

Photo © Sam Ogden

# BEHAVIOR-BASED ROBOTICS

Ronald Craig Arkin

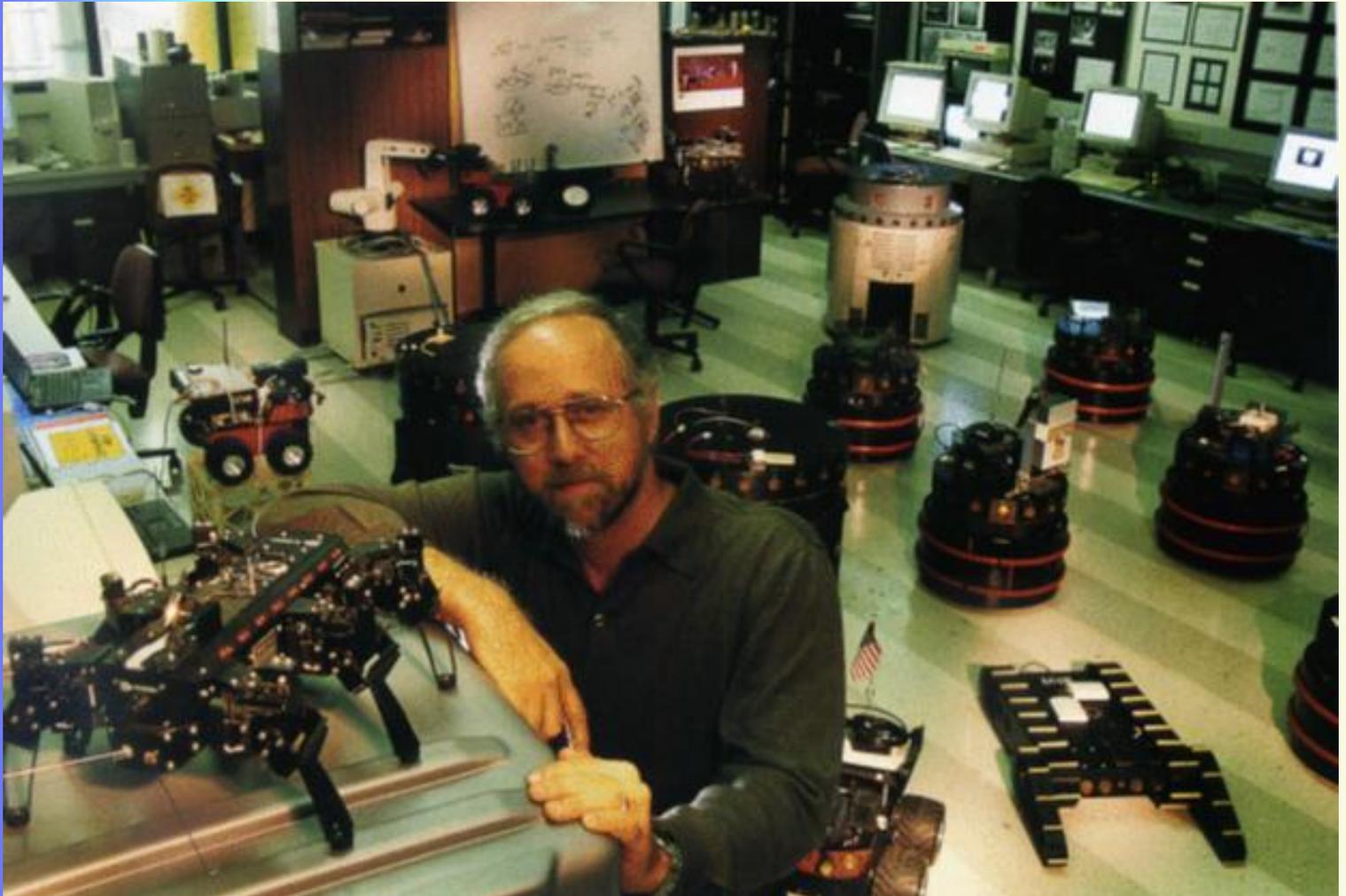
CAP. Nº 3

Juarez, Maurício e Saulo

Continua

# RONALD CRAIG ARKIN

Professor  
College of Computing  
Georgia Institute of Technology



[Voltar.....](#)

# Índice

## 3.1 O que é robótica baseada em comportamentos ?

3.1.1 - Sistemas Reativos

3.1.2 - Um exemplo de Navegação

3.1.3 - Base para a Robótica Baseada em Comportamento

## 3.2 - Representação do Comportamento

3.2.1 - Diagramas de Estímulo-Resposta

3.2.2 - Notação Funcional

3.2.3 - Diagramas de Estado Finito

3.2.4 - Métodos Formais

3.2.4.1 - RS

3.2.4.2 - Situated Automata

## 3.3 - Codificação dos Comportamentos

3.3.1 - Codificação Discreta

3.3.2 - Codificação Contínua

## 3.4 - Construindo Comportamentos (Estratégias)

3.4.1 - Comportamento Emergente

3.4.2 - Notação

3.4.3 - Coordenação do Comportamento

3.4.3.1 - Métodos Competitivos

3.4.3.2 - Métodos Cooperativos

3.4.4 - Montando os Comportamentos

# O que é comportamento em robótica?

Talvez o modo mais simples de representar comportamentos em robôs, seja adotar o conceito defendido pela escola behaviorista da Psicologia, onde, um comportamento, simplesmente, é uma reação a um estímulo. Esta visão pragmática nos permite expressar como um robô deveria interagir com o seu ambiente. Pensando assim, nós estaremos neste capítulo limitando-nos ao estudo de sistemas puramente reativos.

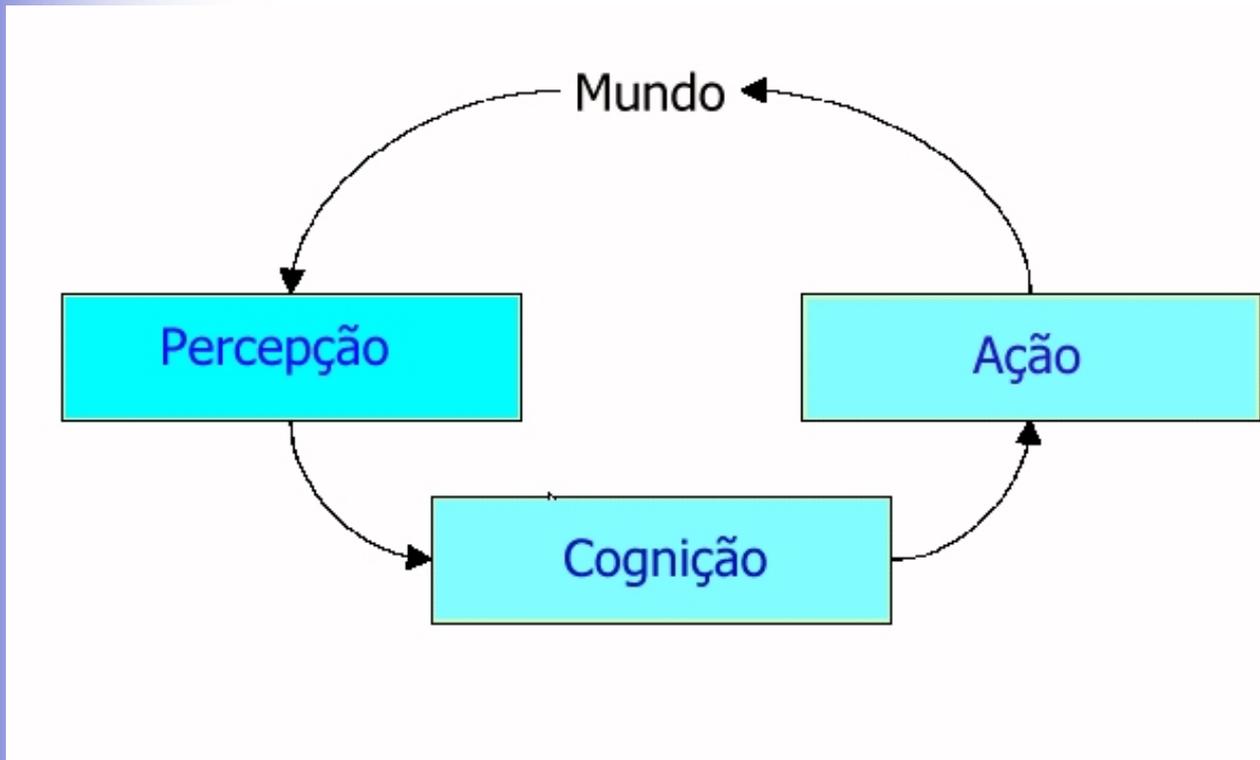
- **Segundo a IA tradicional**
- **Segundo Rodney Brooks**

[voltar.....](#)

[avançar.....](#)



# Segundo a IA tradicional



[Voltar.....](#)

[Exemplos](#)

## Shakey (Stanford Research Institute, 1970)

- Ambiente da oficina, simplificado para ser reconhecido por visão;
- Visão para obter a estrutura 3D do mundo;
- Geração de planos a partir da estrutura 3D, refinamento dos planos e geração de ações atômicas

## Copy-demo (MIT, 1972)

- Mundo de blocos de cores, simplificado;
- Camera e braço robótico;
- Representação 3D.

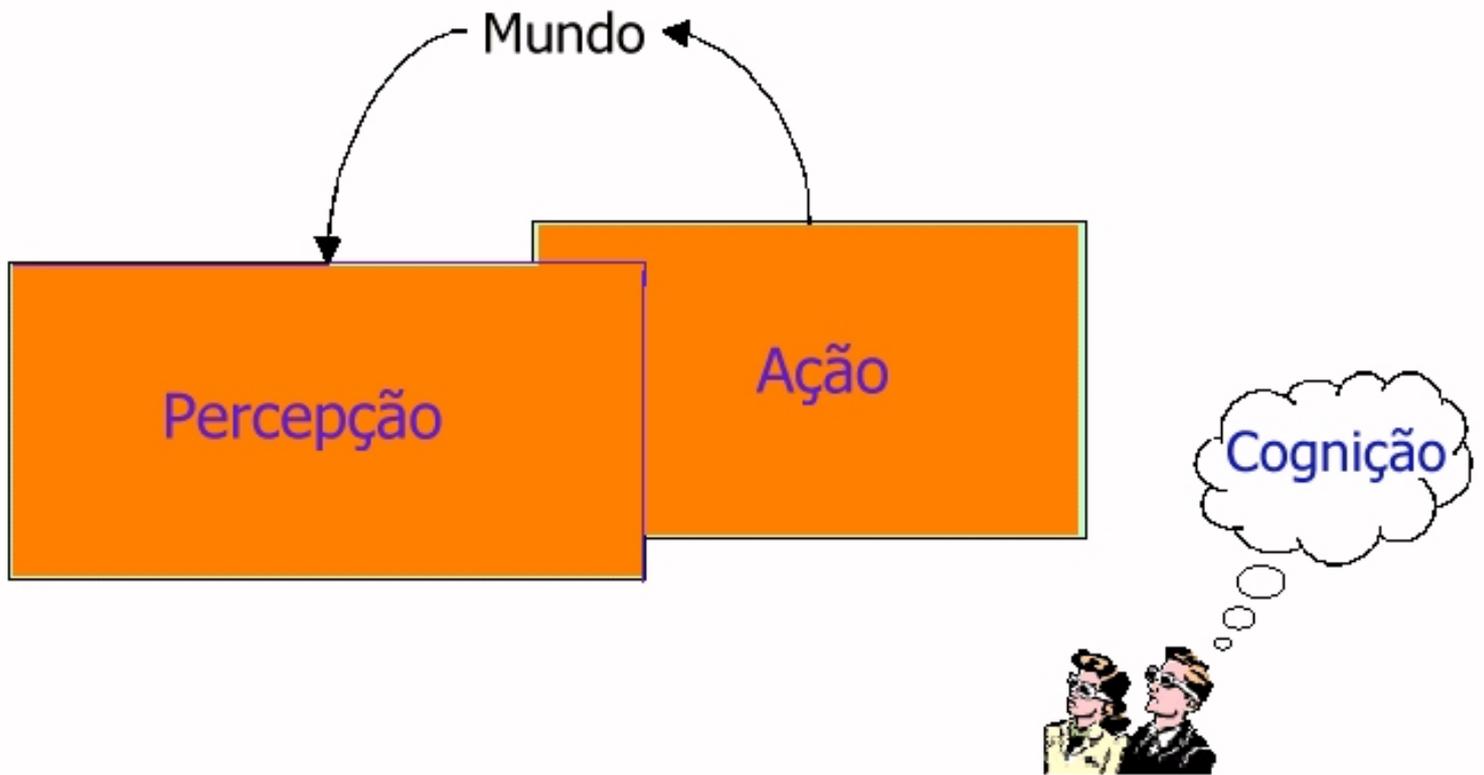
## Na IA Tradicional

- A percepção produz uma representação de propósito geral.
- A cognição se realiza principalmente mediante métodos simbólicos
- A ação é o produto final da cognição.

voltar.....



# Segundo Rodney Brooks



[voltar.....](#)

[Histórico.....](#)



# Os pioneiros (e suas idéias)

## **Agre e Chapman (1987)**

Atividade situada (situated activity): as atividades cotidianas não são o resultado de uma planificação, e sim atividades rotineiras.

## **Rosenschein e Kaelbling (1986)**

É possível definir crenças e objetivos, e compilá-los em programas eficientes baseados em circuitos.

## **Brooks (1986)**

É importante construir agentes com sensores reais que operem no mundo real. Muitas das ações de um agente são separáveis e um comportamento coerente pode emergir da interação independente dos componentes com o mundo real

[Voltar.....](#)

# Rodney A. Brooks



**RODNEY A. BROOKS** is Director of the MIT Artificial Intelligence Laboratory, and Fujitsu Professor of Computer Science.

[Voltar.....](#)

# Sistemas Reativos

Um sistema reativo não contém nenhuma forma central de modelo do ambiente, e não utiliza um raciocínio simbólico complexo. Ele é baseado no princípio da reatividade dos agentes, ou seja, na suposição de que comportamentos inteligentes podem ser gerados sem nenhuma representação simbólica explícita e a suposição de que a inteligência é uma propriedade emergente de certos sistemas complexos. Agentes reativos são capazes de reagir ao ambiente sem utilizar um raciocínio complexo.

Um sistema reativo opera em um nível muito baixo de abstração sem nenhum conhecimento prévio de seu ambiente. Esta característica é em si uma vantagem, bem como uma desvantagem. O tempo de resposta do agente é muito baixo. Isto conduz para uma comunicação eficiente entre dois agentes ou entre o agente e o ambiente.

A desvantagem é que o agente móvel não pode executar análises complexas de seus dados sensoriais. Isto também significa que ele não pode executar operações cognitivas de alto nível envolvendo crenças, desejos e intenções, a não ser que ele tenha acumulado conhecimento suficiente para executar tais funções.

Um sistema reativo em robótica tem as seguintes características:

- A base de comportamento do robô se encontra nas condutas elementares;
- Evita-se o uso de conhecimento simbólico que represente o estado do mundo;
- Os modelos animais de comportamento servem como uma base para estes sistemas;
- Modularidade desde o ponto de vista de projeto de software;
- Os comportamentos complexos são obtidos a partir da coordenação de comportamentos elementares.

[Voltar.....](#)

[Avançar.....](#)



# Navegação Baseada em Comportamento

A navegação baseada em comportamentos foi introduzida por Brooks (1985) e divide a tarefa de navegação em comportamentos básicos. Esta técnica gera caminhos que podem ser difíceis de prever, mas apresentam uma oportunidade de navegação reativa independente de um caminho geométrico baseado em modelos do ambiente, Arkin define este comportamento como “senso - motor”. O conceito de senso - motor originou-se na psicologia e na neurologia para descrever a iteração entre percepção e ação dos seres vivos.

[Voltar.....](#)

[Exemplo.....](#)



# Um exemplo: navegação

**Consideremos que um estudante que vai de uma sala de aula a outra. Deve:**

- **Chegar ao destino a partir da localização atual;**
- **Não se chocar com os obstáculos do ambiente;**
- **Negociar o caminho com outros estudantes que ali se movem, em sua direção ou na contrária;**
- **Tem reação frente às mudanças não esperadas.**

**Os sistemas robóticos baseados em comportamento provêm os meios para um robô navegar em mundo desconhecido, imprevisível e sem planejamento, dotando o robô de comportamentos que negociam metas específicas e coordenadas de um modo planejado.**

**voltar.....**

**avançar.....**



# Base para comportamento de robótica

De onde vêm o comportamento em robótica? Esta pergunta primária conduz a uma série de perguntas subsidiárias que devem ser respondidas para proporcionar a um robô o controle de comportamento:

- O que são os padrões de comportamento para sistemas de robótica?
- O que realmente comportamento primitivo é?
- Como estes comportamentos são coordenados efetivamente?
- Como estes comportamentos são fundamentados a sensor e atuadores?

Os agentes reativos acoplam a percepção à ação sem o uso de representações abstratas ou de seu histórico no tempo ( ex. base de conhecimento). Os modelos e estudos que surgiram a partir da etologia ( estudo do comportamento animal em condições naturais ) são de grande utilidade se o sistema físico à ser modelado tiver comportamentos que dependem diretamente de seus similares na natureza.

Em geral, um projeto poderá tomar diferentes formas, dependendo da base e dos princípios que forem utilizados em sua concepção. Embora não exista uma padronização, existem alguns métodos comuns utilizados atualmente para especificar e projetar comportamentos em agentes físicos, tais como os seguintes:

- Modelos baseados Etologicamente;
- Modelos baseados em Atividade Situada, e;
- Modelos baseados em Experimentos.

Concluindo.

[voltar.....](#)

[avançar.....](#)

# Modelos Baseados Etologicamente

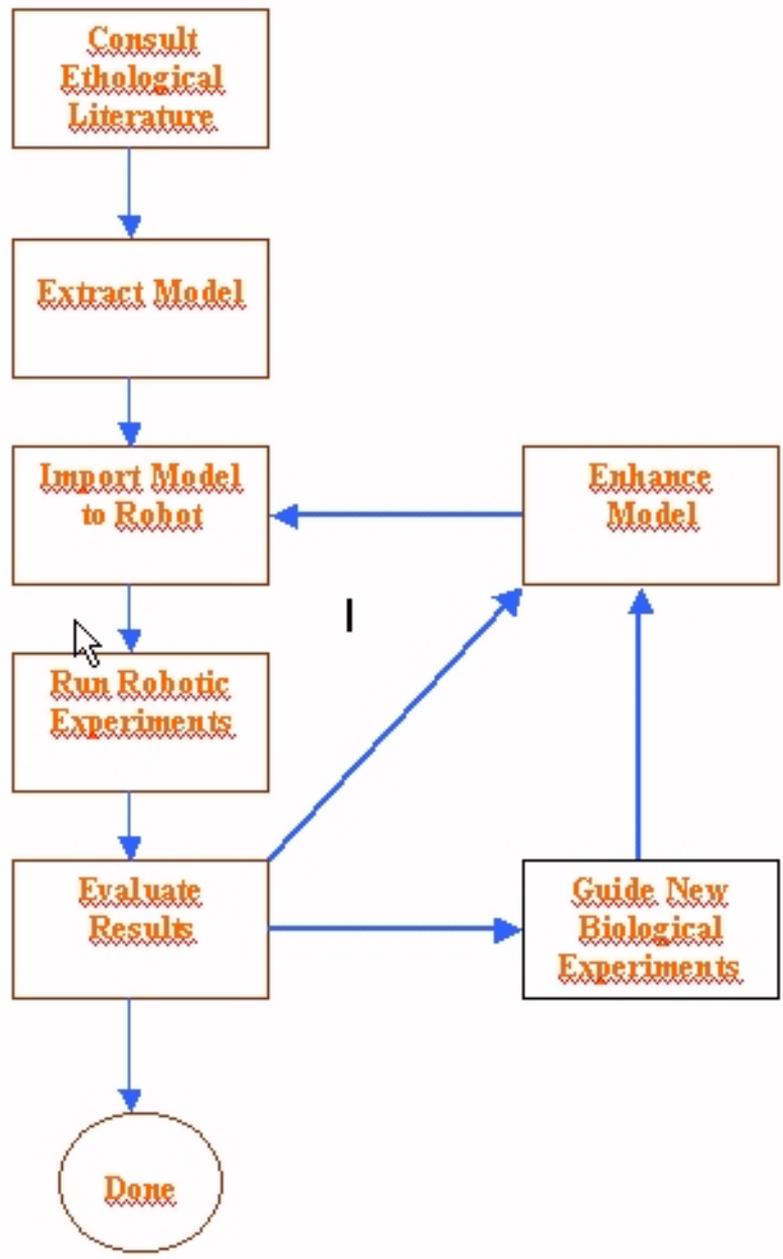
Neste método, se faz uso dos estudos baseados em comportamento animal para proporcionar os aspectos fundamentais nas formas em que se pode construir os comportamentos. Um bom exemplo é o trabalho de Arbib e House (1987) onde estudam o comportamento de navegacional do sapo e sua com o sistema de navegacional.

## Esquema para comportamento baseado etologicamente

Voltar.....

Avançar.....





voltar.....

# Modelos Baseados em Atividade Situada

Neste esquema as ações de um robô são estabelecidas sobre as situações nas quais estas se encontram (Ex. relação direta do agente com o ambiente). Portanto, o problema da percepção se reduz a reconhecer que situações existem para o robô e logo eleger as ações que se deve executar.

Estas situações múltiplas podem ser vistas como microcomportamentos, quer dizer, comportamentos específicos são úteis somente em circunstâncias muito limitadas. Para modelar um robô baseado nesta metodologia é requerido um sólido conhecimento da relação entre o robô e o ambiente o qual está inserido.

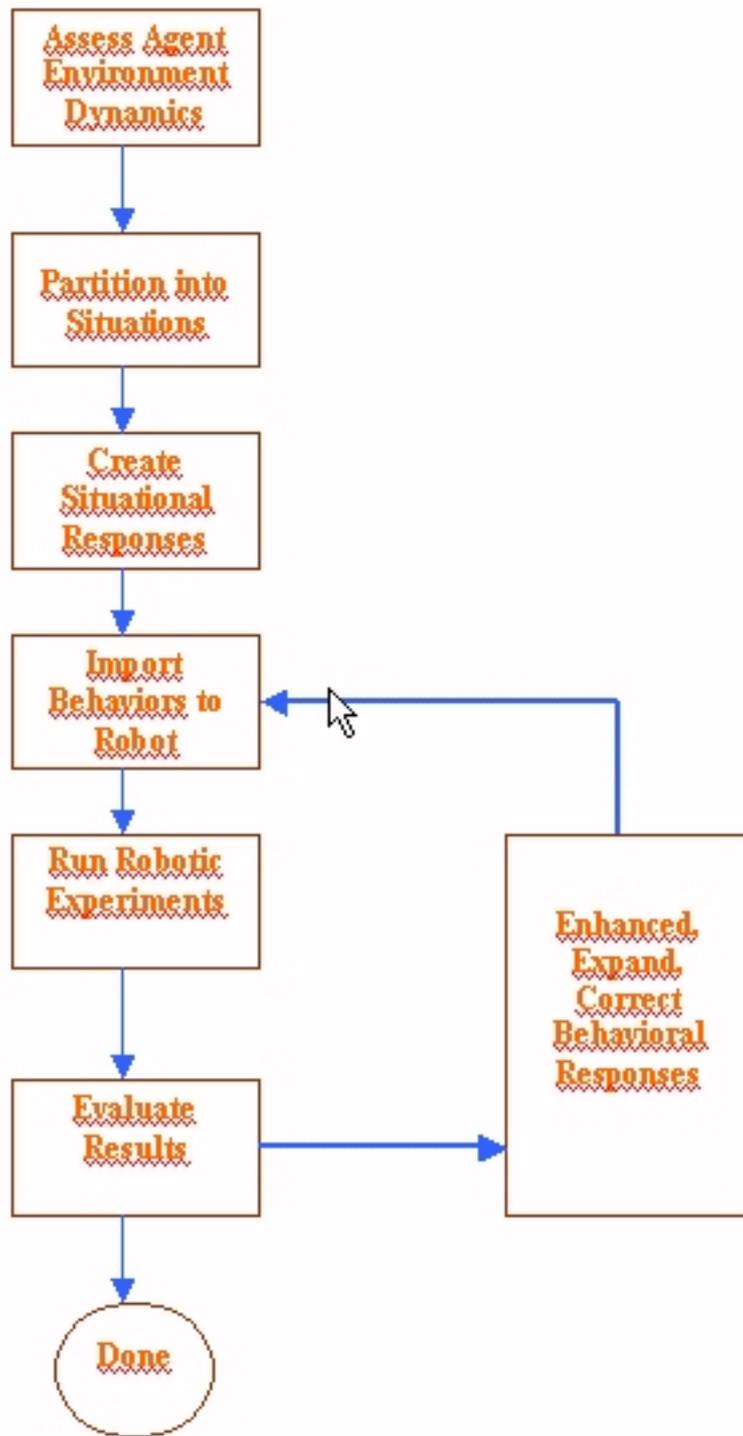
A estratégia de modelagem que caracteriza este método é mostrada a seguir.

Arbitrariamente podem ser criadas situações complexas que podem ser especificadas, mesmo sem uma base biológica. Um exemplo é o Pengi (Agre e Chapman) que é um sistema que caracteriza as situações pelos aspectos indexical/functional delas.

voltar.....

avancar.....





Voltar.....

# Pengi

Usa frases prolongadas para caracterizar situações particulares que exigem certas respostas. Por exemplo (parafrazeando um pouco a situação do Pengi original), "**block-I-need-to-trick-at-the-enemy-is-behind-me**" é uma situação que exige que o agente regresse para obter o objeto em questão.

A coordenação do Pengi é controlada por um mecanismo que arbitra qual ação do candidato será escolhida e aplicada.

**Exemplos: o-bloco-que-necessito-para-golpear-o-meu-inimigo-está-atrás-de-mim**

**estou-no-final-do-corredor-que-me-leva-à-cadeira\_elétrica**

voltar.....



# Modelos Baseados em Experimentação

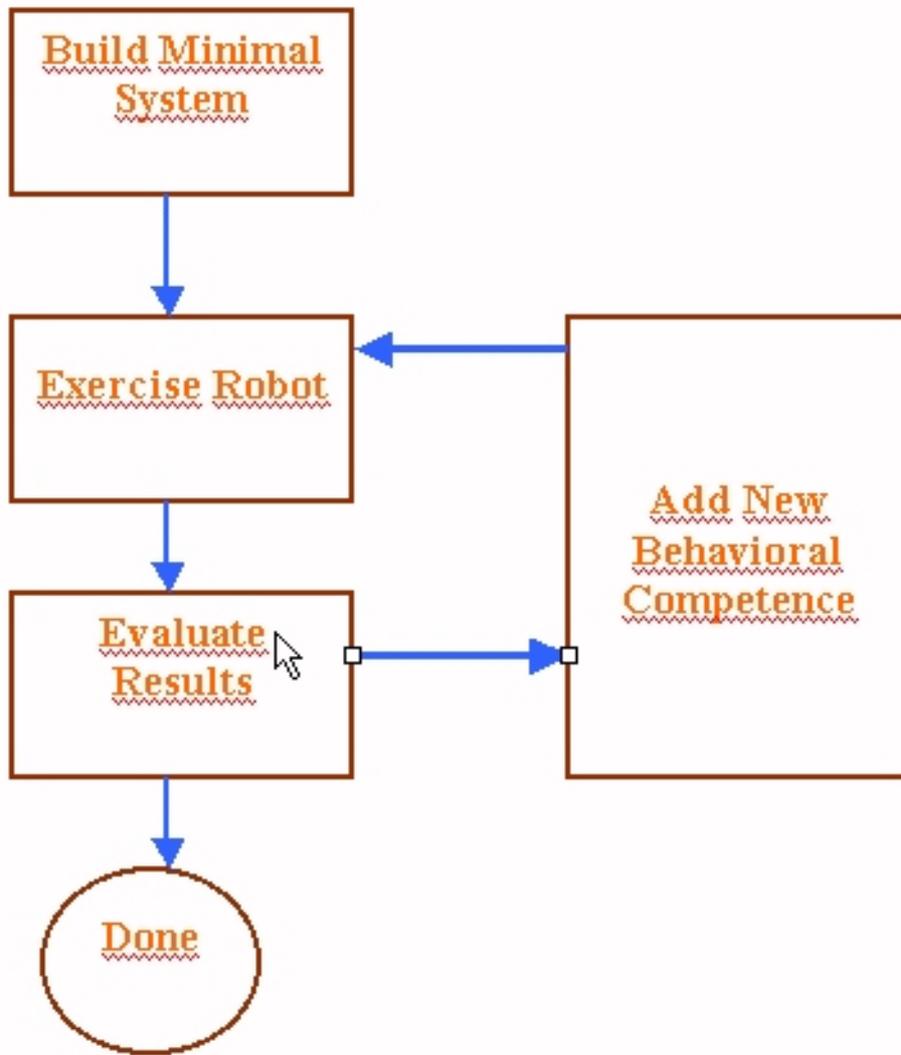
Nesta metodologia, os comportamentos são criados na forma bottom-up, dotando assim o robô de um conjunto limitado de competências que logo provadas no mundo real, permitem ser melhoradas agregando as novas repetidamente até que o sistema mostre um rendimento adequado.

Um excelente exemplo deste paradigma foi o modelo apresentado por Brooks (1989), ao apresentar um robô com a habilidade de levantar-se e efetuar uma caminhada. Inicialmente o robô atingiu seu objetivo plenamente em superfícies lisas porém apresentou problemas em superfícies irregulares.

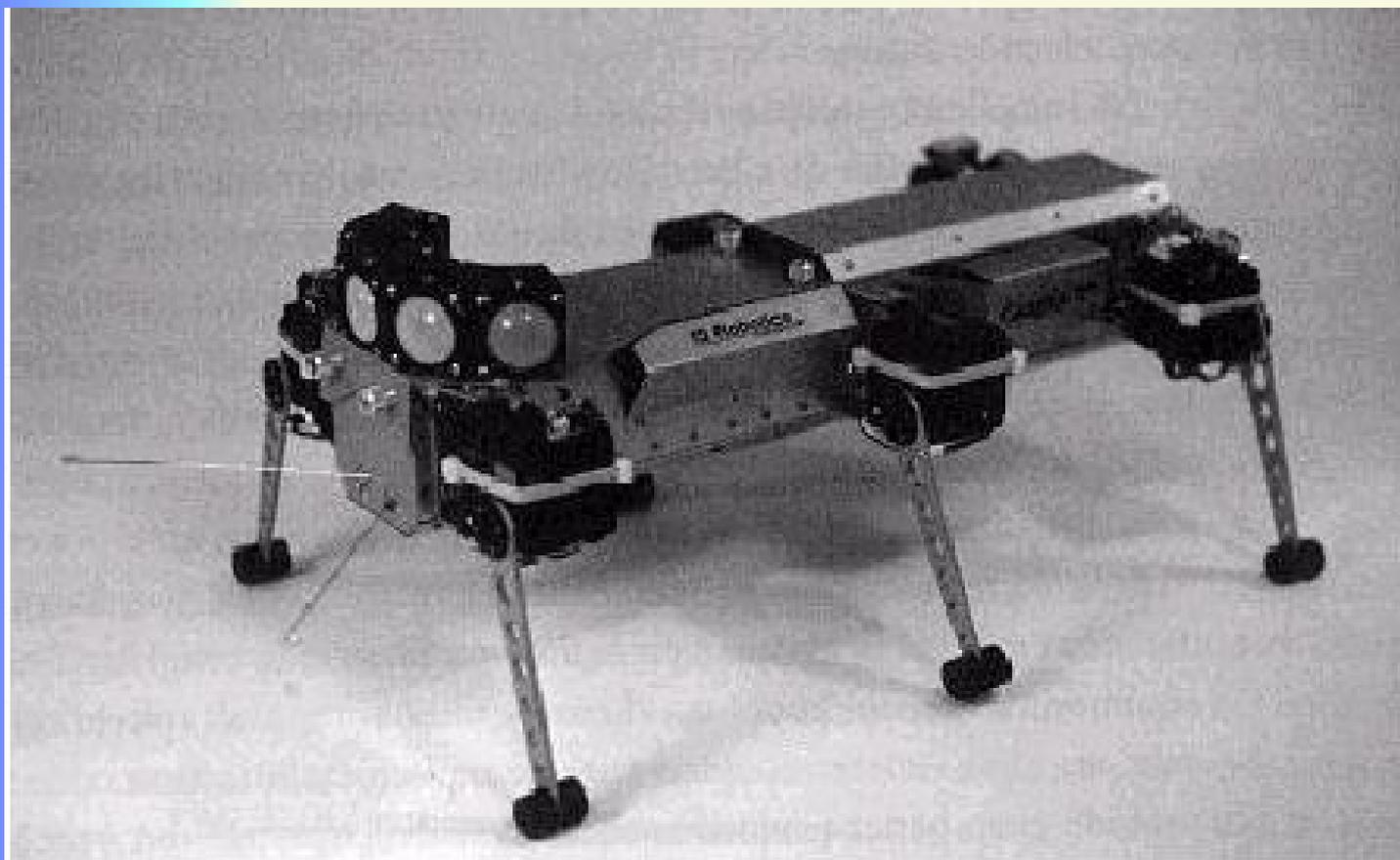
Posteriormente foi aprimorado e foram adicionadas novas características que melhoraram o seu equilíbrio e também outros tipos de sensores ( infra-vermelhos..).

Voltar.....



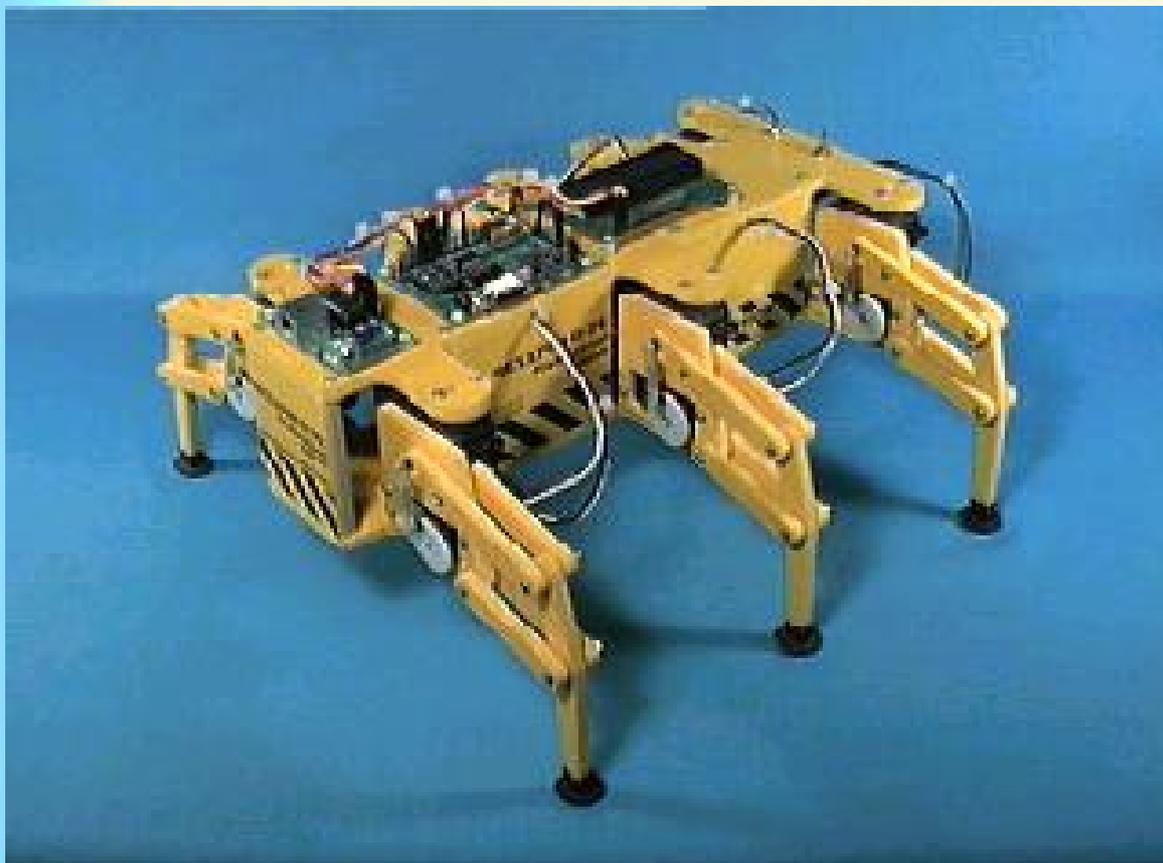


voltar.....



[voltar.....](#)





Voltar.....



**Independente qual seja a base do modelo, existe um certo número de comportamentos típicos que se podem encontrar em agentes robóticos móveis que interagem com o mundo, que incluem os comportamentos relacionados com: exploração, atingir objetivos, proteção, seguir caminhos, posturas, aspectos perceptuais, aspectos sociais e cooperativos, etc.**

**Porém, do ponto de vista do projeto de um sistema baseado em comportamento, se deve considerar vários aspectos importantes: a representação do comportamento, sua codificação, sua coordenação e sua integração com o resto do sistema em uma arquitetura de trabalho.**

# Representação de Comportamentos

Um problema que deve ser considerado no projeto de sistemas baseados em comportamento é a forma como estes podem ser representados logicamente para posteriormente permitir a sua implantação.

Os métodos para expressar o comportamento de um robô reativo devem definir as condições em as que se ativa cada uma dos comportamentos elementares do robô.

Atualmente são explorados quatro métodos fundamentais de representação:

- [Diagramas estímulo-resposta](#)
- [Notação funcional](#)
- [Diagramas de Estados Finitos](#)
- [Métodos Formais](#)

[Voltar .....](#)

[Avançar.....](#)

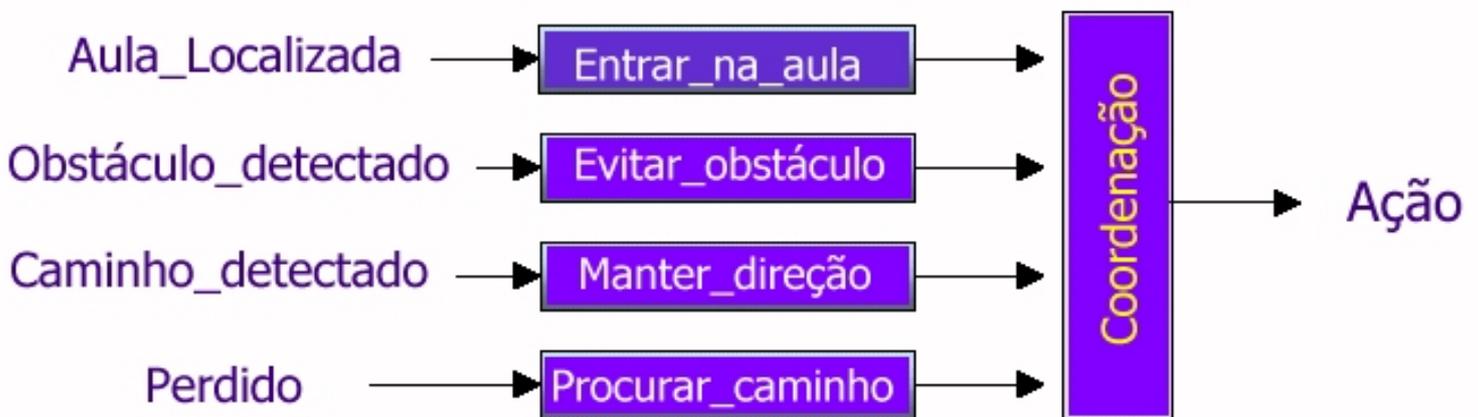


# Diagramas de Estímulo-Resposta

São os mais intuitivos e os menos formais dos métodos de expressão. Qualquer comportamento pode ser representado como uma resposta gerada a um determinado estímulo computado por um comportamento específico. De forma simplificada:



Correspondem a diagramas que representam um comportamento como uma resposta gerada a um estímulo dado. No exemplo abaixo é mostrado um diagrama deste tipo para representar a dinâmica de um agente que deve navegar e encontrar uma sala de aula para entregar objetos.



# Notação funcional

Podem ser utilizados métodos matemáticos para descrever a mesma relação anterior ( diagramas de estados finitos ) utilizando uma notação funcional do tipo:

$$b(s) = r$$

**Comportamento(Estímulo) = Resposta**

Paralelamente, são utilizadas linguagens de programação funcionais como é o caso da linguagem de representação de comportamento "*The Behavior Language*" desenvolvida por Rodney Brooks e sua equipe no MIT no início dos anos 90.

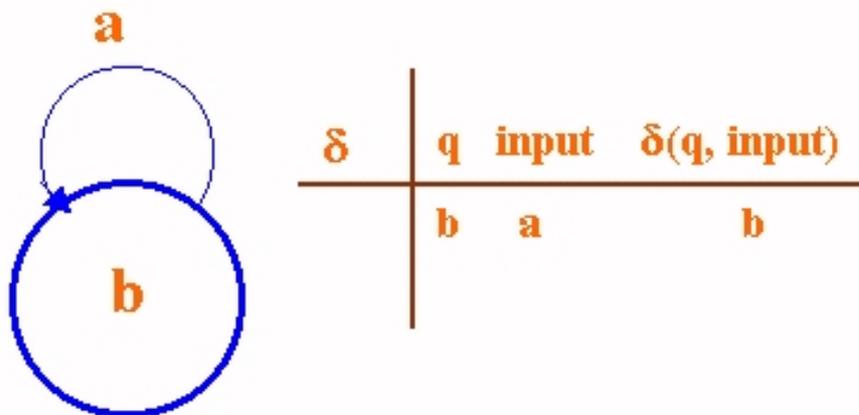
Esta linguagem descreve os comportamentos em termos funcionais e esquemas em linguagem LISP. Embora dependa da arquitetura que se utilize, sua utilização tem ajudado bastante a entender o comportamento total do sistema que se deseja implementar, e por outro lado, permite trabalhar em um nível mais alto se comparado com a especificação necessária em termos de diagramas de estados finitos.

[voltar.....](#)



# Diagramas de Estado Finito

É uma das técnicas tradicionais utilizadas na representação de processos de computação, com a diferença de que os nós representam os estados do comportamento e as transições, os estímulos correspondentes. Uma grande parte de implantação de sistemas de comportamento originalmente empregaram este tipo de método, sem dúvida, dado o conhecimento prévio que se requer para seu uso, alguns pesquisadores tem optado por mecanismos de mais alto nível. Um modelo simples:

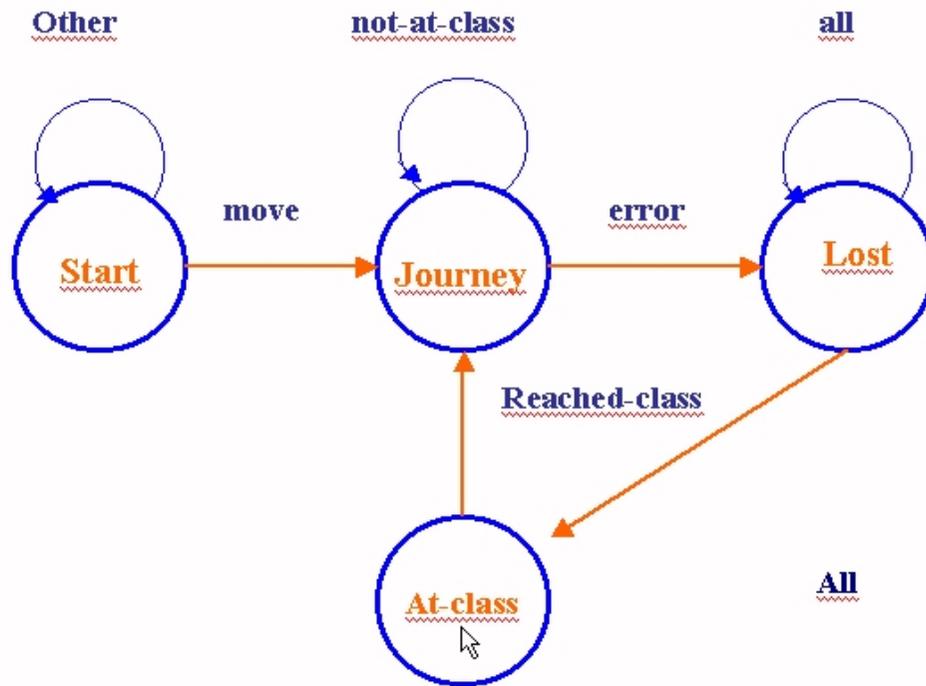


Onde  $q$  representa o conjunto de estados de comportamento permitidos,  $\delta$  é uma função de transição que traça um mapa entre o estado atual e outro, e  $a$  representa todas as entradas.

voltar.....

continuar.....





$\delta$	q	input	$\delta(q.input)$
	início	move	caminho
	início	outro	início
	caminho	erro	perdido
	caminho	<u>fora-da-classe</u>	caminho
	caminho	<u>chegou-na-classe</u>	<u>em-classe</u>
	<u>em-classe</u>	todos	<u>em-classe</u>
	perdido	todos	perdido

# MÉTODOS FORMAIS

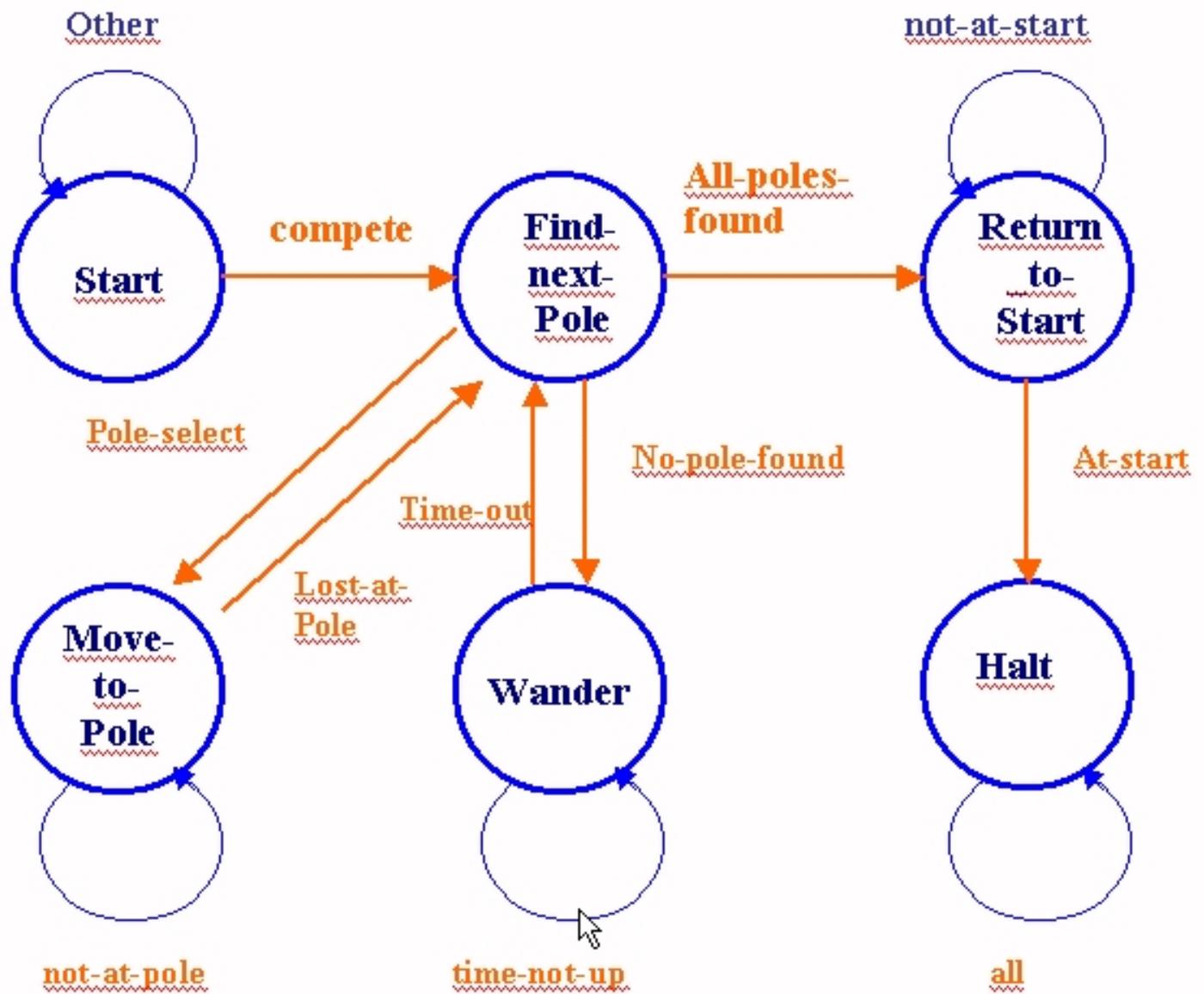
Correspondem a mecanismos formais de especificação de sistemas robóticos baseados em comportamento, e geralmente são utilizados para verificar as intenções do projetista, para facilitar a geração de sistemas de controle robótico, prover um modelo de suporte para o projeto da linguagem de programação de alto nível, etc..

Foram desenvolvidos vários métodos formais para especificar e projetar um sistema robótico baseado em comportamento, porém as duas estratégias mais representativas são: RS ( robot schema) e Situated Automata ( autômato situado).

voltar.....

avançar.....





## FSA para um robô de competição

# RS - Robot Schema

Lyons e Arbib (1989) desenvolveram o modelo RS (robot-schema) como um método para expressar programas de controle de robôs. Um processo algébrico é usado para permitir a formação de uma rede de processos chamada schemas (comportamentos).

A composição dos operadores de processos é definida como uma base para criar estas redes que incluem métodos para condicionamento, sequenciais, paralelos e estruturas interativas

# Autômato Situado

O modelo automata situado, desenvolvido por Kaelbling e Rosenschein (1991), reconhece a relação fundamental do agente como um participante do ambiente interno. O modelo emprega formalismos lógicos como base para o desenvolvimento do diagrama que corresponde às metas e intenções do robô. O uso de lógica habilita o raciocínio em um sistema e pode conduzi-lo ao estabelecimento de propriedades prováveis na obtenção de metas importantes para o projetista.

O REX (Kaelbling-1986), um sistema baseado em LISP, foi o primeiro a incorporar as ferramentas básicas para fornecer as condições necessárias para confecção de programas de controle reativos.

O GAPPs é uma linguagem que foi desenvolvida recentemente, sendo bastante acessível o seu uso.

[Voltar.....](#)

[Continuar.....](#)



## GAPPS (Kaelbling, 1991):

**(ach goal):** ativar o comportamento elementar goal até que seja alcançado o objetivo associado ao comportamento;

**(do goal):** executar o comportamento elementar goal de forma instantânea;

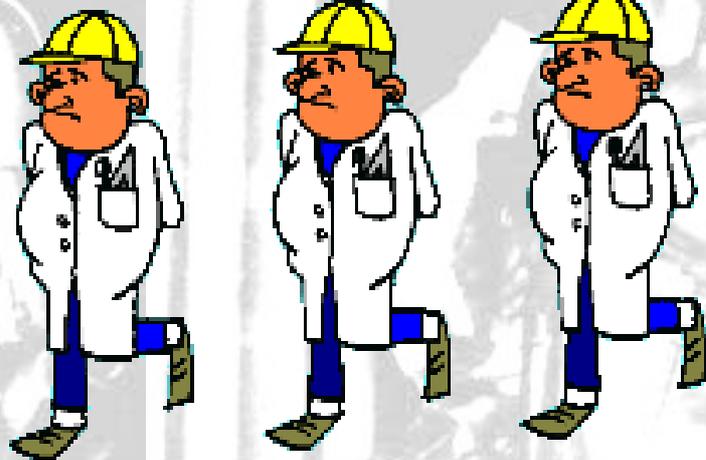
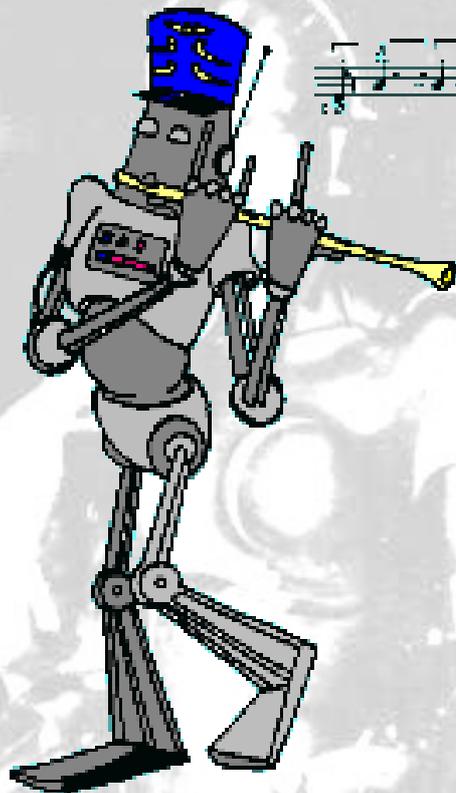
**(maint goal):** manter um determinado comportamento.

### Exemplo:

```
(defgoalr (ach ir-aula)
  (if (not baixa-energia)
    (maint (and (maint entrar-na-aula)
               (maint evitar-obstaculo)
               (maint manter-caminho)
               (maint buscar-caminho))))))
```

[Voltar.....](#)





## 3.3 - CODIFICAÇÃO DOS COMPORTAMENTOS

# Codificação dos Comportamentos

Após a representação do modelo de comportamento do sistema, o passo seguinte é determinar como codificá-las. Para codificar um comportamento deve-se criar um mapa funcional entre o plano de estímulos e o plano de respostas motoras. Com esta finalidade determina-se que a codificação de um determinado comportamento considera os componentes: força e orientação.

Logo podemos representar um comportamento, como o conjunto (  $S$ ,  $R$ ,  $\beta$ ), onde:

- $S$ : domínio de todos estímulos recebidos
- $R$ : faixa de possíveis respostas
- $\beta$ : função de mapeamento dada por  $\beta: S \rightarrow R$

# Codificação dos Comportamentos

Para criar uma resposta de comportamento que um estímulo pode evocar, deve-se criar um mapa funcional entre o plano de estímulos e o plano de respostas motoras.

Respostas de comportamento no espaço físico possuem

Força: magnitude da resposta

Orientação: direção da ação para a resposta

Pode-se representar um comportamento, como o conjunto  $(S, R, \beta)$ , onde:

S: domínio de todos estímulos interpretáveis

R: faixa de possíveis respostas

$\beta$ : função de mapeamento dada por  $\beta: S \rightarrow R$

$\beta$ : As categorias de mapeamento do comportamento são Nulo; Discreto; Contínuo.

[Anterior] [[Próximo](#)]

## ***R - Abrangência das Respostas***

Uma resposta *instantânea*  $r$  (onde  $r \in \hat{R}$ ) de uma reação em um sistema baseado em comportamento pode ser expresso como um vetor de 6 dimensões:

$$r = [x, y, z, \theta, \phi, \psi]$$

onde:

$x, y, z$ : 3 dimensões das coordenadas cartesianas.

$\theta, \phi, \psi$ : 3 graus de liberdade para rotacionais (*roll, pitch, yaw*).

*non-holonomic* - restrição do movimento.

Ex: um robô que anda no chão, onde  $r = [x, y, \theta]$

[\[Anterior\]](#) [\[Próximo\]](#)

## *S - Domínio dos Estímulos*

S consiste do domínio de todos os estímulos perceptíveis.

Cada estímulo individual  $s$  (onde  $s \in S$ ) é representado pela tupla binária  $(p, \lambda)$  onde:

$p$ : classe percentual

$\lambda$ : propriedade da força

$\tau$ : valor limiar

A seleção de todos os  $p$  sobre o domínio  $S$  definem todo o percentual de entradas que um robô pode distinguir.

O  $\tau$  é o percentual da classe  $p$  sobre o que a resposta irá gerar.

O estímulo  $S$  é necessário, mas não é uma condição suficiente para evocar uma resposta.

[\[Anterior\]](#) [\[Próximo\]](#)

## $\beta$ - Mapeamento do Comportamento

Cada comportamento é mapeado entre o domínio dos estímulos e a área da resposta, onde:

$$\beta(s) \rightarrow r$$

$\beta$  pode ser definido arbitrariamente, mas precisa ser definido sobre todos os  $p$  relevantes em  $S$ .

$$\beta:(p,\lambda) \rightarrow \{ \text{para todo } \lambda < r$$

então  $r = [0, 0, 0, 0, 0, 0]$  \(\* sem resposta\*\

senão  $r = \text{função arbitrária}$  \(\* resposta \*\

Associado com  $\beta$ , tem-se o valor de ganho escalar  $g$  (*multiplicador de força*)

$$r' = gr$$

**usado para compor comportamentos múltiplos ou, em caso extremo, desligar um comportamento.**

[\[Anterior\]](#) [\[Próximo\]](#)

## *Codificação Discreta*

Expressa como uma seleção finita de pares de situações-respostas (mapeamento).

O mapeamento muitas vezes usa a forma baseada em regras SE ENTÃO

Exemplos: Gapps (Kaelbling & Rosenschein); Linguagem de Assunção (Brooks); Linguagem de Comportamento (Brooks).

[\[Anterior\]](#) [\[Próximo\]](#)

## *Codificação Contínua*

Respostas contínuas permitem à um robô ter um espaço infinito de reações.

Ao invés de ter uma seleção enumerada de respostas que discretiza o caminho na qual o robô pode se mover, uma função matemática contínua transforma a entrada do sensor em um comportamento de saída.

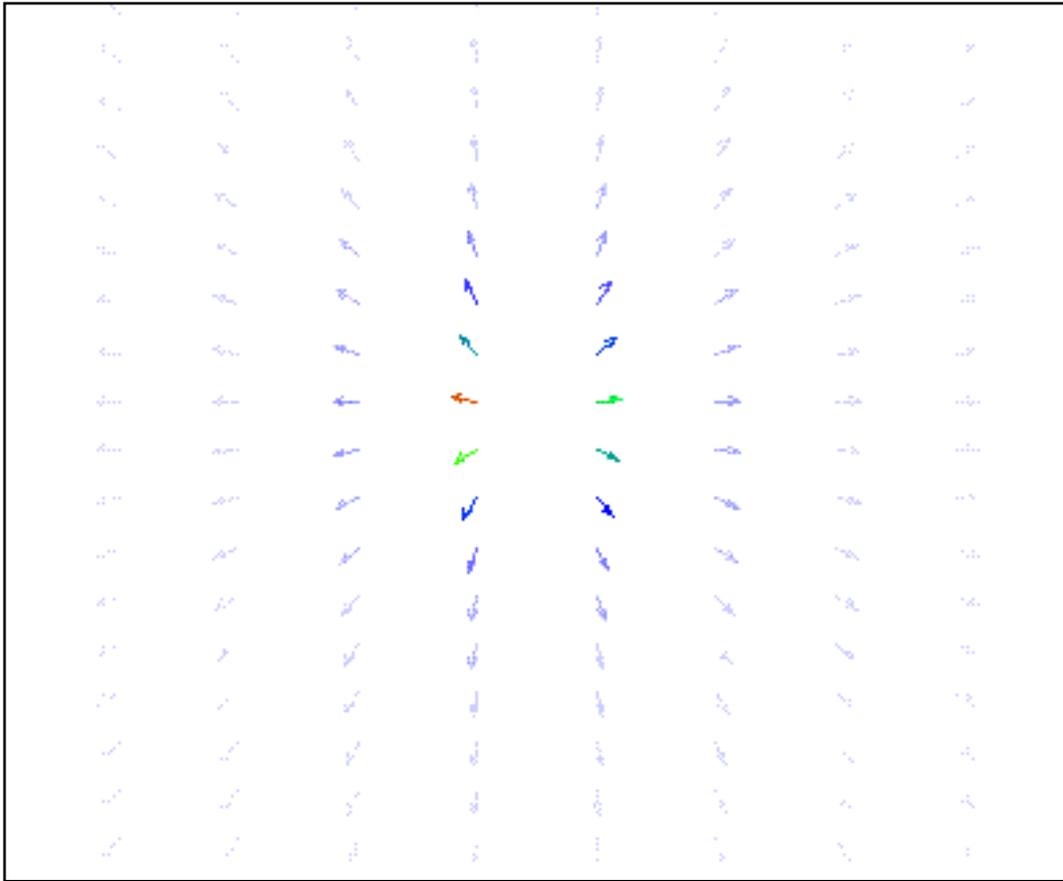
Pode ser simples, com variação de tempo, harmônico, random noise, adaptativo.

Exemplos: [campo potencial](#) (mais comum); esquemas (vector fields); Problemas com a mínima local, máxima, comportamento oscilatório.

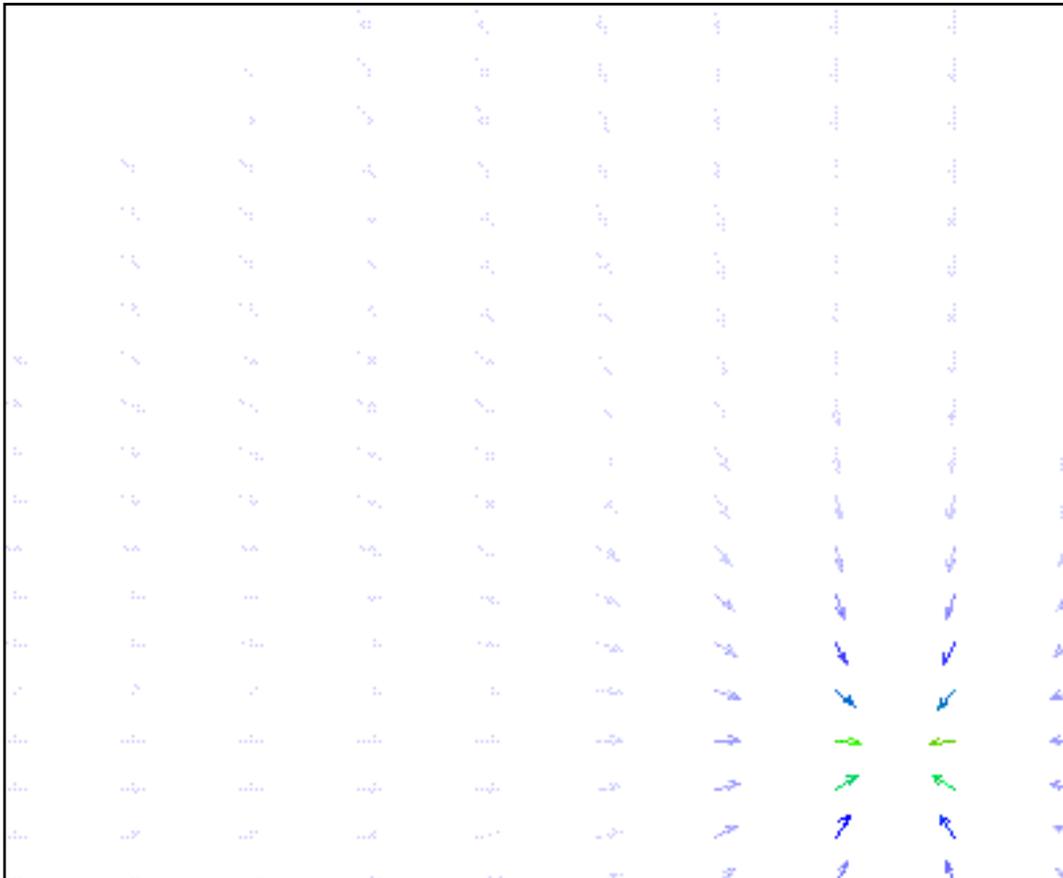
[\[Anterior\]](#) [\[Índice\]](#)

# *Campos Potenciais*

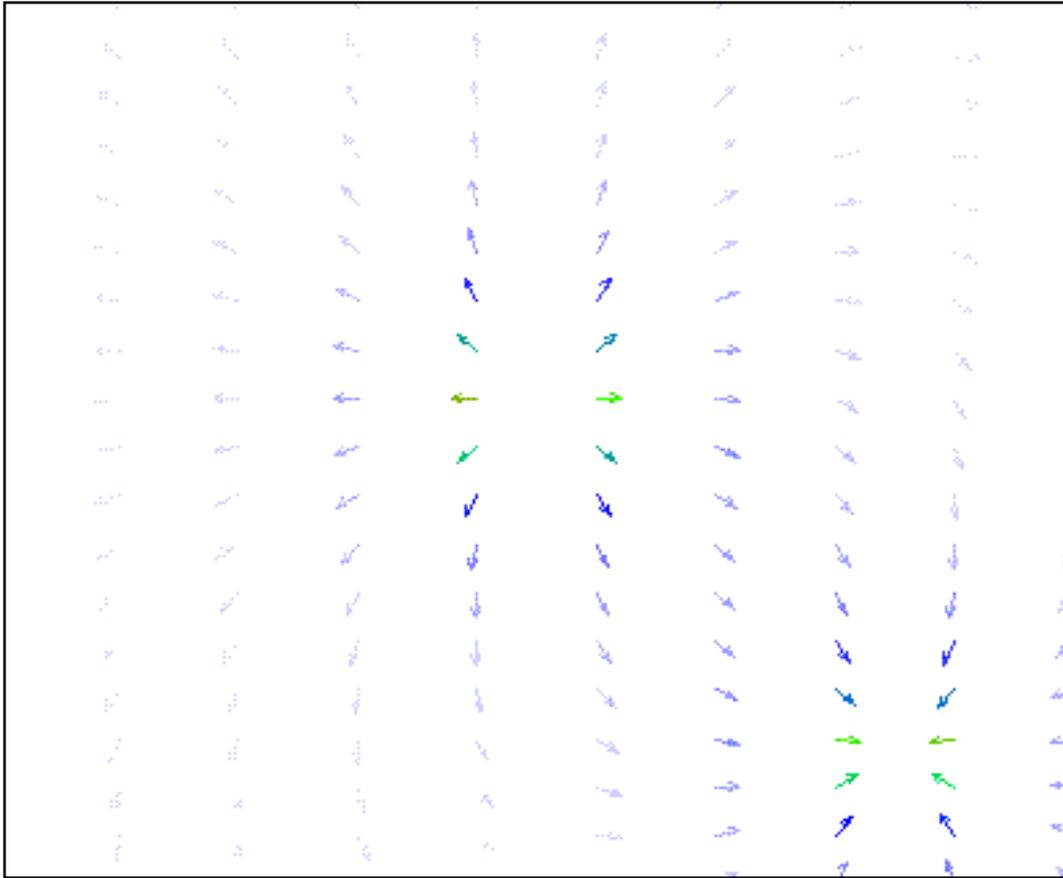
## OBSTÁCULO



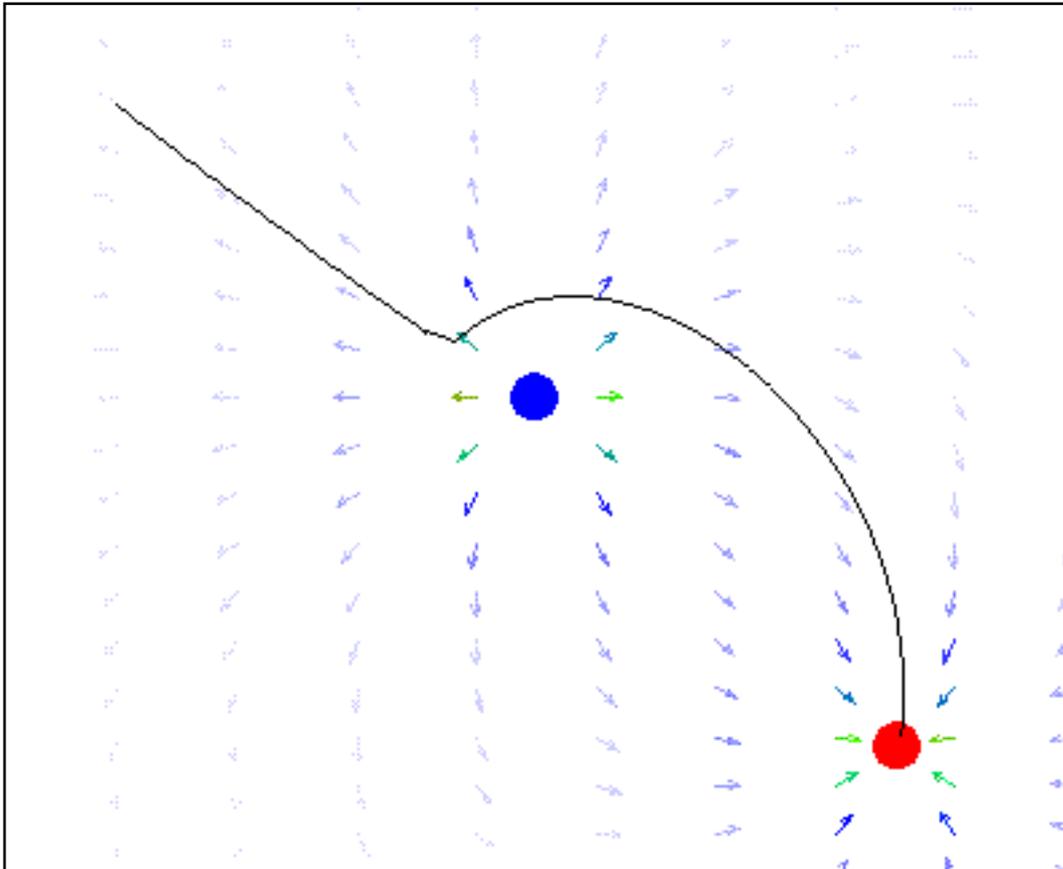
## OBJETIVO



## CAMPO POTENCIAL DO OBSTÁCULO E OBJETIVO



## EXEMPLO DE TRAJETÓRIA



[\[Voltar\]](#)



### 3.4 Construindo/Montando Comportamentos

- estudo de métodos para a construção de sistemas que consistem em múltiplos comportamentos;
- introdução de uma notação formal;
- estudo de diferentes métodos que manipulam atividades comportamentais correntes.

#### 3.4.1 Comportamento Emergente (manifestação)

- é uma propriedade de uma coleção de componentes (comportamentos) que interagem entre si;
- questionamento quanto aos resultados novos ou inesperados; estes, oriundos do conjunto de comportamentos (*behaviors*);
- o resultado, muitas vezes é proveniente de comportamentos que são mais dominantes, i.e., que assumem um grau superior em relação aos demais comportamentos;

#### Aspectos sobre o C.E.

- *Emergence is "the appearance of novel properties in whole systems" (Moravec 1988).*
- *"Intelligence emerges from the interaction of the components of the system" (where the system's functionality, i.e., planning, perception, mobility, etc., results from the behavior-generating components) (Brooks 1991a).*

#### 3.4.2 Notação

Considera-se que múltiplos comportamentos podem atuar de forma concorrente.

- S - vetor de todos os estímulos  $s_i$  para cada comportamento  $B_i$  detectado em um tempo t.
- B - denota um vetor de todos os comportamentos  $B_i$  ativos em um tempo t.
- G - denota um vetor que codifica um ganho  $g_i$  relativo a cada comportamento  $B_i$ .
- R - denota um vetor de todas as respostas  $r_i$ , gerada a partir dos comportamentos.

### Especificação de uma função de controle (C)

$$r = C(G * B(S)) \text{ ou } r = C(G * R)$$

onde,

$$R = \begin{bmatrix} r1 \\ r2 \\ \vdots \\ rn \end{bmatrix} \quad S = \begin{bmatrix} s1 \\ s2 \\ \vdots \\ sn \end{bmatrix} \quad G = \begin{bmatrix} g1 \\ g2 \\ \vdots \\ gn \end{bmatrix} \quad B = \begin{bmatrix} b1 \\ b2 \\ \vdots \\ bn \end{bmatrix}$$

$r$ - vetor que engloba as respostas a serem tomadas pelo robô, representa dos pela forma

$$[x, y, z, \theta, \dot{\theta}, \omega]$$

# Estratégias para a implementação de Comportamentos

- Competitivas

Há um método competitivo simples e puramente arbitrário, onde a saída ( $r_i$ ) referente a um comportamento, é escolhida de  $\mathbf{R}$  e atribuído a um  $P$ . Neste caso uma resposta de  $\mathbf{n}$  disponíveis.

- Cooperativas

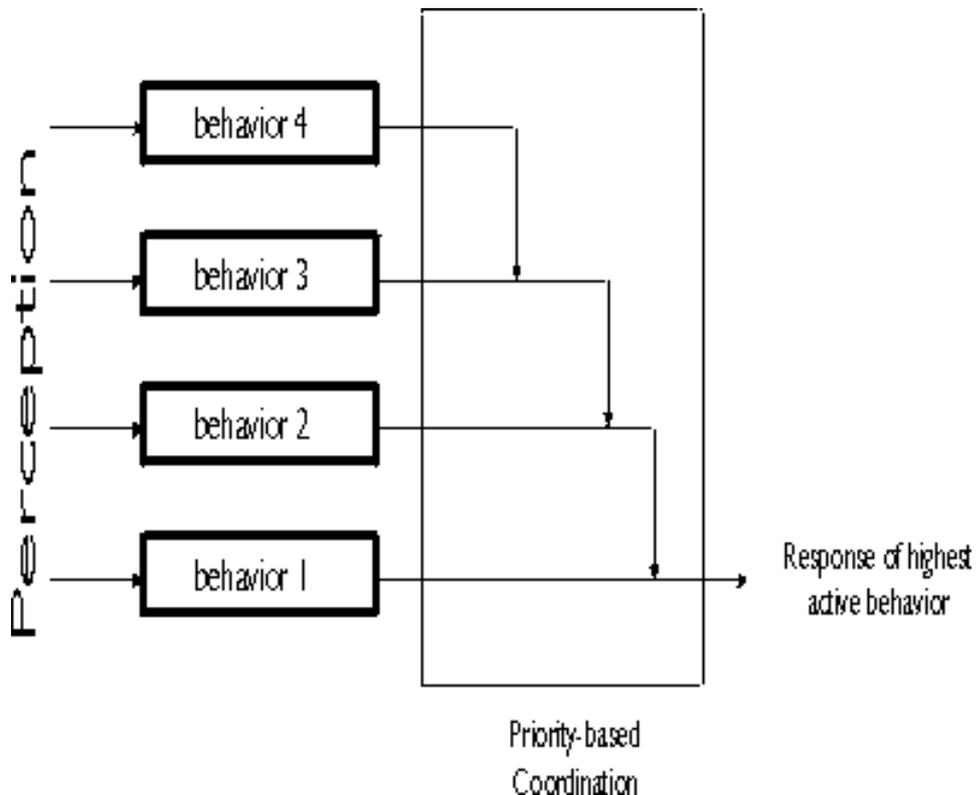
Os métodos cooperativos combinam as saídas referentes aos múltiplos comportamentos.

exemplo pág. 109

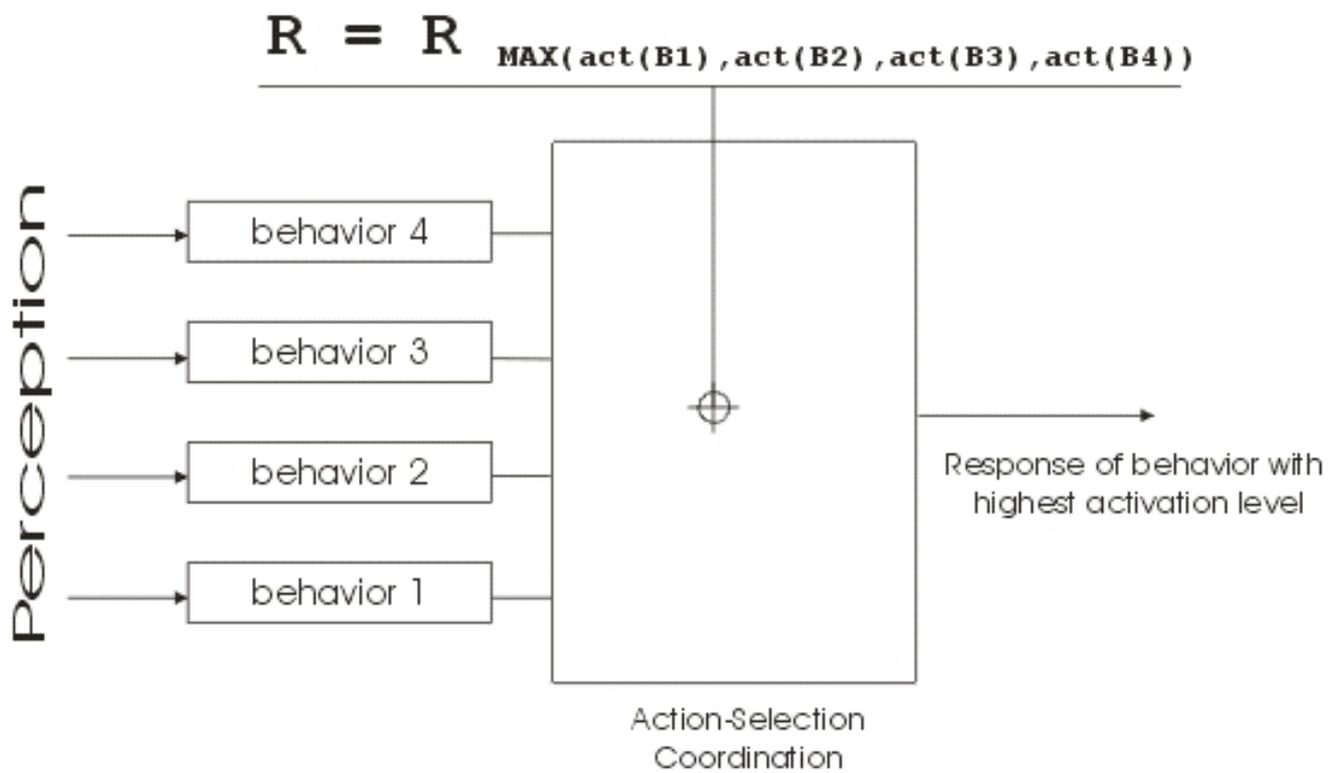
## 3.4.3 Controle dos Comportamentos

- Estudo da natureza da função  $\mathbf{C}$ .
- Subdivisão em duas classes:
- Competitivo

Propõe uma resolução de conflitos em tomadas de decisão. Conflitos podem existir quando dois ou mais comportamentos estão ativos, sendo cada um destes, com uma resposta diferente.

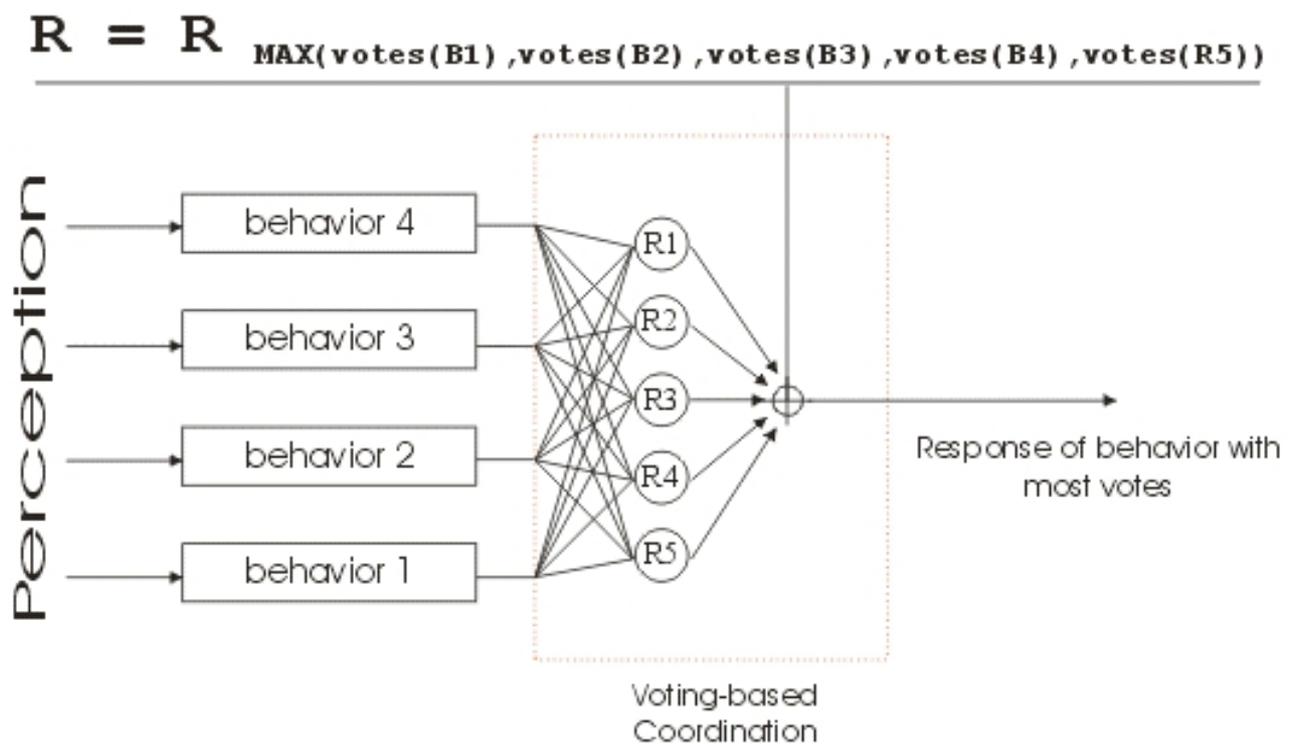


**Decisão por repressão da rede (Brooks 1986)**

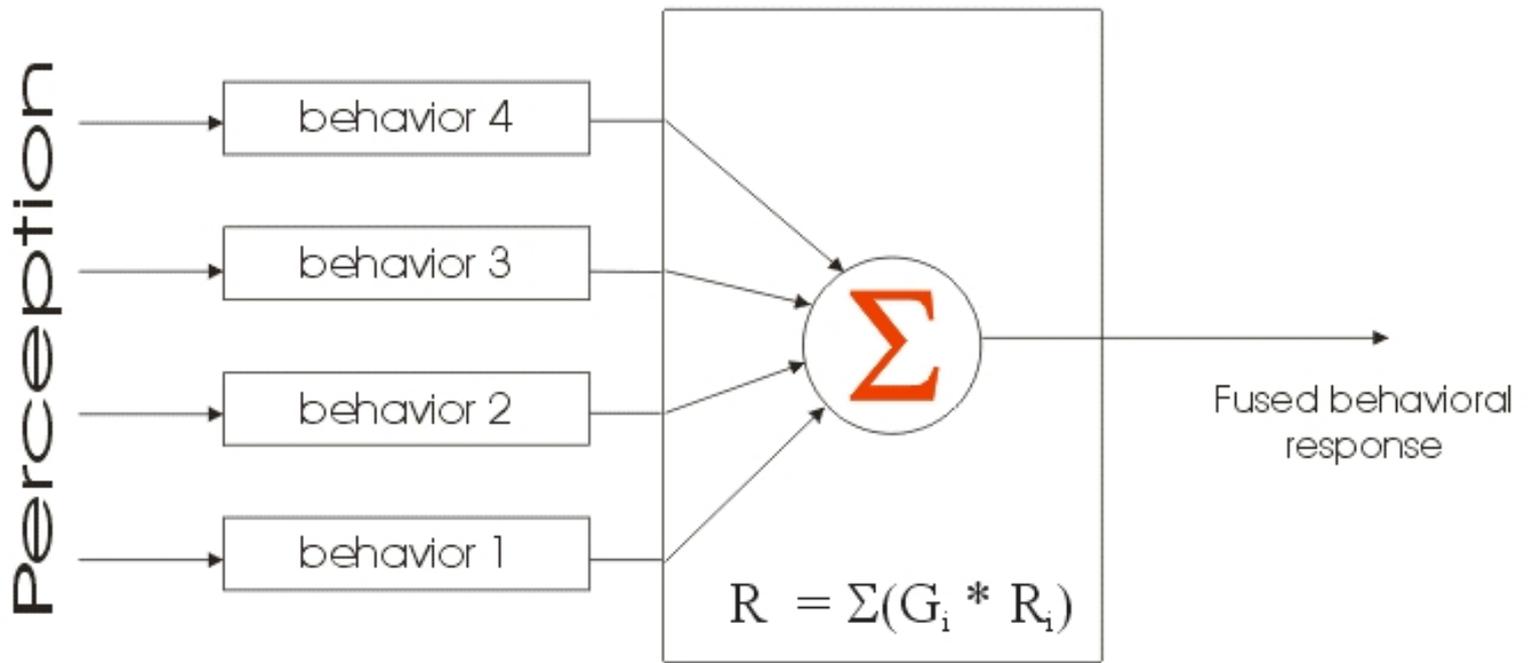


### Decisão por ação-seleção

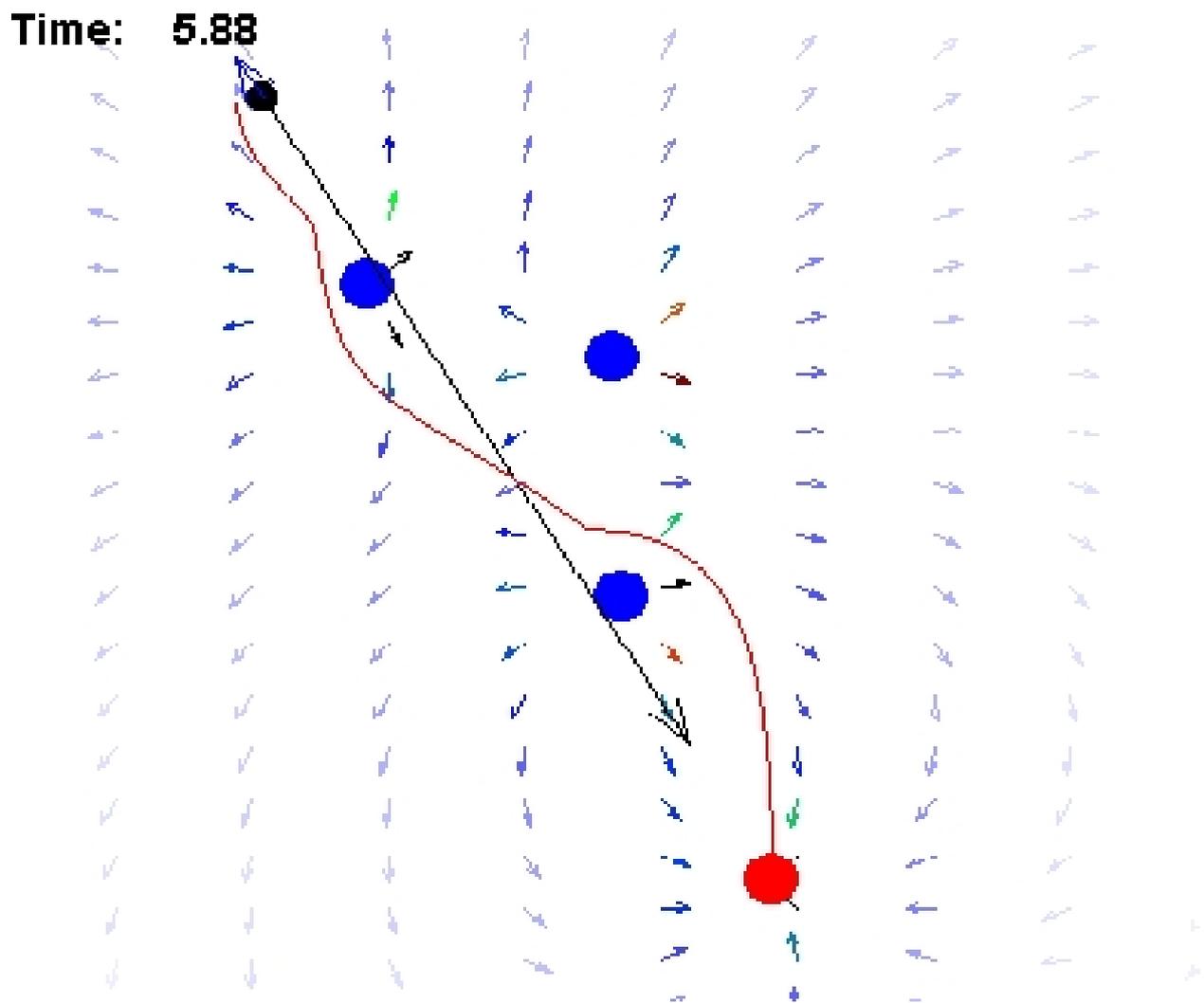
- Cooperativo
- Alternativa aos métodos competitivos;
- A fusão dos comportamentos provê a habilidade de usar concorrentemente as saídas de mais de um comportamento ao mesmo tempo, como por exemplo fugir da luz e procurar comida;
- Decisão por votação;
- Fusão de comportamentos por um vetor de soma;



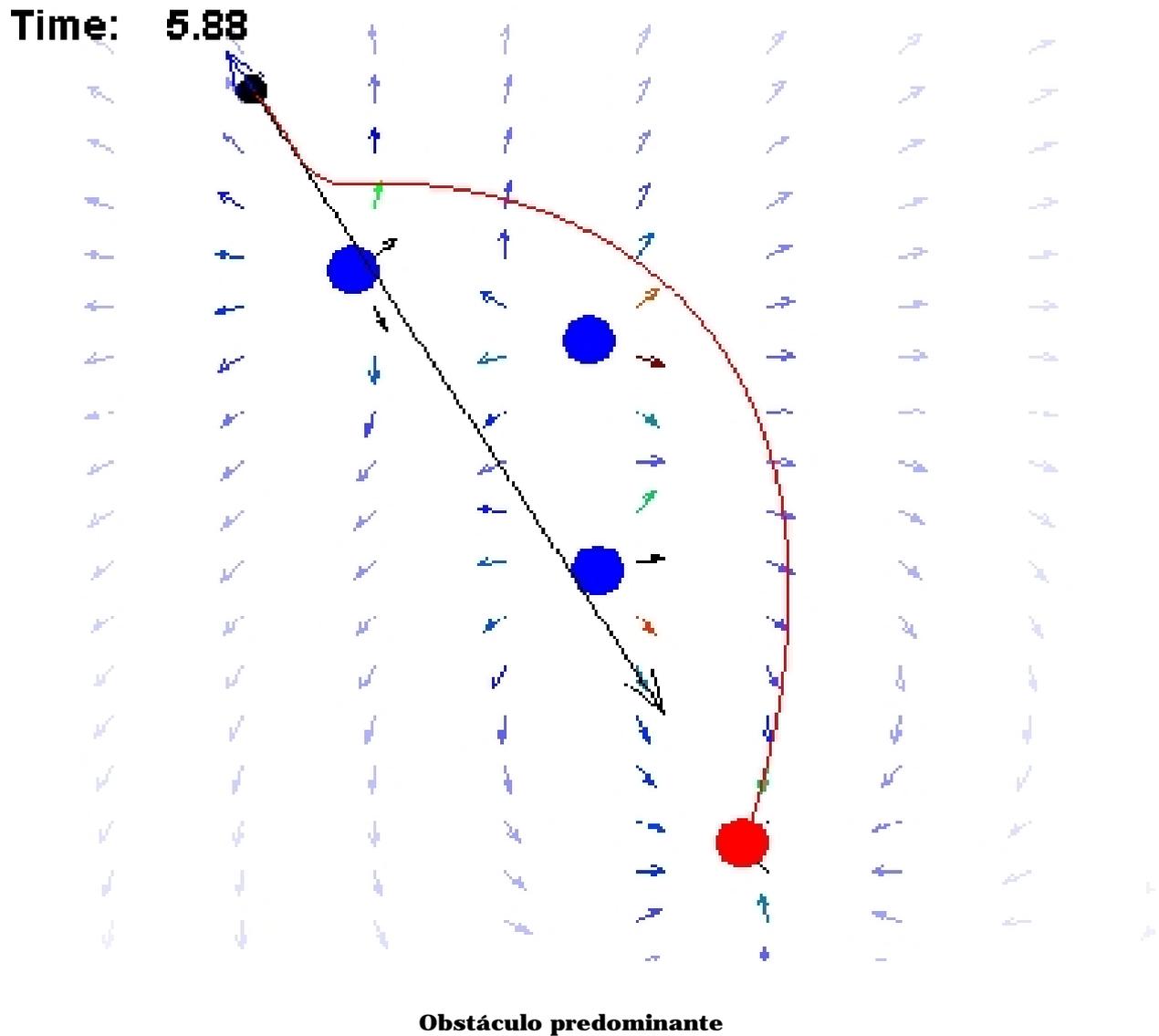
### Decisão por votação



### Fusão de comportamentos



### Objetivo predominante



### 3.4.3 Construindo/Montando os Comportamentos

- Robôs baseados em comportamento são construídos por pacotes;
- Cada pacote consiste em um operador de controle e um número de componentes de comportamento (primitivas).
- Abstração incorporada
- Construção de comportamentos em diversos níveis de abstração;
- Permite a reusabilidade no uso de sistemas modulares que descrevam sistemas baseados em comportamento. Uso de arquiteturas híbridas.
- A construção de comportamentos consiste em agregar comportamentos definidos de forma recursiva.