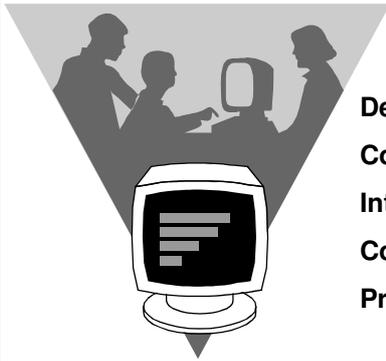


# Sistemas Baseados em Conhecimento (SBCs)



**Definição**

**Conceitos de SBC e SE**

**Introdução a SBC**

**Componentes Básicos de um SBC**

**Processo de Desenvolvimento de SBC**

Thiago A. S. Pardo  
Solange O. Rezende

**Inteligência Artificial** 1

## SBC: definição

*“Programas de computador que usam conhecimento representado explicitamente para resolver problemas”*

2

## Introdução

- ↪ Sistemas Baseados em Conhecimento (SBCs) têm sido utilizados por **mais de 20 anos**
  - ◆ Aplicados a problemas antes resolvidos somente por humanos
- ↪ SBCs são **usados quando**
  - ◆ A formulação genérica do problema a ser resolvido computacionalmente é complexa
  - ◆ Existe uma grande quantidade de conhecimento específico do domínio sobre como resolvê-lo
- ↪ Boa indicação para uso de um SBC: **existência de um especialista humano na área**

3

## Importância do conhecimento

- ↪ **Conhecimento é muito valorizado nas últimas 3 décadas** para resolução de problemas complexos
- ↪ **Maioria dos problemas de IA não permitem estratégias gerais de resolução**
- ↪ *“Para resolver problemas nas áreas em que até o momento o ser humano é melhor sucedido que a máquina, as máquinas precisam saber o que os humanos sabem sobre o assunto” (Rich e Knight, 1993)*

4

## Humanos vs. máquinas

- ↪ (Uma das) vantagens dos **humanos**
  - ◆ Conhecimento
  
- ↪ Vantagens das **máquinas**
  - ◆ Velocidade
  - ◆ Consistência
  
- ↪ Ideal: vantagens dos humanos E das máquinas
  - ◆ **SBCs estão na vanguarda desta lista**
    - Componentes: base de conhecimento e mecanismo de raciocínio

5

## SBCs

- ↪ **Importância dos SBCs**: capacidade de preservar, aproveitar e fazer uso do talento e experiência humanos no processo de tomada de decisão
  
- ↪ **Portanto, deve haver uma maneira de capturar, organizar e disponibilizar (em uma base de conhecimento) o conhecimento humano**
  - ◆ Uma vez na base, conhecimento deve ser acessível e facilmente recuperável

6

## Aplicações de SBCs

- ↵ Atualmente, diversas aplicações de SBCs
  - ◆ Engenharias
  - ◆ Ciências
  - ◆ Medicina
  - ◆ Negócios
  - ◆ Etc.
- ↵ Grande interesse comercial e acadêmico

7

## SBCs vs. sistemas convencionais

- ↵ Para um sistema inteligente ser classificado como SBC, o sistema deve ser capaz de
  - ◆ Questionar o usuário, usando uma linguagem de fácil entendimento, para reunir informações necessárias
  - ◆ Desenvolver uma linha de raciocínio a partir dessas informações e do conhecimento que contém para resolver o problema; deve lidar com regras e informações incompletas, imprecisas e conflitantes
  - ◆ Explicar seu raciocínio
  - ◆ Conviver com seus erros, mas com desempenho satisfatório (similarmente ao especialista humano)
- ↵ As características acima não necessariamente distinguem um SBC de um sistema convencional
  - ◆ Por quê?

8

## SBCs vs. sistemas convencionais

↪ **Reformulando...** SBCs devem possuir as seguintes propriedades

- ◆ Tudo que se sabe sobre o problema deve estar explicitamente representado na base de conhecimento do sistema
- ◆ A base de conhecimento deve ser interpretada por um mecanismo de inferência
- ◆ Os problemas resolvidos por SBCs são aqueles para os quais não é conhecido um procedimento determinístico
  - Em geral, o conhecimento é utilizado para contornar a exponencialidade da formulação genérica

9

## SBCs vs. sistemas convencionais

↪ Diferenças entre SBCs e sistemas convencionais

Sistemas convencionais	SBCs
Estrutura de dados	Representação do conhecimento
Dados e relações entre dados	Conceitos, relações entre conceitos e regras
Algoritmos determinísticos	Busca heurística
Conhecimento embutido no código do programa	Conhecimento representado explicitamente e separado do código que o manipula e interpreta
Explicação do raciocínio é difícil	Podem e devem explicar seu raciocínio

10

## SBCs e SEs

### ↪ SBCs & Sistemas Especialistas (SE)

- ◆ SBCs resolvem problemas usando conhecimento específico sobre o domínio da aplicação
- ◆ SEs são SBCs que resolvem problemas comumente resolvidos por humanos
  - Profunda interação com especialista

### ↪ SBCs podem ser classificados como SEs quando o desenvolvimento do mesmo é voltado para aplicações nas quais o conhecimento a ser manipulado restringe-se a um domínio específico e conta com um alto grau de especialização

- ◆ Mas... os termos SBC e SE são usados indistintamente na área

11

## Sistemas Especialistas (SE)

*“SE são sistemas que são capazes de oferecer soluções para problemas específicos num dado domínio ou que consigam aconselhar (dar conselhos), de uma maneira ou num nível comparável ao de especialistas naquela área.”*

Lucas and van der Gaag

*Princípios de Sistemas Especialistas*

12

## SBC e Sistemas Especialistas



## Construção de SBCs

- ↳ Para a construção de um SBC, deve-se verificar a **viabilidade da aplicação dessa tecnologia** no domínio em questão
- ↳ **Justificativas** para a construção de um SBC
  - ◆ Tarefas remunerativas (para compensar o custo do desenvolvimento do SBC)
  - ◆ Falta/dispersão de especialistas
  - ◆ Domínio bem delimitado
  - ◆ Retenção de conhecimento (que poderia ser perdido)

## Introdução a SBC (cont)

- ↪ Para fazer com que um Sistema Baseado em Conhecimento chegue perto do **desempenho de um especialista humano**, o sistema deve:
  - ♦ ter grande quantidade de conhecimento disponível
  - ♦ conseguir ter acesso a este conhecimento rapidamente e
  - ♦ ser capaz de raciocinar adequadamente com este conhecimento

15

## Tipos de aplicações de SBCs

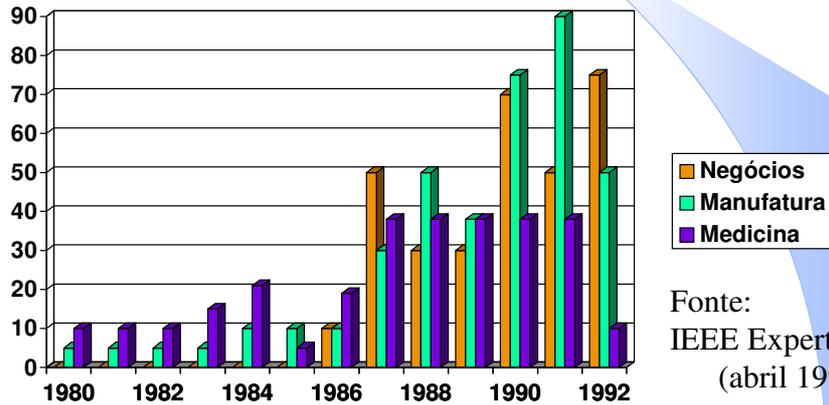
- ↪ **Classes de tarefas** em que os SBCs têm sido aplicados
  - ♦ Interpretação e análise de dados: processamento de imagens, reconhecimento de fala, análise de circuitos elétricos
  - ♦ Classificação: diagnóstico de doenças, determinação de falhas em máquinas
  - ♦ Monitoramento: usinas nucleares, tráfego aéreo
  - ♦ Planejamento: ações de robôs, experimentos em genética, ações militares
  - ♦ Projeto: layout de circuitos e de computadores, tubulações de aviões
- ↪ Surgimento de diversas ferramentas de auxílio a construção e execução de SBCs, linguagens de representação de conhecimento e literatura especializada

16

## Introdução a SBC (cont)

↪ Alguns exemplos incluem: diagnóstico médico, aconselhamento financeiro e criação de produtos

↪ Número de SBCs desenvolvidos por ano



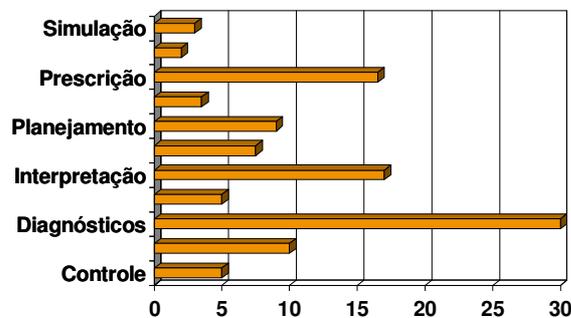
Fonte:  
IEEE Expert  
(abril 1996)

17

## Introdução a SBC (cont)

↪ Como a maioria dos sistemas de IA, um SBC deve trabalhar dentro de sua própria área de conhecimento

↪ Porcentagem de SBC por tipo de problema



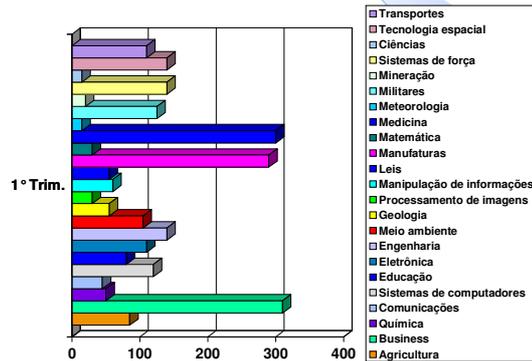
Fonte: IEEE Expert  
(abril 1996)

18

## Introdução a SBC (cont)

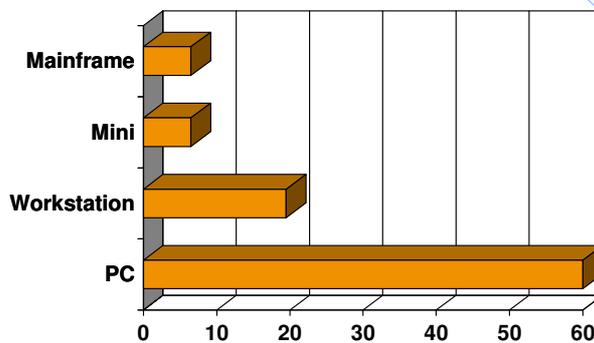
Com o aumento da velocidade computacional e capacidade de armazenamento, SBCs estão sendo aplicados a um conjunto cada vez maior de áreas especializadas.

Fonte: IEEE Expert  
(abril 1996)



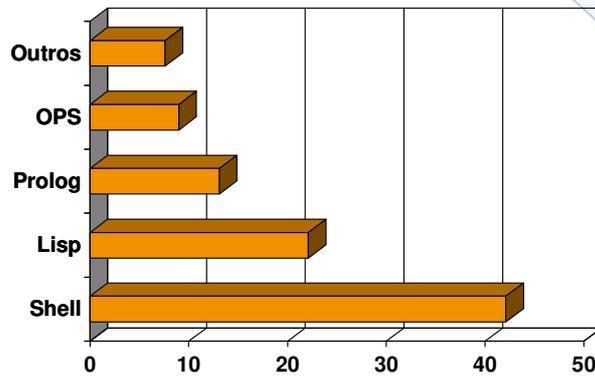
19

## Porcentagem de SBCs desenvolvidos em diferentes plataformas



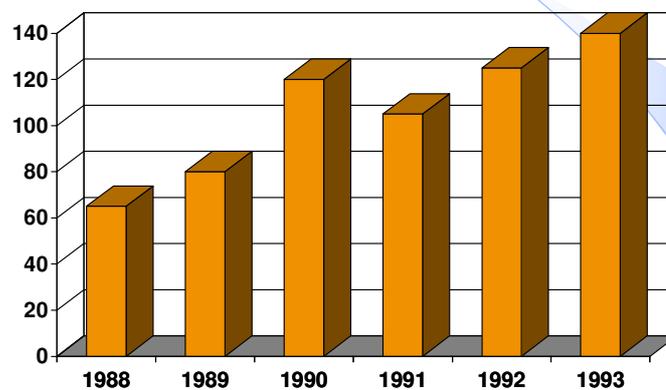
20

## Porcentagem de SBCs desenvolvidos com diferentes softwares



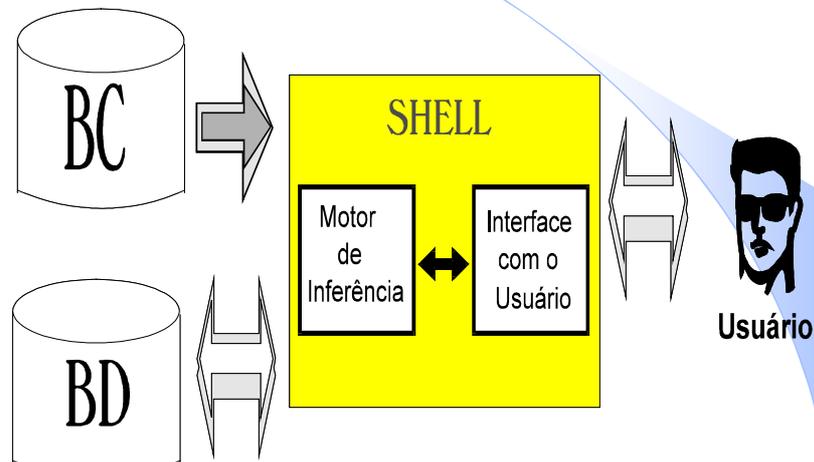
21

## Vendas de ferramentas para desenvolvimento de SBC por ano.



22

## Estrutura Básica de um SBC



23

## Componentes Básicos de SBCs

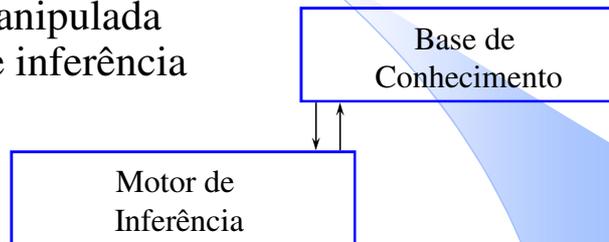
↪ SBC têm **dois** componentes essenciais:

- ♦ Uma *base de conhecimento* que captura o conhecimento específico do domínio e
- ♦ Um *motor de inferência* consistindo de algoritmos para manipular o conhecimento representado na base de conhecimento.

24

## Construindo um SBC

- ↪ Base de conhecimento é consultada/manipulada pelo motor de inferência



25

## A Base de Conhecimento

- ↪ A base de conhecimento é composta de fatos e regras que o sistema conhece sobre o domínio do problema.
- ↪ Conhecimento é representado na base de conhecimento usando uma das técnicas de RC como: **regras de produção**, redes semânticas, frames, scripts, etc.
- ↪ SBCs também podem usar uma mistura de técnicas de RC: estes tipos de sistemas são chamados *Sistemas Híbridos*

26

## Certeza do Conhecimento

- ↪ Conhecimento também pode incluir probabilidades ou fatores de **incerteza**, os quais podem ser usados para:
  - ◆ melhorar a corretude das tomadas de decisões
  - ◆ ajudar a resolver conflitos e
  - ◆ melhorar recursos de explicação
- ↪ Técnicas para lidar com **incerteza** incluem:
  - ◆ Método Bayesiano
  - ◆ Teoria de Evidência de Dempster-Shafer
  - ◆ Teoria da Certeza (Fatores de Certeza) e
  - ◆ Lógica Fuzzy

27

## O Motor de Inferência

- ↪ O motor de inferência inclui um **interpretador**, bem parecido com o motor utilizado num sistema de produção, que aciona regras de uma base de conhecimento e executa itens da agenda
- ↪ Ele também inclui:
  - ◆ Um **agendador** que mantém o controle da agenda
  - ◆ Um **Verificador de Consistência** que tenta manter uma representação consistente da solução que surge

28

## Fazendo Inferências com as Regras de Produção

### ↳ Regras de produção

- ♦ Uma das técnicas mais antigas e bem sucedidas para representar conhecimento em SBCs
  - Regras “se PREMISSAS então CONCLUSÃO”

– Questão: como seria em Prolog?

29

## Fazendo Inferências com as Regras de Produção

### **Regra 1**

se  
o motor está recebendo combustível e  
o motor tenta pegar  
então  
o problema é vela

### **Regra 2**

se  
o motor não tenta pegar e  
as luzes não acendem  
então  
o problema é bateria e cabo

### **Regra 3**

se  
o motor não tente pegar e  
as luzes acendem  
então  
o problema é motor de partida

### **Regra 4**

se  
houver combustível no tanque de  
combustível e  
houver combustível no carburador  
então  
o motor está recebendo combustível

30

## Fazendo Inferências com as Regras de Produção

- ↳ Um SBC que utiliza regras de produção, faz uso das estratégias de raciocínio:
  - ♦ *forward chaining* ou encadeamento progressivo
  - ♦ *backward chaining* ou encadeamento regressivo

31

## Forward Chaining

A direção da busca é dos **dados para as metas** ou hipóteses

*Se* cond1 e cond2 e .. condn

*Então* ação1

→  
direção da busca

O que podemos concluir a partir dos dados?

32

## Backward Chaining

A direção da busca é das metas para os dados

*Se* cond1 e cond2 e .. condn

*Então* ação1

direção da busca  
←

É possível provar as hipóteses a partir dos dados?

33

## Exemplo

- ↗ Voar de São Carlos até Roma, imaginando que não houvesse vôos diretos
- ↗ Encontrar uma seqüência de vôos interligados que começasse em São Carlos e terminasse em Roma.

34

## Exemplo Usando Forward Chaining

- ↪ Começando em São Carlos, e indo para a frente, seria necessário:
- ◆ encontrar todos os vôos saindo de São Carlos, e anotar a cidade onde cada vôo iria aterrissar;
  - ◆ encontrar todos os vôos de cada uma dessas cidades, e anotar a cidade onde cada vôo aterrissa;
  - ◆ continuar fazendo isto até que uma das cidades de destino fosse Roma.

35

## Exemplo Usando Backward Chaining

- ↪ Começando em Roma, e voltando, seria necessário:
- ◆ encontrar todos os vôos que chegam em Roma, e anotar a cidade de origem de cada um;
  - ◆ encontrar todos os vôos que chegam em cada uma destas cidades, e anotar a cidade de origem de cada um;
  - ◆ continuar fazendo isto até que uma das cidades de origem fosse São Carlos.

36

## Exercício: backward chaining

### Regra 1

se  
o motor está recebendo combustível e  
o motor tenta pegar  
então  
o problema é vela

### Regra 2

se  
o motor não tenta pegar e  
as luzes não acendem  
então  
o problema é bateria e cabo

### Regra 3

se  
o motor não tenta pegar e  
as luzes acendem  
então  
o problema é motor de partida

### Regra 4

se  
houver combustível no tanque de  
combustível e  
houver combustível no carburador  
então  
o motor está recebendo combustível

37

↳ 1. Especifica-se objetivo de mais alto nível: problema é X

↳ 2. Dispara regra 1

MT: o motor está recebendo combustível  
o motor tenta pegar  
o problema é vela

↳ 3. Dispara regra 4 com “motor está recebendo combustível”

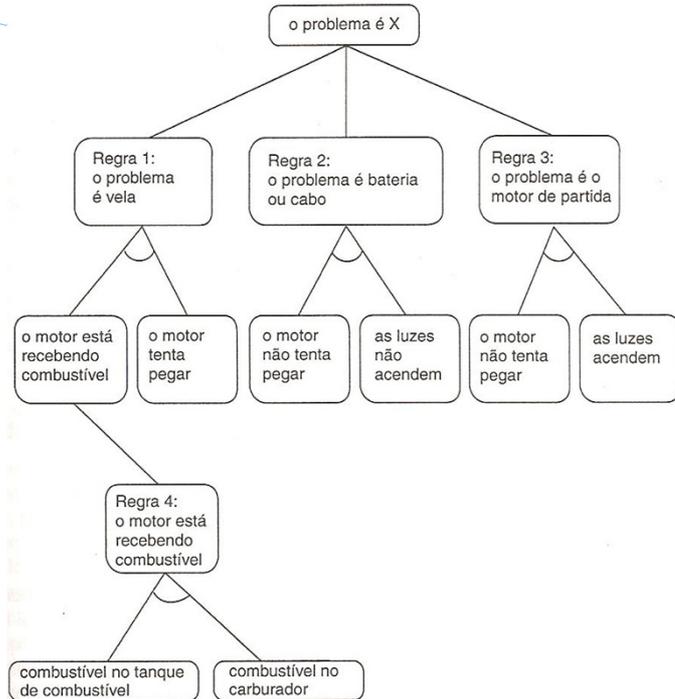
MT: combustível no tanque de combustível  
combustível no carburador  
o motor está recebendo combustível  
o motor tenta pegar  
o problema é vela

↳ 4. Nada casa com regra alguma, então se questiona usuário

Se os três sub-objetivos forem verdadeiros, então o problema é vela

38

## Árvore de busca



## Exercício: forward chaining

### Regra 1

se  
o motor está recebendo combustível e  
o motor tenta pegar  
então  
o problema é vela

### Regra 2

se  
o motor não tenta pegar e  
as luzes não acendem  
então  
o problema é bateria e cabo

### Regra 3

se  
o motor não tenta pegar e  
as luzes acendem  
então  
o problema é motor de partida

### Regra 4

se  
houver combustível no tanque de  
combustível e  
houver combustível no carburador  
então  
o motor está recebendo combustível

## ↳ Esquema geral de funcionamento

- ◆ Ativam-se todas as regras em sequência, várias vezes até que se tenha uma resposta
- ◆ Dados que também são objetivos não são questionados

41

## ↳ 1. Dispara regra 1

→ como “motor está recebendo combustível” é conclusão de outra regra e não pode ser perguntado, essa regra falha

## ↳ 2. Dispara regra 2

→ questiona-se “o motor não tenta pegar”?

resposta “falso”

MT: motor tenta pegar

→ regra 2 falha

## ↳ 3. Dispara e falha regra 3, dado que “motor tenta pegar” está em MT

42

4. Dispara regra 4

→ questionam-se as duas premissas

respostas “verdadeiro”

MT: o motor está recebendo combustível  
há combustível no carburador  
há combustível no tanque de combustível  
o motor tenta pegar

5. Recomeça-se a busca a partir da regra 1

Regra 1 é verdadeira e conclusão é que “o problema é vela”

MT: o problema é vela  
o motor está recebendo combustível  
há combustível no carburador  
há combustível no tanque de combustível  
o motor tenta pegar

43

## Componentes Adicionais

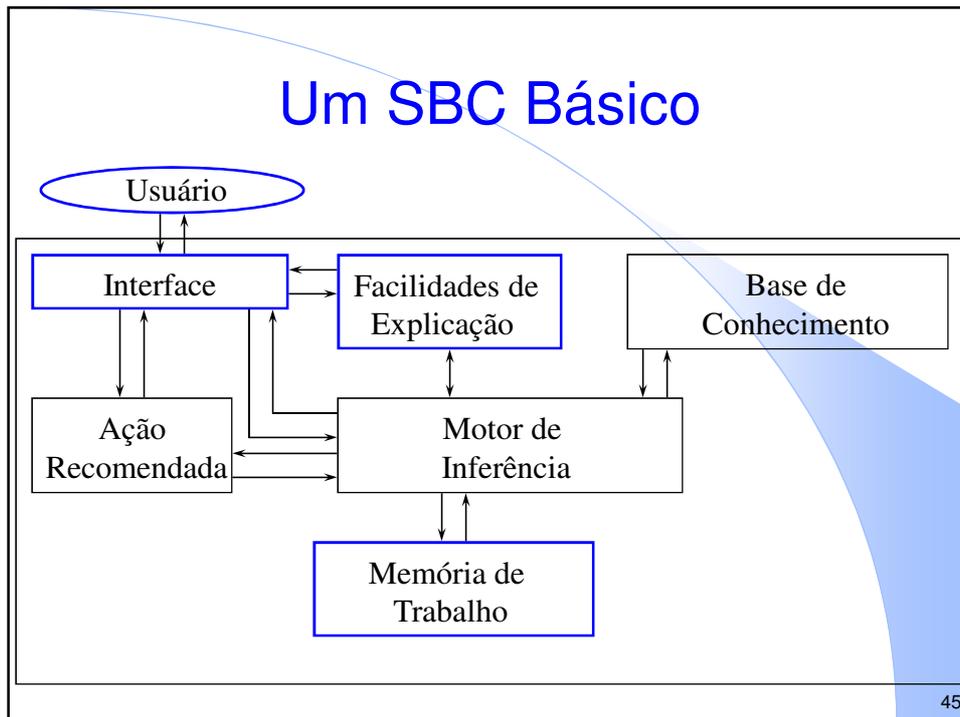
↳ Tendo o núcleo de um SBC, quais outros componentes são necessários?

- ◆ Um **usuário**;
- ◆ Um **modo dos usuários interagirem** com o sistema;
- ◆ Um lugar para armazenar o **conhecimento usado no trabalho**;
- ◆ Uma maneira de se conseguir **ajuda** do sistema.

- Isso é necessário para se interagir com o motor de inferência e a base de conhecimento

44

## Um SBC Básico



## Usuários

↪ SBCs são projetados para interagir com vários **tipos de usuários**, agindo de diferentes maneiras conforme as circunstâncias:

- ◆ Um **leigo** procurando ajuda direta - *modo consultor*;
- ◆ Um **estudante** que quer aprender - *modo instrutor*;
- ◆ Um **construtor de SBC** melhorando ou aumentando a base de conhecimentos - *modo acompanhante*; e
- ◆ Um **especialista** - *modo cooperativo*.

## Interface

- ↗ A **Interface** é um **processador de linguagem** projetado para processar e produzir comunicação orientada a problemas entre o usuário e o sistema.
- ↗ Isto usualmente ocorre numa língua natural, sendo complementada por menus e gráficos.
- ↗ O Processamento de Línguas Naturais é, portanto, uma parte importante dos SBCs.

47

## Interface

- ↗ Linguagem mais abstrata do que a de representação do conhecimento e mais limitada do que a língua humana
  - **Linguagem controlada**
- ↗ **Sistemas convencionais vs. SBC**
  - ◆ Convencionais: interfaces estáticas, com tudo pré-definido
  - ◆ SBC: interfaces mais dinâmicas, que permita a leitura, impressão e manipulação de dados diversos; podem se adaptar ao usuário e seus interesses
    - Questões de IHC
    - Outras linguagens: diagramas, hipermídia, etc.

48

## Memória de Trabalho

- ↪ A **memória de trabalho** do sistema armazena condições iniciais, hipóteses intermediárias e decisões, e soluções finais.
- ↪ Informação é classificada em três tipos:
  - ◆ planos (como resolver o problema);
  - ◆ agendas (ações potenciais a serem executadas); e
  - ◆ soluções (soluções candidatas e hipóteses intermediárias).

49

## Facilidades de Explicação

- ↪ O módulo que facilita a explicação pode **justificar** as conclusões, e auxiliar a explicar o comportamento do SBC.
- ↪ Isto é feito através de questões interativas:
  - ◆ *Por que o sistema faz uma pergunta em particular?*
  - ◆ *Como o sistema alcança a conclusão correta?*
  - ◆ *Por que uma certa alternativa é rejeitada?*
  - ◆ *Qual é a tática atual do sistema para alcançar a conclusão; etc.*

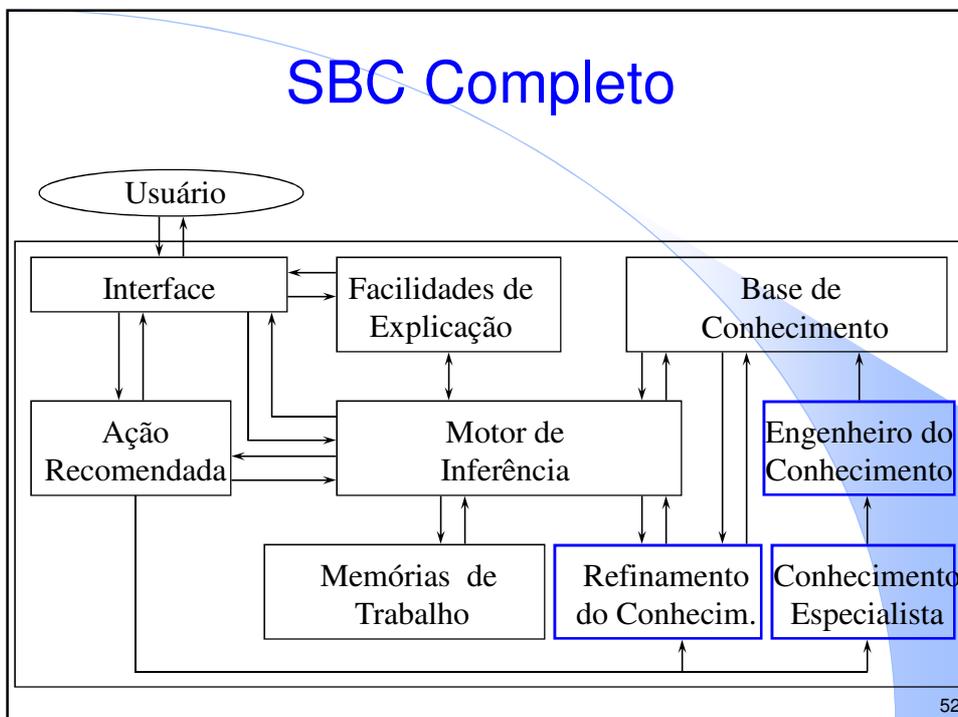
50

## Aquisição de Conhecimento

- ↪ Como o sistema **adquire conhecimento**?
  - ◆ Como o **conhecimento é inicialmente trazido** para dentro do sistema?
  - ◆ Como o sistema adquire **novos conhecimentos**?
  - ◆ Como este **conhecimento pode ser testado**?
- ↪ Quais componentes necessitamos acrescentar ao sistema para realizar estas funções?

51

## SBC Completo



52

## Conhecimento Especialista

↪ Adquirir conhecimento especialista para uma base de um SBC envolve:

- ◆ Obter informação dos especialistas e/ou de fontes documentadas;
- ◆ Classificação desta informação em *declarativa* (factual) e *procedural* ;
- ◆ Codificação desta informação num formato utilizado pelo SBC; e
- ◆ Checagem de consistência do conhecimento codificado com o conhecimento existente no sistema.

53

## O Engenheiro do Conhecimento

↪ O Engenheiro do Conhecimento realiza as seguintes tarefas:

- ◆ Estrutura a área do problema;
- ◆ Interpreta, traduz e integra conhecimento especialista ao sistema;
- ◆ Traça analogias;
- ◆ Apresenta contra exemplos;
- ◆ Traz à luz conceitos difíceis; e
- ◆ Checa a consistência do conhecimento.

54

## Refinamento do Conhecimento

- ↳ Futuros SBC poderão ser capazes de **monitorar, analisar, aprender e melhorar** suas próprias performances, resultando numa base de conhecimento mais aprimorada e num raciocínio mais efetivo.
- ↳ Entretanto, nos SBCs atuais esta tarefa é realizada pelo **Engenheiro de Conhecimento**.

55

## Shell de um SBC

- ↳ **Shell** de um SBC é um SBC com uma base de conhecimento vazia.
- ↳ “Shells são destinadas a permitir que não-programadores usufruam dos esforços de programadores que já resolveram um problema similar.”

56

## Processo de Desenvolvimento SBC

Especificação dos Requisitos



Aquisição de Conhecimento



Projeto



Implementação



Teste



Manutenção

57

## Processo de Desenvolvimento SBC

Especificação dos Requisitos



Aquisição de C



Proje



Implementação



Teste



Manutenção

Estudo de Viabilidade

Desenvolvimento de Metas

Refinamento do Domínio

Escolha da Equipe de Projeto

Identificação de Fontes de Conhecimento

58

## Processo de Desenvolvimento SBC

Especificação dos Requisitos



Aquisição de Conhecimento



Projeto



Implementação



Teste



Manutenção

Adquirir,  
Organizar &  
Estruturar o Conhecimento  
Documentação  
- Dicionário de Conhecimento

59

## Processo de Desenvolvimento SBC

Especificação dos Requisitos



Aquisição de Conhecimento



Projeto



Implementação



Teste



Manutenção

Definição da Estrutura & Organização  
do Conhecimento  
Definir Métodos para Processamento  
Selecionar Ferramenta de Software  
Documentação

60

## Processo de Desenvolvimento SBC

Especificação dos Requisitos



Aquisição de Conhecimento



Projeto



Implementação



Teste



Manutenção

Codificação  
Documentação do Sistema  
Manual do Usuário

61

## Processo de Desenvolvimento SBC

Especificação dos Requisitos



Aquisição de Conhecimento



Projeto



Implementação



Teste



Manutenção

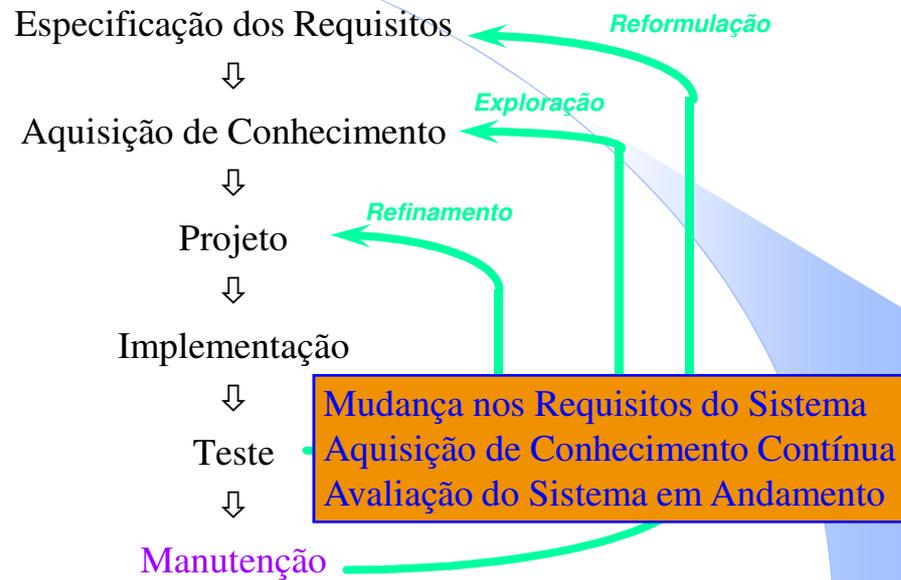
Validação & Verificação  
Processo contínuo.. fases se misturam

*Exploração*

*Refinamento*

62

## Processo de Desenvolvimento SBC



## Verificação e Validação

↳ Há uma necessidade de assegurar que:

- ♦ O sistema funcione
- ♦ O sistema forneça resultados verdadeiros (corretos)
- ♦ O sistema satisfaça os requisitos do cliente

## O IEEE define VV:

- ↪ **Verificação**: processo que determina se os produtos de uma dada fase de desenvolvimento do software satisfaz todos os requisitos estabelecidos durante a fase anterior. “O sistema foi construído corretamente?”
- ↪ **Validação**: processo de avaliar software no fim do processo de desenvolvimento para assegurar que está de acordo com os requisitos do software. “É o sistema certo?”

65

## Verificação e Validação

- ↪ As estratégias de teste mais eficazes são constituídas de vários testes pequenos e separados.
- ↪ O sistema pode ser testado com um conjunto estruturado de problemas teóricos, e com problemas reais.
- ↪ O desempenho pode ser avaliado pela comparação de informações históricas ou com recomendações paralelas de especialistas.

66

## Verificação e Validação

↳ **Técnicas de validação** incluem:

- Simulação
- Teste de Turing
- Revisão por especialista(s)
- Revisão pelos usuários finais

67

## Verificação e Validação

↳ **Problemas:**

- Quais **características** devem ser avaliadas?

68

## Verificação e Validação

### ↳ Problemas:

- Quais **características** devem ser avaliadas?
  - Índice de acerto
  - Velocidade
  - Facilidade de Uso
  - Flexibilidade
  - Robustez
  - Comportamento de Erro

69

## Verificação e Validação

### ↳ Problemas:

- Qual o “**benchmark**” utilizado para desempenho?
  - Tão bom quanto um “especialista razoável”?
  - Tão bom quanto “os melhores”?
  - Melhor que qualquer especialista humano?
  - Como dizer se é ou não?
    - (Quem pode julgar melhor o desempenho do melhor especialista?)

70

## Verificação e Validação

### ↳ Problemas:

- ◆ Escolha de **Dados Exemplo**
  - Os dados de teste são “realistas”?
  - Como o sistema trata **dados inválidos**?
  - E se **dados reais** forem **raros** ou tiverem um “ciclo de resultado” longo?

71

## Alguns Exemplos de SBCs

### ↳ Alguns dos primeiros exemplos são:

- ◆ MYCIN;
- ◆ INTERNIST/CADUCEUS;
- ◆ DENDRAL;
- ◆ DENDRAL HEURÍSTICO; e
- ◆ XCON.

### ↳ Alguns descendentes diretos desses são:

- ◆ NEOMYCIN; e
- ◆ METADENDRAL

72

## Mais Alguns Exemplos

- ↪ Alguns exemplos mais recentes incluem:
  - ◆ PROSPECTOR;
  - ◆ CONSELHEIRO DIPMETER;
  - ◆ FOSSIL;
  - ◆ SPAM;
  - ◆ ACE;
  - ◆ RESEDA;
  - ◆ PUFF; e
  - ◆ CENTAUR;

73

## MYCIN

- ↪ Um dos primeiros, e talvez o melhor Sistema Especialista conhecido, é **MYCIN**.
  - ◆ Acima de 90% de acerto
- ↪ MYCIN é um **Sistema Especialista médico**, projetado para ser capaz de auxiliar médicos no diagnóstico de doenças infecciosas, e então sugerir o possível tratamento.

74

## Dentro do MYCIN

- ↪ Como testes de laboratório podem tomar um certo número de dias e o tratamento de muitas doenças infecciosas necessita de um rápido início, é necessário diagnosticar o paciente o mais rápido possível.
- ↪ MYCIN pode dar diagnósticos antes que os resultados de laboratórios estejam disponíveis.

75

## Elementos do MYCIN

- ↪ MYCIN também inclui:
  - ◆ Uma base de conhecimento consistindo de **500 regras** de produção (regras IF-THEN);
  - ◆ **Regras probabilísticas**, que permitem traçar conclusões plausíveis a partir de evidências incertas;
  - ◆ **Backward chaining** exaustivo para encontrar um diagnóstico;
  - ◆ **Facilidade de explicações** que pode explicar seu raciocínio; e
  - ◆ Uma **interface em linguagem natural**, utilizável por qualquer um.

76

## Uma Sessão Exemplo com MYCIN

↪ MYCIN começa com a inclusão de informações dos sintomas do paciente e sua ficha médica. Então ele pergunta :

MYCIN: Qual é a infecção?

Médico: Endarteritis.

MYCIN: Is Organism-1 a rod or a coccus?

Médico: Rod.

↪ MYCIN então pergunta questões adicionais. . .

77

## Uma Sessão Exemplo com MYCIN (cont)

↪ Para garantir que o médico possa confiar no diagnóstico do MYCIN, o programa fornece as evidências:

Médico: Como você decidiu que o Organism-1 é *E. coli*?

MYCIN: Eu usei a regra 084 para concluir que Organism-1 é *E. coli*. Isto segundo um fator de certeza de 0.51.

↪ Depois de obter um diagnóstico, MYCIN recomendará um tratamento adequado.

78

## MYCIN (cont)

### ↳ Exemplo de regra

```
if the infection is meningitis and  
the type of infection is bacterial and  
the patient has undergone surgery and  
the patient has under gone neurosurgery  
and  
the neurosurgery-time was < 2 months ago  
and  
the patient got a ventricular-urethral-  
shunt  
then infection = e.coli(.8) or  
klebsiella(.75)
```

79

## Desenvolvimento de um SBC

### ↳ Conhecimento essencial, mas...

grande gargalo na construção de um SBC

80

## Exercício para início da próxima aula

### ↳ Em duplas

- ◆ Definir um domínio de conhecimento de interesse da dupla
- ◆ Construir pelo menos 10 regras de produção
  - Garantindo que haja interdependência entre algumas delas, pelo menos
- ◆ Trazer as regras no papel

81

## Exercício para início da próxima aula

### ↳ Aula no LAB 7

- ◆ 1ª metade da turma às 14h
  - Nomes de A a J
- ◆ 2ª metade às 15h
  - Nomes de J em diante

82