

GT-JeDi - Curso de Desenv. de Jogos

IA para Jogos



Fernando Osório

2006/2 - A04

IA para Jogos



Tópicos abordados... Revisão

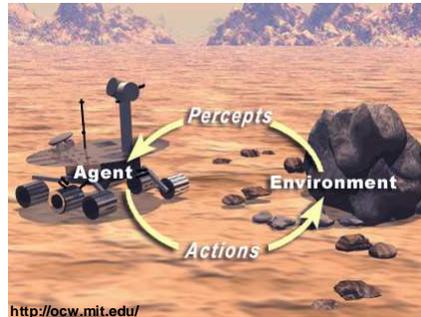
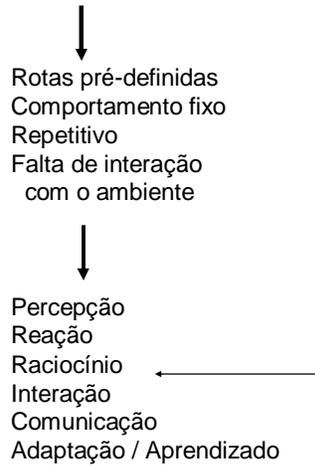
- Introdução à Inteligência Artificial
- IA clássica: **Jogos de Raciocínio**
 - Solução de problemas
 - Jogos de Tabuleiro (*Board games*)
 - Busca em Espaço de Estados (*Trees and adversarial search*)
- IA clássica: **Jogos de Ação em Labirintos**
 - Busca de caminhos (*Path finding*)
 - Planejamento de ações e trajetórias (*Path planning*)
 - Deslocamento / Navegação (*Motion & Navigation*)

Agentes Inteligentes



➤ Agentes Inteligentes

- “Falsa IA” versus “Comportamento inteligente”

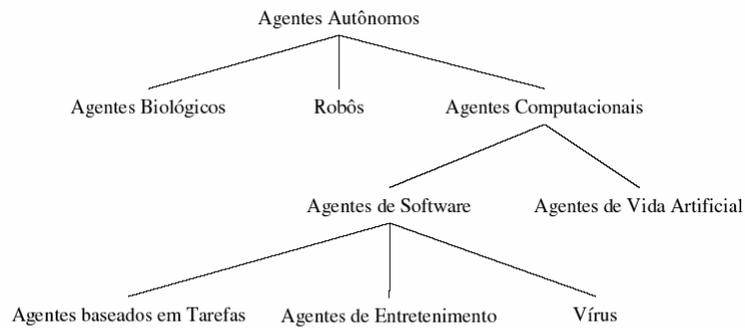


Arquitetura de Controle
Agentes Autônomos
Agentes Inteligentes

Agentes Inteligentes



➤ Agentes Inteligentes - Taxonomia

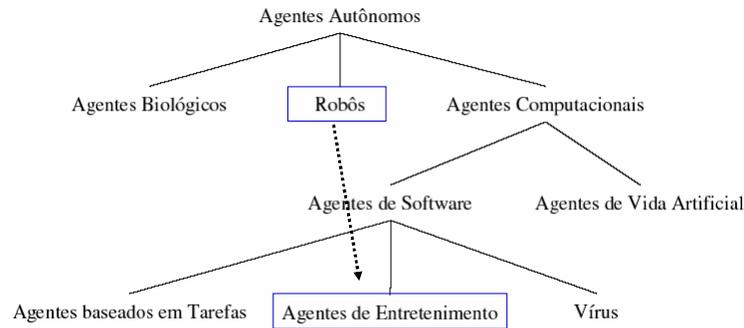


Taxonomia para agentes autônomos [FRA 97]

Agentes Inteligentes



➤ Agentes Inteligentes - Taxonomia



Taxonomia para agentes autônomos [FRA 97]

Sugestão de leitura complementar:

- Dissertação de Mestrado: Farlei Heinen – Sistema de Controle Híbrido para Robôs Móveis Autônomos
- Dissertação de Mestrado: Cássia dos Santos – Ambiente Virtual Inteligente e Adaptativo
- Trabalho de Conclusão: João Bittencourt – Ambiente para Simulação de Múltiplos Agentes Autônomos
- Pesquisas da Profa. Soraia Musse – Proj. CROMOS e HUMUS (<http://www.inf.unisinos.br/~cglab/>)

Agentes Inteligentes



➤ Agentes Inteligentes – Arquiteturas de Controle

- Agentes Cognitivos / Deliberativos
- Agentes Reativos
- Agentes com Arquitetura Hierárquica
- Agentes com Arquitetura Híbrida

➤ Agentes Inteligentes – Interação

- Interação com o Ambiente
- Interação com os outros Agentes

➤ Agentes Inteligentes – Conhecimento

- Sem Memória / Sem Representação Interna
- Estático / Inicial
- Dinâmico / Adquirido

Agentes Inteligentes



➤ Agentes Inteligentes – Arquiteturas de Controle

- **Agentes Cognitivos / Deliberativos - AgCog**
 - Modelo Simbólico e Explícito do Ambiente (mapa)
 - Conhecimento "Total" (posição, obstáculos, etc)
 - Estado Interno
 - Processo Decisório
 - Plano de Ações ⇒ Sequência de Ações
- Agentes Reativos - AgReact
- Agentes com Arquitetura Hierárquica
- Agentes com Arquitetura Híbrida

Agentes Inteligentes



➤ Agentes Inteligentes – Arquiteturas de Controle

- **Agentes Cognitivos / Deliberativos - AgCog**
 - Modelo Simbólico e Explícito do Ambiente (mapa)
 - Conhecimento "Total" (posição, obstáculos, etc)
 - Estado Interno
 - Processo Decisório
 - Plano de Ações ⇒ Sequência de Ações
- **Agentes Reativos - AgReact**
- Agentes com Arquitetura Hierárquica
- Agentes com Arquitetura Híbrida



Agentes Inteligentes



➤ Agentes Inteligentes – Arquiteturas de Controle

- **Agentes Cognitivos / Deliberativos - AgCog**
 - Modelo Simbólico e Explícito do Ambiente (mapa)
 - Conhecimento "Total" (posição, obstáculos, etc)
 - Estado Interno
 - Processo Decisório
 - Plano de Ações ⇒ Sequência de Ações

- **Agentes Reativos - AgReact**
- Agentes com Arquitetura Hierárquica
- Agentes com Arquitetura **Híbrida**



Agentes Cognitivos

? Puros ?

Agentes Reativos

Agentes Inteligentes



➤ Agentes Inteligentes – Arquiteturas de Controle

- **Agentes Cognitivos / Deliberativos - AgCog**
 - Modelo Simbólico e Explícito do Ambiente (mapa)
 - Conhecimento "Total" (posição, obstáculos, etc)
 - Estado Interno
 - Processo Decisório
 - Plano de Ações ⇒ Sequência de Ações
- **Formalismos / Ferramentas Adotadas - AgCog**
 - Autômatos:
 - FSA / FSM – Finite-State Autômata
 - HFSA – Hierarchical Finite State Autômata
 - RdP – Redes de Petri (Hierarchical, Colored, Temporalized)
 - Baseados em Regras:
 - RBS – Rule Based System
(Facts, Rules and Inference)

Agentes Inteligentes



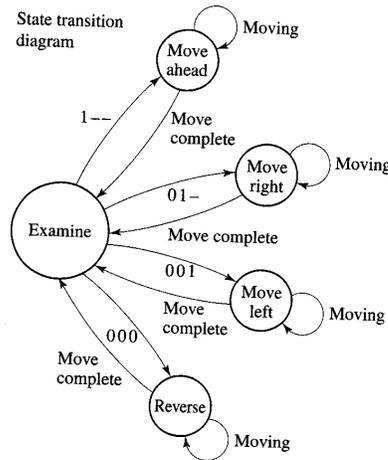
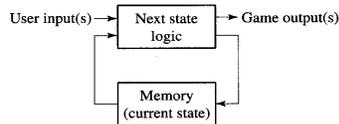
Agentes Inteligentes – Ferramentas

• AgCog - Autômato

Rules

Ahead	Right	Left	Action
Open	-	-	Move ahead
Blocked	Open	-	Move right
Blocked	Blocked	Open	Move left
Blocked	Blocked	Blocked	Reverse direction

implementation *if* ahead = open *then* move_ahead
else if right = open *then* move_right
else ...



3D Games
Watt & Policarpo

Agentes Inteligentes



Agentes Inteligentes – Ferramentas: AgCog - Autômato FSA

- Autômatos finitos – FSA (*Finite-State Automata*): Os NPCs podem ter um comportamento que simula inteligência, mas que na realidade é bastante fixo e pré-definido, através do uso de um autômato. Esta é uma das técnicas de controle mais utilizadas nos jogos na atualidade. Os autômatos definem uma seqüência de estados e de condições para passagem de um estado a outro, fazendo com que o NPC aparentemente execute ações coerentes. Veja um exemplo na Figura 1.12.

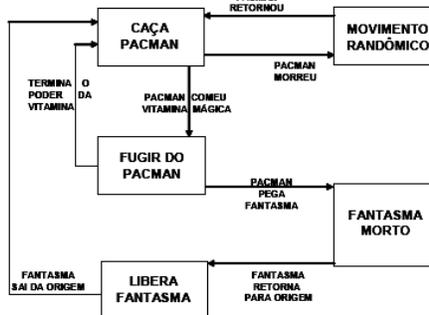


Figura 1.12. Exemplo de um autômato (FSA) usado para o controle do Fantasma no jogo do Pac-Man [adaptado de SCH 04]

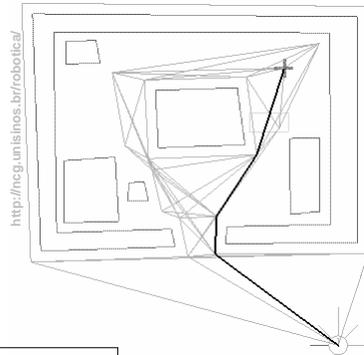
Motores (ERI-MG)
Bittencourt & Osório

Agentes Inteligentes



➤ Agentes Inteligentes – Ferramentas

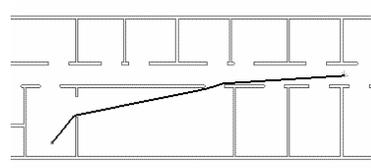
• AgCog – Autômato



SimRob2D

Planejamento:
* Mapa da Geometria do Ambiente
* Grafo de Visibilidade
* Caminho Ótimo (Dijkstra)

Execução:
* Controle Deliberativo / Autômato

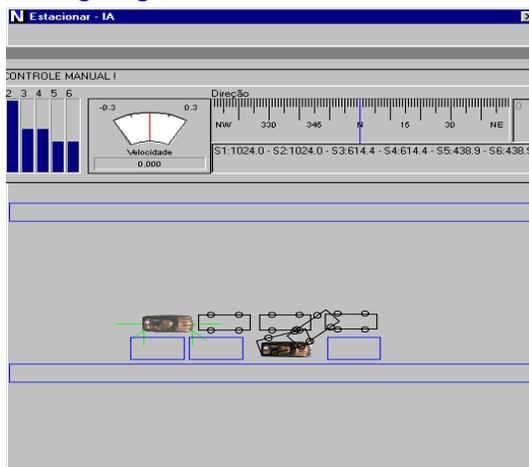


Agentes Inteligentes



➤ Agentes Inteligentes – Ferramentas

• AgCog - Autômato



SEVA - Simulador de Estacionamento de Veículos Autônomos (F. Heinen / PIPCA)

CONTROLE:

SEVA-H
Seva Humano

SEVA-A *Autômato Com Sensores*
Seva Autômato

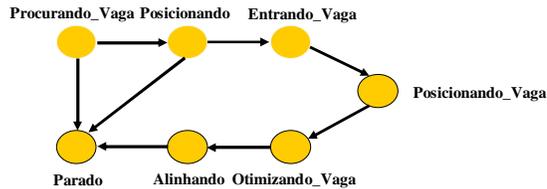
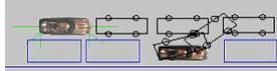
SEVA-N
Seva Neural

Agentes Inteligentes



➤ AgCog – Ferramentas: Autômato

SEVA - Simulador de Estacionamento de Veículos Autônomos (F. Heinen / PIPCA)



```

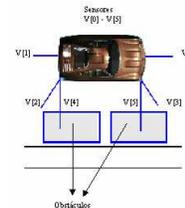
Se Estado_Atual(Procurando_Vaga) e Próximo_ao_Obstáculo(V[4]) e
  Próximo_ao_Obstáculo(V[5])
Então Speed = Avanço_Rápido e RotVel = Direção_Reta;

Se Estado_Atual(Procurando_Vaga) e Longe_do_Obstáculo(V[2])e
  Longe_do_Obstáculo(V[3]) e Longe_do_Obstáculo(V[4]) e
  Longe_do_Obstáculo(V[5])
Então Troca_Estado(Posicionando) e Inicializa(Odômetro);

Se Estado_Atual(Posicionando)
Então Speed = Avanço_Rápido e RotVel = Direção_Reta;

Se Estado_Atual(Posicionando) e Longe_do_Obstáculo(V[4]) e
  Deslocamento_Suficiente(Odômetro)
Então Estado_atual(Entrando_Vaga) e Inicializa(Odômetro);

Se Estado_Atual(Entrando_Vaga)
Então Speed = Ré_Rápida e RotVel = Giro_Esquerda_Max;
    
```

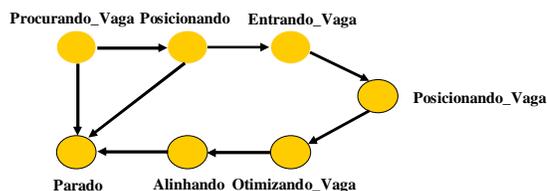
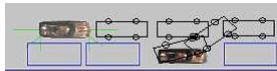


Agentes Inteligentes



➤ AgCog – Ferramentas: Autômato

SEVA - Simulador de Estacionamento de Veículos Autônomos (F. Heinen / PIPCA)



```

Se Estado_Atual(Procurando_Vaga) e Próximo_ao_Obstáculo(V[4]) e
  Próximo_ao_Obstáculo(V[5])
Então Speed = Avanço_Rápido e RotVel = Direção_Reta;

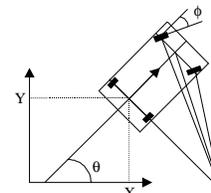
Se Estado_Atual(Procurando_Vaga) e Longe_do_Obstáculo(V[2])e
  Longe_do_Obstáculo(V[3]) e Longe_do_Obstáculo(V[4]) e
  Longe_do_Obstáculo(V[5])
Então Troca_Estado(Posicionando) e Inicializa(Odômetro);

Se Estado_Atual(Posicionando)
Então Speed = Avanço_Rápido e RotVel = Direção_Reta;

Se Estado_Atual(Posicionando) e Longe_do_Obstáculo(V[4]) e
  Deslocamento_Suficiente(Odômetro)
Então Estado_atual(Entrando_Vaga) e Inicializa(Odômetro);

Se Estado_Atual(Entrando_Vaga)
Então Speed = Ré_Rápida e RotVel = Giro_Esquerda_Max;
    
```

$$\begin{aligned}
 X &= V * \cos(\Phi) * \cos(\theta) & \text{Eq. 1} \\
 Y &= V * \sin(\Phi) * \cos(\theta) & \text{Eq. 2} \\
 \theta &= V / L * \sin(\Phi) & \text{Eq. 3}
 \end{aligned}$$

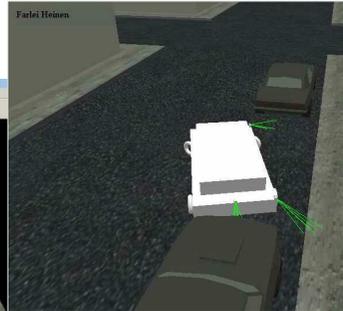
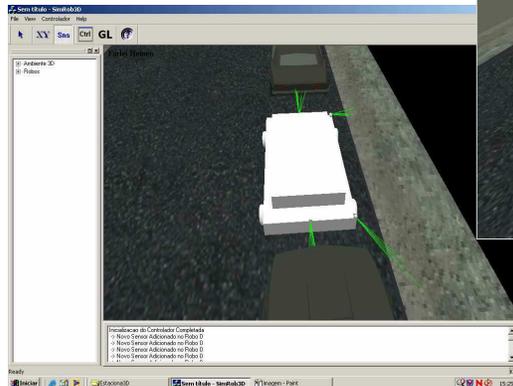


Agentes Inteligentes



➤ AgCog – Ferramentas: Autômato

SEVA 3D - Simulador de Estacionamento de Veículos Autônomos (F. Heinen / PIPCA)



Agentes Inteligentes



➤ Agentes Inteligentes – Ferramentas

• AgCog – HSFA (Hierarchical Finite State Automata)

5.1.2 Hierarchical Finite-state Machines (HFSM)

Finite-state machines (FSMs) consist of a set of states (including an initial state), a set of inputs, a set of outputs, and a state transition function. The state transition function takes the input and the current state and returns a *single* new state and a set of outputs. Since there is only one possible new state, FSMs are used to encode deterministic behavior. It is commonplace, and convenient, to represent FSMs with *state transition diagrams*. A state transition diagram uses circles to represent the states and arrows to represent

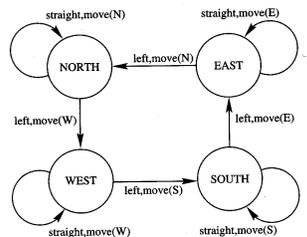


Figure 5.1. The WhichDir FSM.

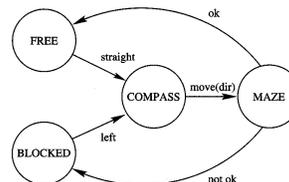


Figure 5.2. HFSM that uses the WhichDir FSM.

the transitions between states. Figure 5.1 depicts an FSM that keeps track of which compass direction a character is heading each time it turns "left."

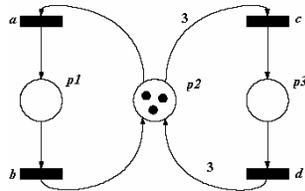
As the name implies, an HFSM is simply a hierarchy of FSMs. That is, each node of an HFSM may itself be an FSM. Just like functions and procedures in a regular programming language, this provides a convenient way to make the design of an FSM more modular. For example, if a character is at coordinates (x, y) , Figure 5.2 depicts an HFSM that uses the FSM in Figure 5.1 as a submodule to calculate the new cell after turning "left," or moving one cell ahead.

Agentes Inteligentes

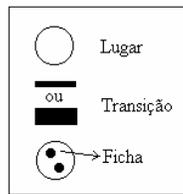


➤ Agentes Inteligentes – Ferramentas

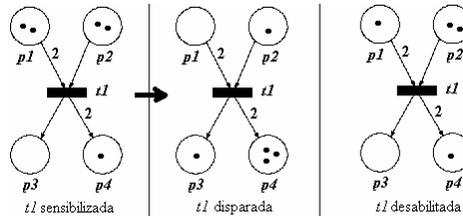
- AgCog – RdP (Rede de Petri)



Um ambiente de modelagem, simulação e controle de processos de manufaturas baseado em Redes de petri coloridas
 Marcos Lemke Ribes
 (Unisinos / TC 2003)



Disparo de Transição



Agentes Inteligentes



➤ Agentes Inteligentes – Ferramentas

- AgCog – RBS (Rule Based System)

RoboForge:
www.roboforge.com



Agentes Inteligentes

RBS (Rule Based System)
RoboForge

<http://www.robforge.com/>

The screenshot shows the RoboForge 'workshop' interface. At the top, there are tabs for 'design', 'moves', 'a.i.', 'profile', and 'test', with 'a.i.' selected. The main area is divided into a 'Nodes' panel on the left and a 'blocks' panel on the right. The 'Nodes' panel shows a flowchart of rules: Start -> Check Zones -> Front is active? -> Frontal Attack -> Find Enemy -> angle in range 5.0° - 355.0° -> Orient Facing only -> The distance to my enemy > 6? -> Distance only. The 'blocks' panel shows a 'decision' block for 'angle (°)' with a circular diagram and the text 'My angle to my enemy' and 'angle range: 5.0° - 355.0°'. Below the diagram is the question 'Is my angle to my enemy in the range 5.0° - 355.0°?'. On the right, there are three 3D renderings of a robot in a dark environment. At the bottom, it says 'A.I. panel activated.' and 'cost: 34,760 creds weight: 3,941 kg health: 22,767 hp'.

Agentes Inteligentes



- Agentes Inteligentes – Ferramentas
 - AgCog – RBS (Rule Based System)

This block contains three images related to the Robocode game. Top left: A close-up of a robot with the number '77.0' and the word 'Fire' below it. Top right: A wide shot of the Robocode arena with the word 'Robocode' in large green letters in the center. Bottom left: A screenshot of the Robocode game window showing a top-down view of the arena with a blue beam of fire from one robot to another. Bottom right: An 'About Robocode' dialog box with the following text: 'Robocode Version: 1.0.6 (c) Copyright 2001 IBM robocode.alphaworks.ibm.com Designed and Programmed by Mathew A. Nelson Graphics by Garrett S. Hourihan Inspired by Brad Schick's excellent game RobotBattle, at http://www.robotbattle.com You are using Java version 1.4.0 by Sun Microsystems Inc. OK'.

Agentes Inteligentes



- Agentes Inteligentes – Ferramentas
 - AgCog – RBS (Rule Based System)



```
package sample;
import robocode.*;
/**
 * MyFirstRobot - a sample robot by Mathew Nelson
 * Moves in a seesaw motion, and spins the gun around at each end
 */
public class MyFirstRobot extends Robot
{
    /** MyFirstRobot's run method - Seesaw */
    public void run()
    {
        while (true) {
            ahead(100);           // Move ahead 100
            turnGunRight(360);    // Spin gun around

            back(100);           // Move back 100
            turnGunRight(360);    // Spin gun around
        }
    }
}
```

```
/** Fire when we see a robot */
public void onScannedRobot(ScannedRobotEvent e)
{
    fire(1);
}

/**
 * We were hit! Turn perpendicular to the bullet,
 * so our seesaw might avoid a future shot.
 */
public void onHitByBullet(HitByBulletEvent e)
{
    turnLeft(90 - e.getBearing());
}
}
```

Agentes Inteligentes



- Agentes Inteligentes – Ferramentas
 - AgCog – RBS (Rule Based System)

GNU Mages - FuzzyF
<http://www.inf.unisinos.br/~jrbit/>



```
#*****
# Definicoes das variaveis de entrada
#*****
#Terreno - pos. 1 (N)
INPUT_VAR = Terreno_N
TERM = Impossivel ConstantFunction -1.0
TERM = Dificil TriangularFunction -1.0 -0.8 -0.6
TERM = Medio TriangularFunction -0.7 -0.5 -0.3
TERM = Facil RighthTriangleFunction -0.5 0
TERM = Alvo ConstantFunction 1.0
END_VAR

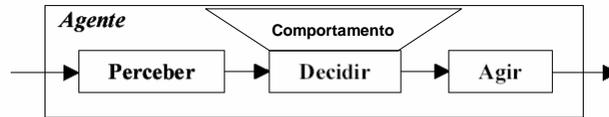
#*****
# Definicoes das Regras - Determinar o status do terreno
#*****
BLOCK_RULES
RL = IF Terreno_N IS Impossivel
    THEN Status_Terreno_N IS Impedido
RL = IF Terreno_N IS Alvo
    THEN Status_Terreno_N IS Impedido
...
RL = IF Direcao IS NO AND NOT Sonar_Dir_Atras IS Nenhum AND
    NOT Status_Terreno_L IS Impedido
    THEN Nova_Direcao IS L CV=0.75
RL = IF Distancia_Bandeira IS Pertissimo
    THEN Acao IS Pegar_bandeira CV=0.97
RL = IF Bot_com_Bandeira IS Sim THEN Acao IS Parar
END_RULES
```

Agentes Inteligentes



➤ Agentes Inteligentes – Ferramentas

- AgReact - Reativo



Arquitetura puramente reativa

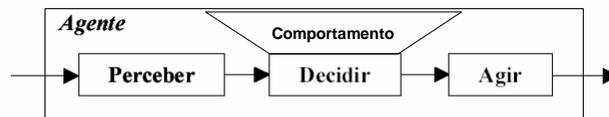
Assim como é difícil ter um agente puramente deliberativo e baseado em uma única ferramenta (FSA, RBS, etc), também é difícil de se ter um agente puramente reativo, sem memória ou estado interno...

Agentes Inteligentes



➤ Agentes Inteligentes – Ferramentas

- AgReact - Reativo



Arquitetura puramente reativa

Assim como é difícil ter um agente puramente deliberativo e baseado em uma única ferramenta (FSA, RBS, etc), também é difícil de se ter um agente puramente reativo, sem memória ou estado interno...

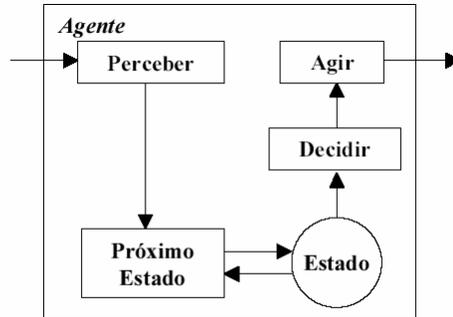
[Agentes Hierárquicos e Híbridos](#)

Agentes Inteligentes



➤ Agentes Inteligentes – Ferramentas

- AgReact – Reativo... com estados



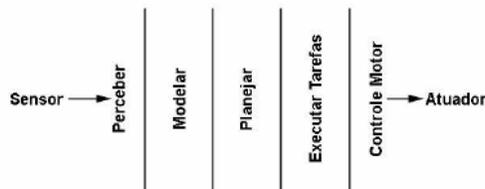
Arquitetura com Estado Interno
sensação de déjà vu :)

Agentes Inteligentes

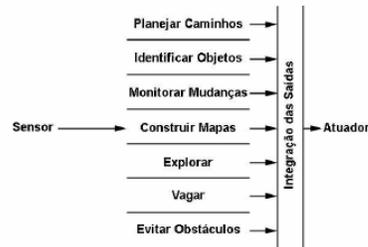


➤ Agentes Inteligentes – Ferramentas

- AgReact – Reativo Hierárquico



Arquitetura Horizontal:
SMPA – Sense, Model, Plan, Act



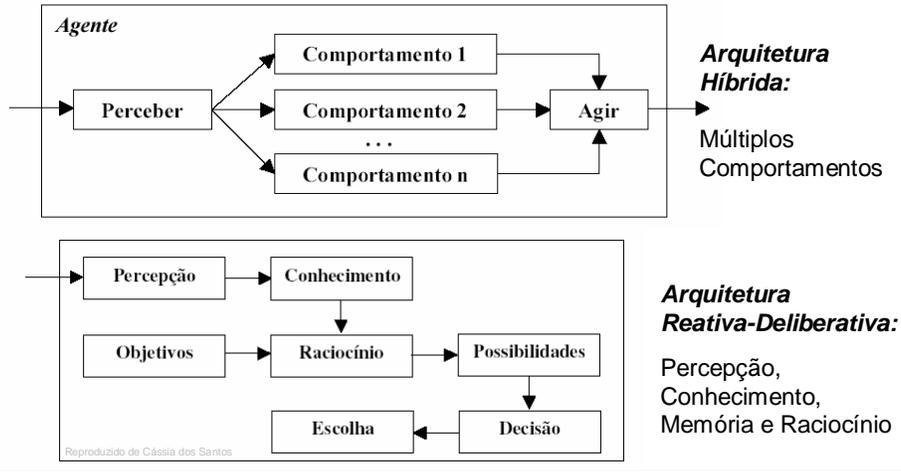
Arquitetura Vertical:
Subsumption - Brooks

Agentes Inteligentes



➤ Agentes Inteligentes – Ferramentas

- Agentes Cognitivos, Reativos e Híbridos

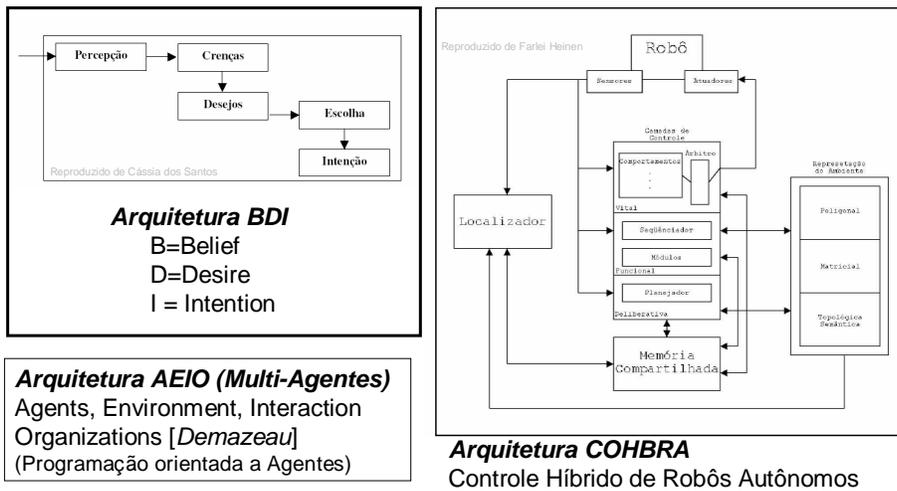


Agentes Inteligentes



➤ Agentes Inteligentes – Ferramentas

- Agentes Cognitivos, Reativos e Híbridos

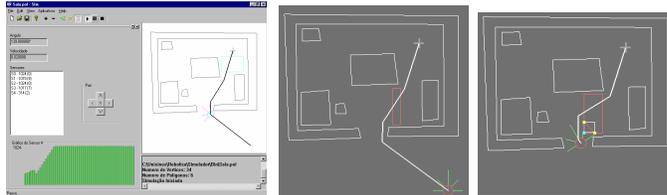


Agentes Inteligentes

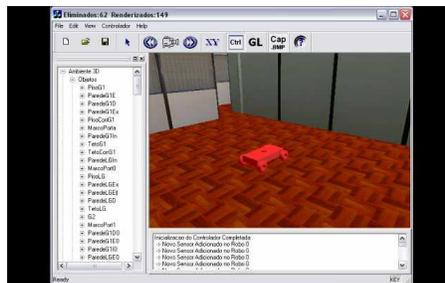
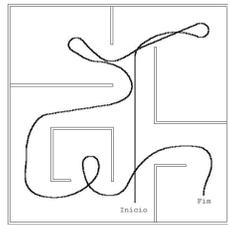


➤ Agentes Inteligentes – Ferramentas: Controle Híbrido

SimRob2D



SimRob3D / COHBRA



Agentes Inteligentes



Representação do Ambiente

Poligonal: Os obstáculos são representados por polígonos. Fornecida pelo usuário. Utilizada principalmente pelo Módulo Localizador.

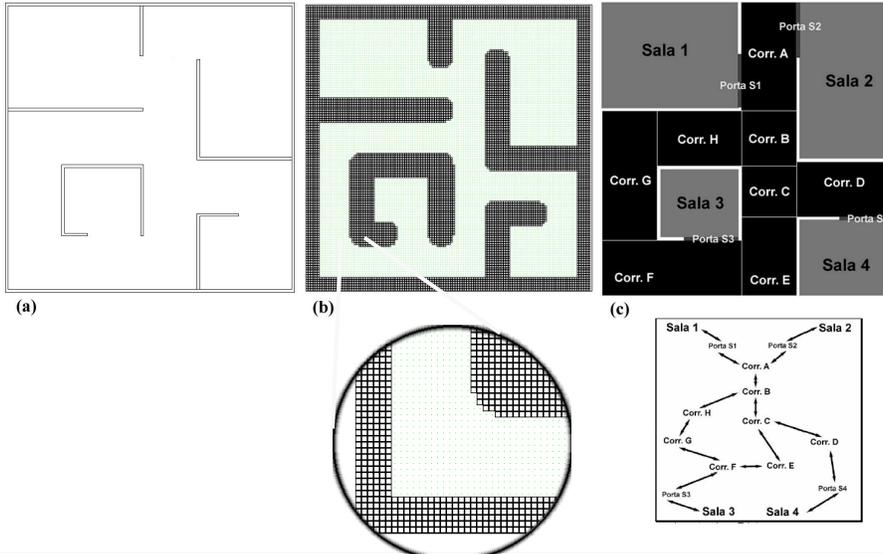
Matricial: Representa o ambiente através de uma matriz. Gerada a partir da representação poligonal. Utilizada principalmente para o planejamento de trajetória.

Topológica/Semântica: Representa as relações topológicas entre diversas áreas do ambiente, e associa a cada área informações semânticas. Fornecida pelo usuário. Utilizada principalmente para otimizar o planejamento de trajetória.

Agentes Inteligentes



Representação do Ambiente



Agentes Inteligentes



Camada Vital

Responsável pelo controle reativo do robô móvel, através de diversos comportamentos primitivos operando em paralelo.

Comportamentos:

- Parar
- Vagar
- Desviar de Obstáculos
- Ir em direção ao Alvo
- Inverter Direção

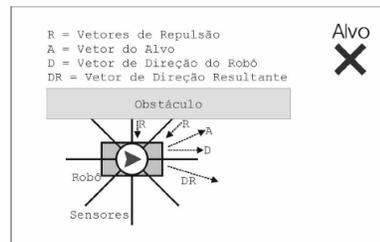


Fig. 6.3 Campos Potenciais

Árbitro: Tem a função de unificar as saídas dos diversos comportamentos em um comando único para os atuadores.

Agentes Inteligentes



Camada Funcional

Autômato: Responsável pelo seqüenciamento dos comportamentos da camada vital.

MMAA – Módulo Monitor de Alterações no Ambiente

Responsável por atualizar a representação do ambiente.

Indica quando ocorre alguma inconsistência e o plano precisa ser recalculado.

MIDA – Módulo Indicador de Direção do Alvo

MMPT – Módulo Monitor de Posição Topológica

Agentes Inteligentes



Camada Funcional

Autômato:
Responsável pelo seqüenciamento dos comportamentos da camada vital.

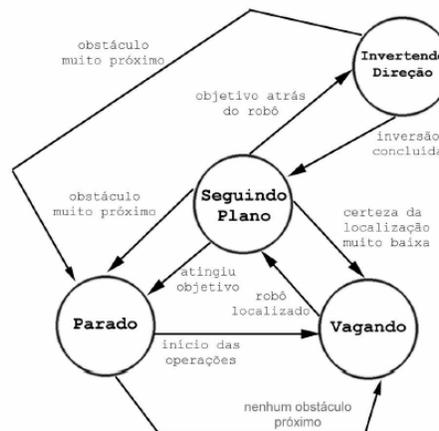


Fig. 6.4 Diagrama de Estados

Agentes Inteligentes



Camada Deliberativa

- Responsável pelo planejamento de trajetória.
- Pré-planejamento utilizando as informações topológicas
- Planejamento final utilizando o **algoritmo A*** na representação matricial do ambiente.

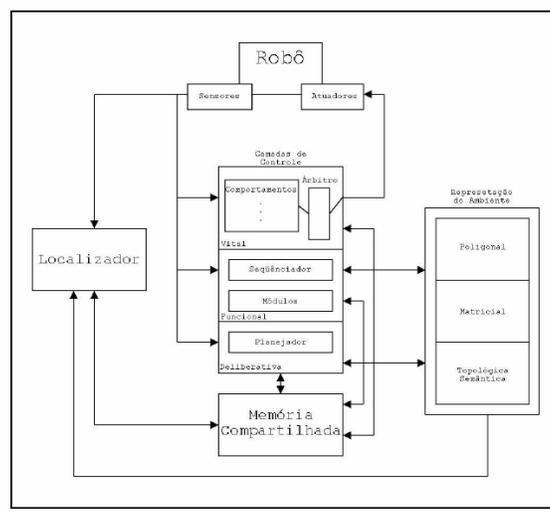
Agentes Inteligentes



Memória Compartilhada

A memória compartilhada é um depósito central de informações que é utilizada para a comunicação entre os diversos módulos.

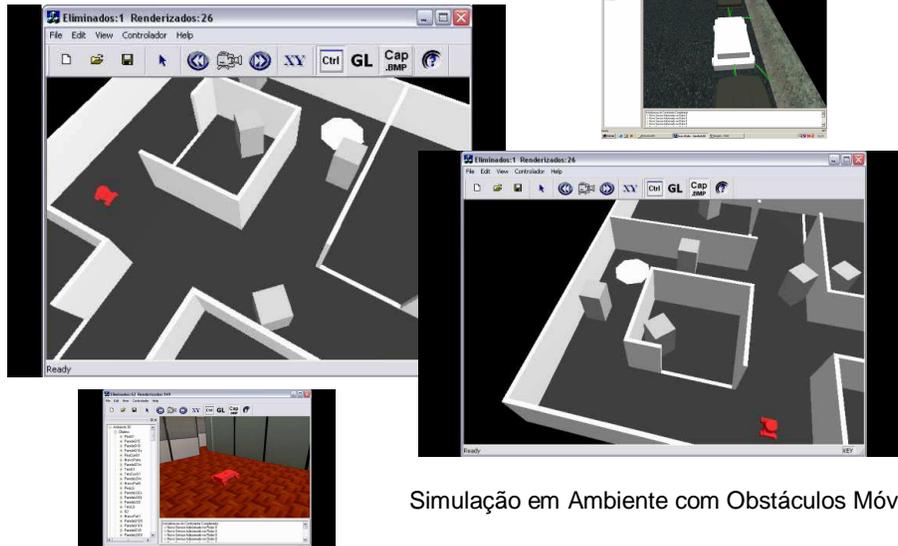
Arquitetura COHBRA
Controle Híbrido de Robôs Autônomos



Agentes Inteligentes



Simulação com o SimRob3D



Simulação em Ambiente com Obstáculos Móveis

Agentes Inteligentes



AGENTES REALMENTE ESPECIAIS:

- Controle Adaptativo: agentes que aprendem
- Exploração do Ambiente e Construção do Mapa
- Agentes Autônomos Inteligentes baseado em Sensores Reais (Visão Simulada)
- Estratégias em Jogos
- * **Multi-Agentes:** Comunicação & Cooperação

Agentes Inteligentes



AGENTES REALMENTE ESPECIAIS:

- Controle Adaptativo: agentes que aprendem
- Exploração do Ambiente e Construção do Mapa
- Agentes Autônomos Inteligentes baseado em Sensores Reais (Visão Simulada)
- Estratégias em Jogos
- * **Multi-Agentes:** Comunicação & Cooperação

**AGENTES ESPECIAIS...
Não erram jamais!**

