

INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL ROBÓTICA

Esther Luna Colombini – Instituto de Computação (Unicamp)

O que é Robótica?

2

□ Área

□ Multidisciplinar

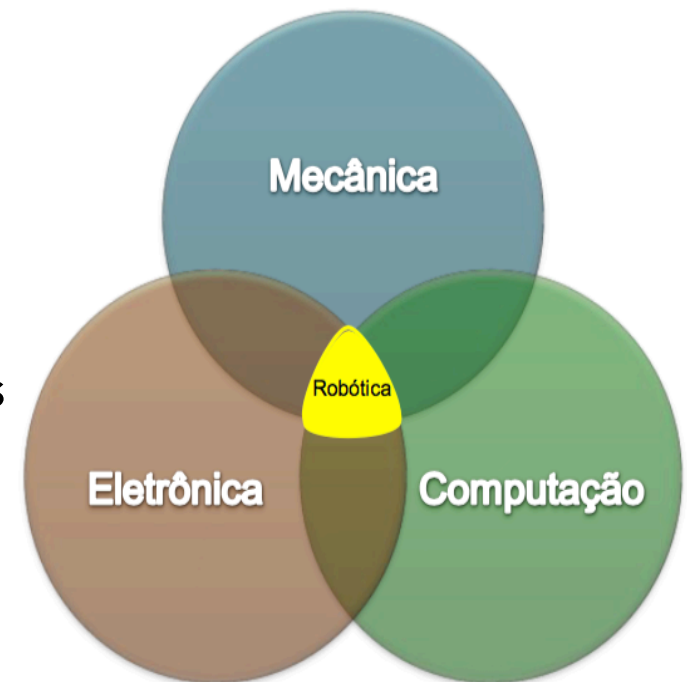
- Pois se envolve e distribui-se por várias disciplinas

□ Interdisciplinar

- Pois estabelece relações entre duas ou mais disciplinas

□ Transdisciplinar

- Pois, pode, existir um pensamento organizador que ultrapassa as próprias disciplinas



Tradicionalmente

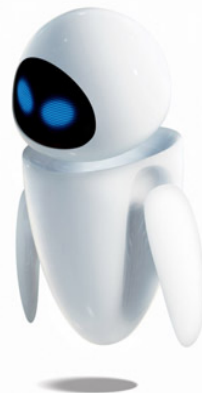
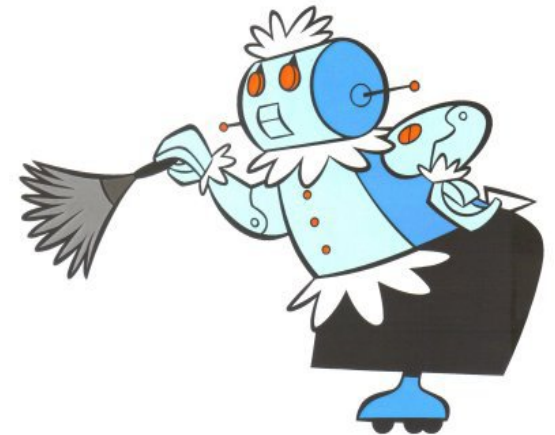
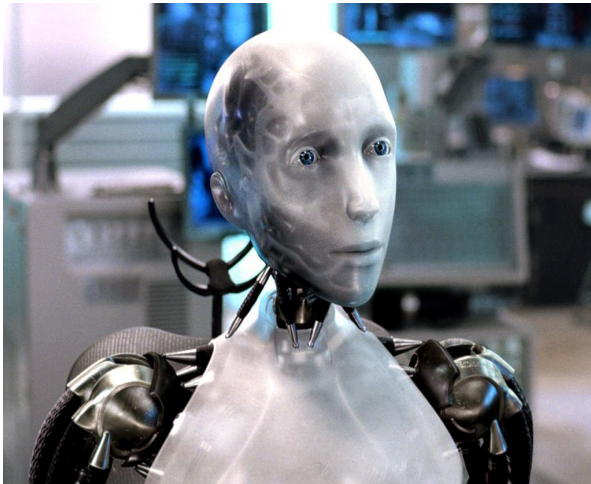
O que é Robótica?

3



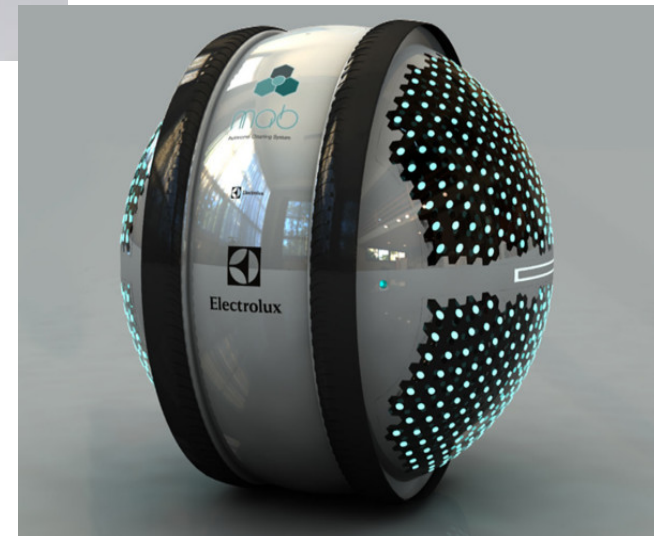
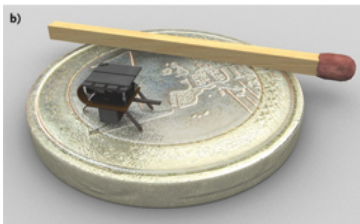
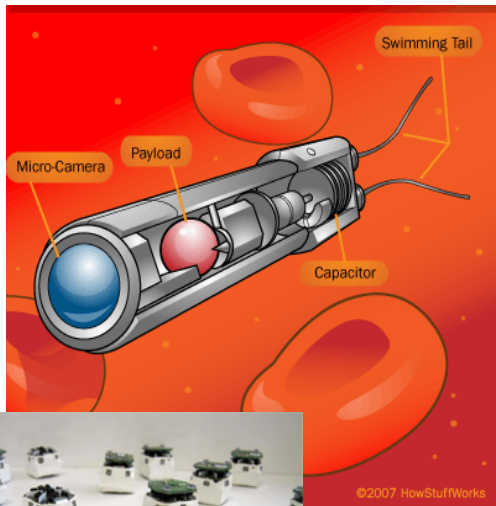
O que é um robô?

4



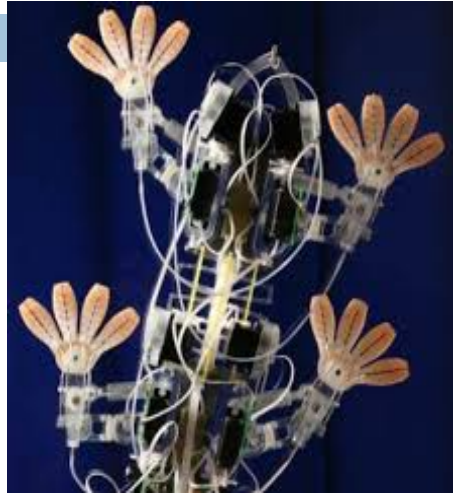
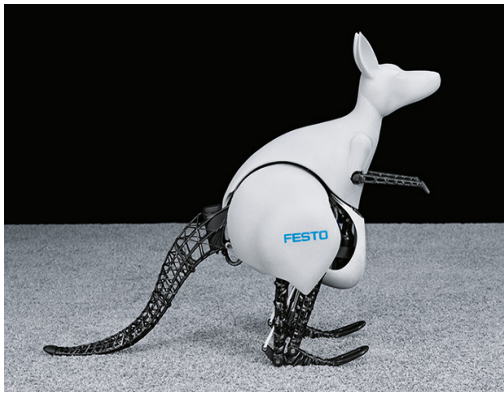
O que é um robô?

5



O que é um robô?

6



O que é um robô?

7

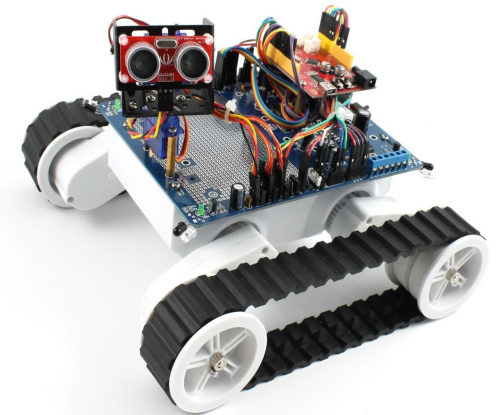
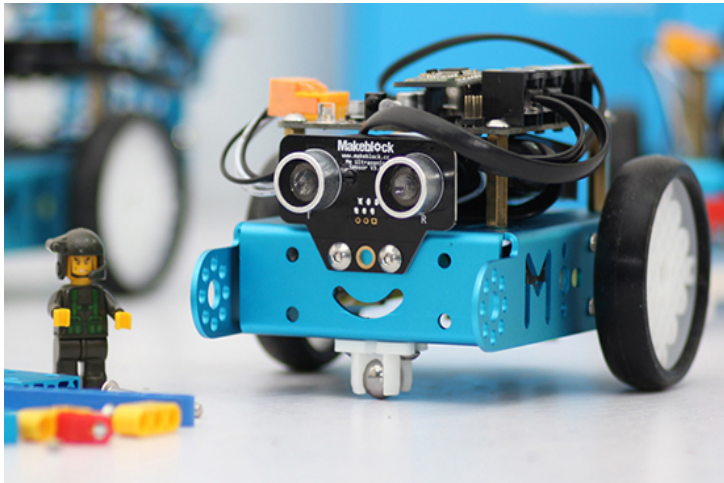


Esther Luna Colombini – Instituto de Computação (Unicamp)

esther@ic.unicamp.br

O que é um robô?

8

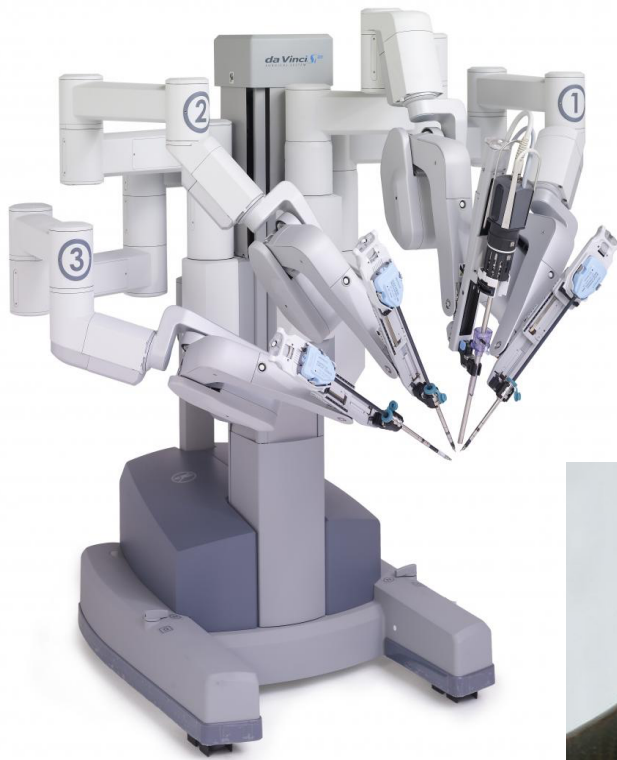


Esther Luna Colombini – Instituto de Computação (Unicamp)

esther@ic.unicamp.br

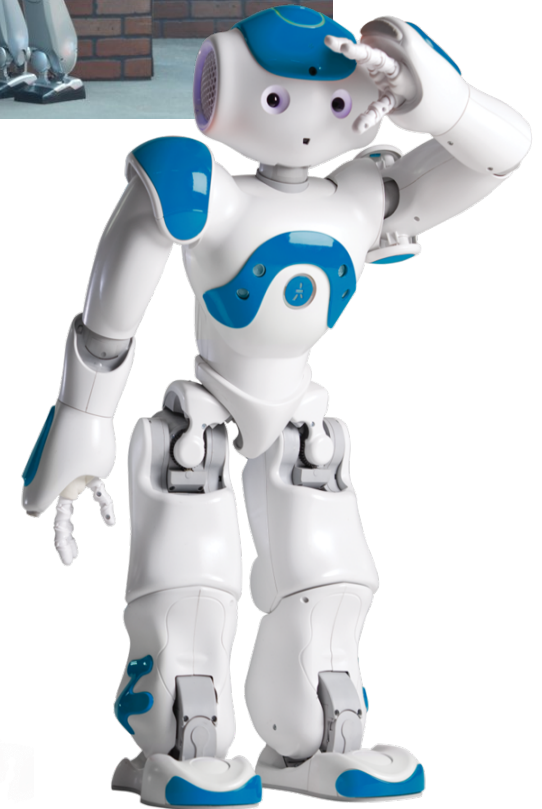
O que é um robô?

9



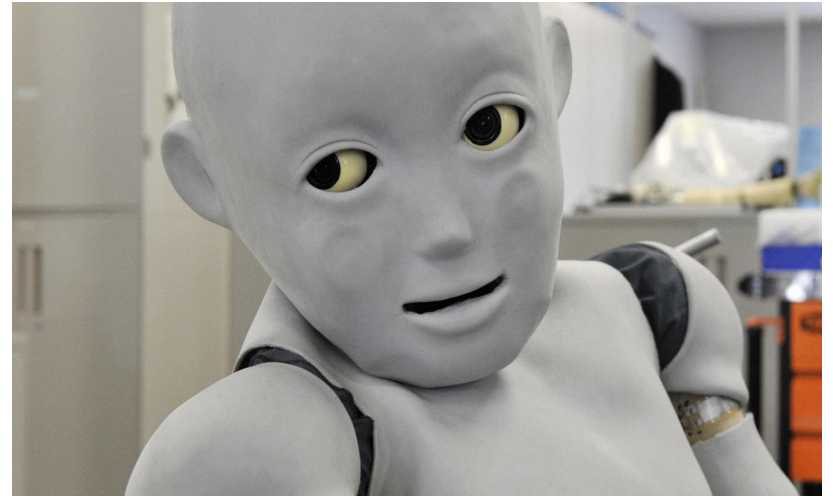
O que é um robô?

10



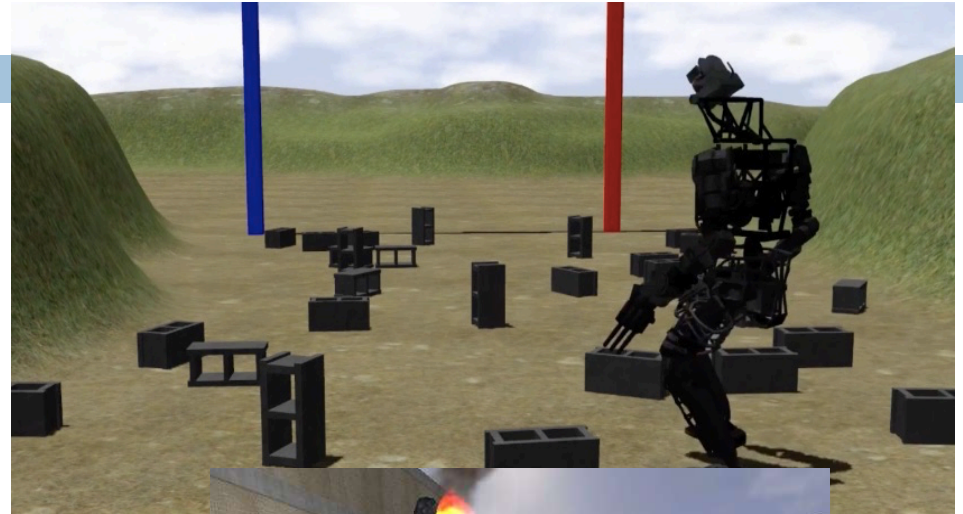
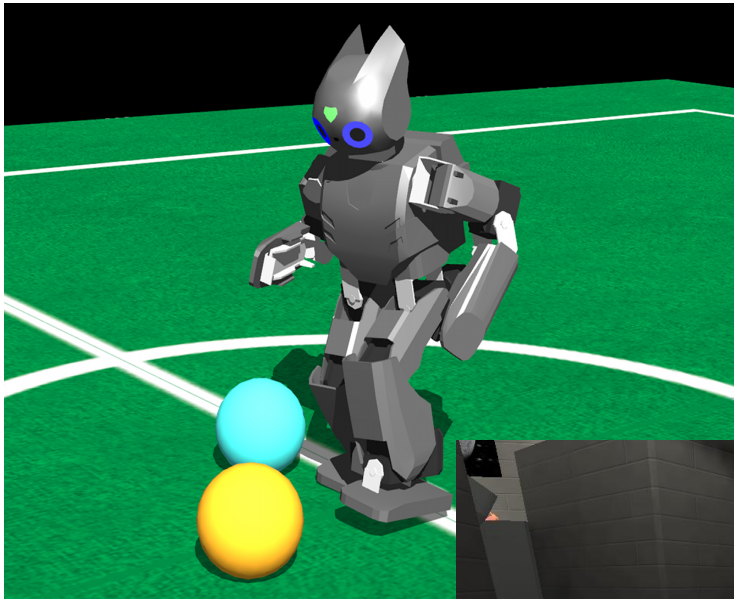
O que é um robô?

11



O que é um robô?

12



O que é um robô?

13

□ Definições históricas:

- A criação da palavra robô é atribuída ao teatrólogo checo **Karel Capek** (1890-1938) em sua peça R.U.R. (Rossum's Universal Robots), de 1920, onde designava um “trabalhador forçado” (homem de metal com aparência humana e sem consciência);
- A palavra robótica é atribuída ao escritor **Isaac Asimov** (1920-1992) em seu famoso livro “I, robot” (1950), onde designava um novo ramo do conhecimento associado à criação e programação de robôs.

O que é um robô?

14

□ Definições modernas:

□ Dicionário Houaiss:

- Máquina, autômato de aspecto humano, capaz de se movimentar e de agir;
- Mecanismo comandado por controle automático;
- Mecanismo automático que efetua operações repetitivas.

□ IFR (*International Federation of Robotics*)

- **Robôs Industriais**
- **Robôs de Serviço**

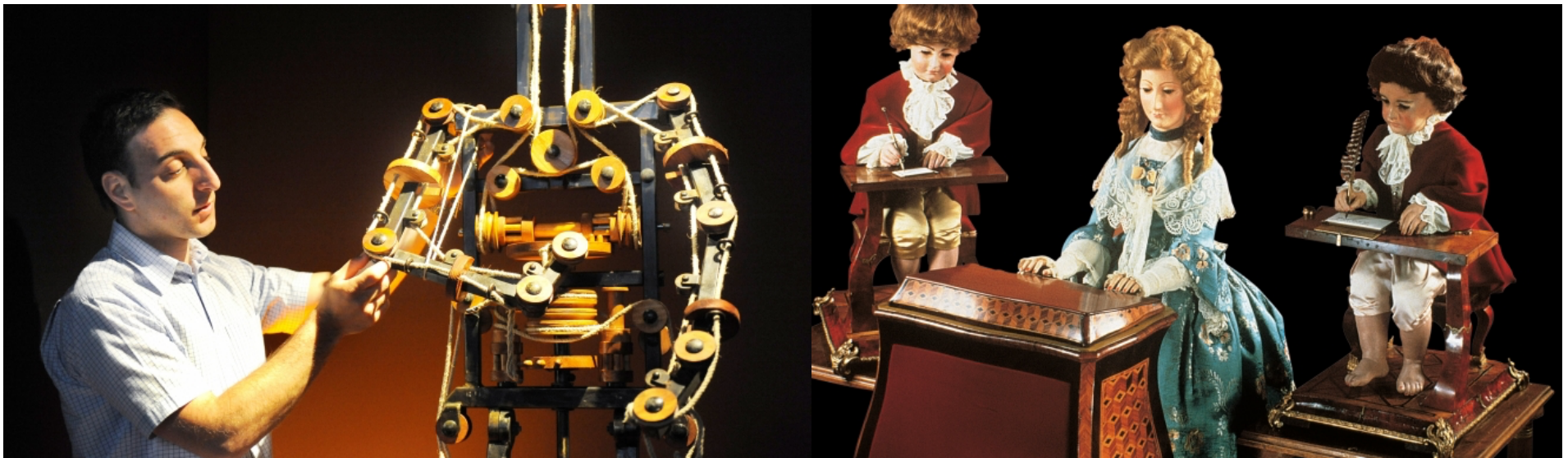
15

Um pouco de História

Evolução da robótica

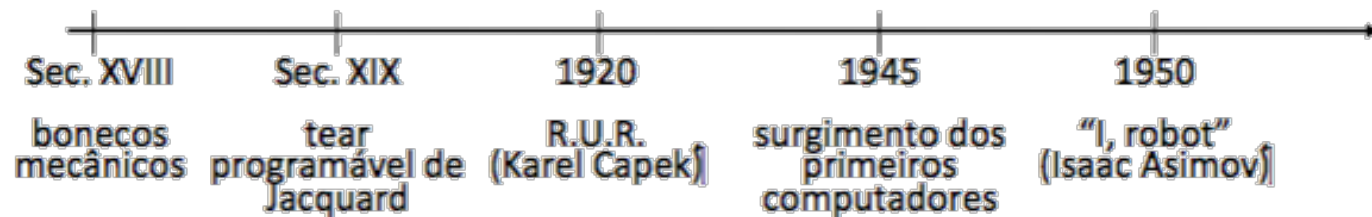
16

- **Gestação da robótica:** trabalhos técnicos rudimentares e grande influência da literatura



Da Vinci's Night

Jaquet-Droz's Three Automatons



Evolução da robótica

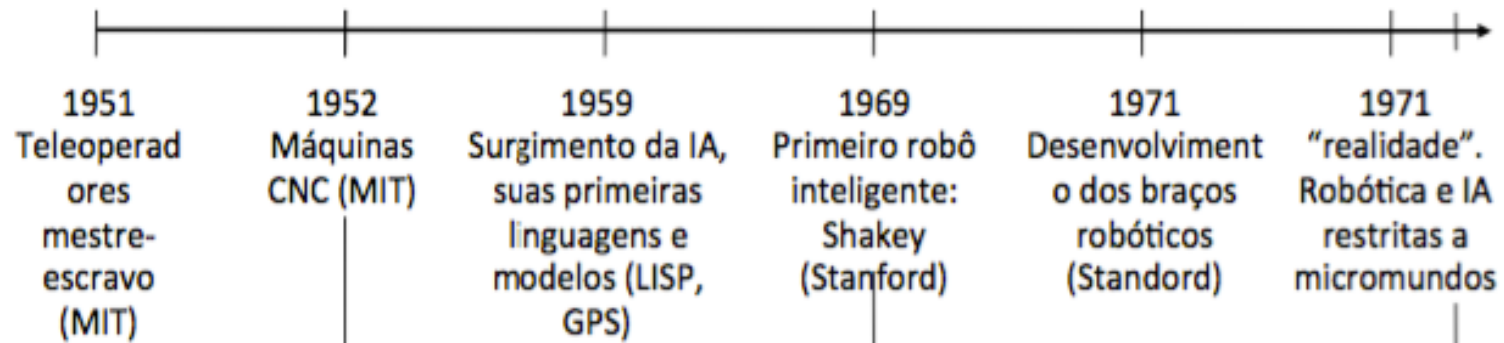
17

- As histórias de Asimov apresentadas no livro “Eu, Robô”, introduziram:
 - ▣ O cérebro positrônico: o “precursor” do microprocessador
 - ▣ As três leis da Robótica
- As três leis da Robótica (1942)
 - ▣ 1a lei: um robô não pode fazer mal a um ser humano e nem permitir que algum mal lhe aconteça.
 - ▣ 2a lei: um robô deve obedecer às ordens dos seres humanos, exceto quando estas contrariarem a primeira lei.
 - ▣ 3a lei: um robô deve proteger a sua integridade física, desde que com isto não contrarie as duas primeiras leis.

Evolução da robótica

18

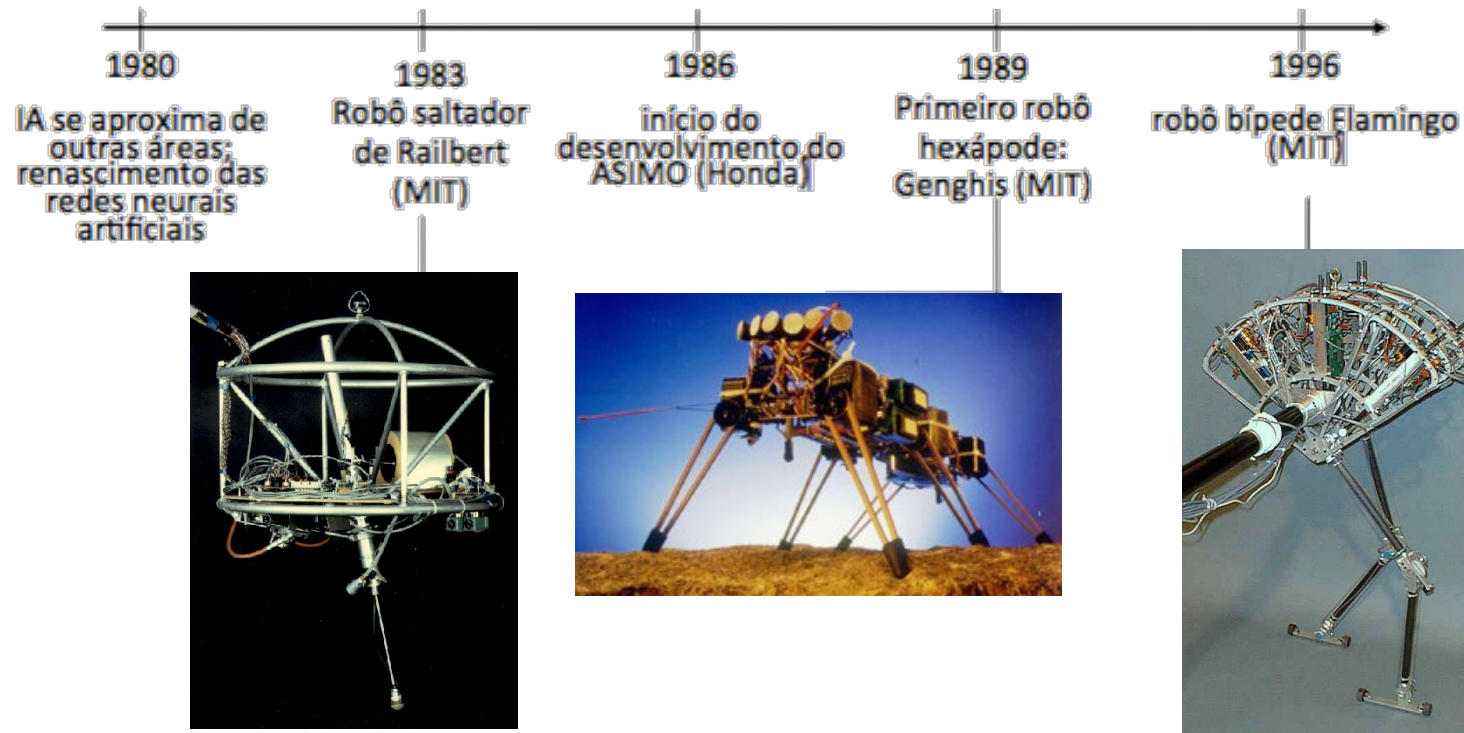
- **1ª geração de robôs:** trabalhos embrionários com IA, presença industrial dos robôs manipuladores proprioceptivos



Evolução da robótica

19

- **2ª geração de robôs:** robôs com sensores proprioceptivos e exteroceptivos (ambientes parcialmente estruturados), forte presença do controle

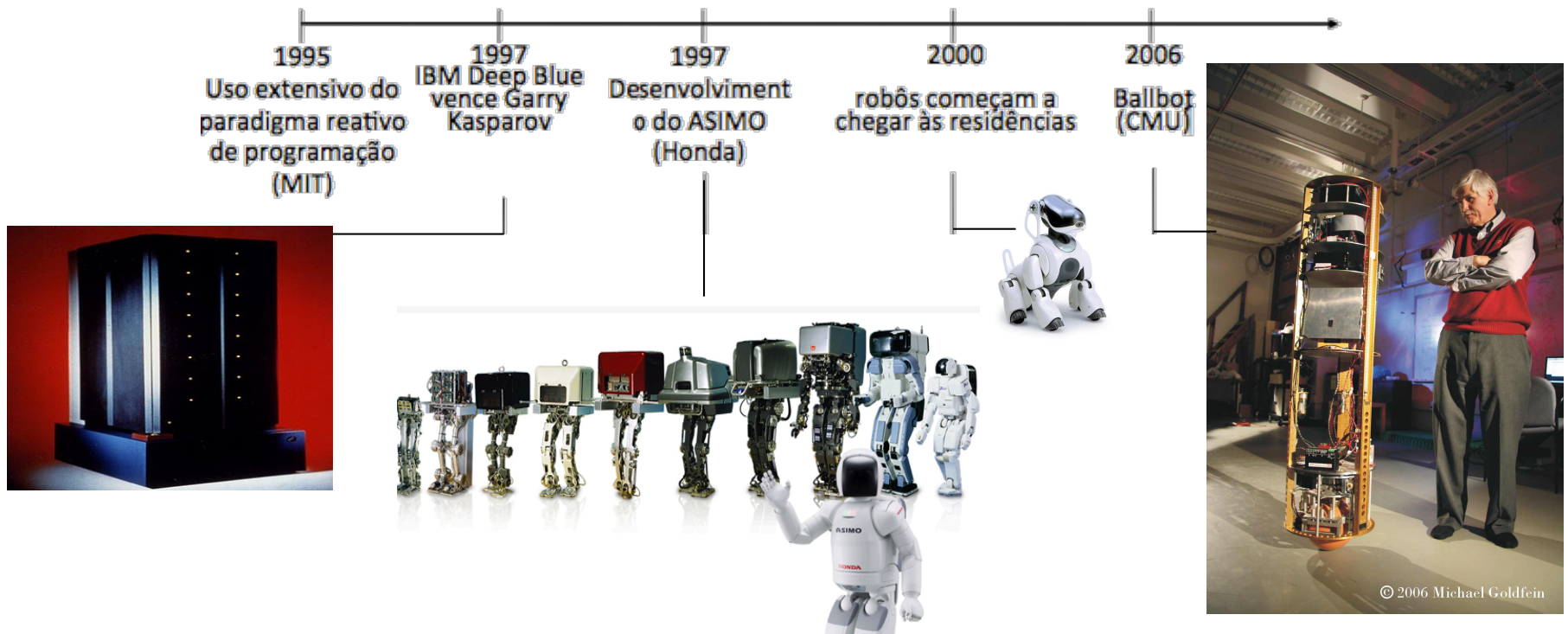


Evolução da robótica

20

- **3ª geração de robôs:** robôs fazem uso intensivo de sensores, algoritmos de percepção, controle e comunicação; crescimento da autonomia; capazes de atuar em ambiente não estruturado

Nasce a RoboCup



Por que Robótica?

21

- Tarefas típicas onde os robôs são empregados:
 - Tarefas repetitivas;
 - Tarefas potencialmente perigosas quando realizadas por humanos;
 - Tarefas onde é necessário reduzir custos;
 - Imitações de seres vivos (entretenimento).
- Implicações sociais: empregos x robôs
 - A evolução da tecnologia é uma tendência irreversível
 - Robôs extinguem postos de trabalho com baixa qualificação;
 - Robôs exigem profissionais mais qualificados;
 - Ética na robótica.

Evolução da robótica

22

□ **A robótica hoje**

□ Segmentos com produtos comerciais:

- auxílio a deficientes
- militar
- automação industrial
- automação de escritório
- entretenimento
- automação residencial

□ Mas ainda... muito fragmentada

Por que Robótica?

23

- ❑ A robótica caminha para se tornar uma das 10 maiores áreas de pesquisa até 2020
- ❑ Até 2014, já foi uma das áreas que mais contratou nas universidades Americanas no contexto da CC
- ❑ **Robótica de manufatura:** crescimento 11,5% a.a. até 2021¹
- ❑ **Robótica de serviço**
 - ❑ robôs semi ou totalmente autônomos realizam serviços, para uso profissional ou pessoal, úteis para o bem-estar dos seres humanos, excluindo as operações de manufatura
 - ❑ crescimento 20% a.a. até 2021¹

¹ Fonte: Wintergreen Research report. Disponível em: <http://www.researchmoz.us/industrial-robot-market-shares-strategy-and-forecasts-worldwide-2015-to-2021-report.html>



Robótica de Serviço: Problemas Gerais

24

- Robôs autônomos
- Robôs móveis (muitos casos)
- Natureza do ambiente
 - ▣ Parcialmente desconhecidos
 - ▣ Imprevisíveis
 - ▣ Não estruturados
- Robôs de manufatura x Robôs de serviço
 - ▣ pré-programação quase impossível
 - ▣ Não se pode determinar antecipadamente qual será o universo de transformações sensoriais e motoras requeridas pelas diversas situações que serão encontradas.
- Como fazer um robô interagir com este ambiente desconhecido?

Robôs móveis

25

- Um **robô móvel** é um dispositivo automático que é capaz de se movimentar e interagir em um ambiente definido (Wikipedia)
- Problemas e desafios da robótica móvel:
 - Como construir robôs?
 - Sensores, atuadores, processamento, estrutura física, etc.;
 - Como controlar robôs?
 - Controle de baixo nível, alto nível (programação, I.A., etc.)
 - Desafios inerentes à robótica móvel:
 - Mapeamento e localização;
 - Reconhecimento de padrões;

Robótica: Primitivas

26

- A relação entre as primitivas sentir, planejar e agir
 - Sentir
 - como, quando e o quê observar?
 - Planejar
 - como determinar as ações corretas?
 - Agir
 - como ativar os atuadores de forma adequada?

27

Paradigmas da Robótica

Paradigmas

28

1. Definição de paradigma
2. Visão dos paradigmas de programação de robôs móveis
 1. Paradigma Deliberativo (Hierárquico)
 - Arquitetura: NHC
 2. Paradigma Reativo
 - Arquiteturas: subsumption / campos potenciais
 3. Paradigma Híbrido
 - Arquitetura: AuRA

Paradigmas

29

- Números romanos (■) e arábicos (■) :

I:1 V:5 X:10 L:50 C:100 D:500 M:1000
XI:11, IX:9, XIX:19, MCMXXVIII:1928

- Realizar a operação:
$$\begin{array}{r} \text{XIX} \\ \times \text{IIV} \\ \hline \text{LVII} \end{array}$$

$$\begin{array}{r} 19 \\ \times 3 \\ \hline 57 \end{array}$$

- Conclusão:

- ▣ No domínio das operações aritméticas, considerando os paradigmas romano e árabe, a resolução do problema fica muito mais simples quando se utiliza o paradigma árabe!

Paradigmas (II)

30

□ Paradigma:

“Conjunto de suposições e/ou técnicas que caracterizam uma abordagem para uma classe de problemas”

(Murphy, 2002)

□ Estudo dos paradigmas computacionais:

- Permite comparar códigos em alto nível;
- Permite reutilização de técnicas já desenvolvidas, aproveitando experiências e desenvolvimentos anteriores

Paradigmas na Robótica

31

- Estabelecem:
 1. A relação entre as primitivas **sentir**, **planejar** e **agir**
 - ▣ Sentir: como, quando e o quê observar?
 - ▣ Planejar: como determinar as ações corretas?
 - ▣ Agir: como ativar os atuadores de forma adequada?

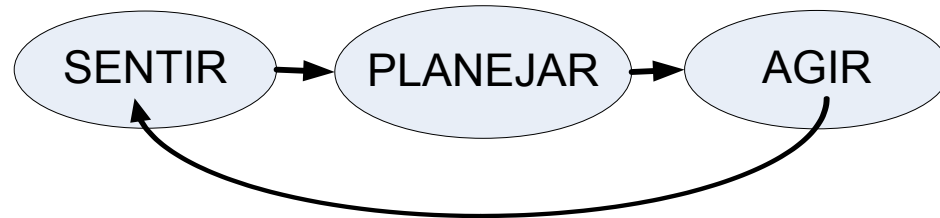
Primitiva	Entrada	Saída
Sentir	Dados	Informações
Planejar	Informações (sentidas ou conhecidas)	Diretivas
Agir	Informações ou diretiva	Comandos aos atuadores

2. A forma como os dados sensórios são processados e distribuídos pelo sistema

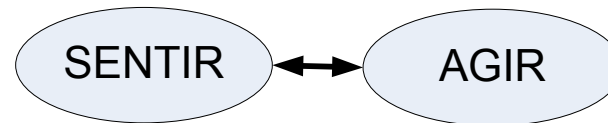
Paradigmas na Robótica (II)

32

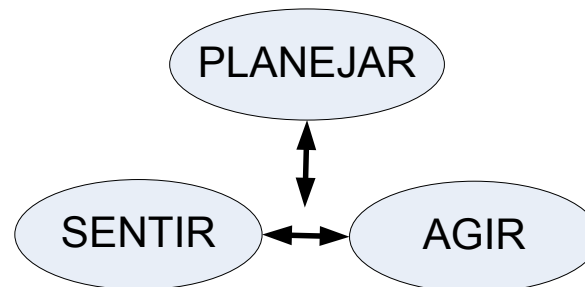
- Hierárquico
(ou deliberativo)



- Reativo



- Híbrido



Paradigma Hierárquico

33

□ (1967-1990)

Primitiva	Entrada	Saída
Sensor	Dados	Informações
Planejamento	Informações	Diretivas
Ação	Diretivas	Comandos aos atuadores

Características

34

□ Sequência de operações típicas:



1. Robô observa o mundo e constrói mapa global



2. O robô planeja todas as diretivas necessárias para atingir seus objetivos



3. O robô age para atender a sua primeira diretiva

□ Sensoriamento é monolítico:

- As observações de todos os sensores são fundidas em uma estrutura de dados global (*modelo do mundo*)

Modelo do mundo

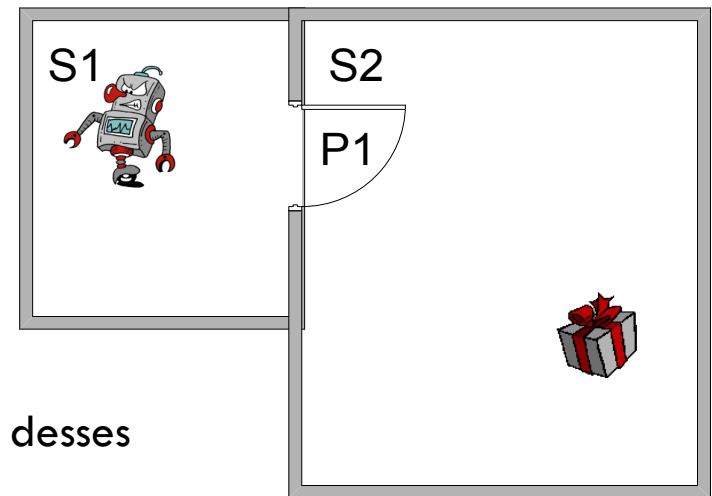
35

- O modelo do mundo tipicamente contém:
 1. Uma representação a priori (previamente adquirida) do ambiente
 2. Informações dos sensores
 3. Qualquer conhecimento cognitivo adicional que pode ser necessário para completar a tarefa

Paradigma Hierárquico: exemplo

36

- **Problema:** um robô em uma sala S1 deseja ir para uma sala S2
- **O robô percebe:**
 - ▣ Objetos móveis na sala, objetos móveis próximos à porta
 - ▣ Situação da porta (aberta ou fechada)
 - ▣ Conexões entre as salas
- **Passos:**
 1. Construir uma representação do mundo
 2. Determinar o estado atual e o objetivo
 3. Definir operadores
 4. Determinar a sequência ideal de aplicação desses operadores



Representação do conhecimento

37

□ Predicados:

- $NA_SALA(x, s)$ onde x é um objeto móvel e s é uma sala
- $PRÓXIMO_A(x, p)$ onde x é um objeto móvel e p é uma porta
- $SITUAÇÃO(p, sp)$ onde p é uma porta e sp assume ABERTA ou FECHADA
- $CONECTA(p, sa, sb)$ onde p é uma porta, sa e sb são salas

Maiúsculo: predicados ou valores, minúsculo: variáveis

Determinação dos estados

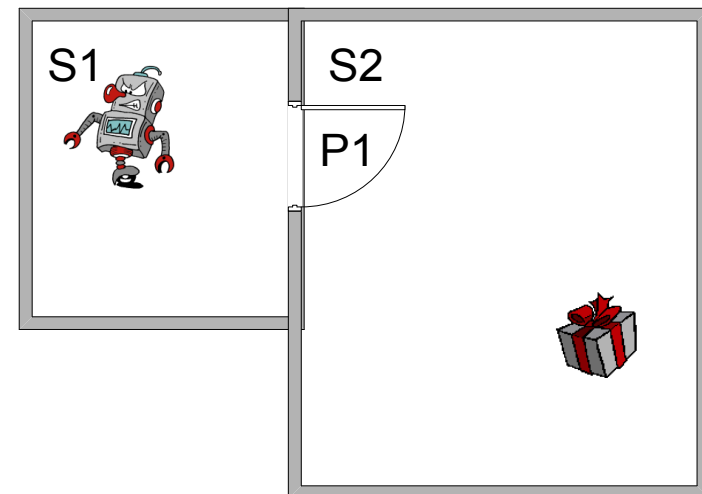
38

▣ Estado inicial:

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

▣ Estado final:

NA_SALA(ROBÔ, S2)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO (P1, ABERTA)



Definição dos operadores

39

Operador	Pré-condição	Adicionar à lista	Retirar da lista
IR_PARA_PORTA (ROBÔ, p)	NA_SALA (ROBÔ, s_a) CONECTADA (p, s_a , s_b)	PRÓXIMO (ROBÔ, p)	
ATRAVESSAR (ROBÔ, p)	CONECTADA (p, s_a , s_b) PRÓXIMO (ROBÔ, p) SITUAÇÃO (p, ABERTA) NA_SALA (ROBÔ, s_a)	NA_SALA (ROBÔ, s_b)	NA_SALA (ROBÔ, s_a)

Como aplicar os operadores?

40

Algoritmo: Resolução de diferenças

1. Computar a diferença entre o estado objetivo e o estado atual usando uma função para determinação de diferença. Se não há diferença, então fim.

Estado inicial:

NA_SALA(ROBÔ, S1) ✗
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

Estado final:

NA_SALA(ROBÔ, S2) ✓
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO (P1, ABERTA)

Resolução de diferenças

41

2. Reduzir a diferença selecionando um operador da tabela de diferença cuja adição à lista inclua um predicado que nega a diferença

Operador	Pré-condição	Adicionar à lista	Retirar da lista
IR_PARA_PORTA (ROBÔ, p)	NA_SALA (ROBÔ, s _a) CONECTADA (p, s _a , s _b)	PRÓXIMO (ROBÔ, p)	
ATRAVESSAR (ROBÔ, p)	CONECTADA (p, s _a , s _b) PRÓXIMO (ROBÔ, p) SITUAÇÃO (p, ABERTA) NA_SALA (ROBÔ, s _a)	NA_SALA (ROBÔ, s _b)	NA_SALA (ROBÔ, s_a)

Estado:

~~NA_SALA(ROBÔ, S1)~~

NA_SALA(PRESENTE, S2)

CONECTADA(P1, S1, S2)

CONECTADA(P1, S2, S1)

SITUAÇÃO(P1, ABERTA)

OBS: variável s_a instanciada com o valor S1

Resolução de diferenças

42

3. Examinar as pré-condições para ver se um conjunto de condições verdadeiras para a variável em questão pode ser encontrado.

Se nem todas forem encontradas, tomar a primeira pré-condição falsa, fazê-la o novo objetivo e armazenar o objetivo anterior no *stack*. Recursivamente reduzir esta diferença repetindo os passos 2 e 3.

Operador	Pré-condição	Adicionar à lista	Retirar da lista
IR_PARA_PORTA (ROBÔ, p)	NA_SALA (ROBÔ, s _a) CONNECTADA (p, s _a , s _b)	PRÓXIMO (ROBÔ, p)	
ATRAVESSAR (ROBÔ, p)	CONNECTADA (p, s_a, s_b) ✓ ✗ PRÓXIMO (ROBÔ, p) SITUAÇÃO (p, ABERTA) ✓ NA_SALA (ROBÔ, s_a) ✓	NA_SALA (ROBÔ, s _b)	NA_SALA (ROBÔ, s _a)

Estado:

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONNECTADA(P1, S1, S2)
CONNECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

Variáveis instanciadas:
 s_a=S1, p=P1, s_b = S2

Resolução de diferenças

43

- Objetivo: PROXIMO (ROBÔ, p)

Operador	Pré-condição	Adicionar à lista	Retirar da lista
IR_PARA_PORTA (ROBÔ, p)	NA_SALA (ROBÔ, s _a) ✓ CONECTADA (p, s _a , s _b) ✓	PRÓXIMO (ROBÔ, p)	
ATRAVESSAR (ROBÔ, p)	CONECTADA (p, s _a , s _b) PRÓXIMO (ROBÔ, p) ← SITUAÇÃO (p, ABERTA) NA_SALA (ROBÔ, s _a)	NA_SALA (ROBÔ, s _b)	NA_SALA (ROBÔ, s _a)

Estado :

NA_SALA(ROBÔ, S1)

NA_SALA(PRESENTE, S2)

CONECTADA(P1, S1, S2)

CONECTADA(P1, S2, S1)

SITUAÇÃO(P1, ABERTA)

Variáveis instanciadas:

s_a: S1, p: P1, s_b: S2

Resolução de diferenças

44

- Objetivo: PROXIMO (ROBÔ, p)

Operador	Pré-condição	Adicionar à lista	Retirar da lista
IR_PARA_PORTA (ROBÔ, p)	NA_SALA (ROBÔ, s _a) CONECTADA (p, s _a , s _b)	PRÓXIMO (ROBÔ, p)	
ATRAVESSAR (ROBÔ, p)	CONECTADA (p, s _a , s _b) ✓ PRÓXIMO (ROBÔ, p) ✓ SITUAÇÃO (p, ABERTA) ✓ NA_SALA (ROBÔ, s _a) ✓	NA_SALA (ROBÔ, s _b)	NA_SALA (ROBÔ, s _a)

Estado :

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)
PROXIMO(ROBÔ, P1)

Variáveis instanciadas:

s_a: S1, p: P1, s_b: S2

Resolução de diferenças

45

4. Quando todas as pré-condições para um operador forem possíveis, colocar o operador no *stack* e atualizar uma cópia do modelo do mundo. Retornar ao operador cujas pré-condições falharam e então aplicar seu operador ou retornar a uma outra pré-condição falha.

Operador	Pré-condição	Adicionar à lista	Retirar da lista
IR_PARA_PORTA (ROBÔ, p)	NA_SALA (ROBÔ, s _a) CONNECTADA (p, s _a , s _b)	PRÓXIMO (ROBÔ, p)	
ATRAVESSAR (ROBÔ, p)	CONNECTADA (p, s _a , s _b) PRÓXIMO (ROBÔ, p) SITUAÇÃO (p, ABERTA) NA_SALA (ROBÔ, s _a)	NA_SALA (ROBÔ, s _b)	NA_SALA (ROBÔ, s _a)

- Plano para o robô:
 - ▣ **IR_PARA_PORTA (ROBÔ, P1)**
 - ▣ **ATRAVESSAR (ROBÔ, P1)**

Execução do plano

46

Estado inicial:

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

Execução do plano

47

Estado inicial:

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

IR_PARA_PORTA (ROBÔ, P1)

Pré-condição:

NA_SALA (ROBÔ, S1)
CONECTADA (P1, S1, S2)
Adiciona: PRÓXIMO (ROBÔ, P1)

Retira: não

Execução do plano

48

Estado inicial:

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

IR_PARA_PORTA (ROBÔ, P1)

Pré-condição:

NA_SALA (ROBÔ, S1) ✓

CONECTADA (P1, S1, S2) ✓

Adiciona: PRÓXIMO (ROBÔ, P1)

Retira: não

Execução do plano

49

Estado inicial:

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

IR_PARA_PORTA (ROBÔ, P1)

Pré-condição:

NA_SALA (ROBÔ, S1)
CONECTADA (P1, S1, S2)
Adiciona: PRÓXIMO (ROBÔ, P1)

Retira: não

Estado :

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)
PRÓXIMO (ROBÔ, P1)

Execução do plano

50

Estado inicial:

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

IR_PARA_PORTA (ROBÔ, P1)

Pré-condição:

NA_SALA (ROBÔ, S1)
CONECTADA (P1, S1, S2)
Adiciona: PRÓXIMO (ROBÔ, P1)

Retira: não

Estado :

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)
PRÓXIMO (ROBÔ, P1)

ATRAVESSAR (ROBÔ, P1)

Pré-condição:

CONECTADA (P1, S1, S2)
PRÓXIMO (ROBÔ, P1)
SITUAÇÃO (P1, ABERTA)
NA_SALA (ROBÔ, S1)
Adiciona: NA_SALA (ROBÔ, S2)
Retira: NA_SALA (ROBÔ, S1)

Execução do plano

51

Estado inicial:

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

IR_PARA_PORTA (ROBÔ, P1)

Pré-condição:

NA_SALA (ROBÔ, S1)
CONECTADA (P1, S1, S2)
Adiciona: PRÓXIMO (ROBÔ, P1)

Retira: não

Estado :

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)
PRÓXIMO (ROBÔ, P1)

ATRAVESSAR (ROBÔ, P1)

Pré-condição:

CONECTADA (P1, S1, S2) ✓
PRÓXIMO (ROBÔ, P1) ✓
SITUAÇÃO (P1, ABERTA) ✓
NA_SALA (ROBÔ, S1)
Adiciona: NA_SALA (ROBÔ, S2)
Retira: NA_SALA (ROBÔ, S1)

Execução do plano

52

Estado inicial:

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)

IR_PARA_PORTA (ROBÔ, P1)

Pré-condição:

NA_SALA (ROBÔ, S1)
CONECTADA (P1, S1, S2)
Adiciona: PRÓXIMO (ROBÔ, P1)

Retira: não

Estado :

NA_SALA(ROBÔ, S1)
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)
PRÓXIMO (ROBÔ, P1)

Estado final :

~~NA_SALA(ROBÔ, S1)~~
NA_SALA(PRESENTE, S2)
CONECTADA(P1, S1, S2)
CONECTADA(P1, S2, S1)
SITUAÇÃO(P1, ABERTA)
PRÓXIMO (ROBÔ, P1)
NA_SALA (ROBÔ, S2)

ATRAVESSAR (ROBÔ, P1)

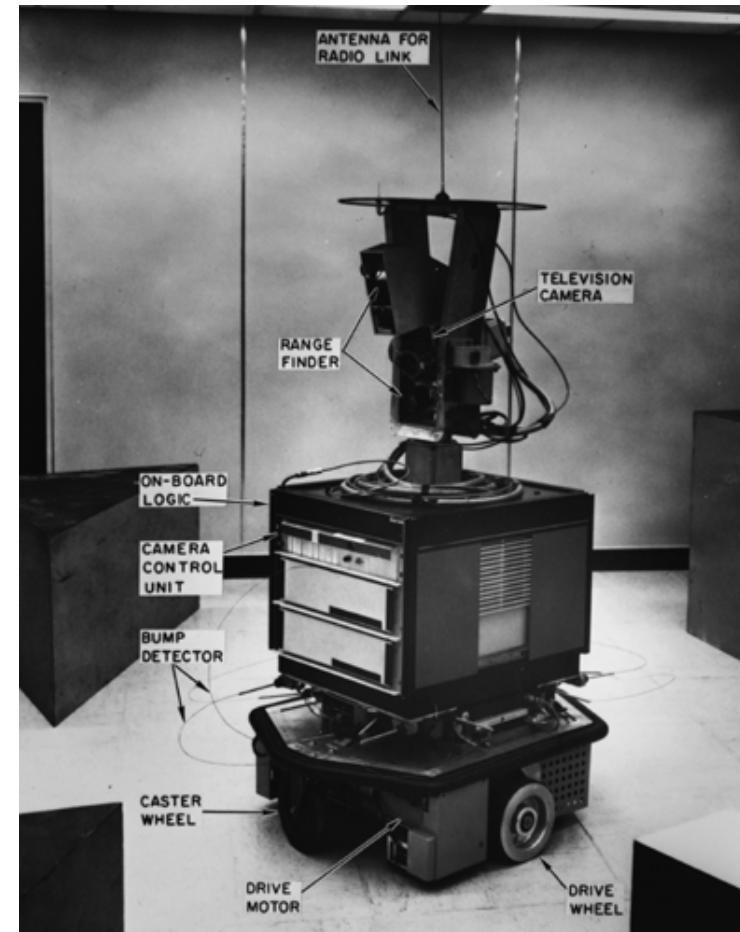
Pré-condição:

CONECTADA (P1, S1, S2)
PRÓXIMO (ROBÔ, P1)
SITUAÇÃO (P1, ABERTA)
NA_SALA (ROBÔ, S1)
Adiciona: NA_SALA (ROBÔ, S2)
Retira: NA_SALA (ROBÔ, S1)

STRIPS

53

- Algoritmo: Strips
- Abordagem:
 - *means-end analysis*: se robô não pode completar a tarefa ou atingir o objetivo em um passo, ele seleciona a ação que vai reduzir a diferença entre o estado em que ele está agora e o objetivo
- Robô Shakey (1969)
 - Primeiro robô inteligente
 - Artificial Intelligence Center Stanford



STRIPS: Avaliação

54

□ ○ Strips sintetiza 2 problemas da abordagem:

1. “Closed world assumption”

○ modelo do mundo contém tudo o que o robô precisa saber. Não pode haver surpresas!



→ Programador humano precisa pensar em tudo!

2. “Frame problem”

○ mundo real é enorme. O número de axiomas que o programa teria que organizar para cada passada na tabela de diferenças seria intratável!



→ Não é uma metodologia aplicável no mundo real!

Além disso, é impossível lidar com as incertezas do ambiente no período em que os olhos do agente estão fechados

Problemas gerais...

55

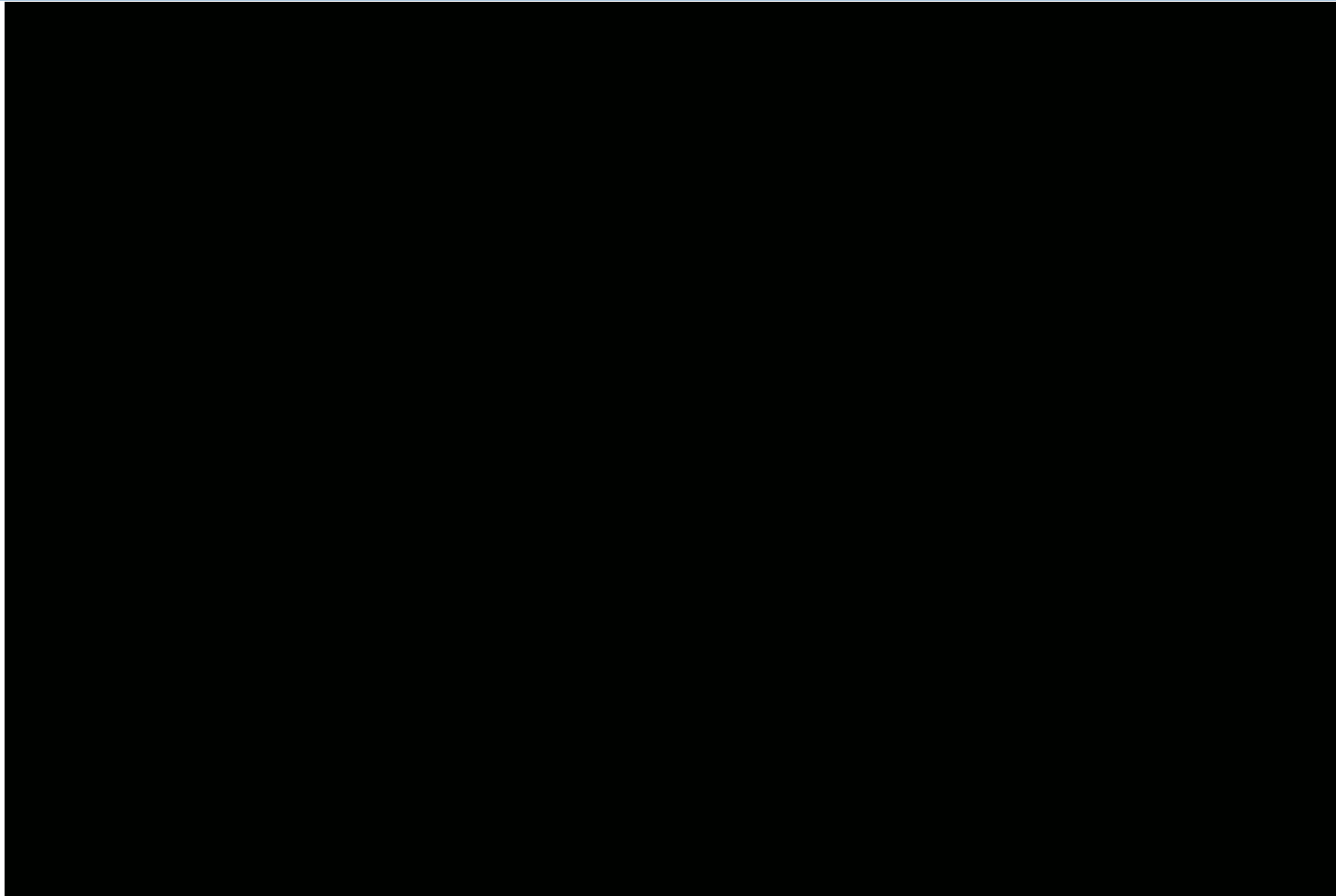
- Impossibilidade de lidar com as incertezas do ambiente no período em que os olhos do robô estão fechados



- Esta metodologia de *dividir-e-conquistar* forma um tipo de planejamento chamado **PLANEJAMENTO LINEAR !!**
- **Planejamento Linear**
 - ▣ Resolve sub-objetivos de forma sequencial e ordenada

Shakey

56



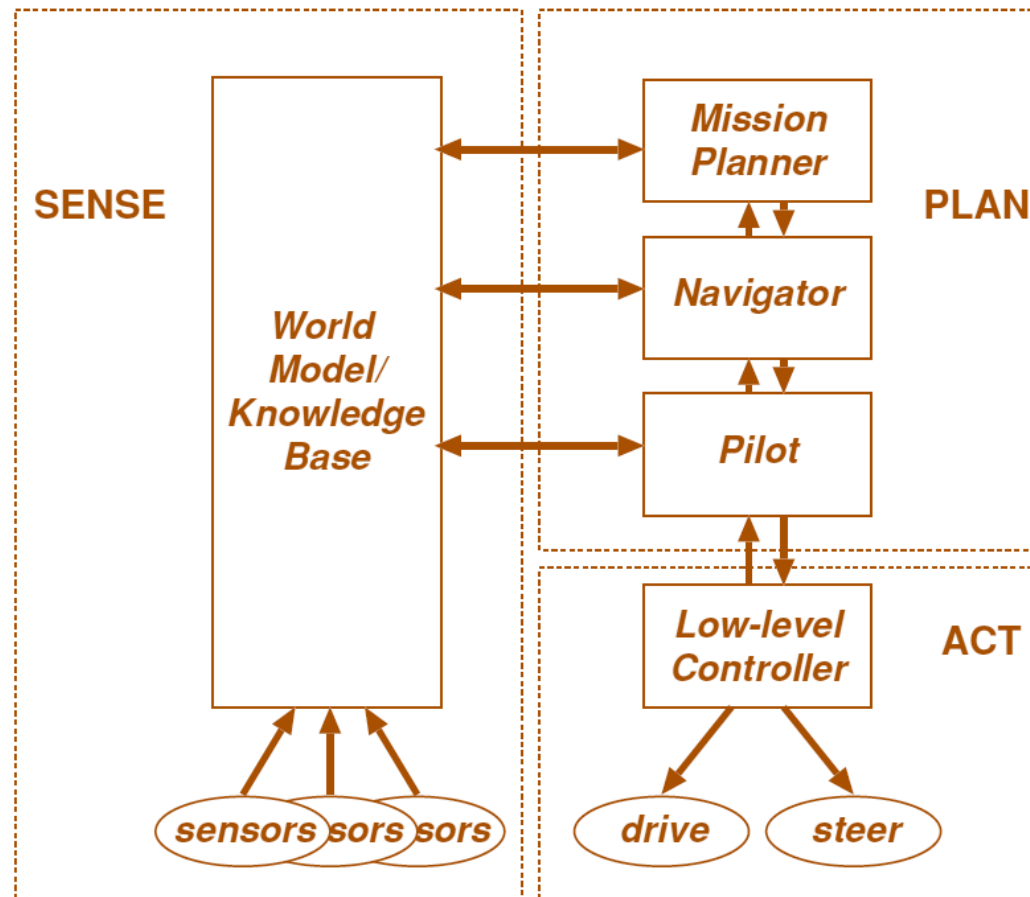
Arquiteturas

57

- Arquiteturas mais representativas do paradigma hierárquico:
 - ▣ Nested Hierarchical Controller (NHC), desenvolvida por Meystel
 - ▣ NIST Realtime Control System (RCS), desenvolvida por Albus/JPL

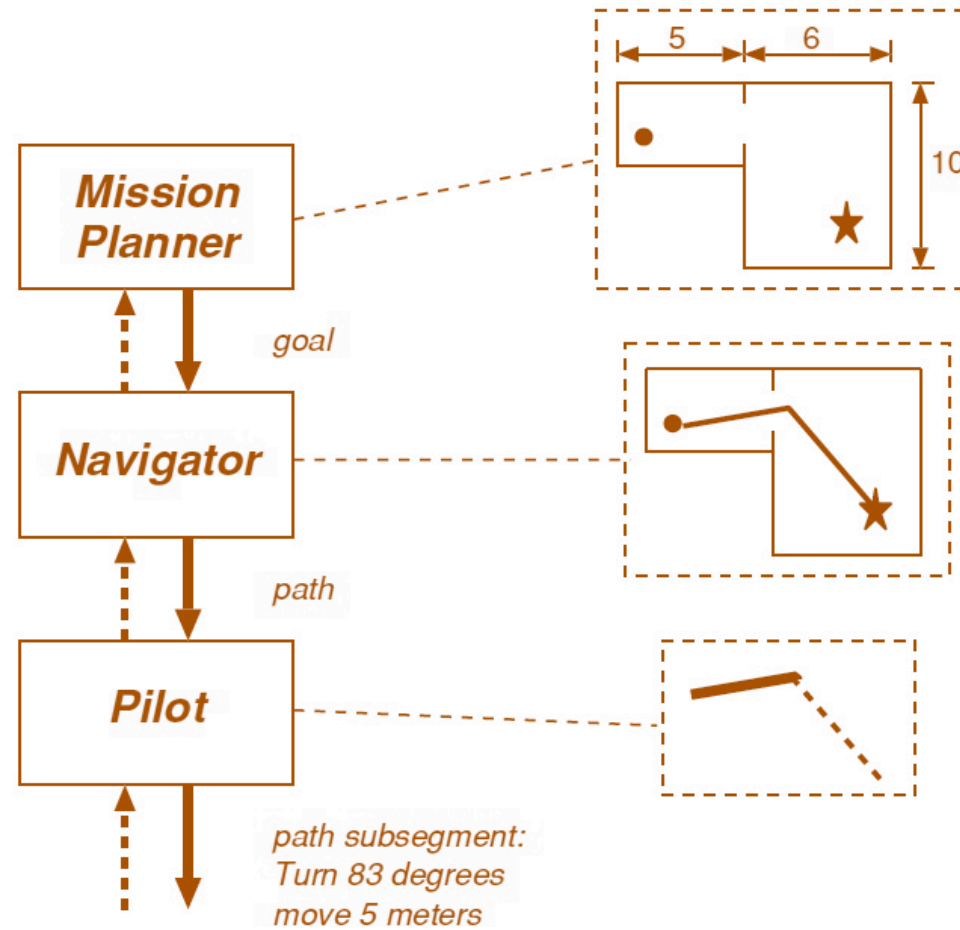
Arquitetura NHC

58



Arquitetura NHC (II)

59



NHC: resumo

60

- Planejamento para navegação consiste de três passos executados pelo:
 - Planejador da missão (localiza objetos no mapa)
 - Navegador (gera um caminho)
 - Piloto (age sobre a primeira parte do caminho)
- Todos os módulos têm acesso ao modelo do mundo para computar sua porção do planejamento
- Nunca foi implementado em um robô real (apenas em simuladores). Custo de hardware muito elevado na época...

NHC: Vantagens e desvantagens

61

- Em função da modularidade:
 - ▣ Não precisa executar o planejador da missão e o navegador a toda iteração se o mundo estiver como planejado
 - ▣ Problemas de atuação (erros de trajetória em relação ao segmento a ser percorrido): correção no próprio módulo piloto
 - ▣ Variações no percurso: replanejamento apenas no segmento correspondente (módulo navegador)
 - ▣ Mudança de plano, novas tarefas e objetivos: replanejamento no módulo planejador e de missão
- OK para tarefas de navegação. Em outros casos a divisão de responsabilidades em uma hierarquia de planejamento pode não ser clara...

RCS

62

- Desenvolvida no NIST – National Institute of Standards and Technology
 - ▣ Auxílio a empresas de manufatura para construir robôs inteligentes
 - ▣ Idealizado como um padrão geral para o projeto de robôs móveis que seguem paradigma hierárquico
- Baseado no NHC, porém:
 - ▣ Modelo de mundo é organizado em uma hierarquia, com níveis de abstração
 - ▣ Módulos extraem características a partir dos sensores em diferentes níveis de abstração para produzir um modelo do mundo
 - ▣ O módulo de planejamento testa o plano gerado (simula) antes de transformá-lo em ações

RCS (III)

64

- Decomposição da função sensorial: extração de características em graus variados de descrição
 - ▣ Níveis mais altos -> descrição de mais alto nível -> atributos com semântica mais natural
- Decomposição do planejador:
 - ▣ Níveis mais altos -> descrição em linguagem de nível mais alto -> planos mais abstratos
- Decomposição das ações:
 - ▣ Níveis mais altos -> macro-operadores
 - ▣ Níveis mais baixos -> ações de controle

Paradigma hierárquico: avaliação

65

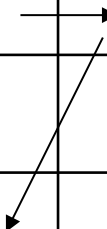
- Provê uma relação ordenada entre as primitivas: sentir, planejar e agir com foco no planejamento
- Desenvolvidos para aplicações específicas
- Algoritmos de *planning* muito pesados. Usualmente requerem linguagens como LISP e PROLOG
- Problemas:
 - Observações dos sensores são fundidas em uma estrutura global (mapa do mundo), onde estão presentes muitas informações irrelevantes, o que dificulta o planejamento
 - “Uma rocha está caindo sobre mim. Para onde devo ir?”

Paradigma reativo

66

□ (1988 - ?)

Primitiva	Entrada	Saída
Sensor	Dados	Observação
-	-	-
Ação	Observação	Comandos aos atuadores

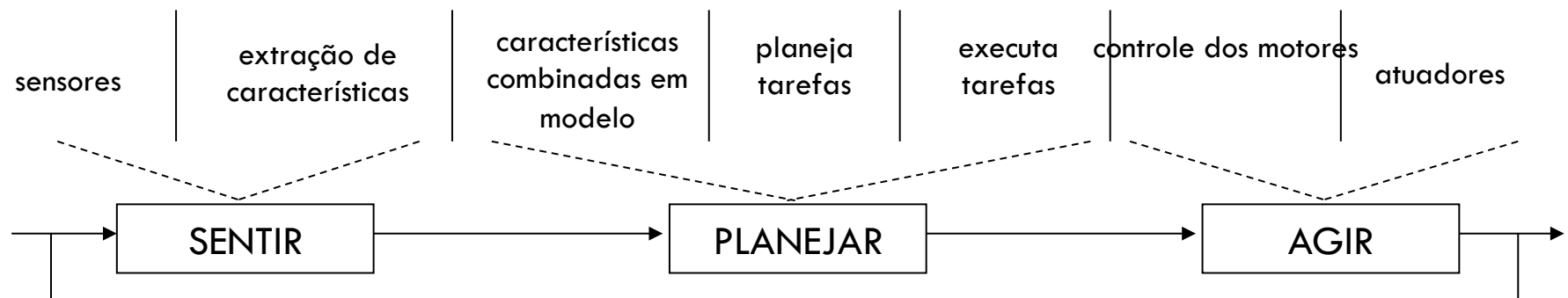


□ Movimento dos pesquisadores de IA investigando biologia, etologia (estudo do comportamento animal) e psicologia cognitiva (estudo de como humanos pensam e representam conhecimento)

Paradigma hierárquico: decomposição

67

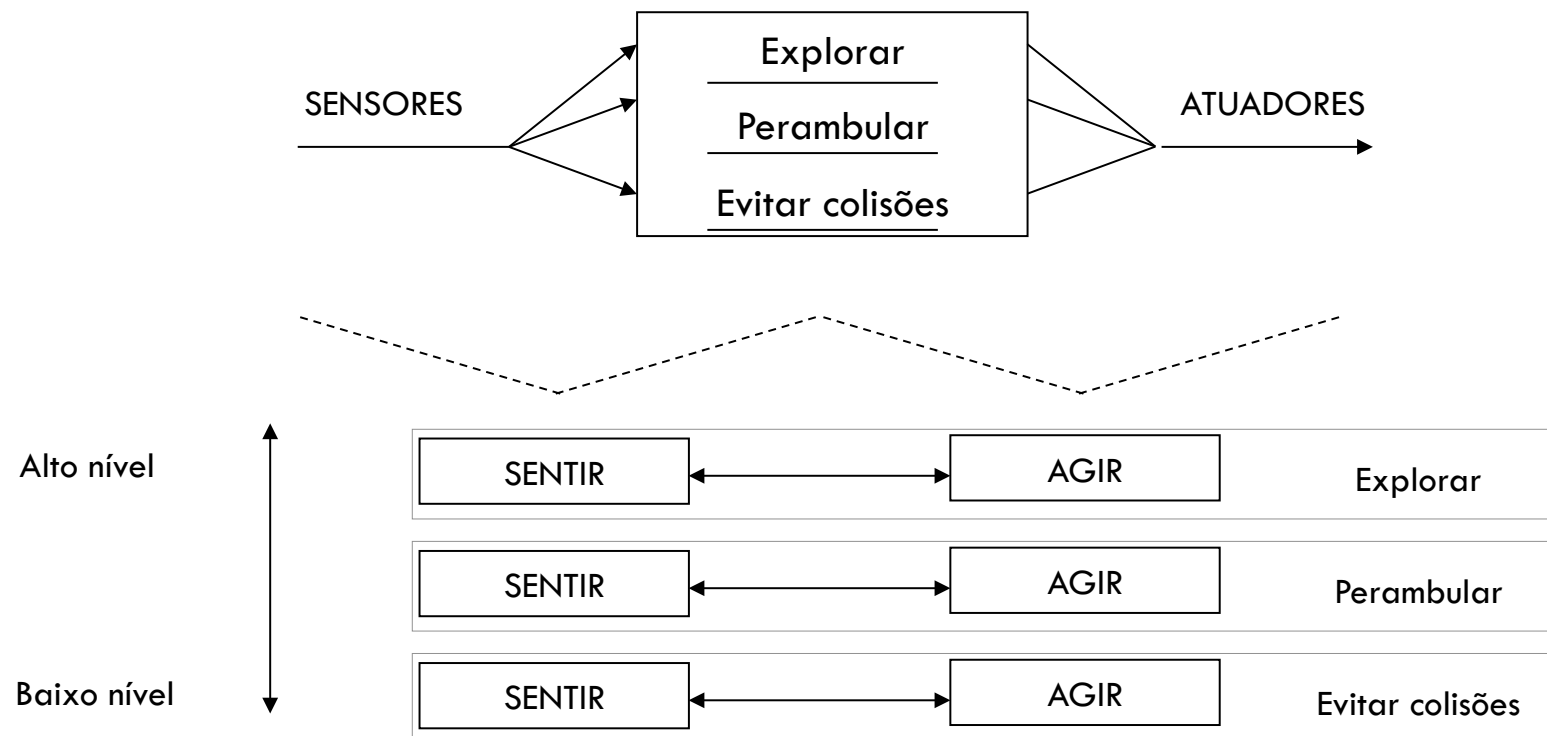
□ Decomposto horizontalmente:



Paradigma reativo: decomposição

68

- A literatura etológica sugere que a inteligência seja formada em camadas verticais:



Paradigma reativo: características

69

- **Comportamento:**
 - ▣ Mapeamentos diretos de entradas sensórias para padrões de ações motoras que são então usadas para realizar uma tarefa
 - ▣ Do ponto de vista matemático: função de transferência que transforma entradas sensoriais em comandos para atuadores

- Camadas provêm comportamentos que vão de um nível mais baixo a um mais alto;
- Se algo acontecer com um comportamento, outros ainda podem operar;
- Os comportamentos são independentes (um comportamento não sabe o que outro faz ou percebe);
- Paradigma reativo não é propício a provas matemáticas demonstrando que alguns comportamentos são suficientes e corretos para uma tarefa;

Paradigma reativo: características (II)

70

- Características compartilhadas por quase todas as arquiteturas reativas:
 - ▣ Robôs são agentes situados operando em um nicho ecológico;
 - ▣ Comportamentos servem como os blocos básicos de construção das ações do robô, e o comportamento resultante do robô é emergente;
 - ▣ Apenas sensoriamento específico para o comportamento é permitido;
 - ▣ Segue inerentemente bons princípios de projeto de software;
 - ▣ Modelos de comportamentos de animais são frequentemente citados como a base para esse sistema ou um comportamento particular.

Arquiteturas

71

- Devem se preocupar com:
 - ▣ Como ocorre o disparo dos comportamentos;
 - ▣ O que acontece quando múltiplos comportamentos são ativos ao mesmo tempo.

- Arquiteturas mais relevantes:
 - ▣ Subsumption
 - ▣ Campos potenciais

Subsumption: características

72

- Robôs com esta arquitetura foram os primeiros a andar, evitar colisões, escalar sem as pausas “move-pensa-move-pensa” do Shakey
- Sua fama decorre do fato de que comportamentos muito naturais foram produzidos pela arquitetura
- Módulos são agrupados em camadas de competência (camadas mais baixas associadas à sobrevivência, enquanto mais altas orientadas aos objetivos do agente)
- Módulos em uma camada podem inibir ou sobrepor saída de comportamentos de outras camadas
- O uso de estados internos é evitado
- Uma tarefa é completada ativando-se a camada apropriada, que ativa então a menor camada abaixo dela, e assim sucessivamente
- Comportamentos são reflexivos. Não possui memória

Exemplo 1

73

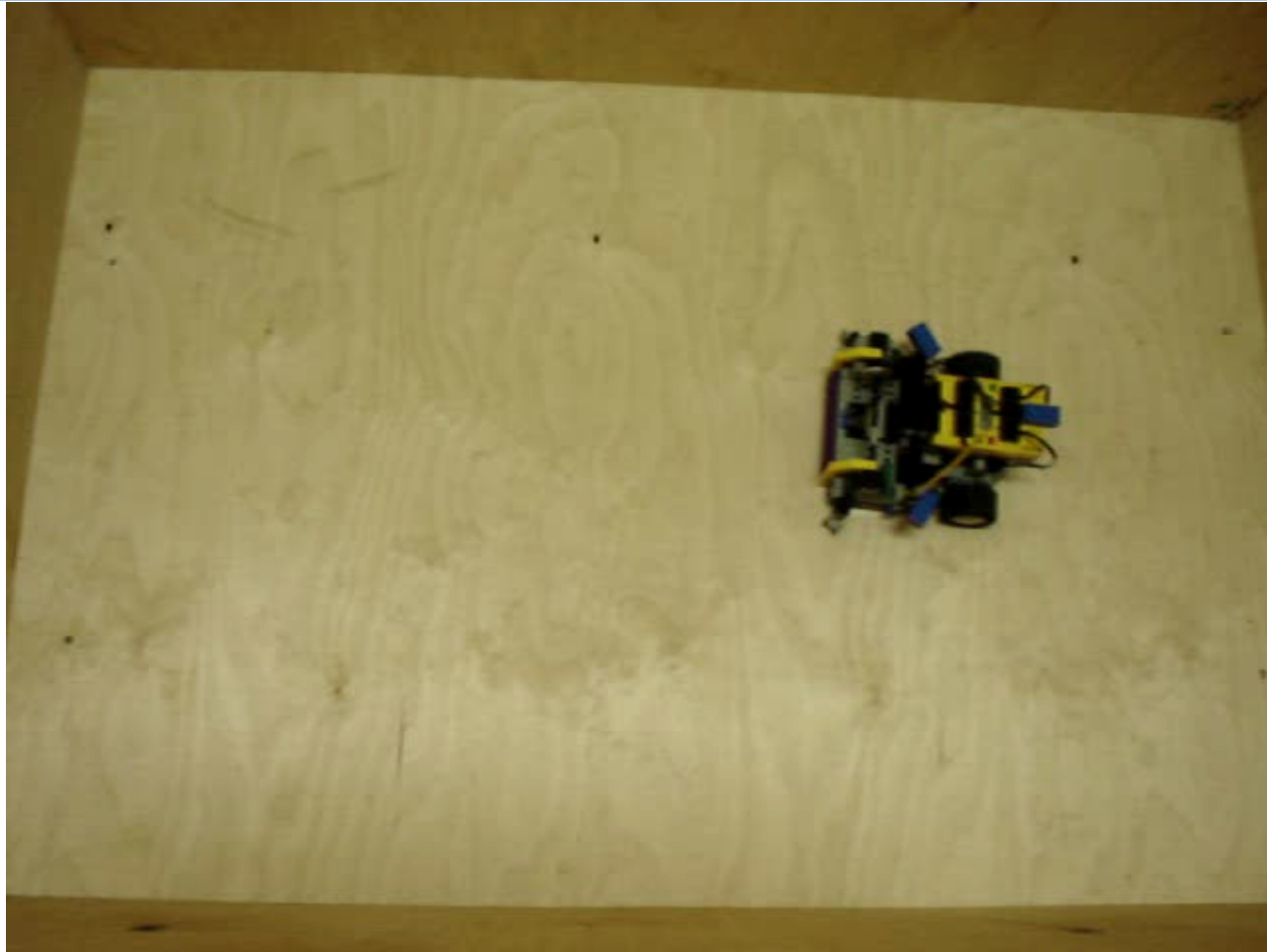


Esther Luna Colombini – Instituto de Computação (Unicamp)

esther@ic.unicamp.br

Exemplo 2

74



Campos potenciais

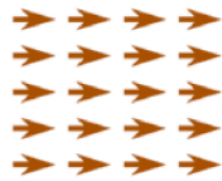
75

- Arquitetura com muitas variantes
 - Comportamentos são modelados usando vetores para representá-los
 - Combinação de vetores produz comportamento emergente
 - Ações motoras de um comportamento representadas na forma de um campo potencial
- Abordagem requer:
 - Que todos os comportamentos sejam implementados como campos potenciais
 - Que a combinação dos comportamentos seja feita não pela soma, mas pela soma vetorial

Campos potenciais básicos

76

- Cinco tipos básicos de campos potenciais que podem ser combinados para construir campos mais complexos: uniforme, perpendicular, atrativo, repulsivo e tangencial



a



b



c



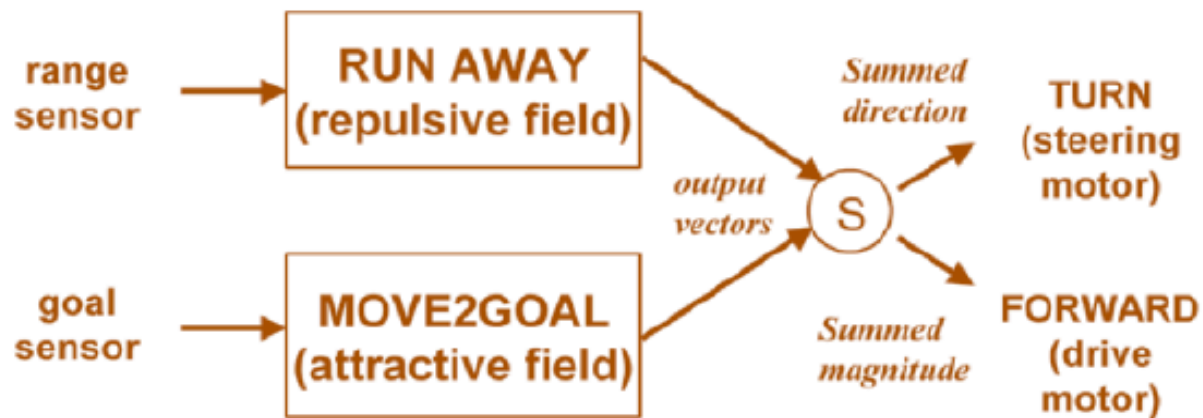
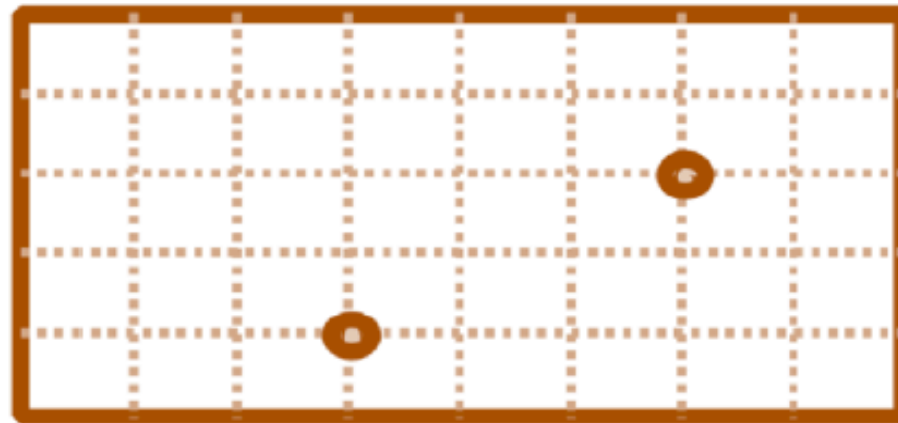
d



e

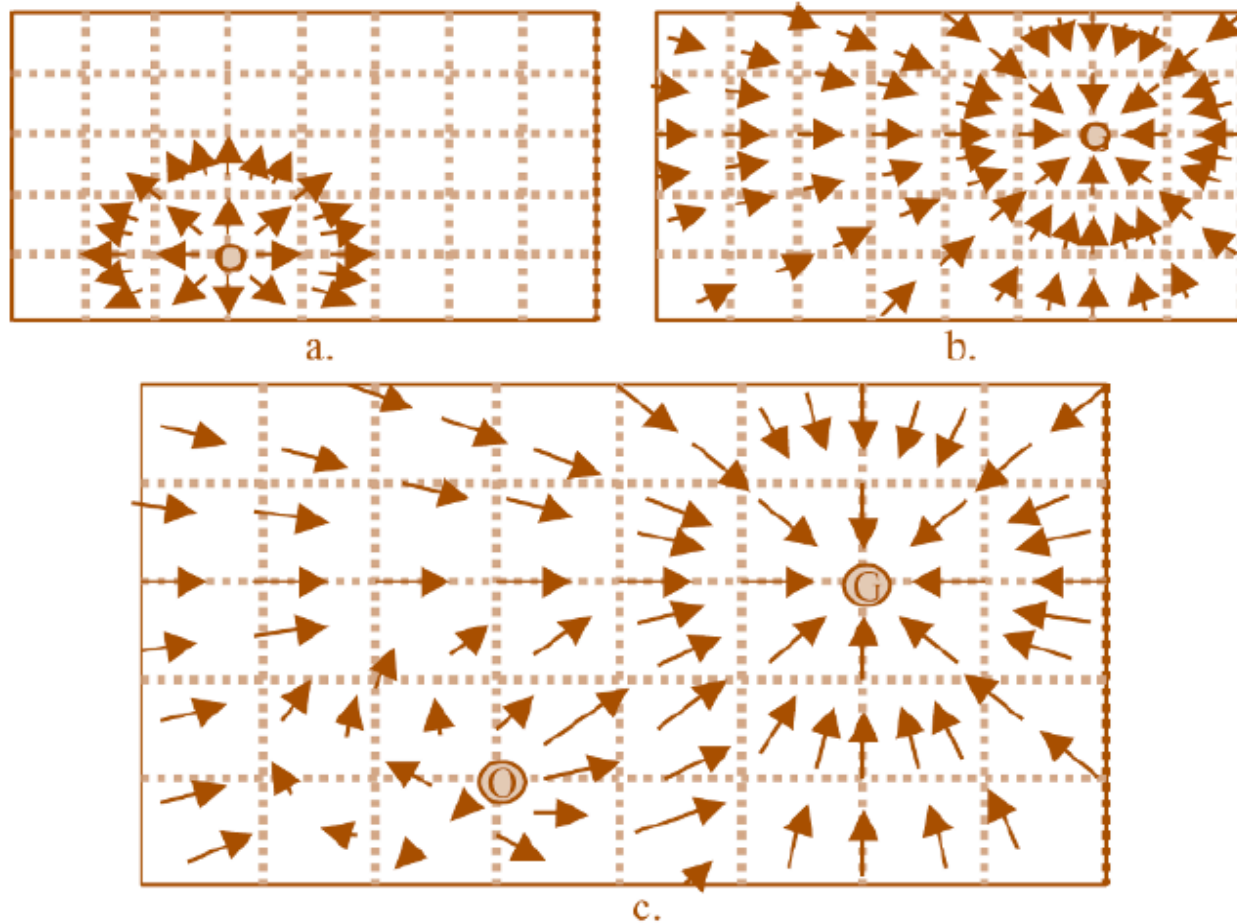
Combinação de campos

77



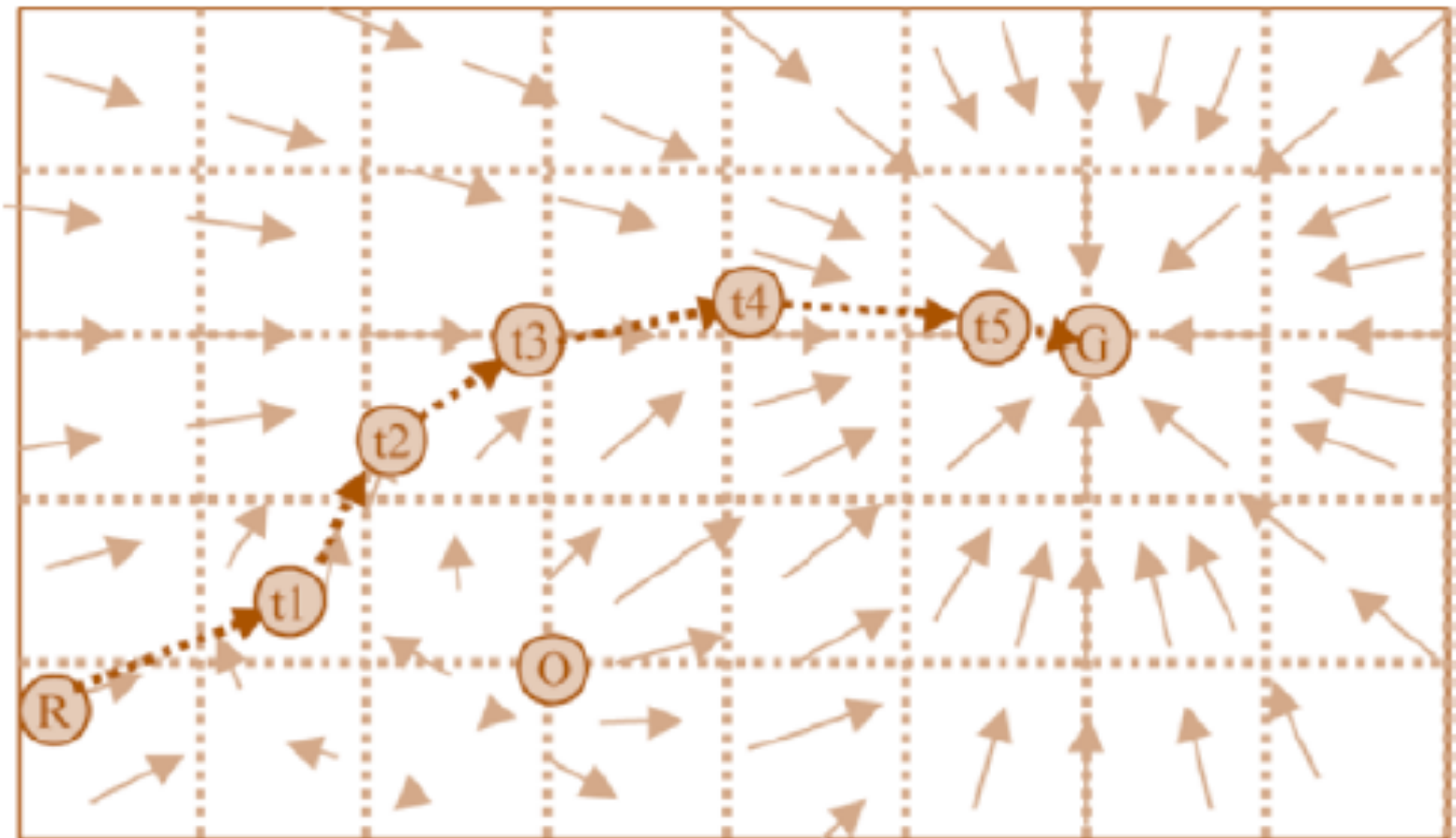
Combinação de campos (II)

78



Combinação de campos (III)

79



Campos potenciais: avaliação

80

- Problemas em regiões com campo nulo
 - ▣ Idealmente deveriam ser apenas no objetivo, mas podem ocorrer em outros pontos
 - ▣ Pode ser resolvido adicionando pequeno grau de ruído aleatório aos vetores de direção
- Problemas com magnitude dos campos
 - ▣ Dependendo das características do terreno do ambiente, escolhas ruins dos padrões de magnitude dos campos podem levar a comportamentos desastrosos em certos caminhos

Subsumption x Campos potenciais

81

- Subsumption
 - Projeto modular, onde cada comportamento pode ser independentemente testado
 - Não voltado à tarefa
- Campos potenciais
 - Tem uma representação de fácil visualização em uma região grande
 - Aceita várias formas de descrição do campo potencial
 - É portátil
 - Problemas de campo nulo

Paradigma reativo: avaliação

82

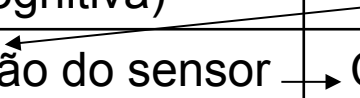
- Vantagens:
 - ▣ Executam em tempo real
 - ▣ Robôs com comportamentos simples podem realizar tarefas relativamente complexas
 - ▣ Não requer representação do estado interno, o que simplifica sua implementação computacional, tornando-os particularmente interessantes para sistemas de baixo custo
 - ▣ Executam bem em ambientes dinâmicos
- Desvantagens:
 - ▣ Dividir o comportamento em comportamentos simples não é trivial
 - ▣ É difícil prever o comportamento emergente
 - ▣ Só podem realizar tarefas que envolvam comportamentos reflexivos (não possuem planejamento e raciocínio)

Paradigma híbrido

83

(1990-?)

Primitiva	Entrada	Saída
Planejamento	Informação (do sensor ou cognitiva)	Diretivas
Sensor-Ação	Observação do sensor	Comandos aos atuadores



- Procura fundir as vantagens dos robôs deliberativos com a simplicidade dos robôs reativos;
- Robô planeja (de forma deliberativa) como cumprir sua missão (usando um modelo global do mundo) e então adota um conjunto de comportamentos (reativos) mais adequados para cumprir seu objetivo (ou parte dele)
- Estabelece uma organização *planejar, sentir-agir*.

Arquiteturas: características

84

- Focam em questões como:
 - ▣ Como distinguir entre reação e deliberação?
 - ▣ Como organizar responsabilidades na porção declarativa?
 - ▣ Como o comportamento geral emerge?

- Componentes comuns:
 - ▣ Sequenciador: gera conjunto de comportamentos em uma ordem
 - ▣ Gerenciador de recursos: aloca recursos a comportamentos
 - ▣ Cartógrafo: cria, ordena e mantém mapa ou informação espacial
 - ▣ Planejador da missão: operacionaliza os comandos em termos do robô
 - ▣ Analisador de desempenho e Resolvedor de problemas: permite que o robô saiba se está ou não fazendo progressos

Arquiteturas: famílias

85

- Famílias de arquiteturas:
 - **Administrativa:** foca na subdivisão da porção deliberativa em camadas baseadas na responsabilidade administrativa de cada função deliberativa
 - Exemplo: AuRA
 - **Hierarquias de estado:** usa o conhecimento do estado do robô para distinguir entre atividades reativas e deliberativas
 - Exemplo: 3-Tiered (3T)
 - **Orientado a modelo:** caracteriza-se por comportamentos que têm acesso a porções do modelo do mundo
 - Exemplo: Saphira

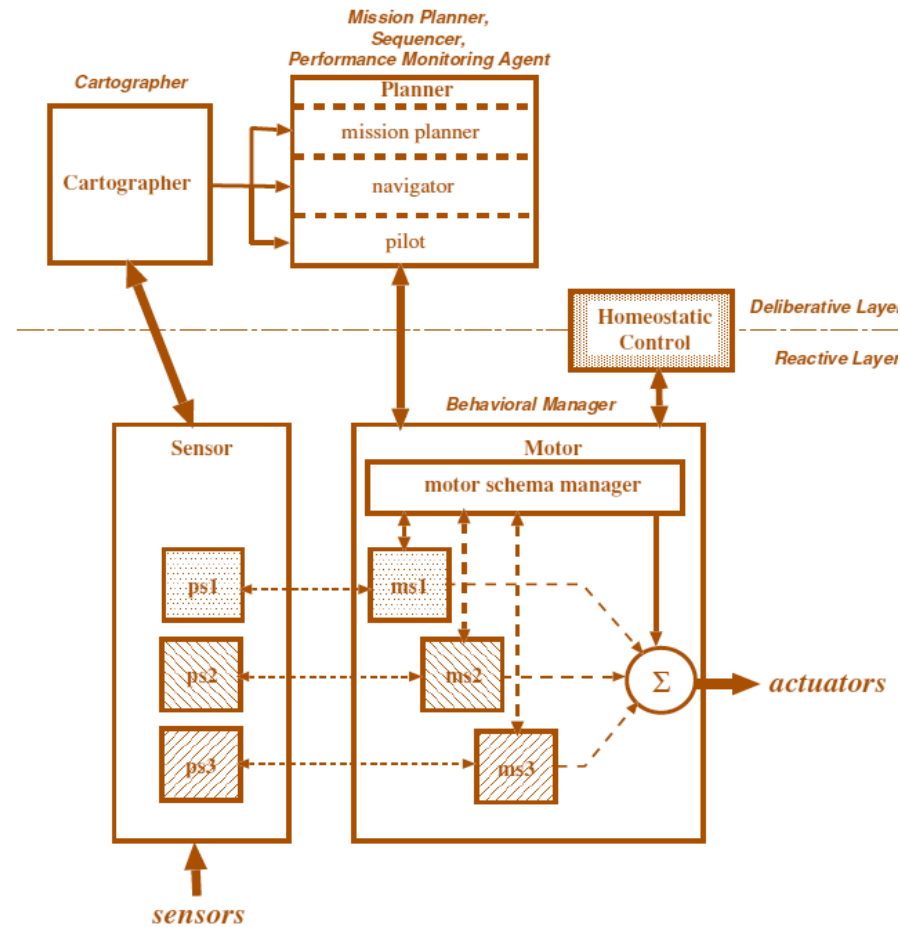
AuRA

86

- Autonomous Robot Architecture (AuRA)
 - ▣ Cinco subsistemas (equivalentes a classes orientadas a objeto)
 - Dois subsistemas deliberativos: planejador e cartógrafo
 - Dois subsistemas reativos: sensório e motor (sistema motor implementado com campos potenciais)
 - Quinto subsistema: controle homeostático, modifica as relações entre comportamentos alterando ganhos

AuRA (II)

87



Paradigma híbrido: avaliação

88

- A integração de deliberação e controle reativo deve ser levada em conta pelo projetista para produzir um sistema flexível, robusto e inteligente
- A integração de deliberação e controle reativo é um problema complexo
 - ▣ Se tentar modelar e pré-planejar todas as eventualidades, corre-se o risco de que o processo de planejamento nunca termine
 - ▣ Por outro lado, não é seguro para o robô fazer suposições grosseiras sobre o mundo, que não reflitam a sua natureza dinâmica

Paradigmas: Conclusão

89

- A evolução do hardware nas últimas décadas tem permitido realizar cada vez mais planejamento. Contudo, a explosão combinacional típica dos problemas não permite sua utilização frequente em problemas de tempo real
- A maior parte das aplicações atuais é voltada para variantes do paradigma híbrido
- Dinâmica do ambiente tem sido tratada por sistemas adaptativos autônomos

Atividades extra-classe

90

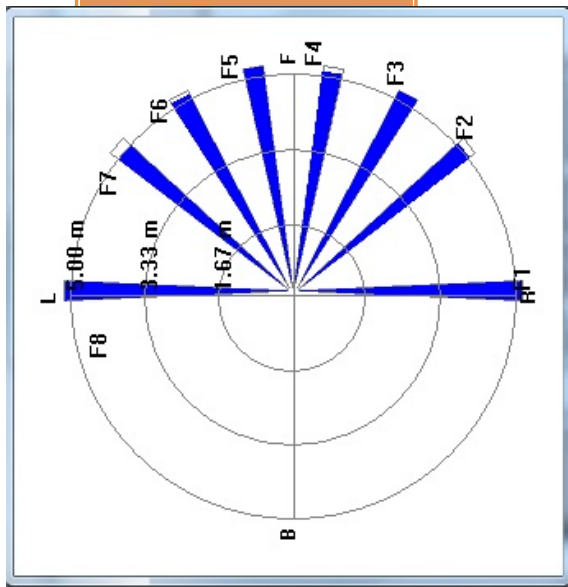
□ Atividades

□ Leitura:

- <http://geology.heroy.smu.edu/~dpa-www/robo/subsumption/>

□ Exercício:

- Considere o material apresentado no site acima. Defina como poderíamos implementar, segundo o paradigma reativo e a arquitetura subsumption, um sistema para controlar um robô que tem como objetivo encontrar uma maçã no ambiente. Além disso, é importante que o robô não colida com os obstáculos da cena. Assuma um robô com 8 sensores de distância distribuídos conforme a figura ao lado (distribuição em torno do robô) e que o mesmo possui um sensor do tipo câmera especial que detecta maçãs e que está posicionado no seu centro, direcionado para frente. Defina que comportamentos seriam necessários, com entradas e saídas, e como estes seriam coordenados na arquitetura de referência. Discuta a sua proposição em uma página.



Referências

91

- MURPHY, R. R. **Introduction to AI robotics**. MIT Press, 2002.
- DUDEK, G.; JENKIN, M. **Computational Principles of mobile robotics**. Cambridge Press, 2000.
- ROMERO, R. A. F.; PRESTES, E.; OSÓRIO, F.; WOLF, D. (Orgs) **Robótica móvel**. LTC, 2014.
- BROOKS, R. **Intelligence without representation**. Artificial Intelligence, 47:139-159, 1991.
- RUSSEL, S. NORVIG, P. **Artificial Intelligence: a modern approach**. Prentice Hall, 2002.
- BRATKO, I. **PROLOG: programming for artificial intelligence**. Addison Wesley, 2nd edition, 1990.
- <http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionListCategory.htm>