

Inteligência Artificial

Apontamentos para as aulas

Luís Miguel Botelho

Departamento de Ciências e Tecnologias da Informação
Instituto Superior de Ciências do Trabalho e da Empresa

Julho de 2015

Sistemas Baseados em Conhecimento: Conhecimento, Representação e Arquitetura

Índice

1	INTRODUÇÃO AOS SISTEMAS BASEADOS EM CONHECIMENTO	4
2	DISTINÇÕES A RESPEITO DE CONHECIMENTO E DE REPRESENTAÇÃO	5
2.1	CONHECIMENTO DECLARATIVO VS. CONHECIMENTO PROCEDIMENTAL	5
2.2	REPRESENTAÇÃO EXPLÍCITA VS. REPRESENTAÇÃO IMPLÍCITA	6
3	ARQUITETURA DE SISTEMAS BASEADOS EM CONHECIMENTO	7
4	MÉTODOS DE RACIOCÍNIO E DE REPRESENTAÇÃO DE CONHECIMENTO	9

Sistemas Baseados em Conhecimento: Conhecimento, Representação e Arquitetura

Uma ideia fundamental subjacente aos sistemas baseados em conhecimento é que os elevados níveis de desempenho na resolução de problemas de um determinado domínio de aplicação se conseguem fazendo uso de conhecimento extensivo do domínio. Esta ideia contrasta com a crença mais antiga segundo a qual a qualidade do desempenho se deve essencialmente a raciocínios muito complexos e a poderosos mecanismos de processamento de informação.

Outra ideia fundamental é a de representação explícita de conhecimento do domínio. A representação explícita de conhecimento é uma tecnologia de concepção de sistemas computacionais em que se usam estruturas de dados que representam conhecimento sobre o domínio da aplicação. O SBC dispõe de um mecanismo de processamento de informação genérico, totalmente independente do problema concreto a que o sistema é aplicado, que usa o conhecimento explicitamente representado nessas estruturas de dados para fazer face aos problemas apresentados ao sistema. Usando esta tecnologia é possível criar sistemas cujo desempenho pode ser alterado e melhorado pela simples alteração ou acréscimo do conhecimento representado explicitamente, sem ter que alterar o mecanismo genérico de processamento de informação.

O conjunto das estruturas de dados que representam o conhecimento de um SBC constitui a sua base de conhecimentos (BC). O mecanismo genérico de processamento de informação que usa o conhecimento armazenado na base de conhecimento para resolver problemas designa-se tradicionalmente motor de inferência.

Existem diversos métodos para representar conhecimento, cada um deles associado a determinados mecanismos de processamento. Um dos métodos de representação de conhecimento mais divulgados permite a representação de regras e a utilização de raciocínio dedutivo.

Geralmente, utilizam-se ferramentas computacionais que facilitam a tarefa de preencher a base de conhecimento de um SBC. A sofisticação e complexidade destas ferramentas variam desde as linguagens de programação da inteligência artificial (tradicionalmente Lisp e Prolog) até aos ambientes de desenvolvimento (“shell”). Quando se usam linguagens de programação de inteligência artificial, o conhecimento do SBC é escrito em ficheiros de texto que são depois convertidos pela linguagem numa coleção de estruturas de representação interna adequadas ao seu processamento posterior. Os ambientes de desenvolvimento dispõem de editores especiais com diversos mecanismos de verificação de sintaxe e até de validação semântica do conhecimento introduzido.

A complexidade das ferramentas de desenvolvimento de SBCs varia também a respeito do método de representação e do tipo de processamento de informação que oferecem. Em geral, as linguagens de programação dispõem de um único método de representação e de um único tipo de raciocínio. Os ambientes de desenvolvimento possibilitam a utilização de diversos métodos de representação de conhecimento e diversos mecanismos de processamento, incluindo diversos tipos de raciocínio.

Na próxima secção, apresenta-se uma panorâmica dos sistemas baseados em conhecimento no âmbito da inteligência artificial. A secção seguinte descreve duas importantes distinções acerca da representação de conhecimento. Por um lado, a distinção entre representação implícita e representação explícita; por outro lado, a distinção entre conhecimento declarativo e conhecimento procedimental.

1 Introdução aos Sistemas Baseados em Conhecimento

Têm sido criados e usados Sistemas Baseados em Conhecimento em diversos domínios de aplicação, entre os quais os seguintes, que se celebrizaram.

- Sistemas para auxiliar a identificação de compostos químicos (e.g., CRYSLIS, DENDRAL)
- Sistemas para auxiliar a tomada de decisão em explorações mineralógicas (PROSPECTOR)
- Sistemas para diagnosticar doenças e prescrever medicação (e.g., MYCIN)
- Sistemas para gerar configurações de computadores mediante as características do cliente (XCON)
- Sistemas para cálculo simbólico, por exemplo, simplificação de polinómios, diferenciação, integração (MACSYMA)

Ao contrário do que sucedeu com os Sistemas Baseados em Conhecimento de primeira geração, os quais foram concebidos isoladamente dos outros sistemas computacionais existentes nas organizações, hoje em dia, a integração é a palavra de ordem. A possibilidade de integração com sistemas existentes e com sistemas que venham a ser criados é uma propriedade desejável dos Sistemas Baseados em Conhecimento e da generalidade dos de sistemas informáticos.

Diversos Sistemas Baseados em Conhecimento com sucesso, efetivamente usados durante períodos de tempo apreciáveis, e cujos resultados foram realmente úteis foram descontinuados porque a tecnologia com que foram criados não se integrava com os outros sistemas de informação existentes na organização. Esta constatação permite concluir que os sistemas inteligentes devem ser implementados usando uma tecnologia que permita a sua fácil integração com outros sistemas. Atualmente, pensa-se que as tecnologias baseadas em componentes e especialmente as tecnologias baseadas em agentes inteligentes são as melhores opções, desse ponto de vista.

Os Sistemas Baseados em Conhecimento é uma tecnologia conceptual e computacional para a criação de Sistemas Inteligentes. Os Sistema Periciais (“*Expert Systems*”) são a classe de Sistemas Inteligentes em que a tecnologia dos SBCs tem sido mais usada. Por vezes, a expressão Sistema Baseado em Conhecimento é usada como sinónimo de Sistema Pericial.

Um Sistema Pericial é um SBC em que o conhecimento representado é obtido de um ou mais profissionais especialistas (peritos) da área de aplicação do sistema. É muito importante que o tipo de raciocínio de um Sistema Pericial seja semelhante ao de um perito, para que o seu comportamento possa ser compreendido pelos profissionais que o utilizam.

2 Distinções a respeito de conhecimento e de representação

Duas distinções têm sido apontadas e debatidas a propósito de conhecimento e de representação. Primeiro, a distinção entre conhecimento declarativo e conhecimento procedimental. Segundo a distinção entre representação explícita e representação implícita. Nesta secção apresentam-se estas duas distinções.

2.1 Conhecimento declarativo vs. conhecimento procedimental

O conhecimento declarativo representa declarações acerca das entidades de um domínio, suas propriedades e relações entre elas. O conhecimento procedimental especifica ações individuais ou sequências de ações que podem ser executadas.

Exemplos de conhecimento declarativo

- X é um número positivo se e só se X é maior do que 0.
- Se X é um coelho e X não tem deformidades então X tem quatro patas.
- Ferrari é uma marca de automóveis
- Isadora Duncan foi uma bailarina
- Se X pertence ao conjunto S1 e S1 está contido em S2 então X pertence ao conjunto S2.

Exemplos de conhecimento procedimental

Procedimento para criar uma cópia de uma disquete

1. Ligar o computador;
2. Inserir a disquete original na *drive* de disquetes;
3. Selecionar e executar a opção “copy disk” do menu de opções do ícone do *drive* de disquetes;
4. Proceder como especificado através das caixas de diálogo geridas pelo programa de cópia de disquetes;

Procedimento para calcular e imprimir a média dos 20 valores mantidos no vetor *Amostras*

1. Inicializar a variável *Soma* com 0
2. Inicializar a variável *Indice* com 1
3. Repetir 20 vezes
 - (a) Somar o valor da variável *Soma* ao valor do elemento de *Amostras* indexado por *Indice*, e atribuir o resultado à variável *Soma*
 - (b) Acrescentar uma unidade ao valor da variável *Indice*
4. Dividir a variável *Soma* por 20 e atribuir o resultado à variável *Media*
5. Imprimir a variável *Media*

Um programa de computador efetua sempre ações, por exemplo adquirir dados (através de uma ação de leitura), imprimir resultados num terminal ou escrevê-los num ficheiro (através de uma ação de escrita), controlar um dispositivo físico (e.g., uma *drive* de DVD), inserir ou remover registos numa base de dados. Consequentemente, qualquer

programa de computador com utilidade tem que ter algum conhecimento procedimental (implícito ou explícito), caso contrário não poderia efetuar nenhuma ação.

2.2 Representação explícita vs. representação implícita

O conhecimento usado num programa de computador pode estar representado explicitamente ou implicitamente. Quando o comportamento de um programa de computador é guiado por estruturas de dados que representam conhecimento, diz-se que o conhecimento está representado explicitamente. No entanto, o mais vulgar é encontrar programas de computador em que o conhecimento está disseminado no seu código só se tornando aparente pela observação do comportamento do programa. Há tecnologias para sistemas inteligentes, como as redes neuronais, que se comportam de acordo com o conhecimento que têm. No entanto, esse conhecimento não está representado explicitamente; está disseminado por todo o sistema, não sendo possível “olhar para lá” e identifica-lo. Nestes casos, o conhecimento do programa está representado implicitamente.

Um das características específicas de programas com conhecimento representado explicitamente é que podemos alterar o seu comportamento pela modificação do conteúdo das estruturas que representam o conhecimento.

Exemplos de representações explícitas e implícitas

O conhecimento representado explicitamente pode ser tanto declarativo como procedimental, o mesmo acontecendo com o conhecimento representado implicitamente. Um programa para determinar o máximo elemento de um conjunto pode ser concebido de diversas formas.

Numa das abordagens, teremos uma estrutura de representação de conhecimento com a definição do conceito *máximo elemento de um conjunto*; e um procedimento que interpreta e aplica essa definição a um conjunto particular para determinar o máximo. Nesta abordagem, o programa contém uma representação explícita de conhecimento declarativo. Além disso, contém um procedimento (representado implicitamente no código) capaz de interpretar representações explícitas (escritas com um determinado formato) e de as utilizar para resolver problemas concretos.

Definição do máximo elemento de um conjunto: um determinado objeto (no sentido lato de entidade) é o maior elemento de um conjunto, aqui designado pelo máximo, se for membro do conjunto e se, nesse conjunto, não existir nenhum elemento maior do que ele.

Representação explícita da definição de máximo de um conjunto

$$\forall \text{max, set} [(\text{Member}(\text{max}, \text{set}) \wedge \neg \exists x(\text{Member}(x, \text{set}) \wedge x > \text{max})) \Rightarrow \text{Max}(\text{set}) = \text{max}]$$

Além desta definição, o programa para determinar o máximo de um conjunto de entrada tem que ter um procedimento capaz de adquirir (ou receber) o conjunto de dados concreto e de usar a definição de máximo para o determinar.

Uma abordagem alternativa em que se usa conhecimento procedimental representado implicitamente é o seguinte programa (descrito em pseudocódigo), o qual representa implicitamente a definição de máximo de um conjunto

Procedimento *Maximo(S)*

Inicializar a variável *Max* com um elemento do conjunto *S*

Para cada *X* pertencente a *S*, repetir

 Se $X > Max$ então atribuir *X* a *Max*

Devolver *Max*

O programa em C apresentado na Figura 1 representa o conhecimento procedimental descrito pelo pseudocódigo acabado de apresentar, apenas para o caso de um conjunto de inteiros.

```
int Maximo(int N, int *S)
{
    int i = 1;
    int Max = S[0];
    while(i < N)
    {
        if( S[i] > Max) Max = S[i];
        i++;
    }
    return Max;
}
```

Figura 1 - Função em C para determinar o máximo de um conjunto

O conhecimento representado na definição de *máximo* apresentada está implícito no código do programa descrito. Além de representação implícita, trata-se de conhecimento procedimental porque especifica o conjunto de passos necessários à determinação do máximo de um conjunto. Há outros tipos de representação implícita não procedimental.

3 Arquitetura de sistemas baseados em conhecimento

Todos os sistemas baseados em conhecimento (SBC) têm uma arquitetura que partilha algumas características comuns, ainda que casos específicos possam ter alguns aspetos diferenciadores. A arquitetura comum aos sistemas baseados em conhecimento é constituída por três blocos: Base de conhecimentos (“knowledge base”), motor de inferência (“inference engine”), e interface. A interface pode funcionar em dois modos, o modo de utilização e o modo manutenção (i.e., criação e atualização da base de conhecimentos).

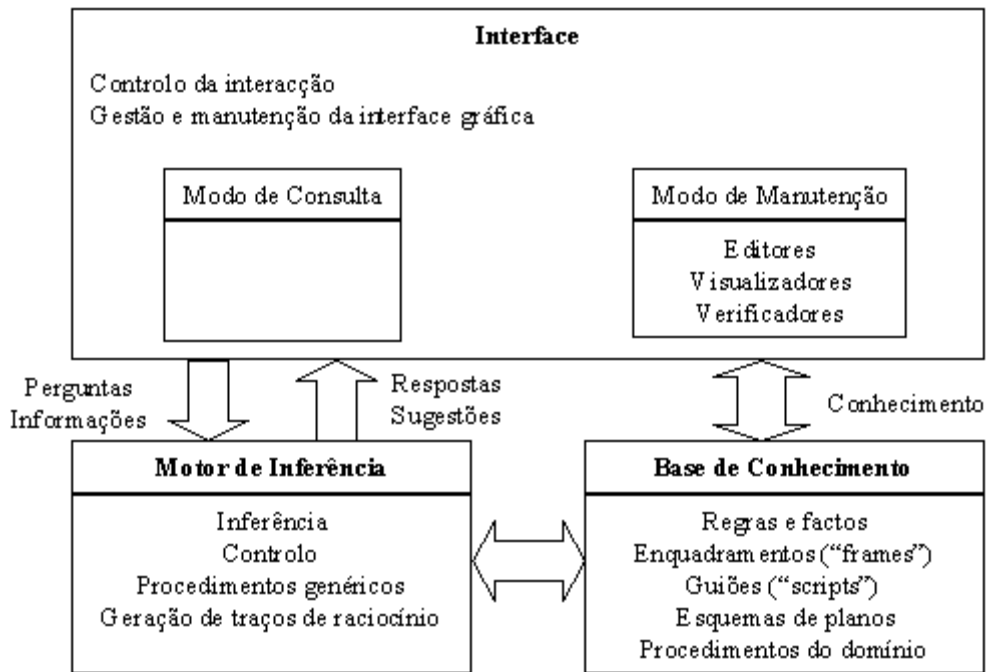


Figura 2 – Arquitetura dos Sistemas Baseados em Conhecimento

Tendo esta arquitetura, em que o conhecimento se representa na base de conhecimento, os sistemas baseados em conhecimento recorrem obrigatoriamente a métodos simbólico de representação explícita. Embora a representação tenha de ser explícita, o tipo de conhecimento usado pode ser declarativo, procedimental ou misto.

Base de Conhecimento

A Base de Conhecimento (BC) contém representações do conhecimento adquirido durante a realização do sistema e durante a sua posterior utilização, por exemplo,

```
IF The site of the culture is blood, and
   The identity of the organism is not known with certainty,
and
   The stain of the organism is grammed, and
   The morphology of the organism is rod, and
   The patient has been seriously burned
THEN There is weakly suggestive evidence (0.4) that the
identity of the organism is pseudonomas.
```

Figura 3 - Uma regra do Mycin

Idealmente, o conhecimento representado na BC deve ser facilmente compreendido por quem o lê, mesmo sem formação especial em computadores. Cada fragmento do conhecimento (por exemplo, uma regra), deve ser independente dos outros e não deve conter qualquer detalhe relativo à ordem pela qual a sua utilização é feita. O conhecimento é representado através de um ou mais métodos de representação, usando ferramentas computacionais adequadas.

Motor de Inferência

Motor de inferência é uma expressão usada historicamente que designa o componente do sistema baseado em conhecimento capaz de inferir nova informação a partir de conhecimento e informação existentes na BC (por exemplo, dedução com encadeamento para trás), capaz de consultar e de criar conhecimento na BC, capaz de

efetuar certas operações pré-definidas (por exemplo, operações aritméticas, operações com conjuntos), e de explicar o raciocínio do sistema.

Interface

A Interface compreende um componente de interação com o utilizador em modo de consulta, e um componente de criação e de atualização da BC. A criação e atualização da BC poderão ser feita pelo próprio utilizador, por um engenheiro do conhecimento, ou pelo técnico especialista de manutenção do sistema (dependendo do tipo do sistema e de restrições impostas à sua utilização).

Tanto o modo de consulta como o modo de criação e atualização têm, em geral, várias opções. A Interface encarrega-se do controlo da interação com essas opções.

O componente de criação e atualização da BC dispõe de visualizadores, editores e depuradores (“debuggers”) do conhecimento que facilitam a sua criação, inspeção e modificação.

4 Métodos de Raciocínio e de Representação de Conhecimento

O conhecimento de um sistema baseado em conhecimento é guardado na Base de Conhecimentos usando um determinado método de representação de conhecimento e é processado pelo Motor de Inferência do sistema usando um mecanismo de processamento de informação adequado ao método de representação usado. Um dos métodos mais conhecidos de representar conhecimento é a representação por regras, o qual está em geral associado a um tipo de raciocínio baseado na dedução. Existem, no entanto outros tipos de representação de conhecimento que se podem usar em Sistemas Baseados em Conhecimento, entre os quais, a representação em lógica, a programação em lógica, as regras, os esquemas – enquadramentos (*frames*), objetos, casos, guiões (*scripts*) –, as redes semânticas, e os grafos de dependência conceptual.

Representação em Lógica: $\forall x \forall y x > y \Rightarrow \text{Maior}(x, y) = x$

Programação em Lógica: $\text{Maior}(X, Y, X) :- X > Y.$

Regras: IF $X > Y$ THEN $\text{Maior}(X, Y) = X$

Como se constata, a representação em lógica, a representação através da programação em lógica e a representação através de regras são todas muito semelhantes. Na verdade, a programação em lógica e as regras são ambos métodos de representação baseados na lógica.

Redes semânticas: A Ana pensa que o João bebe cerveja, mas não é verdade.

Estas afirmações contêm conhecimento que pode ser representado através de redes semânticas como a que se ilustra na Figura 4.

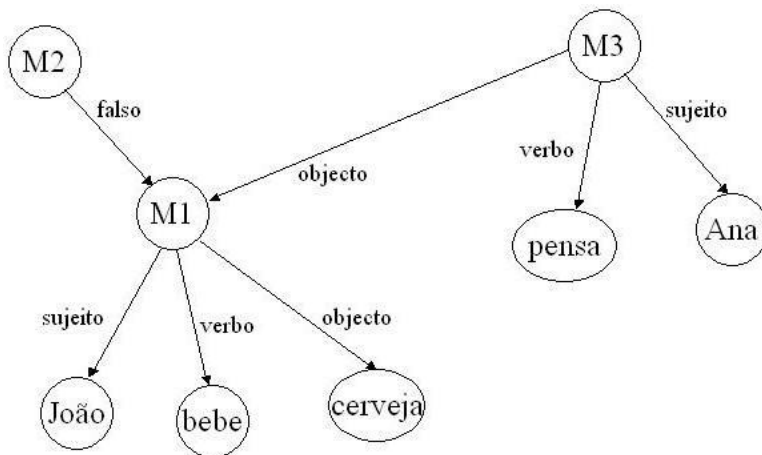


Figura 4 – Rede Semântica

Numa rede semântica, os nós representam conceitos e os arcos representam relações estruturais entre conceitos. As relações conceptuais são também representadas por nós. Os nós João, bebe, cerveja, pensa e Ana representam conceitos atômicos. Os nós M1, M2 e M3 representam conceitos moleculares, isto é, relações conceptuais. M1 representa o conceito “O João bebe cerveja”. M2 representa o conceito “Não é verdade que o João bebe cerveja” e M3 representa o conceito “A Ana pensa que o João bebe cerveja”.

Infelizmente, nem todas as redes semânticas apresentam um esquema de representação de conhecimento tão claro como este. A heterogeneidade das redes semânticas é um dos maiores problemas relativos à sua utilização e disseminação.

Guiões (“Scripts”)

Os guiões são esquemas de eventos. São usados para representar situações típicas, como por exemplo, a ida a um restaurante (Figura 5). Frequentemente, os sistemas baseados em guiões usam-nos para reconhecer exemplos de situações típicas. Outras vezes, os guiões controlam a ação de personagens animadas.

Name: restaurant

Resources: tables, chairs, food, bill, money, tip

Roles: waiter, waitress, customer, cook, cashier, owner

Requisites: customer hungry, customer has money

Results: customer has less money; owner has more money; customer not hungry

Scene 1: Entering

Customer enters restaurant

Customer looks for table

Customer decides where to sit

Customer sits down

Scene 2: Ordering

Scene 3: Eating

Scene 4: Leaving

Figura 5 – Guião da ida a um restaurante

O guião representado na Figura 5 chama-se “restaurant”; tem um conjunto de recursos que vai usar (tables, chairs, food, bill, money, tip); refere um conjunto de papéis (“roles”) desempenhados por agentes (no caso, pessoas); tem um conjunto de condições de utilização (só é usado se o cliente está com fome e se o cliente tiver dinheiro); tem também um conjunto de resultados (o cliente fica com menos dinheiro e sem fome, o dono do restaurante fica com mais dinheiro). Do ponto de vista, procedimental, o guião especifica que há uma entrada no restaurante, seguida da escolha e encomenda dos pratos, depois passa-se para a ingestão da refeição, e finalmente, o cliente sai do restaurante. A entrada é ainda descrita em maior detalhe.

Para aumentar a flexibilidade da representação por guiões, estes possibilitam a especificação de cenas facultativas e de alternativas.

Tal como acontece com as redes semânticas, a heterogeneidade deste sistema de representação é um dos principais fatores da sua fraca disseminação.

Raciocínio Baseado em Casos (CBR, “Case Based Reasoning”)

O raciocínio baseado em casos é uma tecnologia de representação de conhecimento e de raciocínio em que se usam descrições de casos anteriores para resolver novos problemas. De um modo geral, um sistema baseado em casos possui uma base de casos que descrevem problemas já encontrados e resolvidos pelo sistema. Quando surge um novo problema, o sistema usa e possivelmente adapta um caso encontrado anteriormente para resolver o novo problema.

O raciocínio associado a este tipo de representação é a analogia. Numa primeira fase seleciona-se o caso mais semelhante ao novo problema e estabelece-se uma analogia entre os dois, isto é, uma relação que descreve as semelhanças e diferenças entre as descrições dos problemas. Numa segunda fase usa-se esta analogia para produzir uma solução para o novo problema através da possível modificação da solução para o problema anterior. A modificação / adaptação da solução anterior é guiada pela relação de analogia.

New Case	Problem
<ul style="list-style-type: none"> • Symptom: Brake light does not work • Car: Audi 80 • Year: 1989 • Battery Voltage: 12.6V • State of lights: OK 	Symptom: Front light doesn't work Car: VW Golf II, 1.6L Year: 1993 Battery Voltage: 13.6 State of lights: OK State of light switch: OK Solution <ul style="list-style-type: none"> • Diagnostic: Front light fuse defect • Repair: Replace front light fuse

Similarity

Adaptation: Replace brakes light fuse

Figura 6 – Representação de um caso e de um problema em CBR

Na Figura 6 representa-se um caso que descreve um problema anterior em que um automóvel VW tem uma luz da frente que não funciona e em que o diagnóstico do problema foi que o fusível da lâmpada da frente está defeituoso, e em que a solução do problema foi a substituição do fusível da lâmpada da frente; e apresenta-se também um novo problema que deverá ser solucionado pelo sistema: trata-se de um Audi em que a luz do travão não acende. Dada a elevada semelhança entre os dois casos, o sistema adapta a solução do problema anterior (substituir o fusível da lâmpada da frente) para solucionar o novo problema (substituir o fusível da lâmpada do travão).

Implementação

Existem diversas ferramentas computacionais para a implementação de Sistemas Baseados em Conhecimento, as quais variam desde as mais simples até às mais sofisticadas. Do lado simples, existem as linguagens tipicamente usadas na Inteligência Artificial – Lisp e Prolog. Do lado mais sofisticado, existem os ambientes de desenvolvimento (“*Shells*”) – KEE, Golden Works, Kappa PC.

As linguagens de programação usadas em IA são ferramentas mais flexíveis mas menos sofisticadas. Quando muito, disponibilizam um único método de representação e manipulação de conhecimento. É necessário programar outros mecanismos de representação e processamento do conhecimento e os mecanismos de manutenção do sistema (validação, modificação, ...).

Os ambientes de desenvolvimento (“*Shells*”) são bem mais sofisticados do que as linguagens mas não raramente mais inflexíveis. Em geral, incluem diversos métodos de representação e manipulação de conhecimento (regras encadeadas para trás e para a frente, e objetos) e métodos mais ou menos sofisticados de manutenção (browsers de conhecimento, editores especializados, verificadores da base de conhecimento).