

Inteligência Artificial

Agentes Inteligentes

Problema: Auxílio a Compras na Web

Search the Web for documents in any language

audio amplifier

search refine

Click to find related books at Amazon.com.
Acquiring documents match your query.

1. **SIP326BC Stereo Audio Amplifier**
Presents STEREO AUDIO AMPLIFIER, SIP326BC, SCOPE OF THE STUDY. The stereo audio amplifier panel contains 12 fault insertion switches used to teach...
<http://www.cse.cmu.edu/~sug326bc.htm> - size 5K - 23-Jun-96 - English

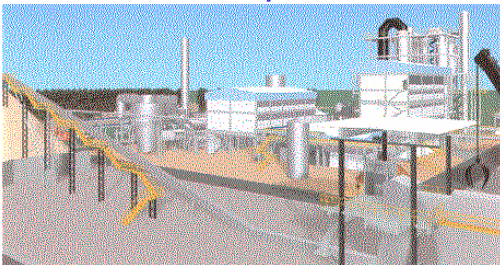
2. **Audio Amplifier Fine Tuning**
Audio Amplifier Fine Tuning. Most ham CW and SSB communication is between 3000Hz to 3000Hz2. The following is a chart reprinted from the SGS data sheet...
<http://www.pan-fax.net/~sunh/receivers/sgs710321.htm> - size 1K - 27-May-97 - English

3. **Audio Amplifier Users**
SB Electronics, Inc. Specialists in Film/Foil Capacitor Design and Manufacturing Since 1959! Audio Amplifier Users. Don't settle for anything but the...
<http://www.sbelectronics.com/user.htm> - size 75K - 8-Aug-97 - English

4. **Detail: CK151 Audio Amplifier Kit**
CK151 Audio Amplifier kit \$16.99. A universal audio amplifier with many applications. Includes a microphone jack. Microphone input sensitivity: 5 mV...
<http://www.shopsite.com/ds-prod067.html> - size 2K - 19-Aug-97 - English

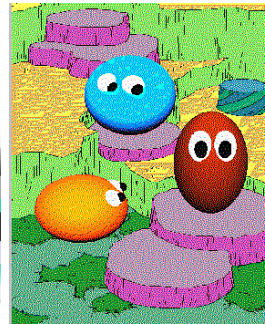
5. **Audio Amplifier**
Audio Amplifier. There have been two audio amplifiers that I have used for this receiver. One using a TDA2002, built exactly like the one in the Beginner...
<http://www.pan-fax.net/~sunh/receivers/sgs710321.htm> - size 2K - 26-May-97 - English

Problema: Automatização de sistemas de potência



objetos: rios, barragens, turbinas, transformadores, linhas, ...

Problema: Produção de histórias interativas



- Criar ilusão da vida (ex. Walt Disney)
- Permitir interação com usuário
- Modelar comportamento e personalidade (ex. tamagotchi)

Problema: navegação autônoma (ALVINN)

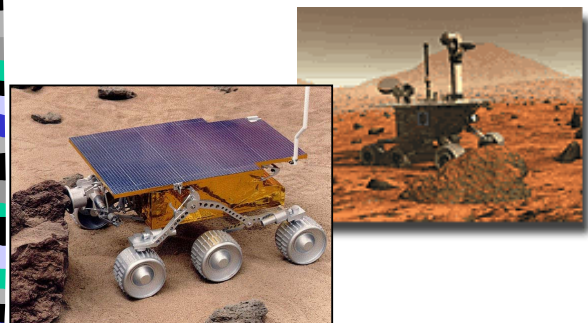


Velocidades → 100km/h

Distâncias → 140km

Usa RN associada a imagens para guiar uma van em rodovias públicas.

Problema: exploração planetária



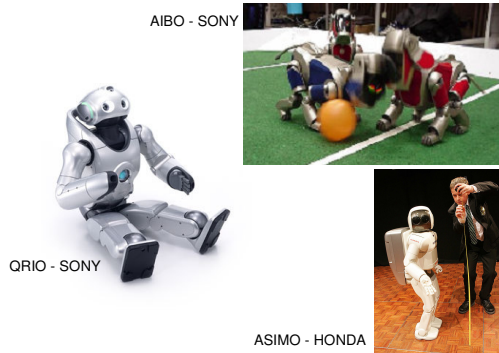
Problema: aspirador de pó automático



2005: 2 milhões de unidades vendidas para uso doméstico

7

Problema: robôs de estimação



8

E aí?

■ O que estes problemas têm em comum?

- Grande complexidade (número, variedade e natureza das tarefas)
- Não há “solução algorítmica”, mas existe conhecimento
- Modelagem do comportamento de um ser inteligente (autonomia, aprendizagem, conhecimento, etc.)

9

O que é “ser inteligente”?

- Ser inteligente é atuar como humanos.
- Ser inteligente é “pensar” como humanos.
- Ser inteligente é “pensar” racionalmente.
- Ser inteligente é atuar racionalmente.

Racionalidade = capacidade de alcançar o sucesso esperado na execução de uma tarefa.

Grau de Sucesso: medida de desempenho a ser maximizada.

10

■ Inteligência Artificial (IA)

- Há 50 anos lida com esses problemas.
- **Objetivo:** construir (e aprender a construir) programas que, segundo critérios definidos, exibem um **comportamento inteligente** na realização de uma dada tarefa.

11

Um programa de IA pode ser visto como um Agente Racional

■ Plano da aula

- O que é um Agente Racional (inteligente)?
- Ambientes e Arquiteturas
- Aplicações
- Estado atual do conceito de agente

12

O que é um agente?

■ Agente é qualquer entidade que:

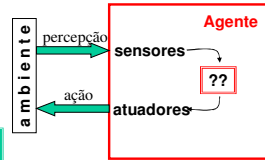
- **percebe** seu ambiente através de sensores (ex. câmeras, microfone, teclado, *finger*, ...)
- **age** sobre ele através de atuadores (ex. vídeo, auto-falante, impressora, braços, *ftp*, ...)

■ Ambiente/agente

- Físico: robôs
- Software: softbots
- Realidade virtual (simulação do ambiente físico): softbots e avatares

■ Efetua o mapeamento:

- sequência perceptiva → ação



13

Medida de Desempenho

■ Critério que define o grau de sucesso de um agente na realização de uma dada tarefa

O quê avaliar, Como avaliar, Quando avaliar

- Esta medida deve ser imposta do exterior
- Má escolha da MD pode acarretar comportamento indesejado
- Compromissos entre objetivos múltiplos conflitantes
- Resta o problema de saber **quando** avaliar o desempenho

14

Agente Racional

■ Agente Racional:

- “Para cada sequência perceptual possível, o agente racional deve **selecionar uma ação** que ele **espera que maximize sua medida de desempenho**, segundo a evidência dada pela **sequência perceptiva** e os eventuais **conhecimentos** que tenha”.

■ Limitações de:

- Sensores, atuadores, “raciocinador” (conhecimento, tempo, etc.)

■ Agente racional deve ser **autônomo**:

- Ter capacidade de raciocínio, decisão e de adaptação a situações novas, para as quais não foi fornecido todo o conhecimento necessário com antecedência

15

A metáfora de agente decompõe:

1) Problema em:

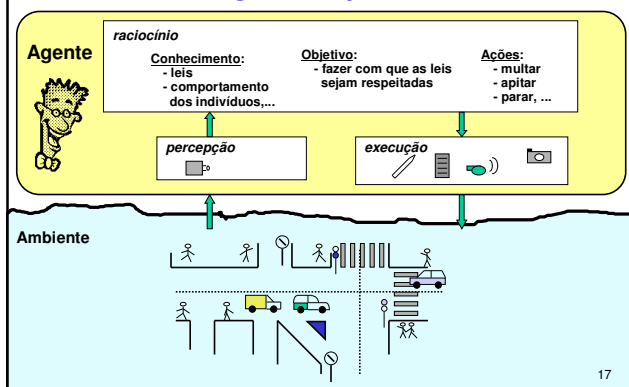
- percepções, ações, objetivos e ambiente (e outros agentes)

2) Tipo de conhecimento em:

- Quais são as propriedades relevantes do mundo
- Como o mundo evolui
- Como identificar os estados desejáveis do mundo
- Como interpretar suas percepções
- Quais as consequências de suas ações no mundo
- Como medir o sucesso de suas ações
- Como avaliar seus próprios conhecimentos

16

Agente de polícia



17

Propriedades do Ambiente

- **totalmente observável x parcialmente observável**
- **determinístico x estocástico** (envolve previsibilidade do próximo estado)
- **episódico x sequencial** (envolve ter ou não consequências futuras decorrentes da decisão atual)
- **estático x dinâmico** (envolve tempo)
- **discreto x contínuo** (aplicado a estado, tempo, ações e/ou percepções)
- **único agente x multiagente** (envolve comunicação, cooperação, competição..)

18

Estrutura do Agente

Agente = arquitetura de HW
+
arquitetura de SW

Arquitetura de HW:

- onde o agente vai ser implementado (dispositivo computacional, sensores e atuadores)

Arquitetura de SW:

- "arquitetura do agente": módulos básicos do programa e suas inter-relações

19

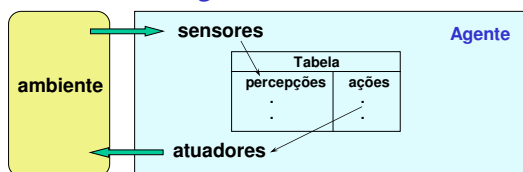
Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- Agente baseado em objetivos
- Agente baseado em utilidade
- Agente aprendiz

autonomia
complexidade

20

Agente tabela

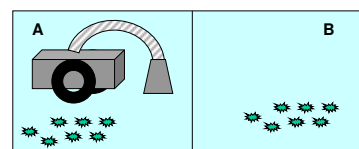


Limitações

- Mesmo para problemas simples → tabelas muito grandes
– ex. xadrez 30*100
- Nem sempre é possível, por ignorância ou questão de tempo, construir a tabela
- Não há autonomia nem flexibilidade

Ambientes

- observável, determinístico, episódico, estático, discreto (e minúsculo!) 21



PERCEPÇÃO [sala, estado]	AÇÃO
[A, limpo]	Ir para a direita
[A, sujo]	Aspirar
[B, limpo]	Ir para a esquerda
[B, sujo]	Aspirar

22

function Agente-Tabela-Aspirador (*percept*) **return** uma ação

static: *percepts* – uma sequência, inicialmente vazia
table – uma tabela de ações indexada pela sequência perceptiva, inicialmente totalmente especificada

append *percept* ao final de *percepts*
ação ← **Lookup** (*percepts*, *table*)
return ação

Uso **MUITO** limitado (**impossível**): seja P o conjunto de percepções possíveis e T, o tempo de vida do agente →
entradas da tabela = $\sum_{t=1..T} |P|^t$

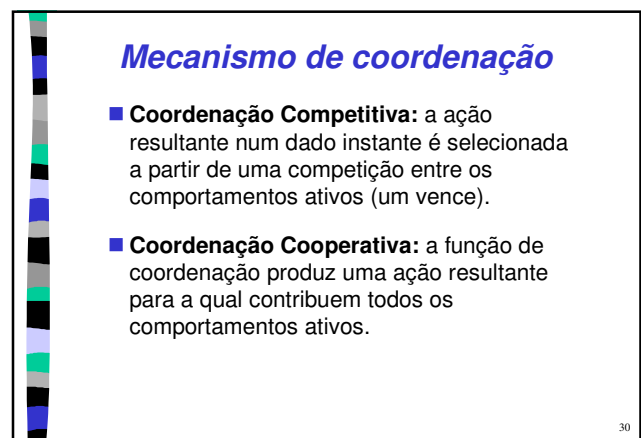
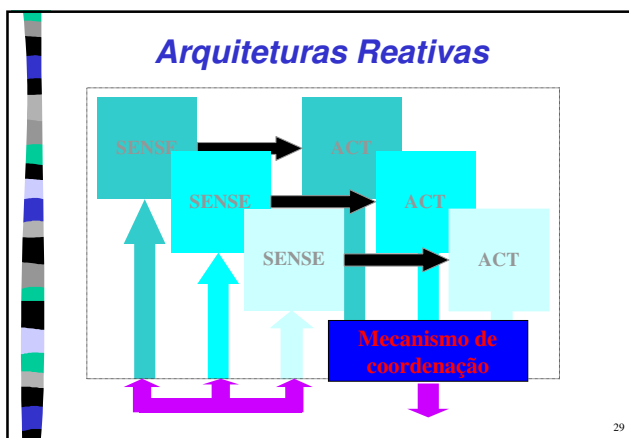
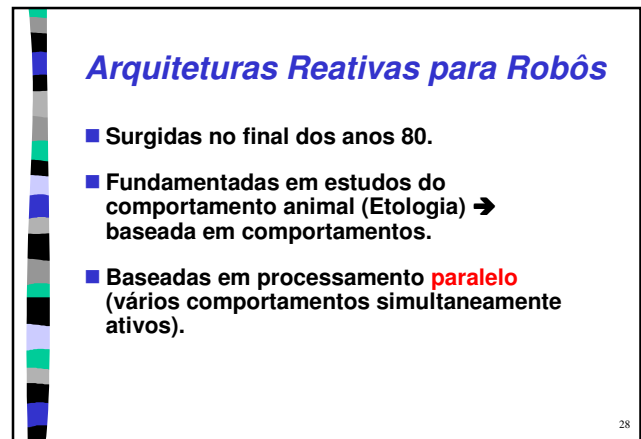
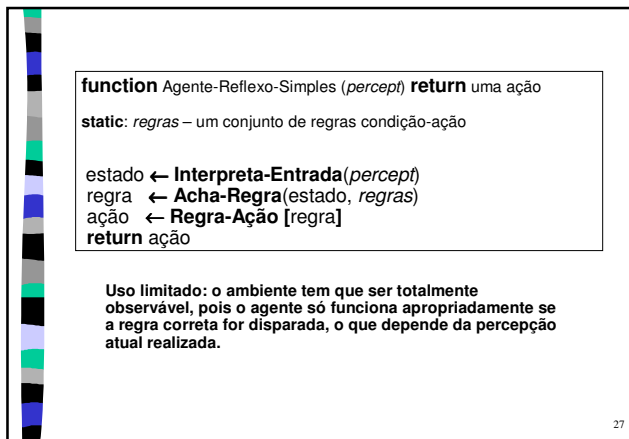
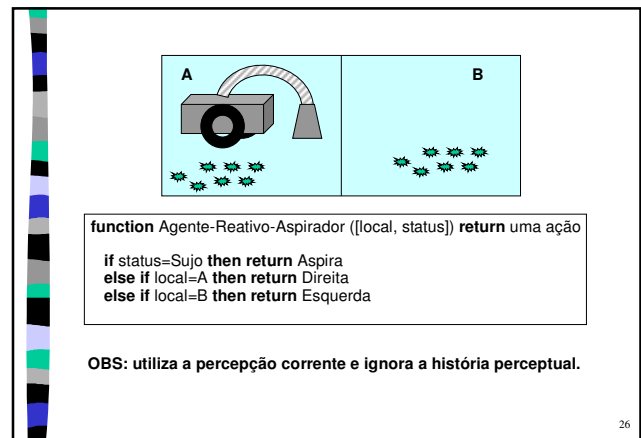
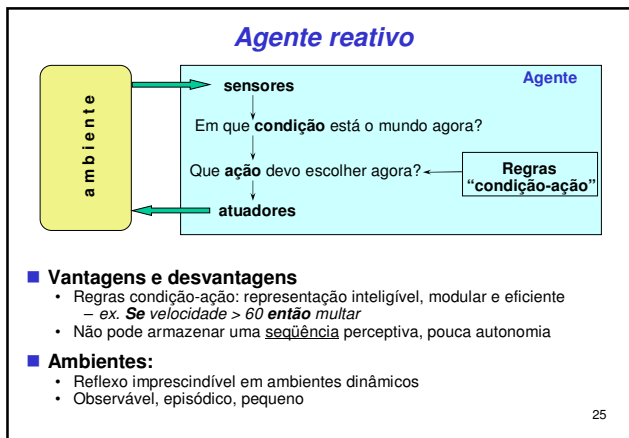
→ Como IA soluciona isso?

23

Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- Agente baseado em objetivos
- Agente baseado em utilidade
- Agente aprendiz

24



Estudo de Caso: REACT Comportamentos Reativos para Robôs Móveis

<http://www.lti.pcs.usp.br/robotics/react>



31

REACT

■ Arquitetura baseada em *Motor Schemas*

■ *Motor Schema* ≡ comportamento

- Comportamentos são divididos em:
 - módulo de percepção
 - módulo de codificação

■ Saída do *Motor Schema* → Vetor

- representa a ação a ser executada
 - Magnitude = velocidade, direção = rotação

32

REACT

■ Módulo de codificação:

- Mapeamento contínuo: percepções → ações
- Usa o Método de Campos Potenciais

■ Coordenação dos comportamentos:

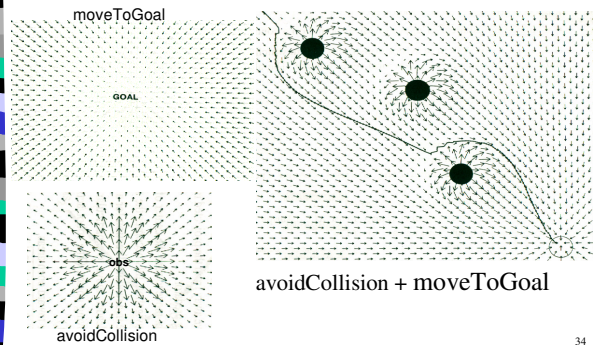
- Abordagem cooperativa → Superposição dos campos de força

■ Alguns comportamentos na REACT:

- avoidCollision
- moveToGoal

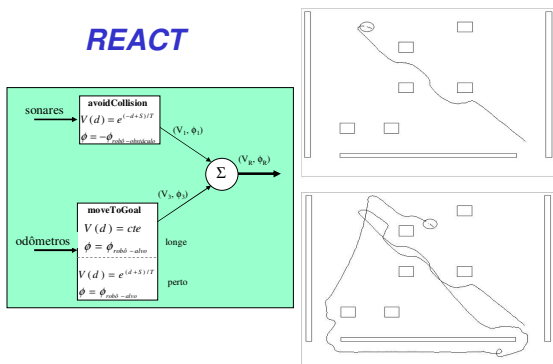
33

REACT



34

REACT



35

Arquiteturas Reativas: Vantagens

- Comportamentos reativos são normalmente simples de projetar.
- Paralelismo, simplicidade de comportamentos individuais e ligação direta *SENSE-ACT* permitem operação em **tempo real**.
- Processamento **local** da informação sensorial.
- Prototipação rápida para poucos comportamentos.

36

Arquiteturas Reativas: Problemas

- Implementação de um grande conjunto de comportamentos é uma tarefa difícil.
- Combinação de comportamentos reativos não garantem sucesso na execução da tarefa (defensores da abordagem falam em *Inteligência emergente*). Ex: pontos de campo nulo.
- Dificil definição de um conjunto mínimo de comportamentos reativos no caso geral.

37

Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- Agente baseado em objetivos
- Agente baseado em utilidade
- Agente aprendiz

38

Necessidade de um modelo (1)

- Um agente puramente reativo que tenha observação parcial do ambiente pode cair em *deadlocks* (ex: robô parado num local de campo potencial nulo) ou *loops* infinitos.
- A forma mais efetiva de lidar com observabilidade parcial é “lembrar e imaginar a parte do mundo que não está observável no momento”, i.e, manter um *estado interno* que dependa da história perceptual passada e reflita (ao menos em parte) aspectos não observados no estado atual.

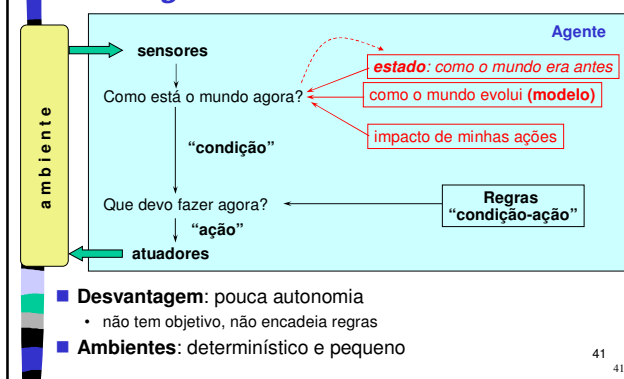
39

Necessidade de um modelo (2)

- Neste novo agente, para determinar como o mundo está num determinado momento, ele usa:
 - informações perceptuais atuais (como o agente reativo)
 - seu estado interno
 - informações a respeito de como o mundo evolui, independentemente de suas ações (*modelo do mundo*)
 - informações a respeito do impacto/efeito de suas próprias ações no mundo
- e, com isso, atualiza seu estado interno.

40

Agente baseado em modelo



41

```
function Agente-Com-Estado (percept) return uma ação
```

```
static: estado – uma descrição do estado corrente do mundo
       regras – um conjunto de regras condição-ação
       ação – a ação mais recente, inicialmente nula
```

```
estado ← Atualiza-Estado(estado, ação, percept)
```

```
regra ← Acha-Regra(estado, regras)
```

```
ação ← Regra-Ação [regra]
```

```
return ação
```

42

Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- **Agente baseado em objetivos**
- Agente baseado em utilidade
- Agente aprendiz

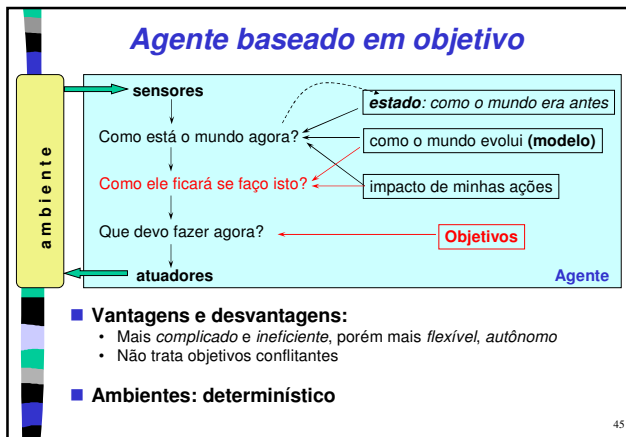
43

Necessidade de metas/objetivos

- Além do estado interno, um agente precisa de alguma informação a respeito de **metas**, indicando situações desejáveis, para decidir a melhor ação a executar.
- Assim, pode **combinar** as informações do impacto de suas ações com seus objetivos, de modo a **fazer considerações acerca do futuro** (predições) e decidir melhor suas ações.
 - O agente poderá ter que considerar longas seqüências de ações encadeadas para poder atingir sua meta → **busca** e **planejamento** são subáreas de IA que visam determinar a **seqüência de ações** que leva o agente ao objetivo.

44

Agente baseado em objetivo



45

Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- Agente baseado em objetivos
- **Agente baseado em utilidade**
- Agente aprendiz

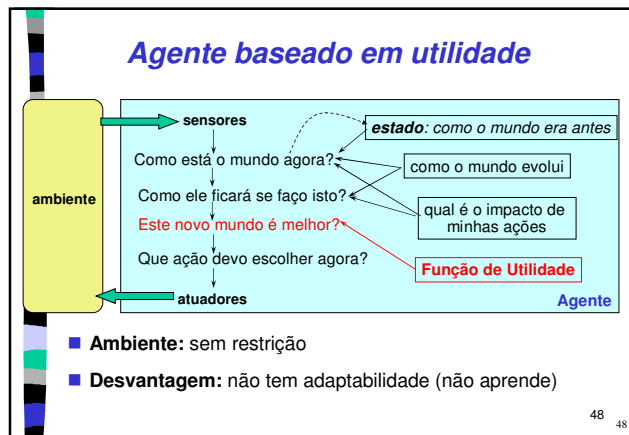
46

Medida de desempenho mais geral: função de utilidade

- Uma função de utilidade mapeia um estado (ou seqüência de estados) em um número real que descreve o **grau de satisfação** associado ao estado.
- Permite decisões racionais quando há:
 - **Objetivos conflitantes** (ex: velocidade e segurança) → neste caso a função de utilidade define um compromisso adequado entre eles.
 - **Múltiplos objetivos**, nenhum dos quais se tem certeza de ser atingido → a função de utilidade permite um balanceamento entre a possibilidade de sucesso com a importância de cada objetivo.

47

Agente baseado em utilidade



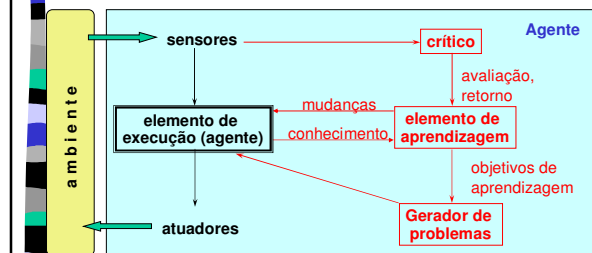
48

Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- Agente baseado em objetivos
- Agente baseado em utilidade
- **Agente aprendiz**

49

Agente que aprende

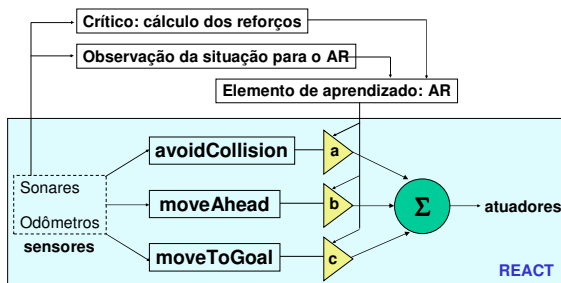


- **Ambiente:** sem restrição
- **Vantagem:** tem adaptabilidade (aprende)

50

50

AAREACT – Aprendizado Autônomo na REACT



51

AAREACT

- **Situações (24):** número finito, dadas por atributos
- **Saída (8):** número finito de ponderações entre as ações dadas pelos comportamentos (a:AC, b:MA, c:MG)

ATRIBUTO	PESOS (a,b,c)
FreeTarget	(0, 0, 1)
BackTarget	(1, 0, 1)
SideObstacle	(0.3, 1, 0)
DiagonalObstacle	(1, 1, 0)
MiddleObstacle	(1, 0.6, 0.4)
NarrowPath	(0.3, 1, 0)
FrontalObstacle	(1, 0.3, 0)
-----	(0.7, 0.5, 1)

Reforço positivo recebido quando:

- 1) Atingir o alvo (valor alto)
- 2) Valor proporcional à composição entre a velocidade média desenvolvida pelo robô e a velocidade média de aproximação do alvo

AR: tentativa e erro, aprendendo a melhor saída para cada situação, dada pela maximização da soma (descontada) dos reforços recebidos.

52

Evolução da noção de agente além das fronteiras da IA....

- **Agentes em IA**
 - Metodologia (metáfora) para projeto de sistemas
 - Sistemas multiagentes e robótica
- **Agentes em Computação**
 - **Adoção de uma nova metáfora** (antropomórfica e sociológica). Extrapolação de OOP
 - Integração de técnicas de IA
 - Novas tecnologias próprias à Web (ex. mobilidade)
 - Marketing (moda)
- **Agentes: técnica ou metodologia ?**

53

Desenvolver agentes inteligentes

- **Projeto:**
 - Modelar tarefa em termos de ambiente, percepções, ações, objetivos e utilidade
 - Identificar o tipo de ambiente
 - Identificar a arquitetura de agente adequada ao ambiente e tarefa
- **Implementação**
 - O gerador e o simulador de ambientes
 - Componentes do agente (vários tipos de conhecimento)
 - Testar o desempenho com diferentes instâncias do ambiente

54