremos que o **art** constrói as suas estruturas de dados enquanto em **kee** as bases funcionais são a estrutura de dados. Isto porque o limite de **kee** reside no facto de originalmente ter nascido como método múltiplo de representação do conhecimento e só depois lhe foram adicionados métodos de raciocínio .

4.4.2.III) Personal consultant plus(PC)- [Martin, 88]

Personal consultant plus é uma ferramenta da Texas Instruments que para a representação do conhecimento usa conjuntos de regras rotuladas como frames . Só pode trabalhar com uma única base de conhecimentos ao mesmo tempo. Usa tipos especiais de informação, as "propriedades" que controlam vários aspectos de uma consulta.

As "propriedades" são associadas às frames aos parâmetros e às regras. Como exemplo as frames têm propriedades tais como **Metas, Ocorrências, e Dados iniciais**. Metas são listas de parâmetros que constituem o objectivo das frames.

As consultas ao Personal consultant plus envolvem a instanciação das frames e determinar as suas metas e ocorrências designa o número de vezes que as frames podem ser instanciadas durante uma consulta.

Por exemplo, se uma frame representa carros num parque, então usando a propriedade das ocorrências, o sistema pericial pode especificar o número máximo de carros que podem ser estacionados no parque. A propriedade dos dados iniciais representa informação que é fornecida pelo utilizador do Sistema Pericial antes do sistema tentar determinar quaisquer valores de parâmetros dentro de uma frame.

Por exemplo, se um parâmetro numa frame representa cidades para serem visitadas, o sistema pericial deve usar a propriedade dos dados iniciais para determinar o mês ou estação, que podem influenciar o valor desse parâmetro.

Entidades, conceitos ou situações (associadas com o domínio) são representadas em frames através do uso de parâmetros. Os parâmetros podem, por sua vez, ser organizados em grupos para guardar conjuntos de parâmetros associados logicamente. O sistema cria automaticamente um grupo de parâmetros para cada frame.

Cada parâmetro tem também um valor numérico, chamado o factor de certeza. O factor de certeza é uma medida da confiança que temos em que o valor do parâmetro esteja correcto.

O terceiro elemento importante da base de conhecimentos do Personal consultant plus é a regra. A finalidade primária das regras é inferir o valor dos parâmetros e comunicar a informação ao utilizador.

As regras são utilizadas para representar as heurísticas e a perícia dos sistemas periciais.

4.4.3 Sistemas com capacidade de indução

Nos últimos anos tem-se assistido ao aparecimento de uma nova geração de **SGSP** tais como **ExpertEase** e **Timm**, não menos poderosos, mas de mais simples utilização, pois evitam que o utilizador se tenha de preocupar com a representação do conhecimento e os métodos de raciocínio. Ao contrário de **kee** e **art** não se destinam sómente ao engenheiro do conhecimento mas sim para o utilizador não especialista de IA. Estas ferramentas são orientadas por exemplos, isto é, são capazes de induzir as regras de inferência que constituem o sistema pericial, bastando para isso apresentar-lhes alguns exemplos, para que possam construir as regras correspondentes. Por outras palavras poderemos dizer que são capazes de aprender regras automaticamente.

4.4.3.1) ExpertEase [Chorafas 87]

O **ExpertEase** foi desenvolvido por Donald Michie na Edinburgh University. O utilizador não tem de escrever as regras que o programa utiliza para tomar decisões sobre os dados. O que o utilizador deve fazer é estruturar o material de forma que o programa possa inferir uma tabela lógica a partir dos exemplos. A habilidade de inferir regras a partir de exemplos pode ser útil para as pessoas que não conhecem a lógica por detrás de condições determinadas, ou eventos, mas suspeitam que os eventos podem estar relacionados e que a relação é consistente. Os pedidos de ajuda são atendidos rapidamente. O **ExpertEase** estimula o utilizador a listar todos os itens possíveis ou valores numéricos que possam aparecer em cada categoria de factores e na lista de decisões. O utilizador só precisa inserir cada item possível ou valor numérico uma vez.

Normalmente depois de os itens ou valores estarem listados, o utilizador desloca-se para um segundo ecrã do **ExpertEase** que lhe permite utilizar os exemplos. Uma vez que todos os nomes dos itens ou valores numéricos dos factores e resultados já estão no programa, o utilizador pode inserir o exemplo rapidamente e economizar tempo e esforço digitando somente uma ou duas letras do nome do item ou do valor numérico desejado. O programa preencherá o resto.

4.4.3.II) Timm- [Martin, 88]

O **Timm** foi desenvolvido da General Research, de McLean, Virgínia, nos inícios de 1980. É um sistema pericial baseado no computador que "pensa" sobre várias maneiras de resolver problemas quando a informação crítica é inadequada ou escassa. O **Timm** auxilia os utilizadores, passo a passo, até completar uma determinada tarefa. O utilizador não necessita de ser engenheiro do conhecimento ou programador de computadores.

Um sistema pericial construído com **Timm** pode ser utilizado para comunicar com outros sistemas periciais ou aceder a bases de dados. Na estrutura de representação do conhecimento, **Timm** permite a utilização de dados simbólicos e numéricos, variáveis discretas e contínuas, e factores de certeza. A herança de valores e decisões é possível entre sistemas ligados. Uma importante característica de **Timm** é a capacidade de executar raciocínio analógico para gerar as regras e assim chegar a conclusões.

Quando se inicia a utilização de **Timm** para a construção de um sistema pericial o utilizador tem de fornecer a seguinte informação:

- O nome do novo sistema pericial
- Uma lista das possíveis decisões que poderão ser alcançadas
- Uma lista dos factores que poderão influenciar a decisão alcançada

• Alguns exemplos ilustrativos nos quais o perito especifica as decisões correctas a tomar para uma combinação dada de factores.

4.4.4 Sistemas com arquitecturas "blackboard"

4.4.4.I)Hearsay-III- [Hayes-Roth 83],[Oli, 84].

Do sistema de compreensão da fala **Hearsay-II**, Lee Erman e Chuck Williams derivaram, em 1980, um instrumento geral para a construção de sistemas periciais baseados no modelo do blackboard: **Hearsay-III**. As hipóteses escritas no blackboard são agora unit que, por sua vez, são condições de <<classes>> das quais só herdam a propriedade. Os arcos que ligam a unit chamam-se <<desempenhos>> e as várias unit ligadas por desempenhos compõem um gráfico no blackboard. Existem dois blackboards, o verdadeiro e próprio ao domínio e o "scheduling" destinado unicamente a orientar as operações das várias fontes de conhecimento(ver cap.2 fig).

As fontes de conhecimento começam com a descrição da situação em que vão operar (quando tal situação se verifica, o scheduler activa-as ou escreve-as no blackboard de scheduling).

Um tipo especial de unit é o dos <<conjuntos de escolha>> (choice set). Um conjunto de escolha representa um grupo de interpretações alternativas, das quais deve ser escolhida uma única. Também estes conjuntos de escolha, tal como as outras unit, podem ser classificados hierarquicamente, permitindo, assim, diversos níveis de hipóteses.

4.4.4.II)Age- [Hayes-Roth 83],[Oli, 84]

Em 1979, Penny Nii realizou em Stanford um sistema chamado **age**, que pode ser utilizado para a construção de hipóteses de forma incremental e também **age** é baseado no modelo de blackboard. O engenheiro do conhecimento que resolve servir-se do **age** deve, antes de mais, fornecer o conjunto das hipóteses possíveis, cada com uma hierarquia de <<valores considerados>>, e depois o elenco das fontes de conhecimento, cada uma com uma base de regras de inferência, tendo associada a situação em que a base deve ser activada e também a estratégia de raciocínio a adoptar na avaliação das regras. Cada fonte de conhecimento tem como objectivo indicar quais são os valores reais e, assim, tornar plausíveis as hipóteses que contemplem tais valores entre os seus valores considerados.

O **age** permite configurar o próprio sistema escolhendo os componentes mais apropriados.

Análise rápida e comparativa entre as shells existentes com mais promenor .

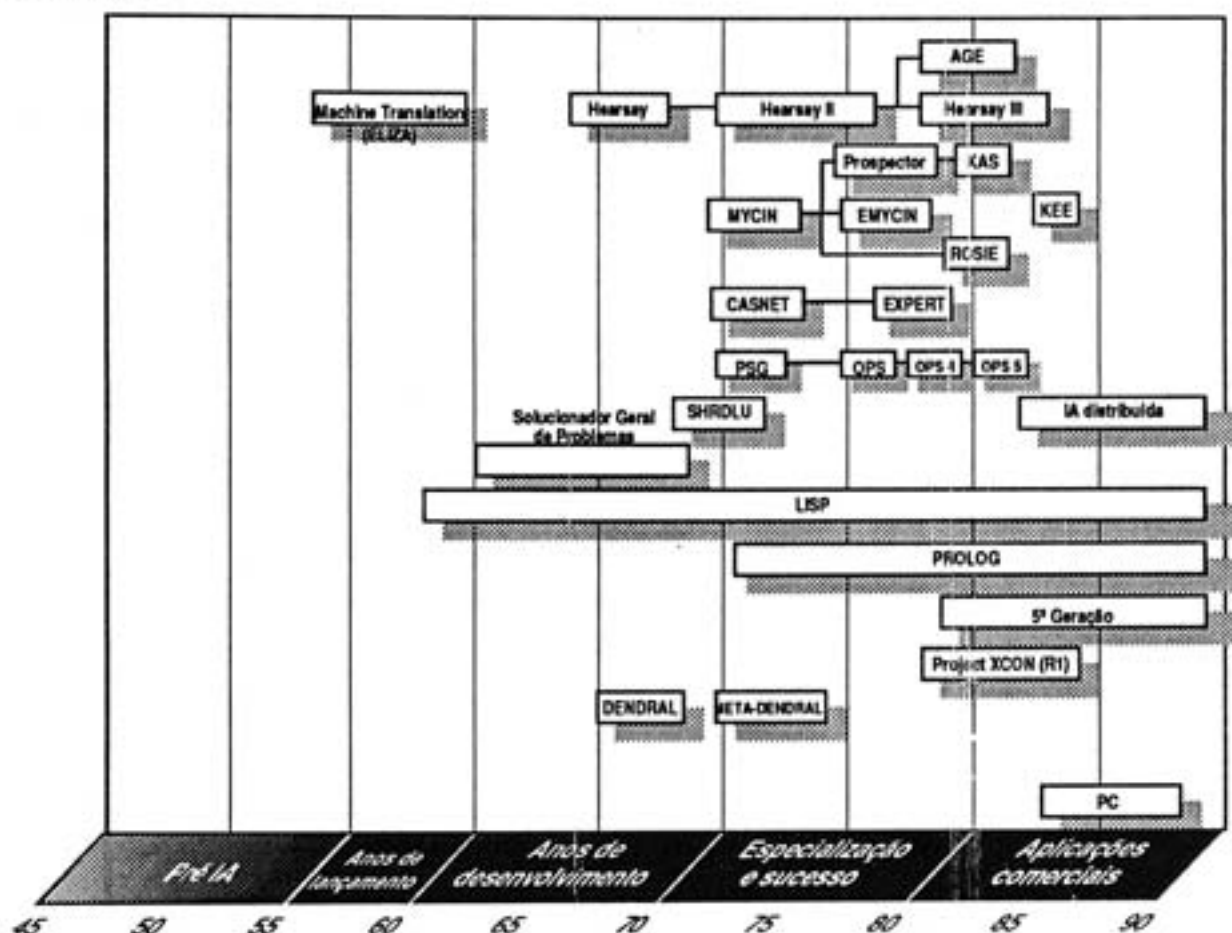


figura 4.4

Os SGSP são as principais ferramentas disponíveis actualmente para tarefas de engenharia do conhecimento. Factores a ter em consideração na avaliação dum SGSP:

- A classe de resolução de problemas que o instrumento permite realizar: diagnóstico, suporte para decisões, planificação, classificação, etc.
- Métodos de representação do conhecimento: regras de inferência, rede semântica, frames ou outros.
- Estratégias de controlo: encadeamento inverso (demonstrar a veracidade de um facto), encadeamento directo (deduzir algo dos factos disponíveis), blackboard (mais raciocínio cooperante).

- Métodos para o tratamento do conhecimento incerto: probabilidade bayesiana, fuzzy logic, etc.
- Configuração: tipo de linguagem (C, LISP, PROLOG, ETC.) e tipo de hardware (computador pessoal, workstation, general purpose, etc.).
- Gestão da base de conhecimento: instrumentos disponíveis para o engenheiro do conhecimento criar, modificar e verificar a base de conhecimento.

4.4.5 terceira geração

O apostar nos sistemas de inteligência artificial distribuída não é uma preocupação de hoje. Consultando os relatórios das workshops realizadas sobre IAD, e alguns artigos, encontramos comunicações e trabalhos com preocupações fundamentais no tema.

São ambientes de programação com características que representam um salto qualitativo importante e nos permitem afirmar que se trata afinal de uma terceira geração que interliga, gere e facilita a cooperação entre sistemas ampliando a sua utilidade e eficácia.

Alguns sistemas implementados com os princípios da IAD

4.4.5.I)-Controlo de tráfego aéreo utilizando técnicas de IAD [McArthur, 84]

Planeamento

. Os agentes podem decidir quando devem satisfazer os seus objectivos, quando devem agir e quando devem deixar os outros agir.

Problemas

- . Acumulação de informação
- . Mudanças dinâmicas no ambiente
- . Quando devem os agentes actualizar o seu modelo

Cooperação

- . Quando fornecer ou pedir informação
- . Que tipos de dados enviar ou que acção realizar
- . Actuação dinâmica, capaz de suspender o planeamento para receber informações e, eventualmente, alterar o plano em execução.

4.4.5 II)-Resolução de conflitos por negoaciação [Sycara, 88].

Sistema encarregado de negociar com vários agentes. Existe um agente único responsável pelas "conversações" e que segue duas estratégias possíveis:

- a) Usando argumentos que possui (case-based reasoning), demove os agentes a aceitarem a proposta apresentada tentando diminuir a importância dos pontos apresentados como não aceitáveis;
- b) Reformula a proposta de modo interactivo

4.4.5.III)-Arquitectura de um Sistema Pericial que suporta cooperação entre agentes heterogéneos levemente acoplados [Giovannini, 87].

Sistema que suporta vários agentes heterogéneos mas que não são autónomos. Cada sistema é especialista num determinado domínio, não sabe nada sobre o domínio dos outros agentes, só resolve sub(problemas) na sua área de competência.

Existe um agente especial responsável pelo controlo, isto é, sabe qual a especialidade dos diferentes agentes e é responsável pela divisão em subproblemas e desempenha a função de schedulling. A comunicação entre agentes faz-se através desse agente especializado nas comunicações e resume-se à troca de mensagens muito simples como parâmetros, resposta, ou indicação de que necessita de ajuda.

Não há qualquer restrição temporal ou de qualquer outro género de recursos na resolução dos problemas.

4.4.5.IV)-Hecodes [Zhang, 89] - Prestemos um pouco mais de atenção a este projecto pois, pelo que me foi possível encontrar numa revisão da literatura sobre este tema, é o mais avançado e completo neste campo e permite a cooperação entre sistemas periciais computadorizados.

Hecodes (HEterogeneous COoperative Distributed Expert Systems) é utilizado para realizar a cooperação entre sistemas periciais heterogéneos distribuídos. **Hecodes** consiste de um sistema de controle centralizado, um sistema blackboard, e módulos de processamento. Todas as partes de **Hecodes** foram especificadas (e implementadas) utilizando as técnicas de programação utilizadas em bases de conhecimento e portanto elas são independentes do domínio. **Hecodes** pode integrar Sistemas Periciais (SPs) variados e pré existentes utilizando tipos de cooperação também variados. Os SPs podem ser heterogeneos no que diz respeito à estratégia de controle, representação do conhecimento e linguagem de programação. Um protótipo foi implementado utilizando Common Lisp numa rede de máquinas Dec Vax algumas das quais estavam a 60 milhas de distância.

Este trabalho é um passo importante em direcção ao teste de gestão de sistemas periciais distribuídos e heterógeneos.

Um utilizador pode submeter uma tarefa a qualquer dos nós de **Hecodes** (isto é, o nó de controle ou qualquer nó de um SP) através de um módulo de interface.

Cada SP resolve somente as subtarefas dentro da sua capacidade. Quando encontra uma subtarefa que não pode realizar (podem ser várias) aplica o scheduler, requerendo a sua intervenção para esta subtarefa ser resolvida. O scheduler é um meta-Sistema Pericial que conhece (através do blackboard) tudo acerca dos SP locais.

Quando o scheduler aceita o pedido de uma aplicação que requiere a solução de uma subtarefa de um SP (ou de um utilizador) A, encaminha esta subtarefa para um SP apropriado B, depois de transformar a subtarefa mostrando a sua heterogeneidade. Os sistemas periciais podem diferir nos seus mecanismos de inferência e nos métodos de representação do conhecimento. Uma vez que o SP B dá o resultado desta subtarefa ao scheduler, o scheduler informará o agente original A dos resultados após as transformações que forem necessárias.

Com vista a conseguir cooperação entre sistemas periciais, foi adoptada uma arquitectura distribuída com controle centralizado.

O Sistema Pericial, o Sistema de Controle e o Sistema de Blackboard podem ser considerados como nós numa rede de computadores ligados por um sistema de telecomunicações. Num mesmo computador podem ser localizados mais do que um nó .

Cada Sistema Pericial possui um domínio específico de conhecimento, tem a capacidade de planificar e resolver problemas independentemente e a capacidade de requerer e oferecer assistência a outros SP (através do sistema de controle). Cada módulo de processamento é um interface entre um sistema e um nó de controle e é responsável pelo controle da cooperação e comunicação entre sistemas periciais com os quais está directamente ligado.

O Sistema Blackboard armazena informação estruturada para o controle do sistema e informação partilhada pelos SPs. O sistema de controle é responsável pela gestão da cooperação e comunicação entre todos os SPs e também pela gestão em tempo-real de **Hecodes**. Para diminuir os custos de comunicação entre o sistema de controle e o sistema blackboard, estes dois sistemas estão colocados no mesmo computador e considera-se ser um nó simples, porque o sistema blackboard é essencialmente subsistema de informação para o nó de controle.

Cada subsistema em Hecodes é composto essencialmente de três módulos funcionais:

- o módulo gestor
- o módulo de comunicação
- o módulo interface entre homem e máquina.

Um sistema de controle é um sistema blackboard tornando-se num nó simples de distribuição do mesmo módulo interface e módulo comunicação.

Todos os módulos utilizam os princípios das bases de conhecimentos, cada módulo é associado com uma base de meta-conhecimentos, e isso significa que Hecodes é um sistema de ferramentas independente do domínio.

O módulo scheduler no sistema de controle é o coração do **Hecodes**. Com os outros 4 módulos e dependendo dos conteúdos da base de meta-conhecimentos (a qual armazena todo o controle de informação para todos os sistemas periciais) pesquisa a globalidade do sistema e gere a cooperação e comunicação entre os sistemas periciais com vista a optimizar o endereço das tarefas no **Hecodes**. O sistema blackboard é controlado pelo módulo gestor. Todas as operações para o blackboard (como procura, leitura e escrita) necessárias por outros nós devem ser enviadas pelo gestor do blackboard.

O módulo de comunicação providencia os meios importantes de cooperação entre SPs. A comunicação entre sistemas periciais deve ser feita através do módulo de comunicação no nó de controlo.

Um módulo de processamento é um interface entre um sistema pericial e o nó de controlo e gere um único sistema pericial.

Controlo Centralizado

A arquitectura de controlo centralizada de **Hecodes** tem várias vantagens como segue:

a) É conveniente para armazenar controlo de informação. Acedemos ao armazenamento do controlo de informação, assim como à informação acerca de todos os sistemas periciais cooperantes e aos estados em processamento que foram armazenados uma única vez no nó de controlo.

b) É relativamente fácil pesquisar tarefas e subtarefas para a resolução de problemas para uma execução eficiente. O nó de controlo tem uma visão global do sistema que o habilita a escolher como e quando procurar tarefas num dos sistemas periciais. A informação necessária ao scheduler inclui as prioridades entre subtarefas, o chamar cada um dos SPs e o estado presente dos SPs.

c) O gestor em tempo real do sistema é simplificado, especialmente no tratamento de falhas, e na gestão da transformação das heterogeneidades.

d) É fácil adicionar um novo SP ao **Hecodes** porque necessitamos de modificar somente a informação no nó de controle e num dos módulos de processamento.

Existem, com certeza, desvantagens no controle centralizado em sistemas distribuídos como seja o engarrafamento da informação. Contudo em **Hecodes** a maior parte das operações podem ser localizadas, isto é, vários SP sózinhos podem realizar determinadas subtarefas envolvidas na resolução de um problema, o que implica uma frequência de comunicação baixa.

RESUMO

Hecodes é a ferramenta charneira na construção de sistemas periciais distribuídos. Os domínios de aplicação potencial do **Hecodes** são os que requerem solução cooperativa de problemas. Tais domínios incluem o diagnóstico médico, explorações petrolíferas e outras, planeamento e controlo militar e previsão meteorológica.

É fácil construir um sistema pericial distribuído prático, usando SPs pré-existentes ligados pelo **Hecodes**, porque foram usadas, para a sua implementação, técnicas de programação de bases de conhecimento (independentes do domínio). Quando precisamos de adicionar um novo SP ao **Hecodes**, obtemos simplesmente uma cópia do módulo de processamento, modificamos o conhecimento base nele e o meta-conhecimento base no scheduler. Os SPs a serem equipados com **Hecodes** podem usar uma variedade de representações de conhecimento e de estratégias de controlo, que podem mesmo ser implementadas em diferentes linguagens de programação. Portanto, **Hecodes** é uma poderosa e flexível ferramenta para a construção de sistemas periciais distribuídos.

Do modo como está a ser correntemente implementado, o **Hecodes** compreende várias características importantes:

- Sustenta várias formas de cooperação;
- Evita "deadlocks";
- Efectua transformações de formatos heterogéneos e modelos de raciocínio inexacto;
- Permite que o sistema seja dinamicamente organizado;
- Proporciona uma forte capacidade de comunicação;
- É fácil de expandir.

5- O S.G.S.P. COM BASES DE CONHECIMENTO COOPERANTES

5.1 INTRODUÇÃO

Neste capítulo pretende-se apresentar detalhadamente o trabalho efectuado que consiste de um **gerador de sistemas periciais com bases de conhecimento cooperantes**, que foi desenvolvido para um computador pessoal e transportável .

O estado da arte actual assenta essencialmente na experiência de criação e utilização de sistemas periciais com conhecimentos em domínios muito específicos e limitadas capacidades de inferência.

Daí a razão e o interesse de estudar sistemas que permitam maior generalidade e flexibilidade.

O domínio científico deste trabalho é a inteligência artificial, mas existe uma forte componente de Engenharia (do conhecimento, de software, de programação), atendendo a que se procurou uma implementação efectiva de um protótipo.

Com este projecto pretendeu-se conseguir uma assimilação das técnicas já dominadas na arte da concepção e construção de "shells" e avançar nalguns aspectos:

- Partindo do "Know How" existente no núcleo de IA da F.E.U.P. foi implementada, num equipamento transportável, uma "shell" que permite a cooperação entre bases de conhecimento, simulando uma consulta em simultâneo às bases de conhecimento existentes e previamente geradas.

- Em alternativa aos habituais interfaces em linguagem natural, foram exploradas as facilidades concedidas pela interface gráfica e de gestão de janelas que nos proporcionam os computadores **apple** (mais propriamente o **Apple macintosh SE**) e assim se conseguiu um mais agradável e fácil diálogo com o utilizador.

O instrumento base utilizado foi a programação em lógica mais especificamente "**lpa mac-prolog**" e o protótipo foi programado utilizando um computador **macintosh SE**.

5.2 POSSIBILIDADES E CARACTERÍSTICAS

O protótipo implementado, no essencial, é capaz de duas tarefas fundamentais:

GERAÇÃO DE SISTEMAS PERICIAIS. — Integração e representação do conhecimento.

CONSULTA A SISTEMAS PERICIAIS GERADOS — Raciocínio e diálogo envolvendo comunicação e cooperação externa com o utilizador e interna entre as bases de conhecimentos.

Para isso recorre a técnicas que ainda constituem no momento campos de investigação:

- Capacidade de raciocínio, de diálogo orientado e de integração com facilidade de novos conhecimentos.
- Contrução dinâmica de "conhecimento-orientador", que facilita a busca da resposta?
- Representação e manipulação de raciocínios incertos.

Neste trabalho, ao implementar um protótipo de um **gerador de sistemas periciais com bases de conhecimento cooperantes**, tivemos sempre como meta o desenvolvimento da tecnologia que lhe está inerente, nomeadamente nos seguintes domínios:

- Sistema de geração modelar permitindo facilmente a criação de mais sistemas periciais e também a capacidade do sistema poder ser optimizado quando tal é considerado conveniente.

- Sistema de consulta exaustiva procurando sistematicamente todas as possibilidades de resposta, de modo cooperante, com o conhecimento interno das bases de conhecimento, que o utilizador está autorizado a consultar.

- O sistema permite uma ampliação para um grande número de Sistemas Periciais diferentes mesmo num computador pessoal e com memória bastante limitada.

- O sistema só necessita de armazenar em memória o "conhecimento" respeitante a uma "base de conhecimentos" de cada vez. As bases de conhecimentos sendo cooperantes, têm implementados sistemas de comunicação eficientes que lhes permitem, libertar o espaço de memória que ocupavam deixando apenas as mensagens e directivas para a "base de conhecimentos seguinte" prosseguir o processo de procura da solução pretendida.

- Quando existe impossibilidade de resposta do sistema com o auto-conhecimento incorporado nas bases de conhecimentos dos diversos sistemas periciais, verifica-se o recurso ao utilizador na tentativa de continuação do processo de inferência.

- O sistema recorre ao mundo exterior (utilizador) só em última instância. Se o conhecimento necessário para a resposta não está incorporado em nenhuma base de conhecimentos, o sistema de inferência pergunta então ao utilizador. Se este responder a esta subquestão, o processo de procura da resposta continua normalmente.

- Consegue-se assim obter uma ferramenta importante, quer para o perito (na criação e consulta), quer para todos aqueles que necessitarem de utilizar o sistema pericial, depois de produzidas as bases de conhecimentos pelo perito.

5.2.1 SUMARIO DAS CARACTERISTICAS IMPORTANTES

- Engloba uma teoria de modelização do domínio, em que o conhecimento assume a forma de conceitos, factos e regras.

- Adquire, interactivamente, as regras, os factos e todos os conceitos, propriedades e valores possíveis para o domínio em causa. Constrói automaticamente durante o processo de geração de uma "base de conhecimentos" o seu "modelo de auto-conhecimento - **mac**" (meta-conhecimento).

- Utiliza mecanismos de herança de valores, pelo que as gamas de valores dos atributos da entidade-pai poderão ser herdadas.

- Recorre a múltiplas bases de conhecimento que são logicamente disjuntas, mas que funcionam como agentes cooperantes, sendo a

comunicação entre bases de conhecimento feita por troca de directivas e de mensagens.

- Constrói automaticamente, durante o processo de consulta, para cada uma das "bases de conhecimento" o que ela deve saber sobre as outras, ou seja o seu "modelo de conhecimento da comunidade - **mcc**" (meta-conhecimento).

- Incorpora desde já um módulo rudimentar do que poderá vir a ser um modelo de utilizador. Nomeadamente os seguintes aspectos estão já contemplados:

- O utilizador deverá ter autorização de consulta;
- O utilizador só pode aceder à informação das bases de conhecimento para que foi autorizado;

- O utilizador é objecto de tratamento em igualdade de circunstâncias com os restantes sistemas periciais durante a sessão de consulta. Quando da criação dos modelos de conhecimento da comunidade para cada um dos agentes em presença também é criado um para o utilizador. Assim quer os agentes quer o utilizador dispõem de conhecimento(informação) sobre o que sabem os outros agentes presentes na sessão de consulta e, por consequência, o que pode ser perguntado e a quem.

- Permite ainda o cálculo de um factor de credibilidade, mesmo que para a conclusão tenha recorrido à interacção recursiva entre distintas bases de conhecimento.

- Gera relatórios pormenorizados sobre a forma como obteve a conclusão, especificando as várias bases de conhecimento a que recorreu.

5.3 DESENVOLVIMENTO DAS CARACTERISTICAS DO SISTEMA

5.3.1 Teoria de modelização do domínio.

Os seres humanos elaboram o seu pensamento diário de forma eficiente. O ideal seria que o computador fosse dotado de expressões lógicas correspondentes ao conhecimento humano de senso comum. De qualquer modo, seria sempre difícil extrair conclusões de forma eficiente, tal como os humanos o fazem. Todos os estudos indicam que, para fornecer potência e rapidez comparáveis às do pensamento humano, o conhecimento deverá ser organizado por parcelas e essas unidades deverão existir em número tão grande quanto possível.

Esta metodologia surgiu das tentativas, da investigação em campos da psicologia, para explicar a resolução de problemas pelos humanos e das investigações sobre a compreensão da linguagem natural[Shirai 82].

Marvin Minsky, do MIT, elaborou uma «teoria geral» a partir destes estudos, que dava orientações para a representação do conhecimento e que foi adoptada em diversas experiências realizadas em vários locais. Estes sistemas utilizam conceitos que ainda hoje são universalmente seguidos tais como: enquadramentos, guiões, rede semântica. Estes conceitos permitem-nos estruturar o conhecimento, fragmentando-o em unidades de conhecimento ligadas entre si[Minsky 75].

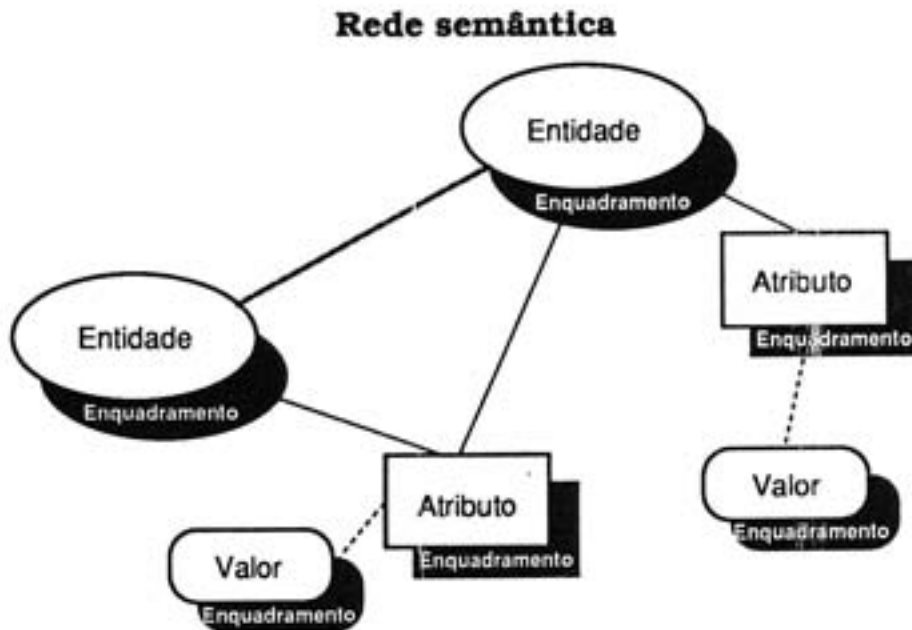


figura 5.1

São fundamentalmente estes princípios que estão presentes na teoria conceptual do domínio que adoptamos no sistema desenvolvido. Uma das vantagens de um modelo adoptando os princípios atrás expostos é a possibilidade de implementar mecanismos de herança. Em redes semânticas especiais que se transformam o conhecimento pode ser estruturado em diversos níveis com relações de ascendência e de descendência. Se tais relações forem utilizadas, então em vez de uma enumeração das propriedades de baixo nível, elas poderão depreender-se da unidade de nível superior. É isto que se chama herança de propriedades.

Definir a "Teoria" consistiu em organizar o conhecimento estrutural, que por sua vez contém o modelo conceptual do domínio. O conhecimento estrutural e o seu modelo de organização constitui o cerne de todo o sistema. O modelo conceptual está organizado como um conjunto de entidades que se relacionam. Uma entidade é caracterizada por um conjunto de propriedades ou atributos que são

comuns a todas as entidades dessa classe. Estão no sistema representados como enquadramentos (frames). O domínio é representado por uma rede de triplos - entidades, atributos e valores. As entidades de um domínio estão agrupadas em classes e a ligação na rede das entidades estabelece uma determinada hierarquia. Cada entidade tem associada a si um grupo de atributos. É também caracterizada pelas relações que tem com as outras entidades. Consideremos ainda os conceitos de valor. Associado a cada atributo, caracterizando-o, existe um conjunto de valores que pode tomar.

5.3.2 Aquisição do conhecimento.

Conhecimento - O conhecimento do domínio foi estruturado no sentido de apoiar o perito na organização do seu saber. Este terá de elaborar um modelo conceptual do seu domínio de conhecimentos. Na construção de um Sistema Pericial além do modelo, o perito terá de fornecer os factos, as regras e as relações. Para melhor elucidação de como funciona o SGSP vamos utilizar um exemplo que engloba a geração de quatro Bases de Conhecimento. Para começar e a fim de explicitar as opções adoptadas, vamos considerar a base de conhecimentos seguinte a que chamamos fauna_flora:

fauna_flora	regras	• Se grau de humidade igual grande e risco de erosão igual pouco e (equilíbrio de fauna igual bom ou equilíbrio de flora igual bom) então qualidade de microclimas igual notável FC = 0.7	
	factos	grau de humidade igual grande MC = 0.6 MD = 0.0 risco de erosão igual pouco MC = 0.3 MD = 0.0 equilíbrio de fauna igual bom MC = 0.0 MD = 0.2 equilíbrio de flora igual bom MC = 0.7 MD = 0.0	

figura 5.2

Além dos factos e regras representadas a base é ainda constituída pela descrição das seguintes entidades, atributos e valores:

entidades	atributos	valores
humidade	grau	grande
erosão	risco	pouco
fauna	equilíbrio	bom
flora	equilíbrio	bom
microclimas	qualidade	notável

5.3.3 Conhecimento Estrutural.

O conjunto de todas as entidades, atributos e valores e as suas relações mútuas constituem o conhecimento **estrutural**. É essencialmente uma rede semântica cujos nós são enquadramentos e permitem-nos representar o modelo conceptual. O conhecimento estrutural contém assim o modelo conceptual do domínio de uma base de conhecimentos e permite que o sistema se conheça a si próprio. As cláusulas unitárias prolog constituíram o tipo de representação adoptado no trabalho para guardar o conhecimento estrutural.

5.3.4 Conhecimento Extensivo.

Conhecimento extensivo ou factual engloba os factos do domínio. Os factos constituem a componente estática do domínio, e o conjunto de factos é habitualmente designado por base de dados.

5.3.5 Conhecimento Compreensivo.

As regras com as premissas, conclusões e relações mútuas constituem o conhecimento compreensivo.

As regras permitem-nos inferir novo conhecimento do domínio a partir do conhecimento já existente. Elas são uma peça importante para a resolução dos problemas por inferência.

5.3.6 Metaconhecimento

Está implementado um esquema que permite de modo dinâmico sintetizar os conhecimentos a que o SP está habilitado a responder cooperando com os outros sistemas em presença. Este conhecimento é denominado **autoconhecimento**.

Trata-se de Metaconhecimento pois o modo como é organizado, dá ao sistema uma capacidade introspectiva.

Está organizado em dois níveis:

- nível de uma base de conhecimentos (auto conhecimento). O que deve saber essa base de conhecimentos resulta da consulta de todo o outro conhecimento (estrutural, extensivo e compreensivo) que foi fornecido pelo perito, depois de devidamente organizado segundo determinadas regras.

- nível de uma comunidade de "Bases de Conhecimentos". Trata-se do conhecimento por um elemento da comunidade de tudo aquilo que cada um dos outros elementos da comunidade sabe. É o resultado da leitura e organização do conhecimento englobado nos modelos de autoconhecimento.

O conhecimento da comunidade de Bases de Conhecimento (agentes) durante uma sessão de consulta é:

- válido apenas durante uma sessão de consulta específica;

- diferente conforme o número de agentes presentes durante a sessão de consulta;

- trata-se de meta-conhecimento a que a "BC" recorre quando só por si não tem hipótese de responder à questão e precisa da cooperação das outras bases de conhecimentos.

Resumindo:

- A informação é essencialmente expressa por um conjunto de triplos constituídos por entidade, atributos e valores;

- A caracterização de uma entidade é também feita pelos atributos que lhe estão associados e pelas relações que tem com as outras entidades;

- O conjunto de entidades, atributos e valores formam uma rede semântica cujos nós são enquadramentos.

- Quer uma entidade, quer um atributo, quer um valor, constituem um enquadramento, que é um nó da rede semântica.

A base de conhecimentos fica assim estruturada de modo adequado. O modo como essa estruturação é feita cria as condições para uma utilização eficiente do conhecimento, pois permite conhecer o próprio conhecimento (Metaconhecimento). Esta característica vai proporcionar um bom desempenho do sistema na condução da pesquisa das soluções, na medida em que diminui a amplitude do espaço de pesquisa. Consegue-se deste modo uma influência do próprio conhecimento do domínio no processo de inferência.

A teoria de modelização do conhecimento do domínio tem uma generalidade suficiente, que lhe permitirá abranger uma vasta gama de domínios. Teremos contudo de reconhecer que, para alguns ramos de saber, ela terá as suas limitações.

5.4 Representação do conhecimento

Vamos esclarecer promenorizadamente como tudo está implementado, utilizando o exemplo da base de conhecimentos **fauna_flora** anteriormente apresentado.

Formalismos de representação a que recorremos:

- triplos O-A-V para representar o conhecimento factual;

```
facto(triplo(grau, humidade, igual, grande)-fc(0.6, 0)).
facto(triplo(risco, erosão, igual, pouco)-fc(0.3, 0)).
facto(triplo(equilibrio, fauna, igual, bom)-fc(0, 0.2)).
facto(triplo(equilibrio, flora, igual, bom)-fc(0.7, 0.2)).
```

figura 5.3

- regras de produção para representar o conhecimento compreensivo;

```
regra(1, triplo(qualidade, microclimas, igual, notável)-0.7,
[flora, erosão, humidade, fauna],
[[triplo(grau, humidade, igual, grande),
triplo(risco, erosão, igual, pouco),
triplo(equilibrio, fauna, igual, bom)],
[triplo(grau, humidade, igual, grande),
triplo(risco, erosão, igual, pouco),
triplo(equilibrio, flora, igual, bom)]]).
```

figura 5.4

- o auto-conhecimento (Metaconhecimento) foi representado utilizando clausulas unitárias prolog.

```
quem_sou(fauna_flora).
minha_fila(fauna_flora, fara).
que_concluo(fauna_flora, [(qualidade,microclimas)]).
que_sei(fauna_flora, [(grau,humidade), (risco,erosão), (equilibrio_fauna),
(equilibrio_flora)]).
descricao_capacidade(fauna_flora,(1,qualidade,microclimas),
[(equilibrio_fauna), (grau,humidade),(risco,erosão),(equilibrio_fauna)], . . . ).
```

figura 5.5

- rede semântica cujos nós são enquadramentos para representar o conhecimento estrutural (modelo conceptual do dominio);

```
termo(qquer, conj_valores, propriedades(qq, [], [], [])).
termo(int, conj_valores, propriedades(i, [-9999999, 9999999], [], [])).
termo(nat, conj_valores, propriedades(n, [0, 9999999999], [], [])).
termo(cadeia, conj_valores, propriedades(c, [], [], [])).
termo(simNao, conj_valores, propriedades(sn, [], [sim, s, nao, n], [])).
termo(natureza, dominio_conhecimento, propriedades(fc)).
termo(grau, atributo, propriedades(qq, [humidade], [])).
termo(qq, conj_valores, propriedades(qq, [], [], [])).
termo(risco, atributo, propriedades(qquer, [erosão], [])).
termo(equilibrio, atributo, propriedades(cadeia, [fauna, flora], [])).
termo(notável, atributo, propriedades(cadeia, [microclimas], [])).
termo(qualidade, atributo, propriedades(qquer, [microclimas], [])).
termo(microclimas, entidade, propriedades(instancia, [], [], [],
[qualidade])).
termo(microclimas, entidade, propriedades(_8265, [], [], [1], [notável])).
termo(flora, entidade, propriedades(_, humidade, [1], [], [equilibrio])).
termo(erosão, entidade, propriedades(_8265, [], [1], [], [risco])).
termo(humidade, entidade, propriedades(_8265, [], [1], [], [grau])).
termo(fauna, entidade, propriedades(_8265, [], [1], [], [equilibrio])).
```

figura 5.6

Os enquadramentos serviram para estruturar o modelo conceptual do domínio. O prolog foi a linguagem utilizada para implementar um enquadramento à custa de uma estrutura prolog. A estrutura utilizada é constituída por predicados com três argumentos. O último argumento é também um predicado com vários argumentos.

Os enquadramentos para as **entidades** são constituídos por:

nome;

identificação;

características:

classe ou instância;

entidade ascendente;

nº das regras em que figura como premissa;

nº das regras em que figura como conclusão;

enumeração dos atributos.

exemplo:termo(humidade, entidade, propriedades(_, [], [1], [], [grau]))

Os enquadramentos para os atributos são constituídos por:

nome;

identificação;

características:

atributo_tipo;

enumeração das entidades a que está ligado.

nº das regras em que figura como conclusão;

exemplo:termo(notável, atributo, propriedades(cadeia, [microclimas], [])).

Os enquadramentos para os valores são constituídos por:

nome;

identificação;

características:

nome_tipo;

mínimo e máximo dos valores possíveis;

enumeração dos valores possíveis;

enumeração dos valores.

exemplo: termo(qquer, conj_valores, propriedades(qq, [], [], [])).

Conjuntos de valores pré-definidos.

O SGSP permite a definição de todos os tipos de valores que desejarmos, mas possui conjuntos de valores pré-definidos que lhe permitem validar os dados introduzidos:

INT - conjunto dos números inteiros.

NAT - conjunto dos números naturais.

SIMNÃO - Os elementos do conjunto podem ser SIM ou NÃO.

CADEIA - Os elementos são cadeias de caracteres.

QQUER - Os elementos são de qualquer tipo.

5.5 Motor de inferência

O motor de inferência permite ao SP inferir novo conhecimento a partir do já existente, ou seja integra os mecanismos necessários à resolução de problemas. O processo de inferência levado a cabo por este sistema é no fundamental um mecanismo de encadeamento inverso, incorporado na linguagem de programação prolog. As alterações relativamente ao mecanismo inerente ao prolog limitam-se a um interpretador de regras de produção funcionando passo a passo e que pode ser interrompido quando se pretenda. Esta possibilidade poderá ser aproveitada para um desenvolvimento futuro em multi-tarefa.

A introdução de raciocínio empregando conhecimento incerto obrigou a implementar, uma alteração ao motor de inferência que não pode parar quando atinge uma solução, mas tem de procurar todas as soluções possíveis e, posteriormente, combinar as várias evidências, nomeadamente na atribuição de um factor de certeza final quando esta solução é alcançada por mais de uma via.

5.6 Raciocínio Incerto

O sistema de raciocínio utilizando factores de certeza pode ser sintetizado do seguinte modo:

- A cada facto é associado um factor de certeza. Esse factor de certeza é expresso por dois valores:

- (MC) medida da confiança;
- (MD) medida da desconfiança.

Esse factor de certeza é introduzido pelo perito quando da geração da base e expressa a confiança do sistema nesse facto.

As regras contêm também factores de certeza que expressam o grau de confiança do perito em atingir a conclusão a partir daqueles antecedentes.

As regras têm a forma:

se p_1 F p_2 F p_3 F.... p_i então (Conclusão, F_c)

sendo F_c o factor de certeza associado à regra e F um conector lógico.

As premissas têm associado também um factor de certeza que é dado por dois valores(MC, MD).

Quando a regra é activada, o factor de certeza dos antecedentes influencia o factor de certeza final da regra e está implementado no sistema para as seguintes situações:

- Conjunção de premissas
- Disjunção de premissas
- Conclusão obtida por mais de uma regra

Para as combinações e propagação da incerteza vamos utilizar as seguintes quatro funções:

1) O corpo de uma regra pode consistir em uma conjunção de evidências derivadas de um conjunto de observações feitas pelo utilizador sendo elas próprias incertas ou imprecisas,. Chamando ao conjunto dessas observações p' , a combinação das medidas de "crença" e "descrença" é feita da seguinte maneira:

$$MC[p_1 \& p_2 \& p_3 \& \dots p_n, p'] = \min(MC[p_1, p'], MC[p_2, p'], \dots, MC[p_n, p'])$$

$$MD[p_1 \& p_2 \& p_3 \& \dots p_n, p'] = \max(MD[p_1, p'], MD[p_2, p'], \dots, MD[p_n, p'])$$

Se ☐	F ☐	MC = 0.6	MD = 0.3	} MC = 0.5 MD = 0.3	
e ☐	F ☐	MC = 0.5	MD = 0.2		
		min MC = 0.5	max MD = 0.3		

então ☐ F ☐

figura 5.7

2) No caso de ser uma disjunção

$$MC[p_1 \& p_2 \& p_3 \& \dots p_n, p'] = \max(MC[p_1, p'], MC[p_2, p'], \dots, MC[p_n, p'])$$

$$MD[p_1 \& p_2 \& p_3 \& \dots p_n, p'] = \min(MD[p_1, p'], MD[p_2, p'], \dots, MD[p_n, p'])$$

Se ☐	F ☐	MC = 0.6	MD = 0.3	} MC = 0.6 MD = 0.2	
ou ☐	F ☐	MC = 0.5	MD = 0.2		
		max MC = 0.6	min MD = 0.2		

então ☐ F ☐

figura 5.8

3) Força evidencial:

Sabe-se um $FC[h, p]$ dado pelo perito para a regra em causa, mas não se sabe qual o FC associado à regra, caso não se tenha a certeza de "p" estar presente. O utilizador deve exprimir então a sua convicção em "p" baseada nas observações "p'", fornecendo $MC[p, p']$ e $MD[p, p']$, que são combinados para formar $FC[p, p'] = MC[p, p'] - MD[p, p']$; este factor é por sua vez combinado com o FC original da regra (dado pelo perito na altura em que as regras foram construídas) da seguinte maneira:

$$a) MC[h, p'] = FC[h] * \max(0, FC[p, p'])$$

$$b) MD[h, p'] = FC[h] * \min(0, FC[p, p'])$$

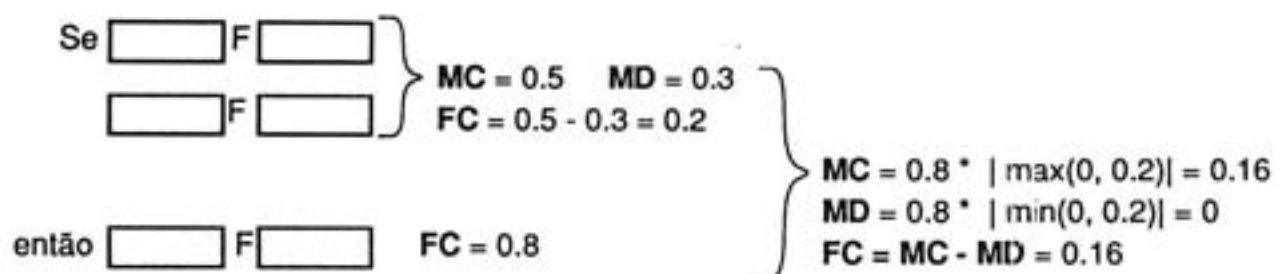


figura 5.9

A anterior função combinatória justifica-se de modo intuitivo, devido ao facto de a combinação feita poder representar realmente um reforço positivo ou negativo tanto maior dos valores prévios quanto maiores as medidas de crédito ou descrédito nas evidências que se vão combinando incrementalmente.

4) No caso de chegar à conclusão por mais de uma regra:

e_1 e e_2 são ambas evidências para h , podendo fazer parte eventualmente de um corpo de evidências maiores.

Dispomos dos valores de $FC(h, e_1)$. Quando se observa depois e_2 e se calcula $MC(h, e_2)$ e $MD(h, e_2)$, esta nova informação deve ser combinada com a já existente anteriormente:

$$MC[h, e] = \begin{cases} 0 & \text{se } MD[h, e_1 \& e_2] = 1 \\ MC[h, e_1] + MC[h, e_1] * (1 - MC[h, e_2]) & \text{nos outros casos} \end{cases}$$

$$MC[h, e] = \begin{cases} 0 & \text{se } MC[h, e_1 \& e_2] = 1 \\ MD[h, e_1] + MD[h, e_1] * (1 - MD[h, e_2]) & \text{nos outros casos} \end{cases}$$

Se uma Factor de Certeza de uma conclusão obtida a partir de uma regra é $FC_1 = 0.6$, e se a mesma conclusão é confirmada por uma segunda regra com $FC_2 = 0.4$ então o $FC = 0.6 + 0.4(1 - 0.6) = 0.76$

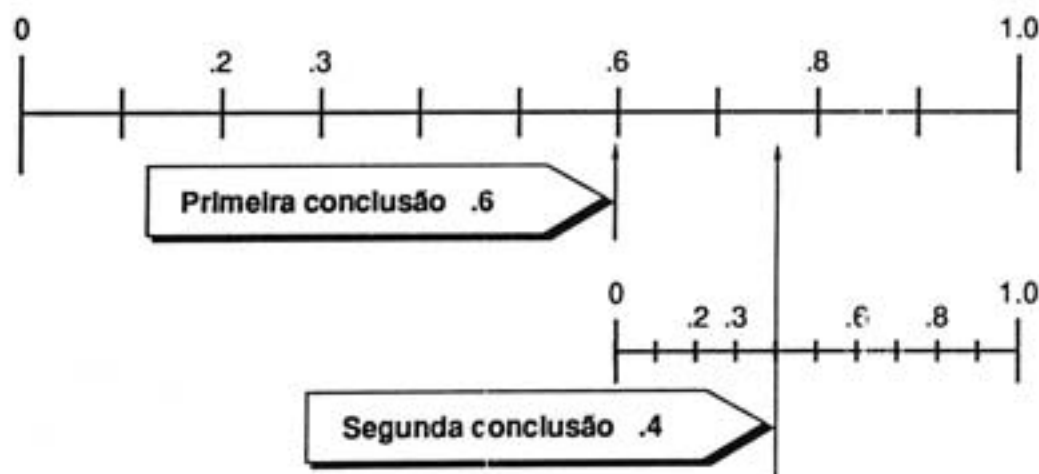


figura 5.10

5.7 O acesso aos diferentes ambientes.

O sistema usa uma interface baseada na ergonomia dos computadores apple disponibilizando um encadeamento de menus que permite ao perito e utilizador navegar nos diferentes ambientes que o sistema proporciona.

Navegabilidade no Sistema

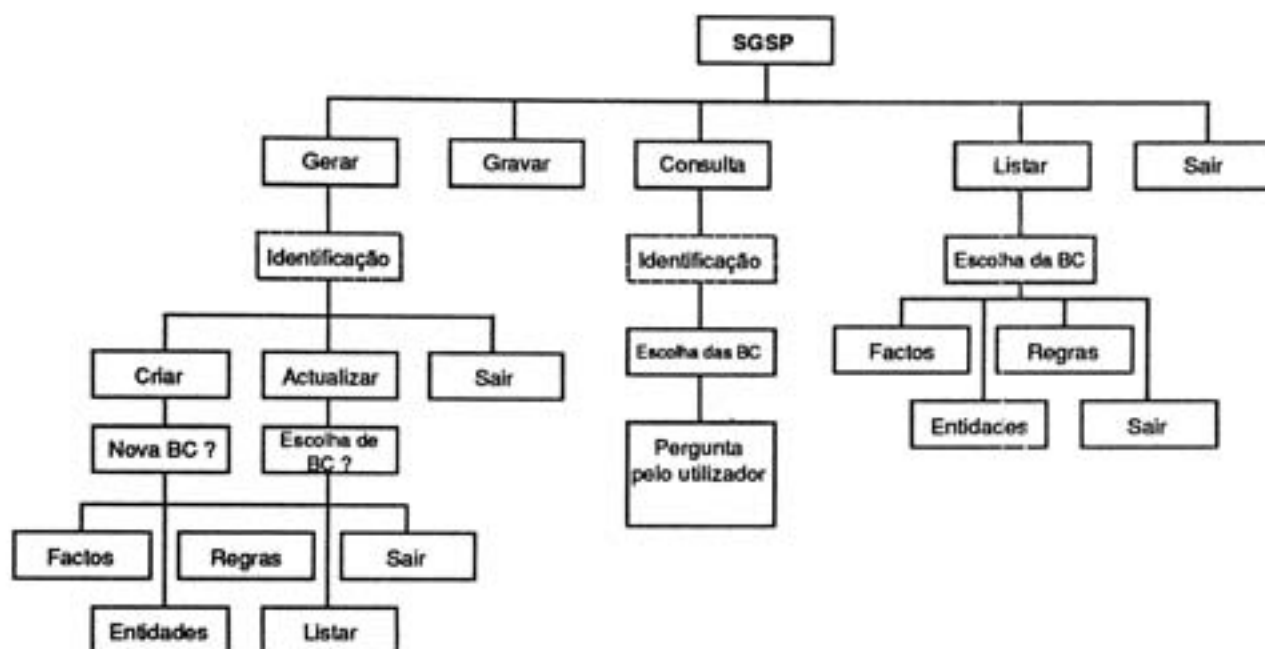


figura 5.11

A Geração de novos sistemas é apenas reservada a peritos mediante a utilização de palavra-chave pois só assim será possível garantir a coerência do modelo guardado.

A consulta também só é possível a utilizadores autorizados. A autorização de um utilizador é feita pelo gestor do sistema escrevendo num ficheiro o nome das bases de conhecimentos que ele pode consultar. Na figura 5.11 representamos estas protecções implementadas a ponteados.

5.8 GERAÇÃO DE SISTEMAS PERICIAIS

O processo de geração de novas bases de conhecimento é um processo reservado a peritos no domínio. Este terá previamente de organizar o seu conhecimento levando em linha de conta o "modelo do domínio" que o sistema aceita, mas a partir daí o processo é interactivo.

explicativo e permite assim com grande facilidade, quer a criação de novas bases de conhecimentos quer a utilização das já existentes. No manual do sistema em anexo serão dados mais pormenores.

Validação do conhecimento introduzido durante a fase de geração:

- Quer em termos de sintaxe, quer em termos de estrutura do conhecimento, o sistema faz a validação do conhecimento introduzido antes da sua aceitação. Como exemplo: se tentarmos introduzir um facto, sem ter definido previamente o conceito utilizado, o sistema vai-nos-lo pedir. Por outro lado, se tentarmos introduzir um conceito já existente, vai-nos avisar, pois interpreta tal facto como uma modificação do previamente existente. Outro tipo de validação é a dos valores possíveis dos atributos ou quais as gamas em que os podem tomar.

- Uma atenção especial para o agrupamento de entidades em classes, pois está previsto aquilo que designamos por mecanismos de herança de valores.

Quando da geração de uma base de conhecimentos são criados quatro ficheiros:

<BC>est - com o conhecimento estrutural.

<BC>ext - com o conhecimento extensivo.

<BC>com - com o conhecimento compreensivo.

<BC>mac - com o modelo do auto-conhecimento.

O Modelo de auto conhecimento(**mac**)

Este módulo incorpora o auto-conhecimento (resumo do que ela sabe):

Organizado com os seguintes tópicos:

quem_sou - nome da B.C.

minha_fila - ficheiro onde deve ir ler directivas e mensagens.

que_concluo - pares - atributos-conceito das conclusões das regras

que_sei - pares atributo-conceito dos factos

descriptor_capacidades - pares atributo-conceito dos factos e das conclusões das regras

5.9 CONSULTA DE SISTEMAS PERICIAIS

Vamos simular uma sessão de consulta cooperante às seguintes 4 bases de conhecimento anteriormente geradas:

ambiente	regras	• Se qualidade de solo igual arável e riqueza de subsolo igual grande e paisagens de local igual notáveis então tipologia de habitats_humanos igual qualidade FC = 0.7 • Se equilíbrio de vegetação igual bom e riquezas_florísticas de natureza igual muitas então tipologia de habitats_humanos igual qualidade FC = 0.4	
	factos	qualidade de solo igual arável MC = 0.6 MD = 0.4 riqueza de subsolo igual grande MC = 0.7 MD = 0.3 equilíbrio de vegetação igual bom MC = 0.8 MD = 0.2 riquezas_florísticas de natureza igual muitas MC = 0.7 MD = 0.4 paisagens de local igual notáveis MC = 0.9 MD = 0.2	
eco_humana	regras	• Se tipologia de habitats_humanos igual qualidade e equilíbrio de comunidades_humanas igual conforto_climático e densidade de população igual aceitável então resistência de comunidades animais igual muita FC = 0.9 MD = 0.2	
	factos	densidade de população igual aceitável MC = 0.6 MD = 0.3	
clima	regras	• Se qualidade de microclimas igual notável e estado de zona_climática igual homogénea e risco de acidentes_climáticos igual pouco então equilíbrio de comunidades_humanas igual conforto_climático 0.8	
	factos	estado de zona_climática igual homogénea MC = 0.6 MD = 0.2 risco de acidentes_climáticos igual pouco MC = 0.6 MD = 0.3	
fauna_flora	regras	• Se grau de humidade igual grande e risco de erosão igual pouco e (equilíbrio de fauna igual bom ou equilíbrio de flora igual bom) então qualidade de microclimas igual notável FC = 0.7	
	factos	grau de humidade igual grande MC = 0.6 MD = 0.0 risco de erosão igual pouco MC = 0.3 MD = 0.0 equilíbrio de fauna igual bom MC = 0.0 MD = 0.2 equilíbrio de flora igual bom MC = 0.7 MD = 0.0	

figura 5.12

Uma observação atenta ao quadro da figura 5.2 mostra de imediato que para uma resposta direccionada à conclusão da regra da base de conhecimentos *eco_humana* ela terá, para responder, de confirmar as premissas dessa regra que nas outras BCs em presença e do seguinte modo: a primeira premissa é confirmada pela base *ambiente*; a segunda premissa é confirmada pela BC *clima*, que vai ainda necessitar de recorrer à ajuda da BC *fauna_flora*.

Ao iniciar a consulta de modo dinâmico e temporário são imediatamente gerados os ficheiros que contêm o "modelo do conhecimento da comunidade" - «BC»*mcc* - sobre o conhecimento que interessa a cada uma das bases e que está disponível nas outras Bases de Conhecimento. Esses ficheiros têm uma estrutura que vamos exemplificar com dois exemplos:

5.9.1 Geração do conhecimento da comunidade

• caso do ficheiro *fauna_floramcc*

```
presentes(eco_humana).  
fila_dele(eco_humana, ecna).  
presentes(ambiente).  
fila_dele(ambiente, amte).  
presentes(clima).  
fila_dele(clima, clima).  
presentes(utilizador).  
fila_dele(utilizador, quser).  
interesses([clima], (qualidade, microclimas)).
```

figura 5.13

• caso do ficheiro **eco_humanamcc**

```

presentes(ambiente).
fila_dele(ambiente, amte).
presentes(clima).
fila_dele(clima, clma).
presentes(fauna_flora).
fila_dele(fauna_flora, fara).
presentes(utilizador).
fila_dele(utilizador, quser).
sabe_acerca([ambiente], (tipologia,habitats_humanos)).
sabe_acerca([clima], (equilibrio,comunidades_humanas)).
descreve_capacidades(ambiente, (tipologia,habitats_humanos), [], -, -, -).
descreve_capacidades(ambiente, (tipologia,habitats_humanos), [], -, -, -).
descreve_capacidades(clima,(equilibrio,comunidades_humanas),[], -, -, -).

```

figura 5.14

O módulo conhecimento_comunidade é um ficheiro ligado a cada elemento da comunidade de sistemas periciais e engloba conhecimento dos outros sistemas incorporados na mesma comunidade, o qual é criado dinamicamente por critérios que implicam que ele seja de interesse para uma interactividade e cooperação efectiva. A análise dos dois exemplos apresentados anteriormente, conjuntamente com o conhecimento incorporado nas quatro bases de conhecimento que estão presentes na sessão de consulta é esclarecedora do modo de funcionamento e critérios de criação do modelo de conhecimento da comunidade.

A sua geração dinâmica durante o lançar de um processo de consulta é feito mediante a leitura por cada um dos agentes, do auto-conhecimento dos restantes agentes presentes, seleccionando aquela informação que de um modo ou outro poderá complementar e enriquecer a informação que ele próprio possui.

Este conhecimento está organizado nos seguintes tópicos:

- **presentes** - base de conhecimentos presente
- **fila_dele** - ficheiro para onde lhe enviar correspondência
- **sabe_acerca** - factos conhecidos pelos outros sistemas.
- **interesses** - o parâmetro aponta a informação que esse sistema específico pode e deve enviar, para um outro agente, com vista a ajudá-lo a resolver uma determinada tarefa.
- **descreve_capacidades** - refere as capacidades dos outros Sistemas periciais em presença. Somente representaremos as capacidades com interesse para o S.P. específico.

5.9.2 Fases da resposta

Após o que foi dito, vamos sistematizar a resposta a uma questão, que implica o recurso a todas as capacidades disponíveis no sistema e entre elas: comunicação e cooperação

No quadro da figura nº 5.12 já está esquematizado o sistema de cooperação entre as bases de conhecimento. Essa cooperação pode ser visualizada também através da figura 5.15. Se lhe for endereçada pelo utilizador qualquer das seguintes questões :

- 1) resistência de VAR igual VAR.
- 2) resistência de comunidades_animais igual VAR.

O resultado é que, para o sistema, elas são equivalentes e, através de uma análise focada nos modelos de auto-conhecimento(mac), o sistema endereça a questão para a B.C. "eco_humana" que vai, por sua vez, recorrer a outras, pois tal torna-se necessário.

Podemos visualizar todo o processo de funcionamento interno através do seguinte esquema:

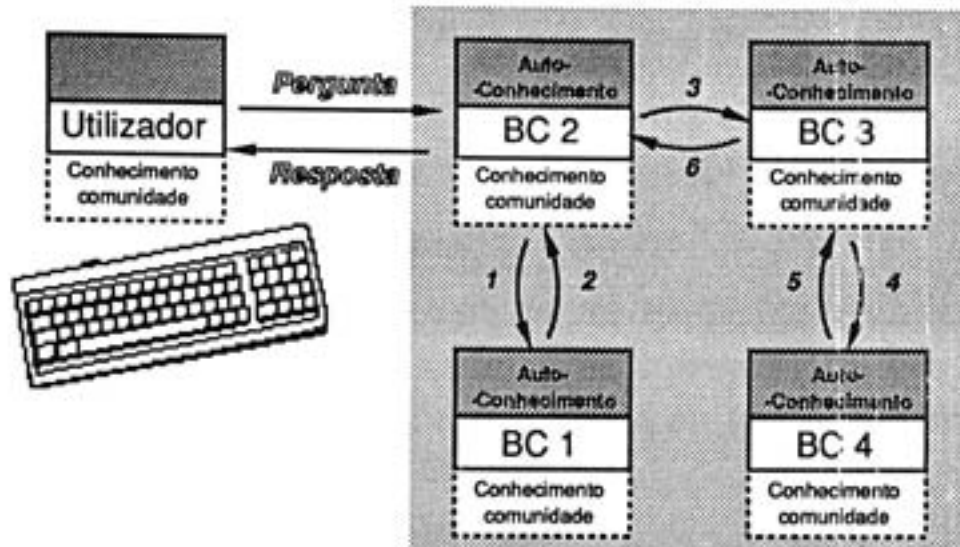


figura 5.15

Vamos também aproveitar o esquema para explicar todo o processo de inferência seguindo os passos lá indicados.

I - Pergunta colocada pelo utilizador ao Sistema.

A existência dos "modelos do auto-conhecimento" «BC»mac permite ao sistema endereçar a questão à BC2. Permite também ao sistema, a partir deles construir os "modelos de conhecimento de comunidade" «BC»mcc.

II - Resposta da BC2 ao utilizador que engloba os seguintes passos:

- 1) BC2 interroga BC1
- 2) BC1 responde a BC2
- 3) BC2 interroga BC3.
- 4) BC3 interroga BC4
- 5) BC4 responde a BC3
- 6) BC3 responde a BC2

5.9.3 MECANISMOS DE COOPERAÇÃO

O sistema incorpora como já foi explicado dois importantes módulos que correspondem a camadas de software que vão permitir e facilitar todo o processo de cooperação entre as bases de conhecimento [Jennings, 89] [Jennings, 90] [Ca, Oli, 90]:

O autoconhecimento — «BC»**mac**.

O conhecimento da comunidade — «BC»**mcc**.

Os dois módulos são indispensáveis para a troca de conhecimentos, visando a complementação da informação necessária à busca das soluções para as perguntas colocadas.

O motor de inferência global do sistema socorre-se deles e da informação aí contida como anteriormente foi explicado.

Mas para a cooperação não existiria se a comunicação entre os agentes não fosse eficiente.

5.9.4 COMUNICAÇÃO

Os mecanismos que permitem a comunicação e portanto tornam possível a cooperação são, entre outros, a troca de **directivas** e de **mensagens** entre bases de conhecimentos.

Mensagens - são respostas dadas por um agente a outro que lhe tinha endereçado uma pergunta.

Tipos de mensagens:

Desconhecido

Falhou

Em espera

Encontrada - encontrou a solução

Terminada - não tem mais soluções.

Directivas - são questões ou (ordens) enviadas por um agente para outro. Têm um formato idêntico às mensagens com algumas diferenças.

tipos de directivas:

Correndo

Corre a tarefa

Impossível

Tarefa impossível

Próxima solução

Não encontrei solução

Sem mais soluções

Tanto nas directivas, como nas mensagens, consta sempre quer o destinatário, quer o remetente. É este acto de troca de correspondência (que é escrita em ficheiros temporários por uma das bases e lida por outra) que permite a cooperação. Uma base de conhecimentos pode assim obter resposta para uma questão que desconhecia. Esta estrutura inspirada nos modelos "blackboard" permite-me chamar a atenção para algumas semelhanças existentes com os ditos modelos. Trata-se de uma solução a que em sentido figurado me permitirei chamar "blackboard sem blackboard", pois não sendo um blackboard de facto, muita da sua filosofia básica está presente.

5.10 PROJECTO E IMPLEMENTAÇÃO.

De seguida será feita a especificação do trabalho utilizando técnicas de Engenharia de Software. O projecto para a implementação foi feito utilizando uma análise de projecto estruturada, que permitiu assim desenhar e posteriormente construir um sistema modular. Pretendeu-se utilizar uma ferramenta poderosa que tornou mais clara e simples a compreensão do esquema de trabalho a desenvolver. Também agora para a explanação do trabalho e uma explicação global recorreremos ao mesmo tipo de técnicas[Ward, 87].

Análise Estruturada

Descrição funcional do sistema utilizando

Diagramas de Fluxo de Informação

O sistema contém cinco entradas e duas saídas, descritas seguidamente.

Entradas:

- **Conhecimento** - Informação organizada.
- **Pedido de descrição** Pedido de explicação do modo de funcionamento interno do sistema pericial.
- **Informação auxiliar**- Informação temporária complementar a pedido do sistema durante uma sessão de consulta.
- **Pedido de explicação** -Pedido de explicação referente a um pormenor do diálogo.
- **Consulta** - Pedido de resposta a questões nas quais o sistema substitui o perito.

Saídas:

- **Informação sobre a base de conhecimentos** - Esclarece o utilizador sobre o modo de funcionamento interno do sistema.
- **Resposta ao problema do utilizador** - Resposta do sistema a um pedido de consulta.

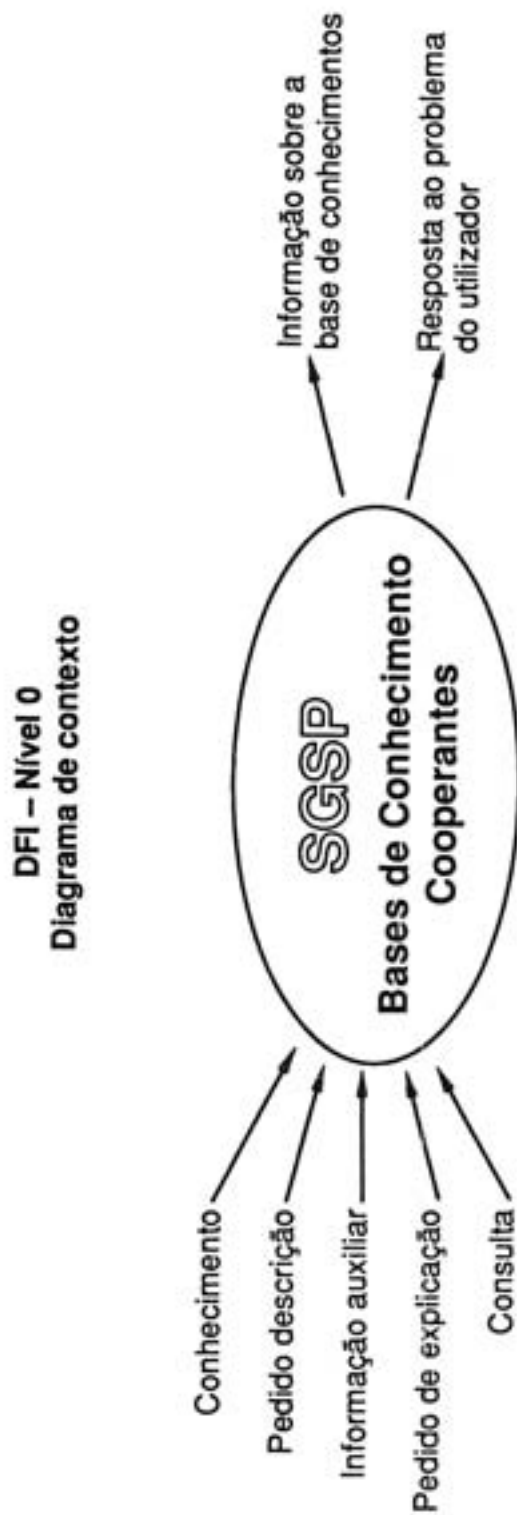


figura 5.16

Processos no Diagrama de Fluxo de Informação de nível 1

1.1 - Módulo de geração - O sistema gerador permite que o sistema aceite qualquer asserção de conhecimento nas formas previstas.

1.2 - Módulo de consulta - finalidade do Sistema Pericial é fornecer informações sobre o domínio, quer a peritos, quer a não peritos.

1.3 - Módulo de gestão de ecrans - Responsável pelo interface com o utilizador ,será essencialmente um interface gráfico com gestão de menus.

1.4 - Camada de cooperação - O sistema permite uma eficiente cooperação entre as várias Bases de Conhecimentos:

- Escolhe as Bases de Conhecimento Necessárias para responder.
- Atribui estratégias de Controlo às B.C.
- Passa informações de umas para as outras.

1.5 - Módulo de listagem - O objectivo fundamental a atingir é o de conseguir a listagem dos conhecimentos de uma Base de Conhecimentos. Por outras palavras, o Sistema lista as regras, os factos e as entidades, assim como os atributos e respectivos valores de uma Base de Conhecimentos.

DFI — Nível 1
Sistema gerador de sistemas periciais
com bases de conhecimento cooperantes

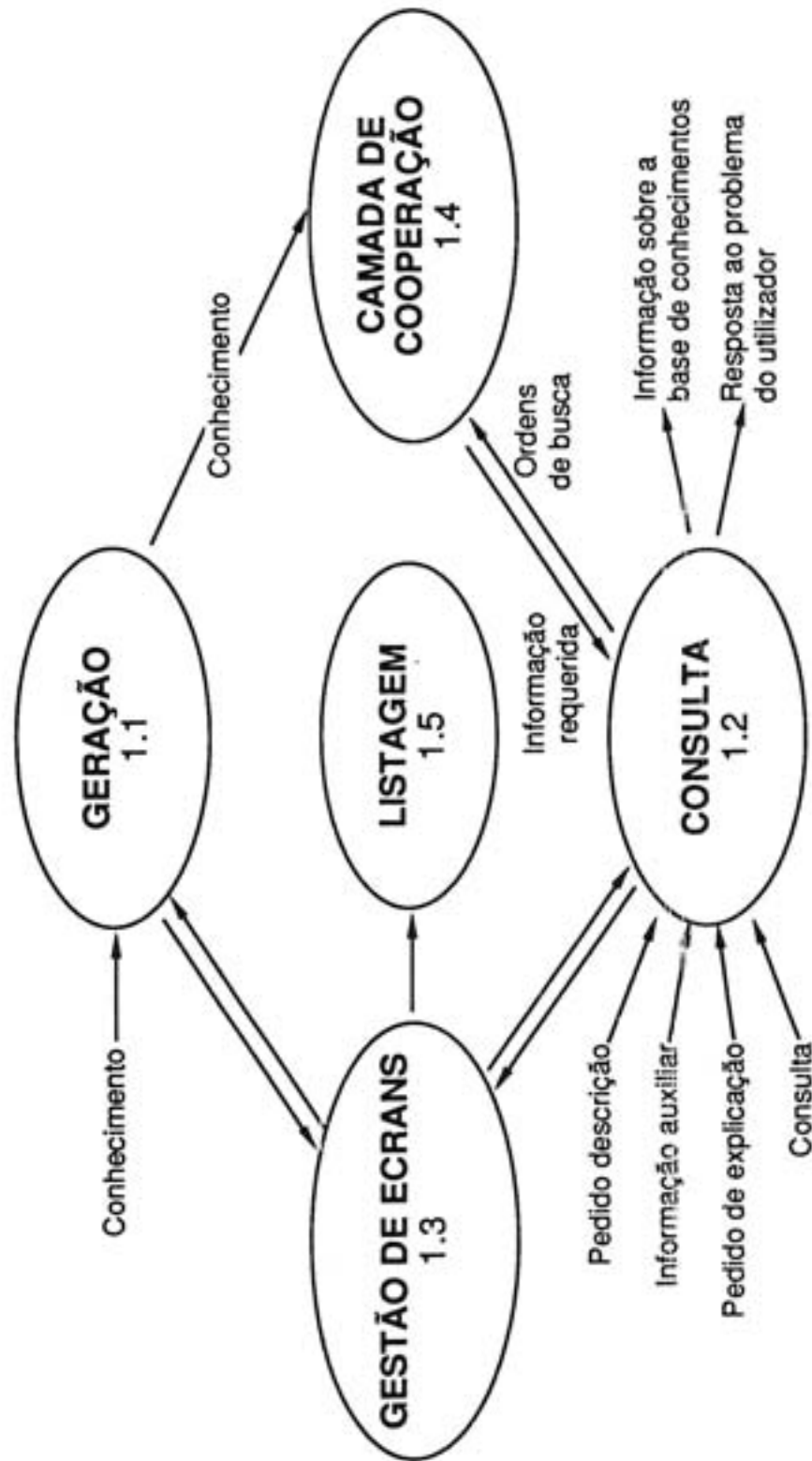


figura 5.17

1.1 - Módulo de geração

Responsável pela integração de novos conhecimentos

Por integração do conhecimento pretendemos que o sistema gerador seja capaz de estruturar os conhecimentos que lhe são transmitidos pelos peritos, incorporando-os nas suas bases de conhecimentos, de molde a posteriormente facilitar a consulta pelos utilizadores.

1.1.1 - Criação - Esta integração, quando feita pela primeira vez, cria um novo Sistema Pericial.

1.1.2 - Actualização - Quando o Sistema Pericial já existe, trata-se apenas da actualização das bases de conhecimento.

Quanto às formas de representação do conhecimento incorporadas nas bases de conhecimentos, procurou-se conseguir que o sistema tivesse uma capacidade suficientemente genérica de compreensão.

São utilizadas as seguintes:

- regras
- factos
- enquadramentos
- rede semântica

O sistema gerador permite que o sistema aceite qualquer asserção nas formas de conhecimento previstas, mas para tal, ele executa uma série de tarefas, primeiro, com o fim de testar a consistência do novo conhecimento com o modelo do domínio . Se ele for consistente, então o novo conhecimento será integrado.

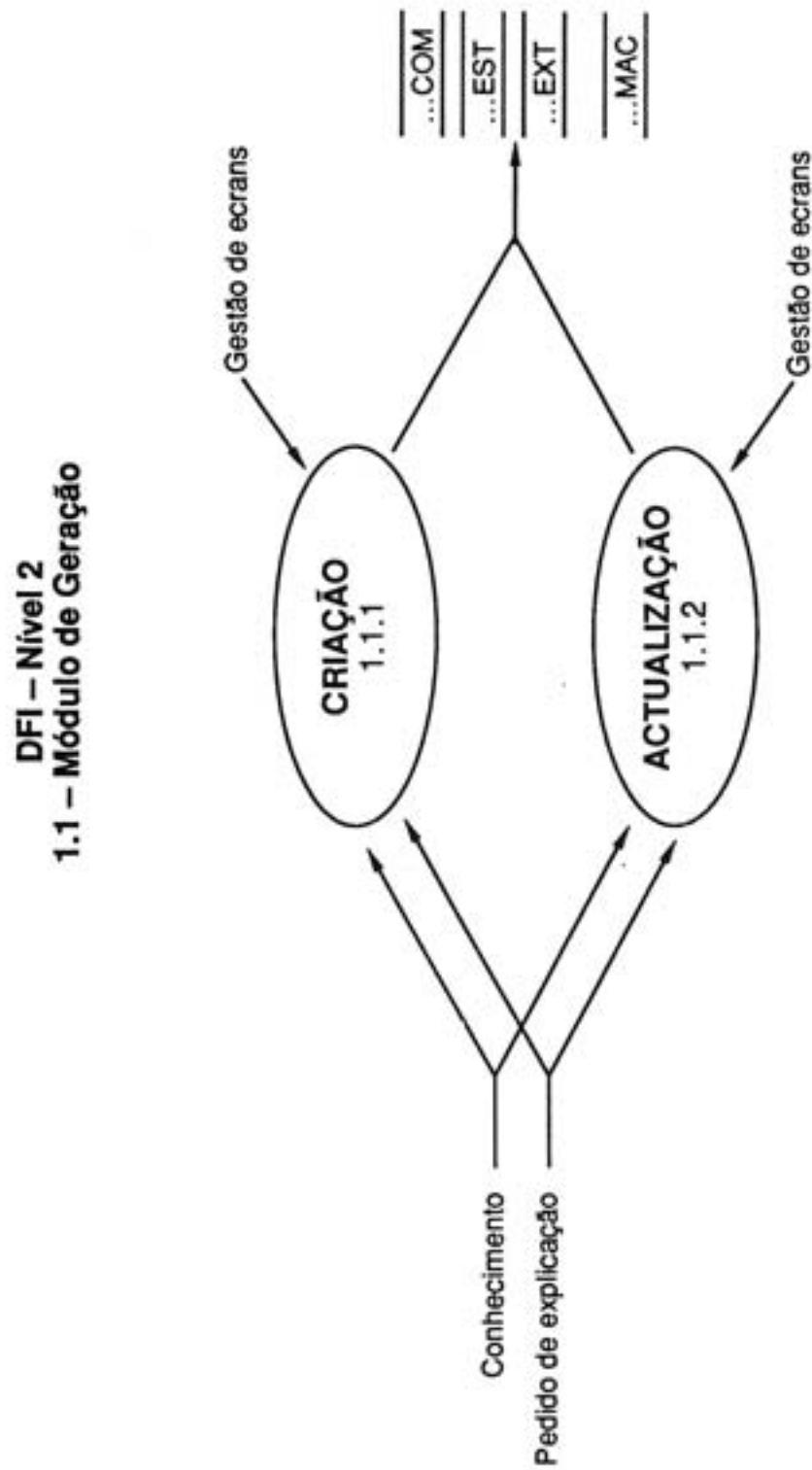


figura 5.18

1.2 - Módulo de consulta

Responsável por todo o tipo de respostas a dar ao utilizador que recorre ao Sistema Pericial como substituto do perito.

A fim de conseguir um agradável INTERFACE VISUAL COM O UTILIZADOR e desejando aproveitar as capacidades gráficas oferecidas pelos computadores da apple , para a consulta e todo o diálogo, não foi adoptada a solução mais habitual - interface de linguagem natural - mas um interface gráfico com gestão de menus .

1.2.1 - Descreve genericamente o funcionamento do Sistema elaborando um relatório do modo como se chegou à solução.

1.2.2 - Auxilia o utilizador durante o processo de consulta sempre que existirem dúvidas, nas fases essenciais da consulta.

1.2.3 - Informações fornecidas pelo consultor para complemento das respostas do sistema e, a pedido deste, para não bloquear o sistema de inferência.

1.2.4 - Processo normal de Consulta do Sistema Pericial.

Inclui a pergunta, a resposta e a explicação do modo como foi atingida.

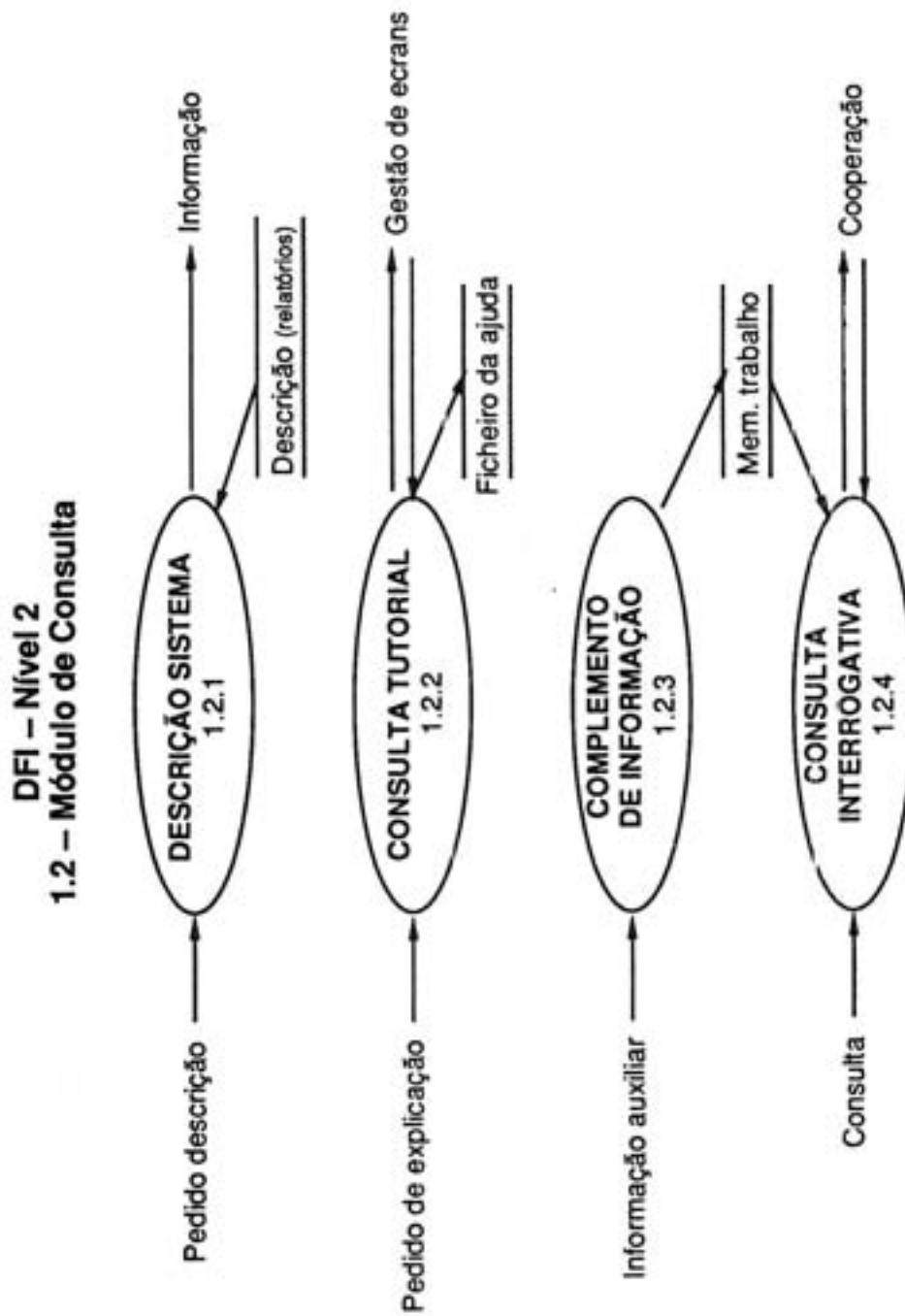


figura 5.19

1.3 - Gestão de ecrans

Com a finalidade de proporcionar um fácil diálogo com o utilizador, o sistema tem capacidade de explicar todo o processo de funcionamento, esclarecendo o utilizador quando este tem várias opções, assim como todo o seu esquema de raciocínio, visualizando nas janelas dos Sistemas Periciais as tarefas que eles vão executando em cada momento.

Pretende-se conseguir dar explicações convincentes e claras, seleccionadas (comprimindo a informação), de toda a dedução e mostrando os passos mais relevantes (elaborando relatórios).

Para isso ele está em perfeita sintonia com os restantes processos presentes.

Para uma compreensão total é necessária uma explicação estruturada completa, que inclui os detalhes que o sistema, quando atinge a resposta para uma solução que lhe tinha sido colocada, dentro de certos limites, é capaz de fornecer.

No caso de não conseguir responder a uma questão fornece explicações negativas, acerca das razões do falhanço de uma dedução, ou seja conseguimos localizar qual a regra ou qual a premissa que não foi confirmada.

DFI – Nível 2
1.3 – Gestão de écrans

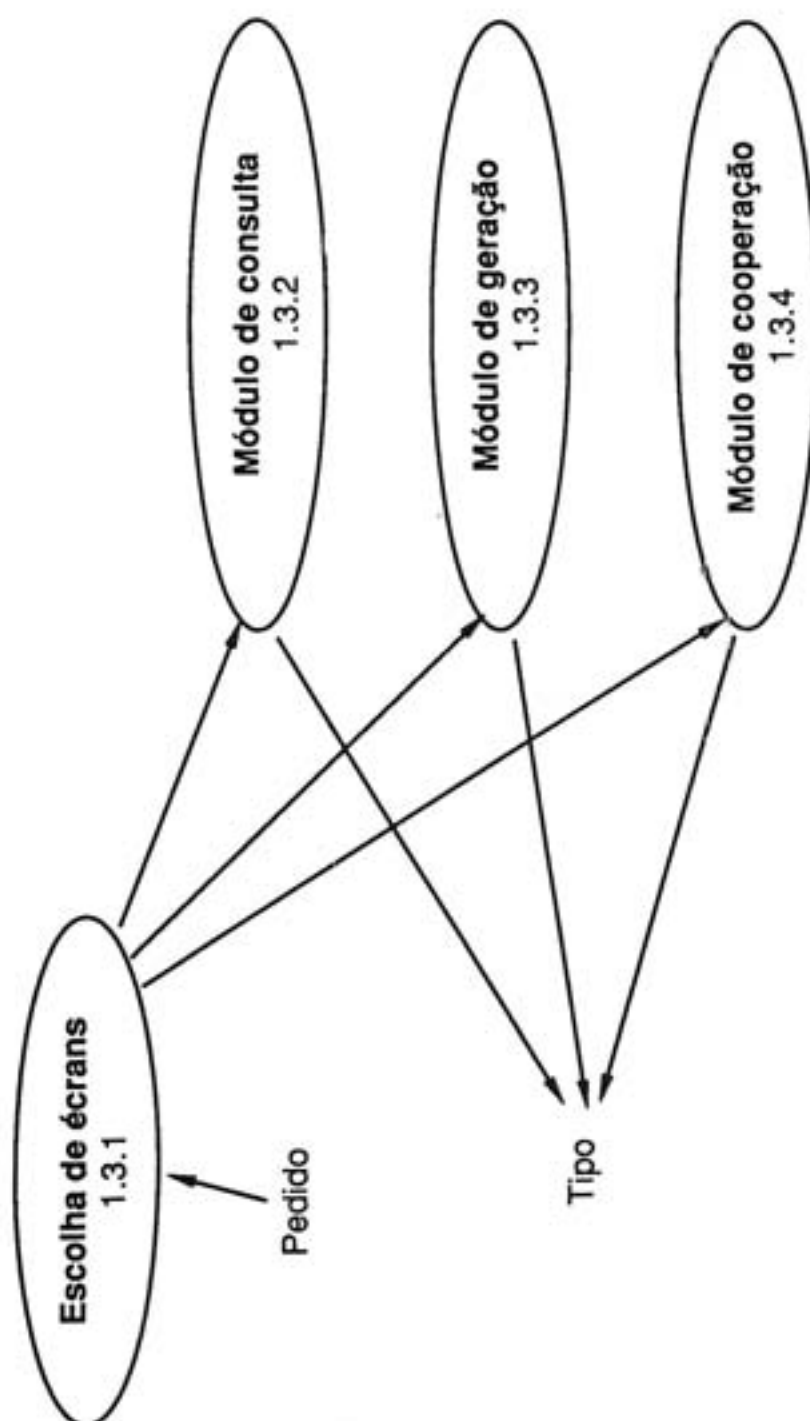


figura 5.20

1.4 - Camada de cooperação

1.4.1 - Módulo de Gestão Global

O Módulo de Gestão Global é o verdadeiro cérebro do sistema e tem uma influência central em todo o funcionamento gerindo quer a aquisição dinâmica de novos conhecimentos auxiliares para o processo de inferência, quer a pesquisa das respostas às questões colocadas.

Controla por sua vez o módulo de gestão dos Sistemas Periciais.

1.4.2 - Gestão dos Sistemas periciais

Atendendo a que necessitam de manipular grandes quantidades de conhecimento, os sistemas periciais são, quando vistos a partir deste nível bases de conhecimentos, ou seja, no fundamental, armazéns de dados contendo o conhecimento subdividido em várias categorias de que fazem a gestão. Essa tarefa é missão deste módulo.

1.4.3 - Construção do auto-conhecimento

Este módulo é responsável pela criação, durante a fase de geração, de um ficheiro contendo um resumo, segundo determinados critérios, do conhecimento incorporado (factos, e conclusões das regras) da Base de Conhecimentos.

Sendo o raciocínio uma acção como outra qualquer, só que interna em vez de externa, é necessário proceder-se ao seu planeamento e é isso que nos facilita o "modelo do auto-conhecimento".

Neste planeamento utilizamos não só o chamado auto-conhecimento mas também, como veremos em seguida, o "modelo do conhecimento da comunidade".

É o "modelo do auto-conhecimento" que vai permitir melhorar a eficiência do processo de consulta, pois é uma síntese e uma indicação acerca do conhecimento incorporado na Base de Conhecimentos.

1.4.4 - Construção do "modelo de conhecimento da comunidade"

Para isso este módulo cria, de modo dinâmico e temporariamente apenas, durante uma sessão de consulta, um ficheiro em que guarda informação sobre o que sabem as outras bases de conhecimentos que lhe poderão ajudar a completar o raciocínio.

A dedução consegue assim exibir eficiência, de modo a conseguir evitar que o processo de busca entre num ciclo ou num caminho sem controlo.

Consegue também deste modo carregar em memória apenas uma base de conhecimentos de cada vez e assim conseguir performances consideráveis em computadores pessoais e transportáveis.

A fim de dotar o SGSP com um sistema de raciocínio, o mais parecido possível com o raciocínio humano, houve o cuidado de implementar também um esquema de raciocínio sobre informação inexacta, dando ao utilizador especialista a possibilidade de introduzir informação imprecisa e conceitos vagos.

DFI – Nível 2
1.4 — Camada de Cooperação

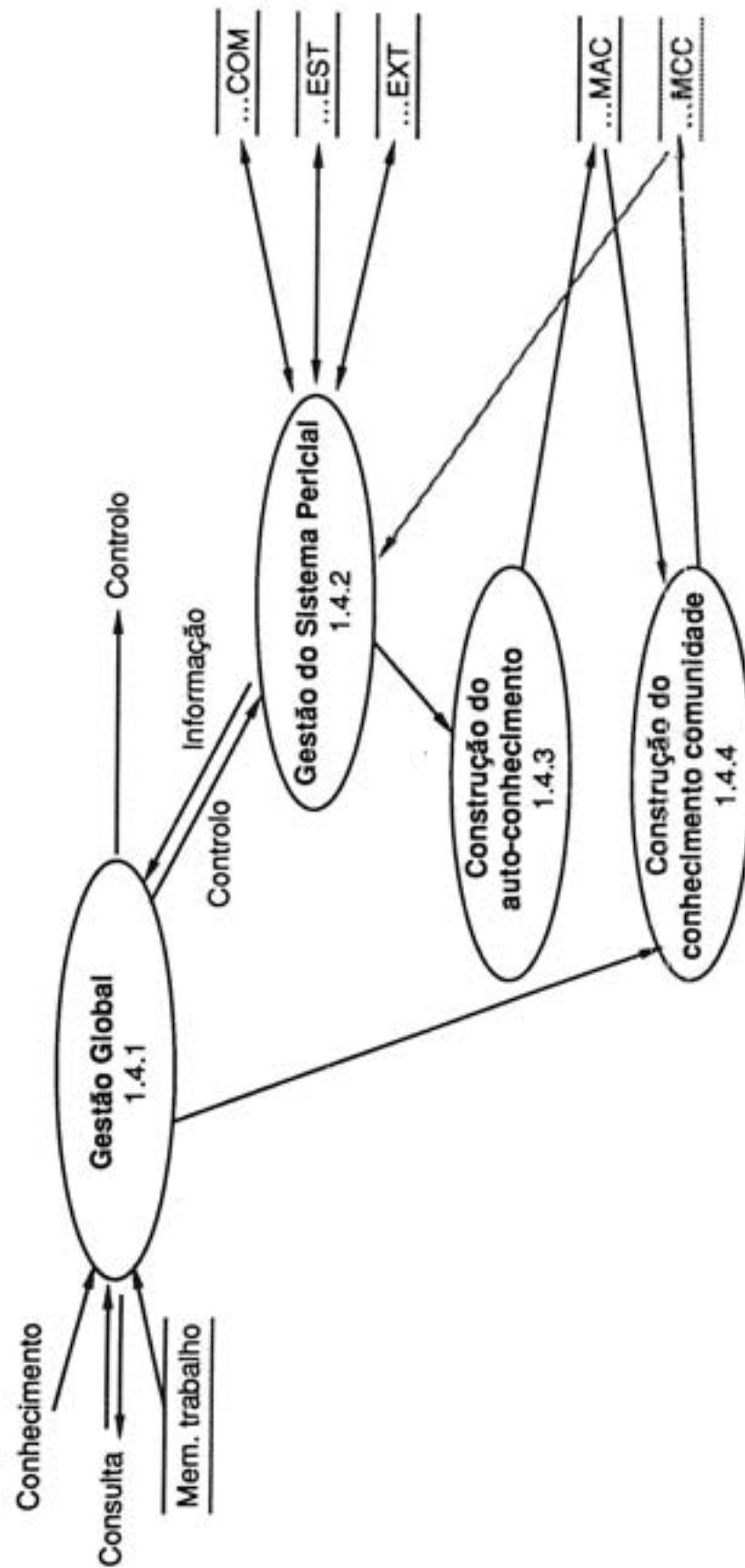


figura 5.21

ARQUITECTURA DE BASES COOPERANTES

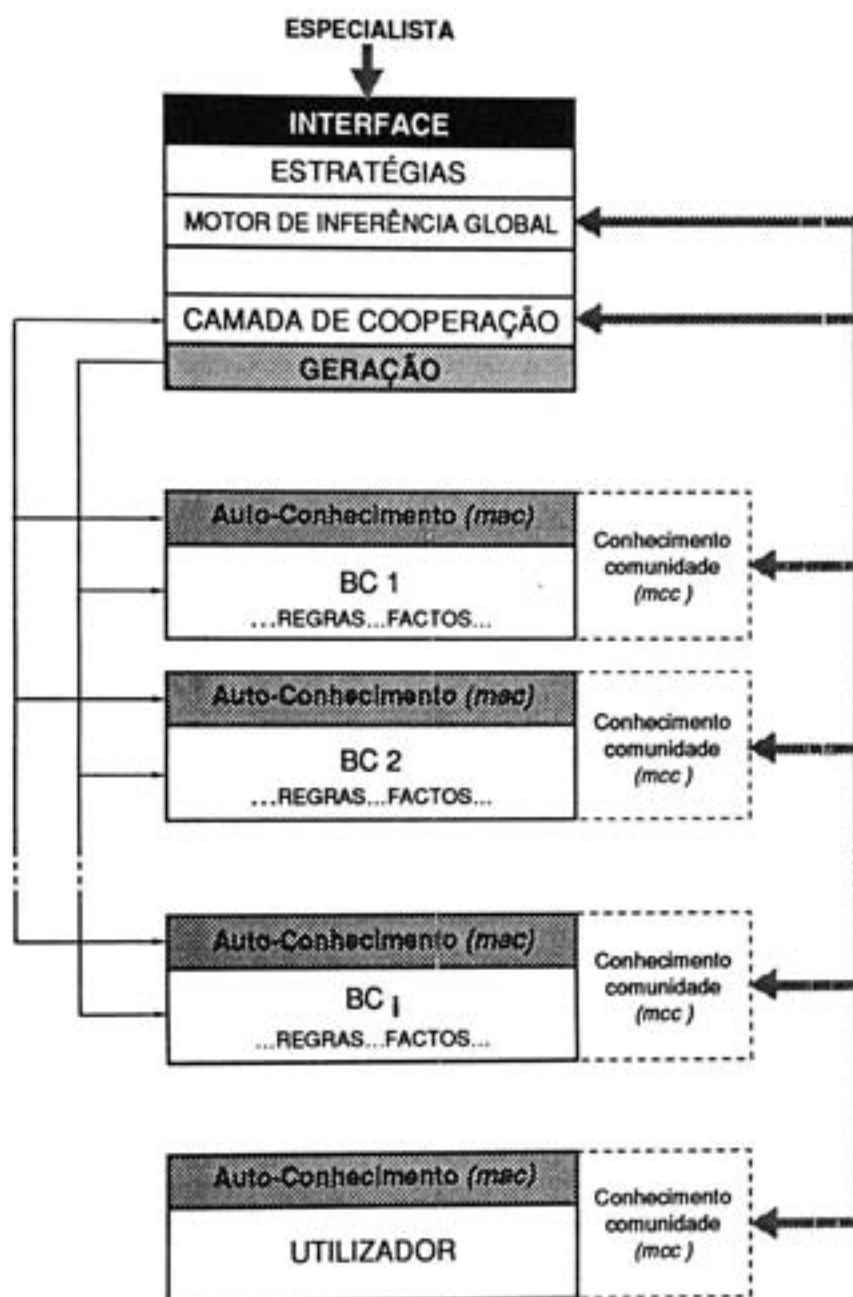


figura 5.22

5.11 CONCLUSÕES

5.11.1 Desenvolvimento

O desenvolvimento deste sistema passou por duas fases distintas:

- projecto promenorizado recorrendo a técnicas de especificação funcional de sistemas;

- implementação do projecto.

Na primeira fase o problema foi decomposto em subproblemas de âmbito mais restrito e cada um deles mereceu uma atenção específica durante um certo período de tempo. Resultaram essencialmente dois módulos :

- o módulo gerador;

- o módulo consultor.

- O modulo gerador permite criar Bases de Conhecimentos.

- O módulo consultor permite consultar as BC previamente criadas, accionando o motor de inferência.

- As Bases de Conhecimentos são geradas com base numa "Teoria" de modelização do conhecimento de domínio. A teoria de modelização de conhecimento permite guardar o conhecimento com base em critérios devidamente definidos e estudados e ao mesmo tempo facilitar a transferência do conhecimento do perito para o sistema.

Uma particularidade do sistema gerador é englobar a possibilidade de definir factores de certeza. Os domínios abrangidos para exprimir o seu conhecimento necessitam muitas vezes de factores de certeza, já que as afirmações não podem ser feitas de um modo categórico. Assim, quer as regras, quer os factos têm associado um factor de certeza, se e só se o perito o entender.

Toda esta parte do processo implica o recurso a técnicas da IA, nomeadamente a Aquisição e Representação do Conhecimento.

Uma vez que dispunha já de um sistema que criava bases de conhecimento, a preocupação seguinte foi desenvolver o sistema consultor. Esta tarefa foi desenvolvida em duas fases:

- consulta a uma única Base de Conhecimentos;

- consulta a várias Bases de Conhecimentos.

O cérebro e núcleo deste sistema é o motor de inferência, que foi também ele desenvolvido em duas fases acompanhando o sistema

consultor. Enquanto a primeira fase nos obrigou a um estudo de estratégias de pesquisa em IA, a segunda representou um salto qualitativo importante obrigando a um aprofundamento teórico nas áreas da IAD e, acima de tudo, cooperação e controlo da cooperação.

Desde a fase de projecto, passando por toda a fase de desenvolvimento e codificação, houve sempre a preocupação de estruturar devidamente o sistema, de molde a este resultar perfeitamente modelar. A linguagem utilizada - o LPA-Mac prolog - ao permitir a utilização de várias janelas de código em simultâneo, facilita bastante a tarefa neste campo, pois está orientado para isso mesmo. O resultado foi que cada subsistema, mesmo elementar, fica independente dos demais. Uma melhoria parcial e localizada é fácil e tem reflexos no trabalho. Por outro lado vai permitir a qualquer outro programador implementar módulos de modo independente. Por tudo isto houve uma grande preocupação em comentar de modo promenorizado todo o código o que facilitará futuros desenvolvimentos e melhorias.

5.11.2 Desenvolvimentos futuros

Entre os desenvolvimentos futuros possíveis surgem:

- Modelo do utilizador
- Bases de conhecimento distribuídas por diferentes equipamentos. Actualmente o sistema possui uma separação e cooperação de facto entre as bases de conhecimento, mas estas estão fisicamente localizadas na mesma máquina.
- Lançamento das bases de conhecimentos como processos paralelos. Para isso torna-se necessário recorrer a um sistema operativo multi-tarefa. O anunciado lançamento para breve do sistema 7 poderá facilitar esse desenvolvimento do sistema. A possibilidade de o motor de inferência ser interrompido, permite com facilidade (sem grandes alterações) adaptá-lo à pretendida situação de processamento multi-tarefa.
- Aplicação do SGSP a um domínio apropriado.

BIBLIOGRAFIA

- [Aiello, 81]** Nelleke Aiello, Conrad Bock, H. Penny Nii, William C. White -
"Joy of AGE-ing: An Introduction to the Age-1 System"
October, 81, Heuristic Programming Project, Computer
Science Department, Stanford University
- [Barbara, 85]** Barbara, Hayes - Roth - "Heuristic Programming Project,
Stanford University, Stanford, CA 94305, U.S.A., in Artificial
Intelligence in International Journal, pgs. 251-321
- [Bobrow 83]** Bobrow, Daniel G. ; Stefik, Mark " The Loops manual"
Dezembro 83, by Xerox Corporation.
- [Bond, 88]** Bond, Alan H. ; Gasser, les. "Readings in Distributed Artificial
Intelligence" Morgan Kaufmann Publishers, inc. San Mateo,
California.
- [BraKto, 86]** Brakto, Ivan - "Prolog Programming for Artificial
Intelligence", Addison Wesley, 1986
- [Brazdil 87]** Brazdil, Pavel B., "Aprendizagem em Sistemas Informáticos"
in "Revista de Informática", Vol. 6, num 3, Maio 1987, pgs.
24-32
- [Camacho 88]** Camacho, Rui, "Representação de Conhecimento num
Sistema Pericial com Perspectivas de Aprendizagem Simbólica

Automática", Tese de Mestrado, Instituto Superior Técnico, Lisboa, 1988

[Chorafas, 87] Chorafas, Dimitris N. "Applying Expert Systems in Business" McGraw-Hill, Inc.

[Coelho 86] Coelho, Helder, "Actualidade Tecnológica das Bases de Conhecimentos", Junho 86, Publicação do LNEC

[Coelho 88] Coelho, Helder, "Engenharia do Conhecimento I" Publicação do LNEC

[Costa 88] Costa, Ernesto, "Aprendizagem: Conceitos, Estratégias e Problemas", Departamento de Engenharia Electrotécnica da Universidade de Coimbra

[Engelmore 88] Engelmore, Robert; Morgan, Tony, "Blackboard Systems", Addison-Wesley Publishing Company, 1988

[Fain, 82] Fain, J. ; Hayes-Roth, F.; Sowizral, H. ; Waterman, D. - "Programing in Rosie: An Introduction by Means of Examples", The Defense Advanced Research Projects Agency

[Fehling, 89] Fehling, Michael; Herman, Leer - "Report on the Third Annual Workshop on Distributed Artificial Inteligence", pgs 3 e 4

[Forgy, 81] Forgy, Charles L. - "OPS5 USER's MANUAL", July 1981, Department of Computer Science Carnegie - Mellon University

- [Gevarter, 84]** Gevarter, William B. - "An Overview of Artificial Intelligence and Robotics", Prepared for: National Aeronautics and Space Administration, Janeiro 1984
- [Giovannini, 87]** Giovannini, F. ; Cortese, G. - "An Expert System Architecture Supporting Cooperation Among Loosely-competed Heterogeneous Agents" in IASTED, pgs. 179-183, 1987
- [Giozza 86]** Giozza, William Ferreira; Araújo, José Fábio Marinho; Moura, José Antônio Beltrão; Suavé, Jacques Philippe "Redes locais de Computadores" MacGraw-Hill - São Paulo - 1986
- [Hayes-Roth 83]** Hayes-Roth, Frederick; "Waterman, Donald A.; Lenat, Douglas A., "Building Expert Systems" Addison-Wesley Publishing Company, 1983
- [Harmon, 85]** Harmon, Paul- "An Exciting new Tool and its Implications for the Marketplace" in "Expert Systems Strategies", Vol. 1, Set. 1985, pgs. 10-12
- [Jennings, 89]** Jennings, N.R. "Relationship between User Models & Agent Acquaintance Models " Departement of Electrical & Electronic Engineering, Queen Mary & Wesfield College (University of London)
- [Jennings, 90]** Jennings, N.R. "Representing Agent Acquaintance Models Within Archon" Departement of Electrical & Electronic Engineering, Queen Mary & Wesfield College (University of London)

- [Leitão, 88]** Leitão, Maria Helena Ribeiro Barbosa - "Uso da Programação em Lógica para Definição de Processos de Tratamento de Minérios", Tese de Mestrado, Universidade do Minho, 1988
- [Lucena, 87]** Lucena, Carlos - "Inteligência Artificial e Engenharia de Software", Publicações Académico-Científicas, 1987, Jorge Zahar Editor
- [Macedo, 90]** Macedo, Joaquim ; Freitas, Vasco - "Um Ambiente de Desenvolvimento para a Gestão em Sistemas Distribuídos" in Actas do 6º Congresso Português de Informática, Associação Portuguesa de Informática 1990, comunicação nº 39, 1990, pgs. 612-627
- [Martin 88]** Martin, James; Oxman, Steven, "Building Expert Systems - A Tutorial" Prentice Hall, Englewood Cliffs, New Jersey 1988
- [Martins, 87]** Martins, João Pavão; Morgado, Ernesto Marques. "Ferramentas de Programação em Inteligência Artificial" in Revista de Informática, volume 6, número 3, Maio de 1987, pgs 10-17
- [McArthur, 84]** McArthur, Dave ; Steeb, Randall; Cammarata, Stephanie - Rand "Distributed Intelligence for Air Fleet Control" in Report on the "Third Annual Workshops on Distributed Artificial Intelligence
- [McDermott, 80]** McDermott, John - "R1: A Rule-Based Configurer of Computer Systems", April, 1980, Department of Computer Science Carnegie - Mellon University

- [Melle, 81] Melle, W. Van; Scott, A.C. ; Bennett, J.S. ; Peairs, M. - "The Emycin Manual" Department of Computer Science, Stanford University
- [Minsky 75] Minsky, M. "A Framework for representing knowledge", The Psychology of Computer Vision, McGraw-Hill, Nova Iorque, 1975
- [Morgado, 88] Morgado, Ernesto ; Martins, João Pavão - "Representação do Conhecimento", Publicação do Instituto Superior Técnico
- [Neves, 88] Neves, Maria da Conceição carvalho. "Sistema Gerador de Sistemas Periciais" Tese de Mestrado, Fevereiro, 1988.
- [Oli, 84] Oliveira, Eugénio da Costa - "Engenharia do Conhecimento: Sistemas Periciais e Prolog", Tese de Doutoramento - 1984
- [Oli, 86] Oliveira, Eugénio da Costa, "Bases de Conhecimento: Metaconhecimento e Dinâmica" in "Actas do 4º Congresso Português de Informática", Associação Portuguesa de Informática 1986, Comunicação nº 12, 1986, pgs. 198-219
- [Oli, 87] Oliveira, Eugénio da Costa - "Inteligência Artificial e Conhecimento", in "Revista de Informática", Vol. 6 num 3, Maio 1987, pgs. 4-10
- [Oli, 90] Oliveira, Eugénio da Costa; Camacho, Rui; Ramos Carlos, "UPSHELL - Rational and Brief Description" - Version 0.1 - Technical Note: UP/ARCHON/TN - 008/06-90 - Faculdade de Engenharia da Universidade do Porto, 1990

- [Patterson 90]** Patterson, W. Dan, "Introduction to Artificial Intelligence & Expert Systems", Prentice Hall, Englewood Cliffs, New Jersey 1990
- [Pereira, 86]** Pereira, Luis Moniz ; Porto, António - "Vantagens da Programação em Lógica para Sistemas de Conhecimentos" in Actas do 4º Congresso Português de Informática, Associação Portuguesa de Informática 1986, comunicação nº6, 1986, pgs. 118-130
- [Porto 85]** Porto, António, "Projecto de Desenvolvimento de Tecnologia de Bases de Conhecimentos", Prova Complementar de Doutoramento. Departamento de Informática da Universidade Nova de Lisboa
- [Porto 88]** Porto, António, "Representação de Conhecimentos e Programação em Lógica", Departamento de Informática da Universidade Nova de Lisboa, Outubro 1988
- [Prerau 90]** Prerau, David S., "Developing and Managing Expert Systems" Addison-Wesley Publishing Company, 1990
- [Rich 83]** Rich E., "Artificial Intelligence", McGraw-Hill, 1983
- [Shirai, 82]** Shirai, Joshiaki; Tsujii, Jun-ichi. "Inteligência Artificial, conceitos, técnicas e aplicações" Publicações Europa-América, Lda. 1988.
- [Shortliffe 76]** Shortliffe, E., "Computer Based Medical Consultation - Mycin", Elsevier, 1976

- [Smith, 81]** Smith, Reid G. ; Davis , Randall - "Frameworks for Cooperation in Distributed Problem Solving" in Distributed Artificial Intelligence, Edited by Alan H. Bond and Les Gasser, 1988, pgs.61-70
- [Stefik 83]** Stefik, Mark; Bell, Alain G. ; Bobrow, Daniel G. "Rule Oriented Programing in Loops" June 1983 by Xerox Cooperation.
- [Sterling 87]** Sterling, Leon; Shapiro, Ehud " The Art of Prolog" The MIT Press Cambrige Massachussetts ; London England
- [Sycara, 88]** Sycara, Katia - "Resolving Goal Conflits via Negotiation" - in AAAT - 88, pgs. 245-250, 1988 .
- [Ward 87]** Ward, Paul T. " Desenvolvendo Sistemas sem complicação" Livros Técnicos e Científicos S.A." Rio de Janeiro 1987
- [Waterman, 86]** Waterman, Donald A. - "A Guide to Expert Systems", Addison - Wesley, 1986
- [Williams 84]** Williams, Chuk "ART - The Advanced Reasoning Tool - Conceptual Overview" publicação da Inference Corporation, Los Angeles, Califórnia
- [Zhang, 89]** Zhang, C. and Bell, D.A. - "HECODES: a FrameWork for Heterogeneous Cooperative Distributed Expert System", 1989

ANEXO A	1
A1. INTRODUÇÃO	1
A2. S.G.S.P.	2
A2.0 ASPECTOS GERAIS.....	2
A2.1 LPA-MacProlog.....	3
A2.2 Entrada.....	4
A2.3 Menu geração.....	5
A2.2.1 entidades.....	7
A2.2.2 factos.....	8
A2.2.3 regras.....	10
A2.2.4 listagem.....	11
A2.2.5 gravar.....	11
A2.4 Menu consulta.....	11
A2.5 Menu listagem.....	17
A2.6 Sair.....	17

ANEXO A

A1. INTRODUÇÃO

O SGSP está implementado em LPA-Mac PROLOG, podendo ser executado num computador Apple com mais de 1 Mb de memória Ram.

É composto pelos seguintes ficheiros:

Concha

Ajuda

Para ser executado necessita ainda dos ficheiros complementares:

Todasbc

utilizadormac

<Utilizador>usr

Devido à sua extensão o código é dividido por várias janelas. Esta característica permite uma organização modular e uma melhor compreensão do código fonte e facilita também bastante a sua alteração em qualquer altura. As janelas em que está dividido o código fonte são as seguintes:

**ecrans
busca
gestor0
gera0**

**utilitários
Mcomum
gera1
gestor1**

**motorFc
Maconhecimento
fa_certeza
consulta**

De forma a melhorar o desempenho do sistema deve existir disponível um mínimo de 2 Mbyte de RAM e o computador deve possuir disco rígido.

Os ficheiros necessários ao funcionamento do SGSP deverão estar todos localizados no mesmo directório por uma questão de organização. Nesse directório deverão também estar os ficheiros respeitantes às bases de conhecimento geradas: quatro por cada base de conhecimentos.

O arranque do sistema é efectuado com um duplo clique no ficheiro:

concha

A2. O SISTEMA GERADOR DE SISTEMAS PERICIAIS

A2.0 Aspectos gerais

-Está escrito em Prolog e engloba o conhecimento sobre a forma de conceitos, factos e regras.

Na fase de Geração adquire, a partir de um interface amigável, todos os conceitos, atributos e valores, assim como os factos e as regras, através de um diálogo com o perito.

Na fase de consulta tenta responder às perguntas, explorando até à exaustão essa possibilidade nas bases de conhecimento que o utilizador está autorizado a consultar.

Pretende-se que:

Um utilizador não especialista em I.A. ou Engenharia do Conhecimento possa construir os seus próprios sistemas periciais. Para isso, o sistema incorpora nos passos mais importantes explicações

sobre a forma como a sequência das mais diversas operações deverá ser efectuada correctamente.

O S.G.S.P. pretende ser uma linguagem de alto nível que dispensa aprendizagem prévia, logo apresenta facilidades acrescidas e ajudas à programação. Apresenta por isso a diminuição da maleabilidade e o apertar do leque de opções quanto à forma como o conhecimento tem de ser planificado antes da sua incorporação.

Num sistema pericial tradicional o engenheiro do conhecimento faz uma incursão no campo do perito, para poder posteriormente vir a incorporar o conhecimento no sistema. Com o S.G.S.P., o perito terá que sair um pouco do seu mundo e fazer ele próprio uma incursão no campo da informática, utilizando as ferramentas informáticas para a criação do seu próprio sistema pericial.

Resumindo -:

Na criação dum S.P., o Engenheiro do Conhecimento tem um papel activo e o perito um papel mais passivo.

Com o S.G.S.P., o Engenheiro do Conhecimento passará a ter um papel mais passivo e o perito ganhará uma maior autonomia, inclusive durante a fase de geração.

A 2.2.1 LPA - MacProlog

LPA-Prolog é um interpretador Prolog para o Macintosh, munido de uma caixa de ferramentas gráficas excepcional. As ferramentas gráficas permitem a prototipagem rápida de interfaces gráficas com toda a ergonomia Macintosh.

LPA-Prolog é uma implementação completa do Prolog de Edimburgo.

. Integra a possibilidade de construção de analisadores da sintaxe pelo método de gramáticas de cláusulas definidas, assim como os predicados bagof e setof, que formam conjuntos de soluções . O código fonte é repartido em janelas. O código de cada janela é declarado compilado ou interpretado segundo a dinâmica desejada e poderá ser gravado como texto ou sob a forma pré-compilada.

Velocidade de execução do LPA:

18,6 KLIPS (Kilo-inferências lógicas/segundo) no Mac II em modo compilado e 6 em modo interpretado.

6,75 e 2,25 no Mac SE

5,6 e 1,9 no Mac Plus

A2.2 Entrada

O acesso ao sistema é efectuado através do écran seguinte no qual está integrado o menu principal em que o utilizador escolhe uma das opções disponíveis.

O menu principal contém as opções indicadas na figura , cujas funções se descrevem nos capítulos seguintes.

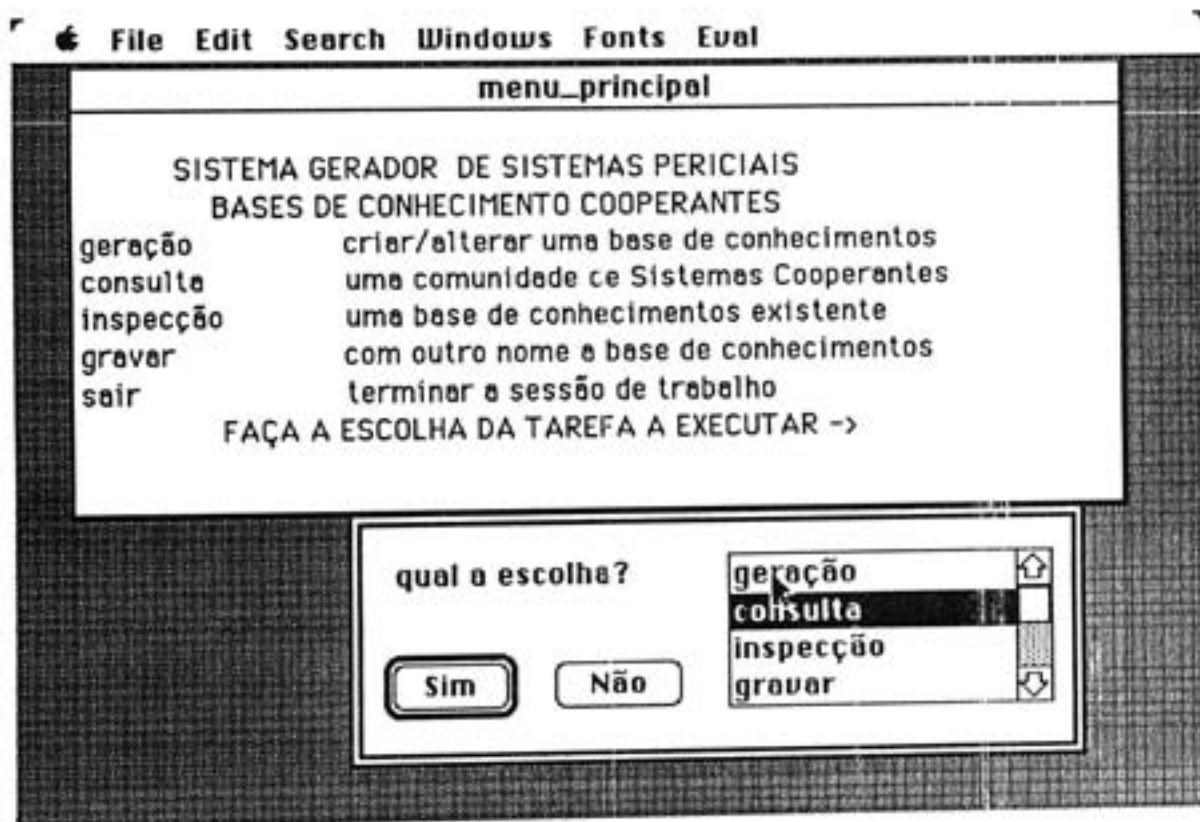


figura A-1 - Menu Principal

Conforme indicado no próprio écran para passar de uma opção para outra dentro do menu disponível basta apontar com o rato escolher a opção desejada e posteriormente confirmar.

Se o utilizador pretender terminar a execução do programa sem entrar no menu inicial, basta optar pela opção **sair**.

A2.3 Menu geração

Quando da Geração de um Sistema Pericial são criados quatro ficheiros em disco:

<BC>com incorpora regras

<BC>ext incorpora factos

<BC>est incorpora entidades, atributos, respectivos valores ,relações mútuas e o nome do domínio de aplicação(modelo conceptual).

<BC>mac incorpora o autoconhecimento (o que ela sabe):

Os conceitos e os atributos deverão ser introduzidos antes dos factos e regras, ou então, se não estiverem definidos, o sistema alertará para o facto.

Seleccionada esta opção no menu inicial, obtém-se o menu de geração que contém as opções : **criar, actualizar, sair**.

As opções criar e actualizar são idênticas com a diferença que **criar** se refere a uma nova base de conhecimentos e **actualizar** se refere a uma base de conhecimentos pré-existente. A opção de **sair** permite regressar ao menu anterior.

Para criar uma nova base de conhecimentos é necessário, por razões óbvias é conhecer a palavra chave.

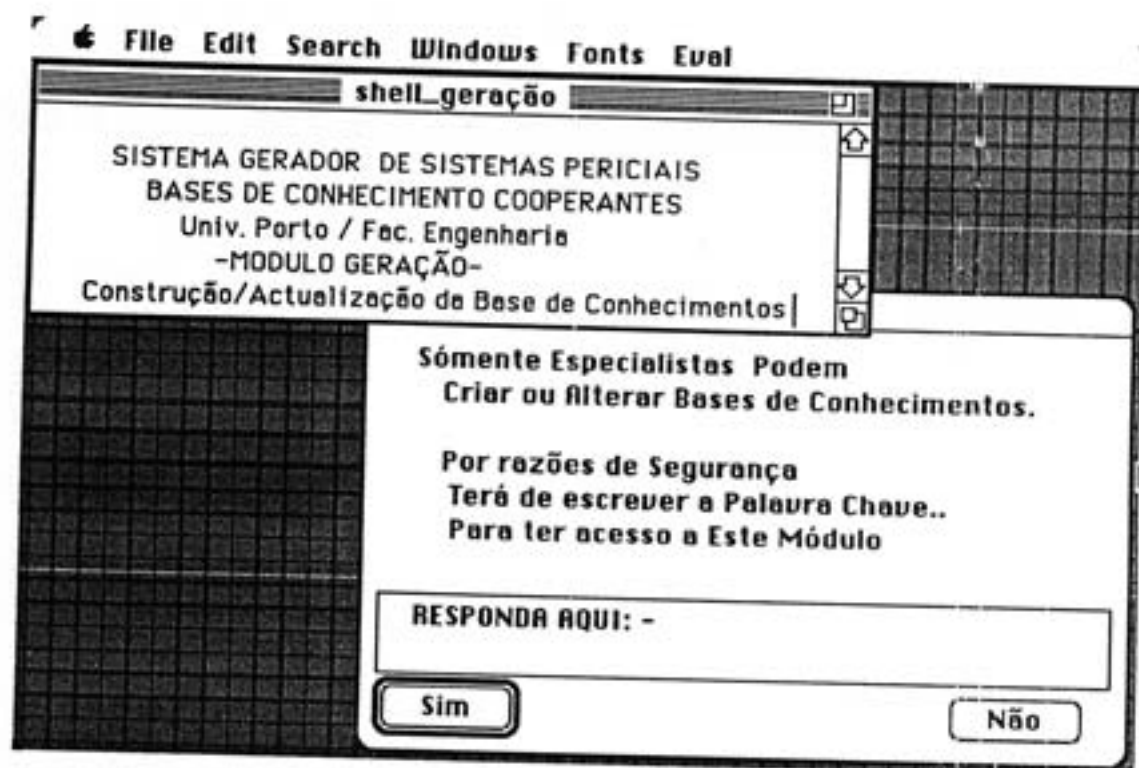


figura A-2 - password

Fases do diálogo que se segue à opção **criar** uma nova base de conhecimentos:

Nome da nova base de conhecimentos?

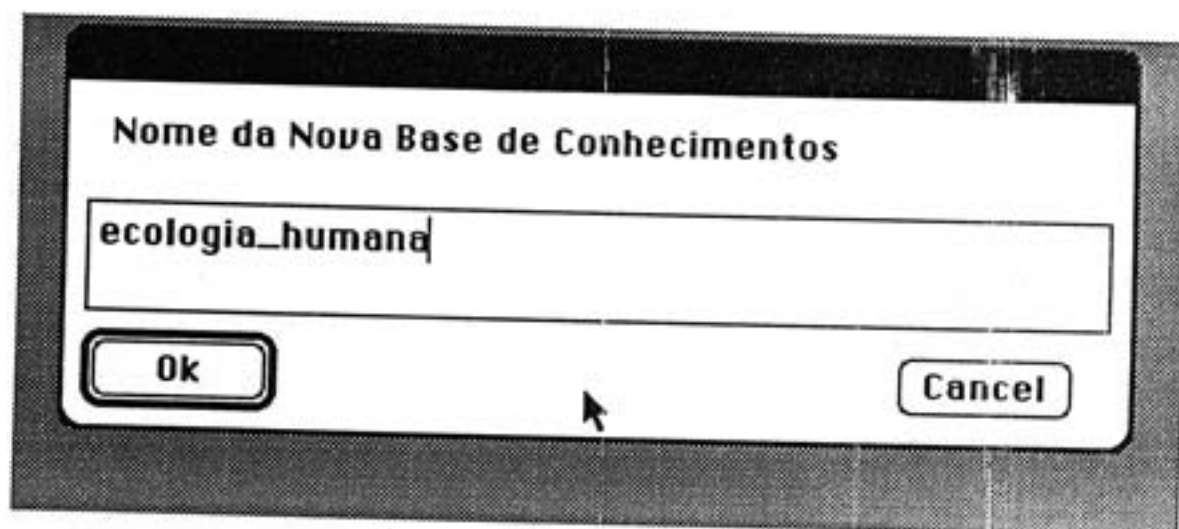


figura A-3 - Nova Base

Designação do domínio da nova base de conhecimentos?

Resposta

Factor de certeza (sem,fc)?

Resposta

- Logo de seguida é visualizado o seguinte écran

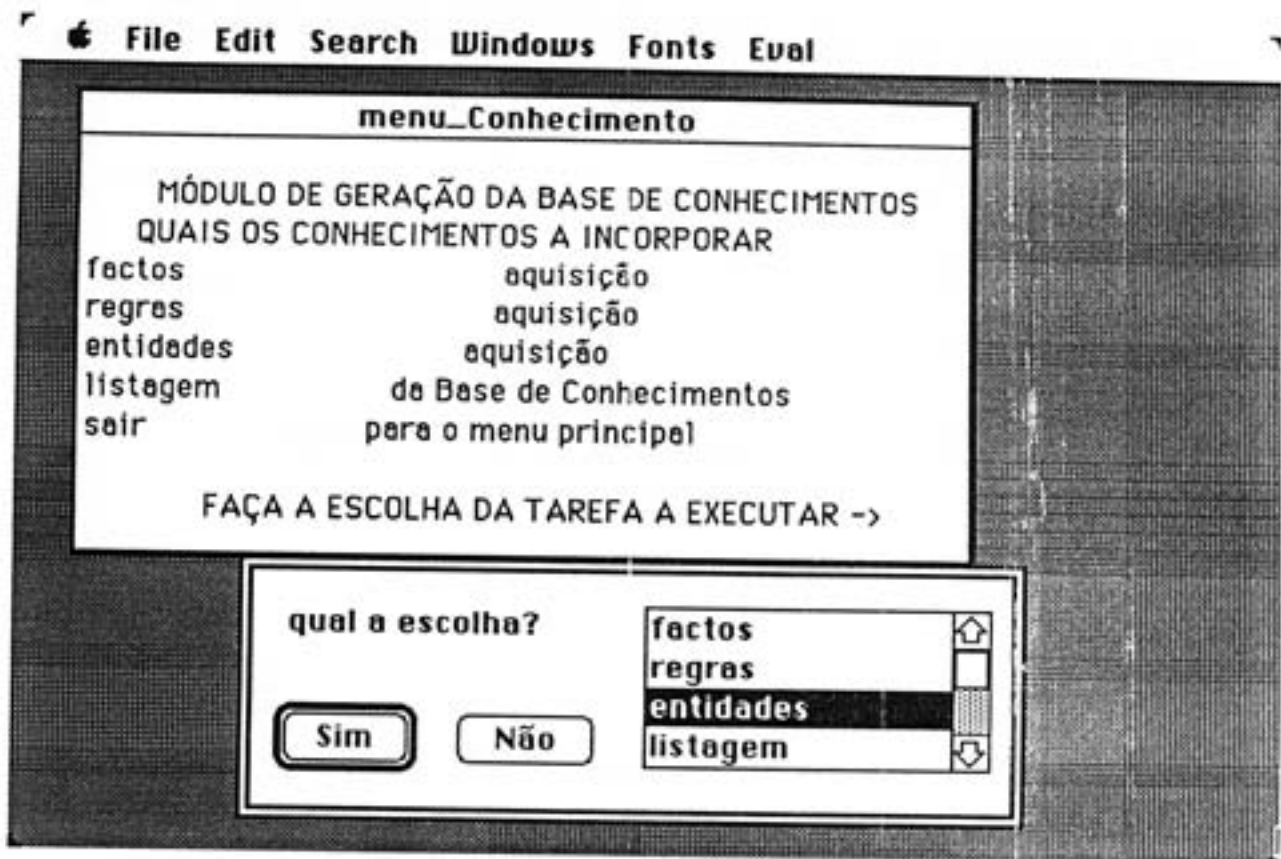


figura A-4 - Menu Conhecimento

A2.3.1 entidades

Permite a introdução voluntária por parte do utilizador de entidades para o domínio.

O sistema solicita a introdução do nome do conceito. Após a introdução do conceito, o sistema solicitará a introdução dos atributos e valores para cada um dos atributos.

Após a introdução dos valores de todos os atributos, o sistema faz a confirmação dos dados introduzidos.

No caso de ser detectada alguma incoerência com os tipos de dados previstos, o utilizador é avisado podendo recomeçar o processo.

A figura seguinte elucida como se processa o diálogo:

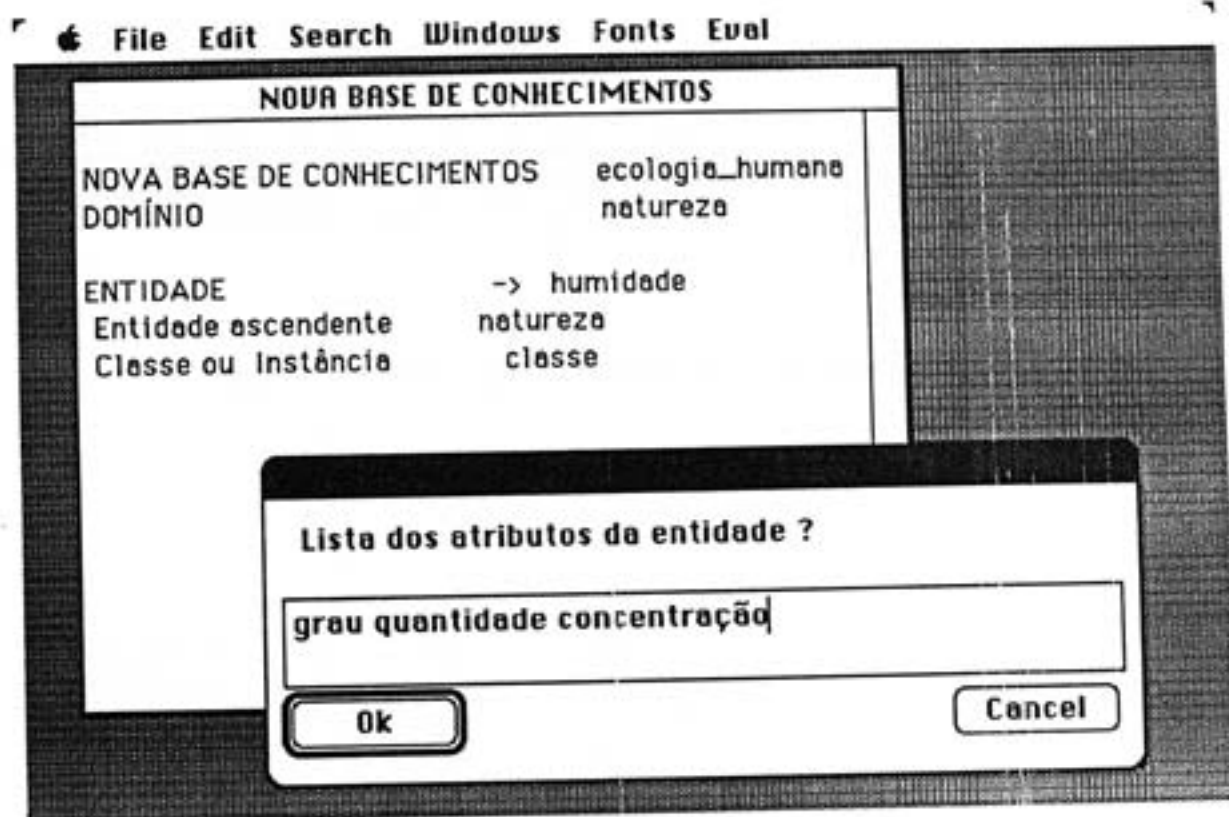


figura A-5 - Aquisição de entidades

A2.3.2 factos

A introdução inicia-se com uma janela mostrando ao utilizador a sintaxe permitida pelo Sistema:

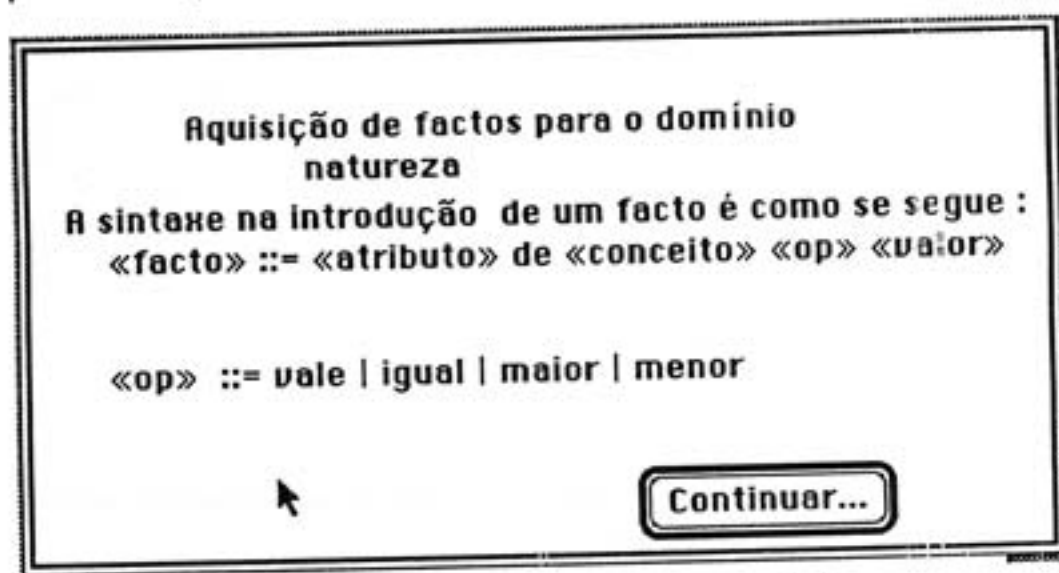


figura A-6 - Menu Principal

e pelo pedido da introdução de factos:

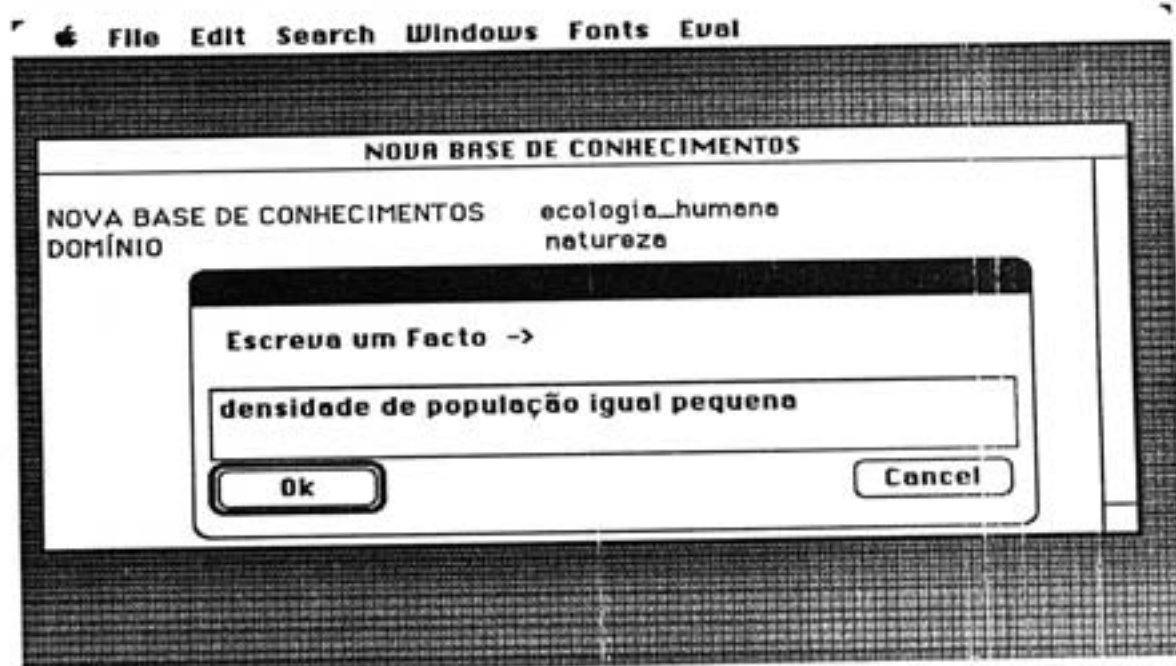


figura A-7 - Introdução de Factos

Como está prevista a utilização de factores de certeza, depois de introduzir um facto, segue-se a pergunta:

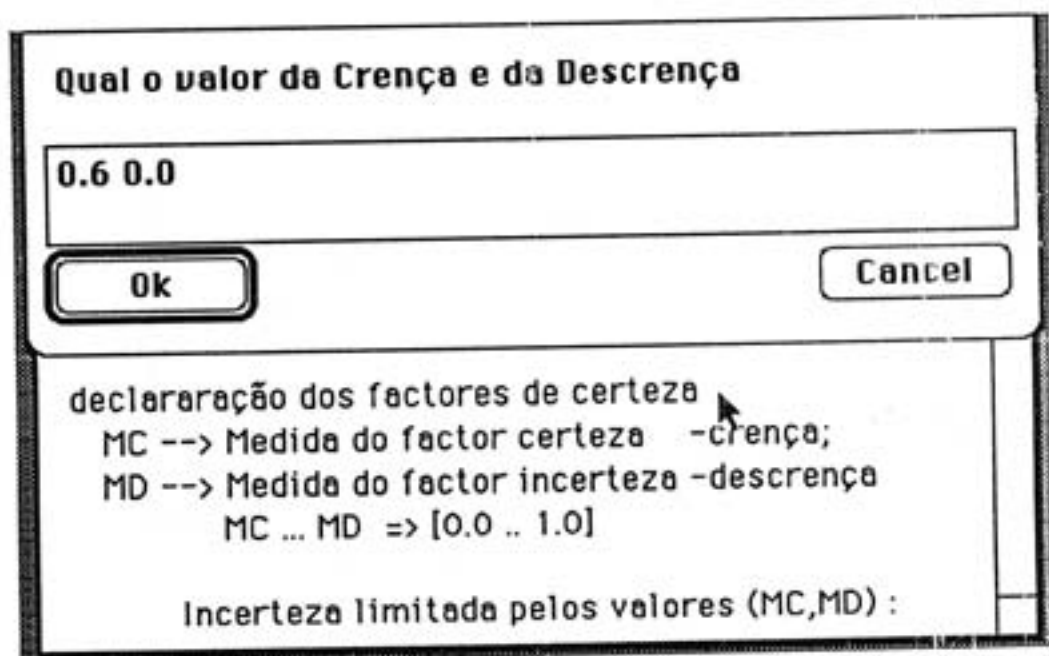


figura A-8 - Factor de Certeza

A2.3.3 regras

As regras são introduzidas em três fases:

- Primeiro as premissas com uma sintaxe idêntica à dos factos;

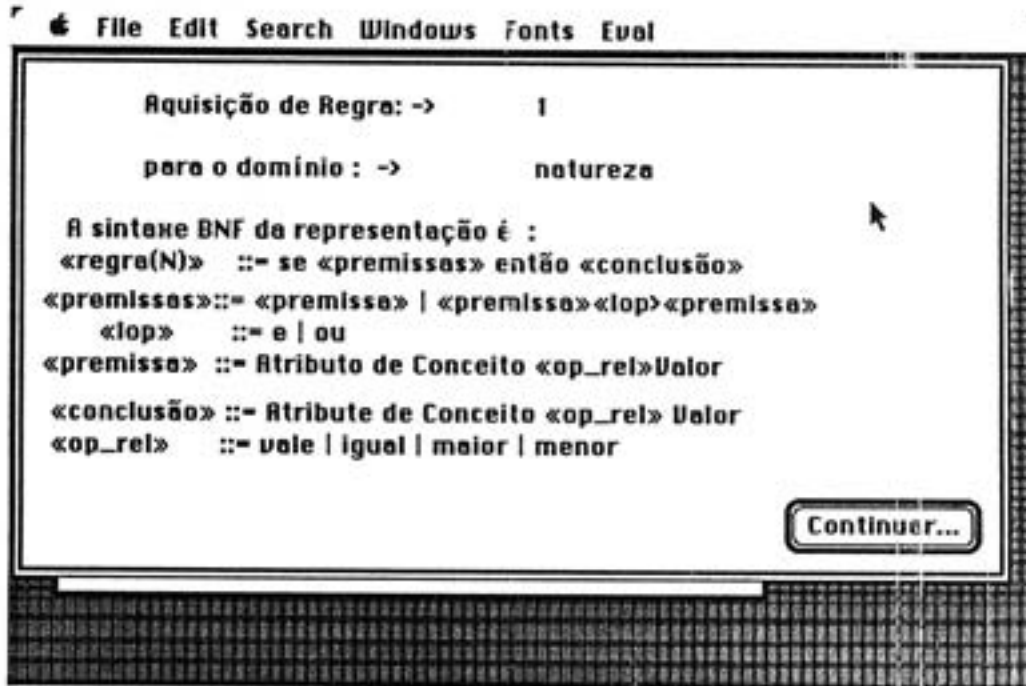


figura A-9 - Sintaxe das Regras

- De seguida a combinação das premissas(conjunção, disjunção ou associações de conjunções com disjunções);

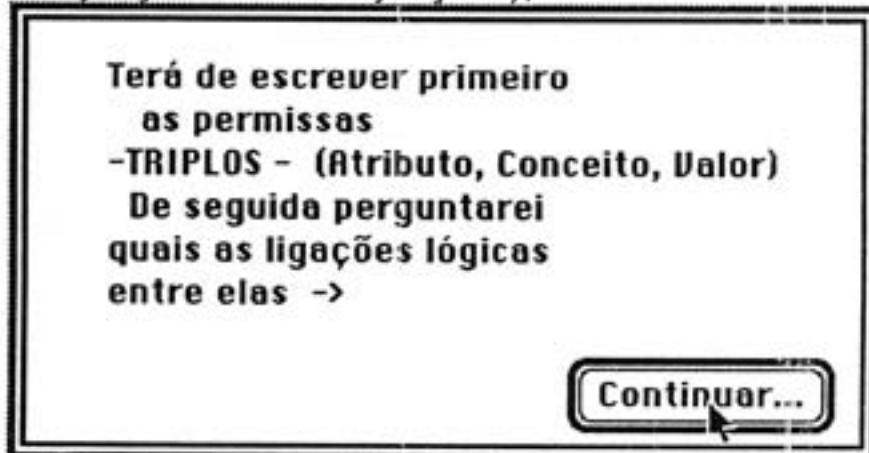


figura A-10 -ligações lógicas

- Para terminar é introduzida a conclusão.

Durante as várias fases de introdução das regras o sistema vai mostrando ao utilizador a sintaxe permitida.

A2.3.4 listagem

A selecção desta opção permite inquirir o sistema sobre o conjunto de triplos O.A.V introduzidos.

Depois de seleccionada surge uma janela com as Bases de Conhecimento existentes. O utilizador deve então indicar o nome da que deseja pelo simples apontar do rato e o sistema responde com uma nova janela contendo a listagem solicitada.

A2.3.5 gravar

Esta opção é utilizada após a criação ou actualização de uma base de conhecimentos e permite gravar os quatro ficheiros que a constituem.

Terminada a primeira fase (introdução de conceitos, factos e regras), segue-se a possibilidade de gravar a Base de Conhecimentos e, tal como foi anteriormente referido, eles ficam assim distribuídos por ficheiros:

<BC> com ;<BC> ext ;<BC> est ;<BC>mac

Este último ficheiro é criado de modo dinâmico e incorpora uma síntese do auto-conhecimento.

A2.4 Menu consulta

Quando um utilizador inicia uma sessão de consulta, ele só a poderá fazer se for um utilizador autorizado pelo gestor sendo o nível

de autorização variável. Significa isto que cada utilizador só poderá utilizar algumas das bases de conhecimento.

Partindo do princípio que o utilizador está autorizado e que sabe quais as bases de conhecimento que pode consultar, ele poderá, ao iniciar o processo de consulta, fazê-lo só a algumas dessas Bases de Conhecimento. O lançamento dos agentes chamados e disponíveis para a sessão de consulta é visualizado por janelas (uma por cada) no monitor do computador e corresponde, em termos de processo interno a gerar de modo dinâmico os ficheiros do "modelo do conhecimento da comunidade".

Os "modelo do conhecimento da comunidade" são gerados com base nos "modelos do auto conhecimento".

Descrição da Sequência de Acções durante uma sessão de perguntas ao S.G.S.P.

- Colocada uma questão, o sistema tenta encontrar uma das bases de conhecimento que permita o matching com o conceito e atributo ou apenas um deles, no caso de o outro ser uma variável.

- Se tal não for possível, não é aceite a pergunta que terá de ser refeita pelo utilizador.

- Após o aceitar da questão, por duas das BC, duas hipóteses são possíveis:

1. Existia um facto com o qual se verificou o matching e, nesse caso, a resposta é imediata.

2. O matching foi feito com a conclusão duma regra e, nesse caso, terão de se comprovar as premissas.

O processo continua procurando essa Base de Conhecimentos no seu próprio conhecimento ou então recorrendo à cooperação com a restante comunidade de agentes presente na sessão de consulta para completar o processo de dedução.

Se a determinada altura do processo de inferência não fôr possível a uma das Bases de Conhecimento confirmar a questão colocada por outra Base, antes de enviar a mensagem "não encontrei solução" ela recorre ao utilizador, que colocou a questão inicial, pois nada garante que ele não possa responder a essa (sub)questão e assim poder então terminar o processo de inferência com êxito.

Só mediante uma resposta negativa do utilizador, o sistema responderá que não tem solução para a pergunta que lhe foi colocada.

Conforme cada agente está a tentar responder a uma (sub)questão é visualizada uma mensagem na janela correspondente, permitindo ao utilizador ter uma ideia do modo como a interacção entre os agentes é estabelecida e de qual foi a contribuição de cada um deles para a solução final.

Em simultâneo, é criado um ficheiro com o nome "relatório" em que é feita uma recolha de todas as fases do processo. Aqui são guardadas todas as etapas da resposta, incluindo as questões colocadas no decorrer do processo e quem lhes respondeu.

Vejamos com um exemplo como o processo se desenvolve:

Após a selecção desta alternativa (consulta) no menu principal, aparece uma janela de diálogo para identificação do utilizador.

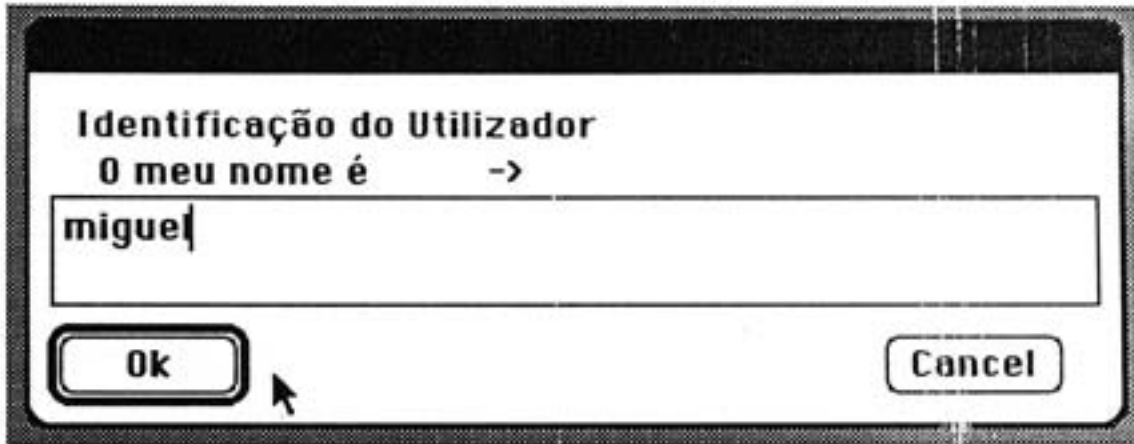


figura A-11 - Identificação do utilizador

De seguida continua o diálogo perguntando quantas e quais as bases de conhecimento que pretende consultar e mostrando em simultâneo todas as existentes.

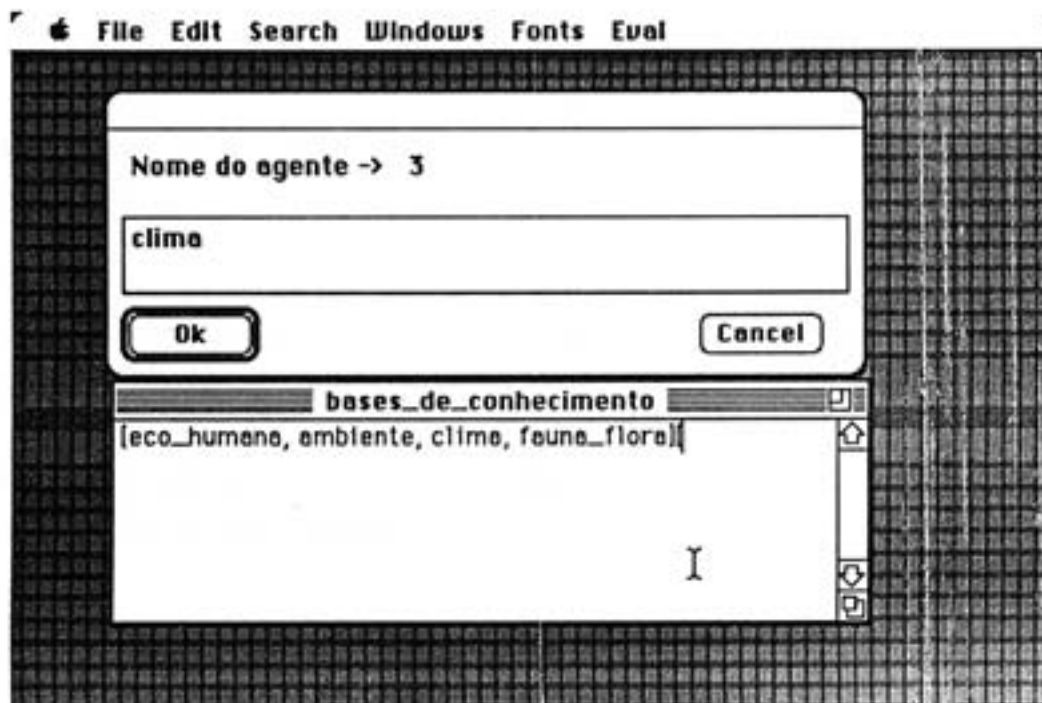


figura A-12 - Agentes Presentes

Terminada esta fase, é reservada uma janela de diálogo, para cada um dos agentes cooperantes apresentando o monitor o seguinte aspecto.

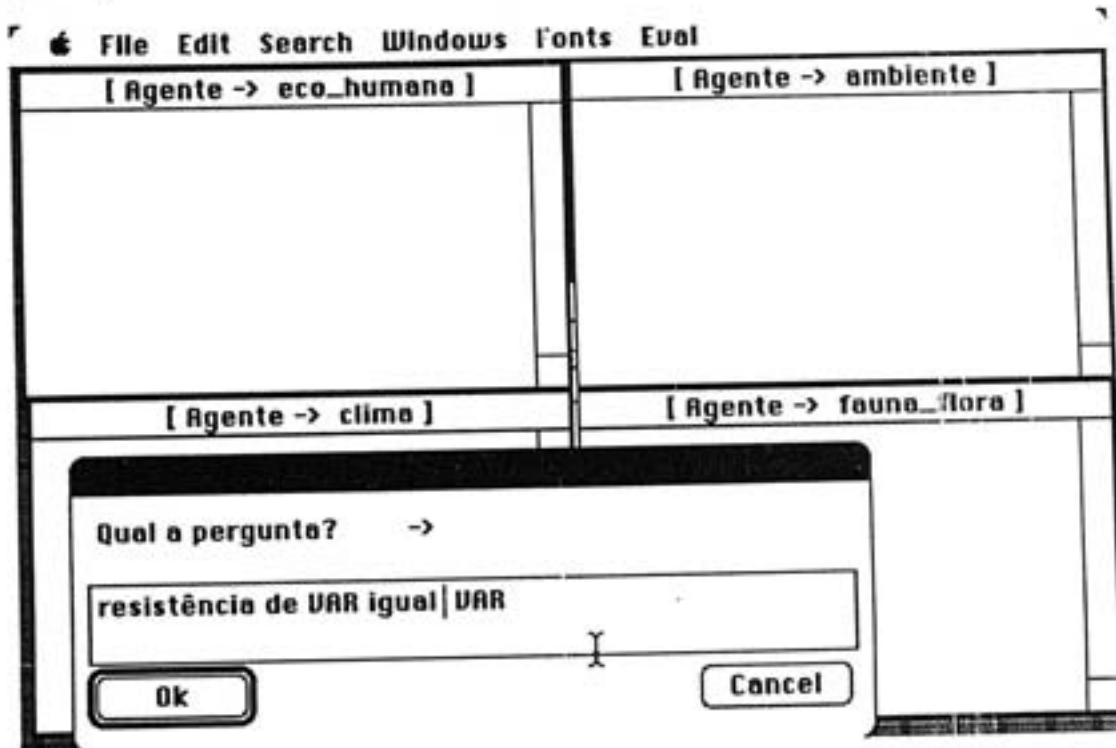


figura A-13 - Pergunta

Nesta fase estão criadas as condições para o objectivo essencial duma sessão de consulta que é colocar perguntas a fim de obter respostas.

Recordemos que estão geradas as quatro bases de conhecimento que constam da figura 5.12 e que têm vindo a ser utilizadas na demonstração do funcionamento do sistema

Mediante a pergunta que foi colocada, a resposta que se segue é a que consta da figura seguinte:

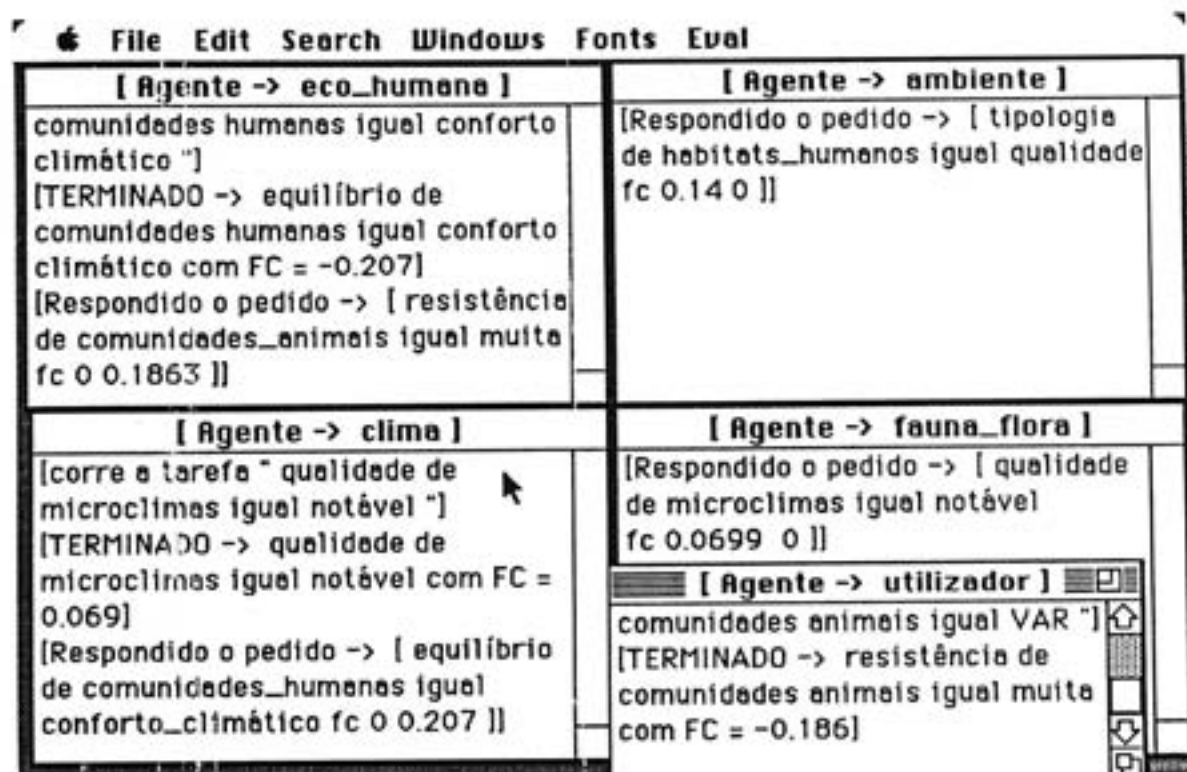


figura A-14 - Resposta do Sistema

Terminada a sessão o sistema elabora um relatório bastante promenorizado cujo texto integral é:

```

utilizador
Corre a tarefa " resistência de comunidades animais igual VAR "
eco_humana
Corre a tarefa " tipologia de habitats humanos igual qualidade "
ambiente
Respondido o pedido -> [ tipologia de habitats_humanos igual qualidade fc 0.14 0 ]
eco_humana
TERMINADO -> tipologia de habitats humanos igual qualidade com FC = 0.14
eco_humana
Corre a tarefa " equilíbrio de comunidades humanas igual conforto climático "
clima
Corre a tarefa " qualidade de microclimas igual notável "
fauna_flora
Respondido o pedido -> [ qualidade de microclimas igual notável fc 0.069 0 ]
clima
  
```

TERMINADO -> *qualidade de microclimas igual notável com FC = 0.069*

eco_humana

Corre a tarefa " equilíbrio de comunidades humanas igual conforto climático "

clima

Respondido o pedido -> [equilíbrio de comunidades humanas igual conforto climático fc 0 0.207]

eco_humana

TERMINADO -> *equilíbrio de comunidades humanas igual conforto climático com FC = -0.207*

utilizador

Corre a tarefa " resistência de comunidades animais igual VAR "

eco_humana

Respondido o pedido -> [resistência de comunidades animais igual multa fc 0 0.1863]

utilizador

TERMINADO -> *resistência de comunidades animais igual multa com FC = -0.186*

A2.5 Menu listagem

Este menu permite aceder à informação contida no sistema e referente à descrição dos triplos - atributo conceito valor, de acordo com o descrito em A.2.3.4

A2.6 Sair

Permite terminar a execução do programa e retornar ao sistema operativo.





FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



000034735