

# Tópicos Especiais I - Jogos

## IA para Jogos



Game AI  
02

Fernando Osório

07/10/2003

## IA para Jogos – Aula 02



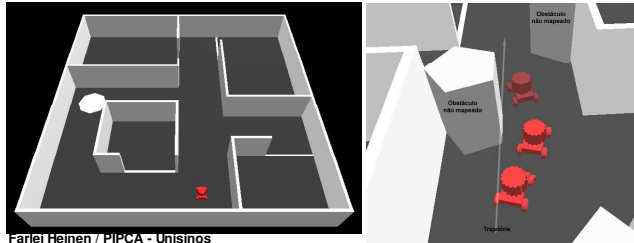
Tópicos abordados...

- Agentes Inteligentes
  - Comportamento:  
“falsa IA” e o “comportamento inteligente”
  - Agentes Reativos
  - Agentes Cognitivos / Deliberativos
  - Agentes com Arquitetura Hierárquica e Híbridos
  - Controle baseado em Autômatos (FSA, RdP)
  - Controle baseado em Regras (RBS – Rule Based)
  - Controle Adaptativo: agentes que aprendem
  - Agentes Autônomos Inteligentes
  - Estratégias em Jogos

## IA para Jogos



- IA clássica: **Jogos de Ação (Labirintos, Corridas, ...)**
  - Busca de caminhos (*Path finding*)
  - Planejamento de ações e trajetórias (*Path planning*)
  - Deslocamento / Navegação (*Motion & Navigation*)



Farlei Heinen / PIPCA - Unisinos

>> Da Busca de Caminhos (A\*) à Movimentação Inteligente <<  
Arquitetura de Controle para Agentes Autônomos

## IA para Jogos



- IA clássica: **Jogos de Ação (Labirintos, Corridas, ...)**
  - Busca de caminhos (*Path finding*)
  - Planejamento de ações e trajetórias (*Path planning*)
  - Deslocamento / Navegação (*Motion & Navigation*)



>> Da Busca de Caminhos (A\*) à Movimentação Inteligente <<  
Arquitetura de Controle para Agentes Autônomos

# Agentes Inteligentes

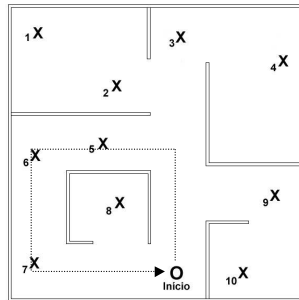


## ➤ Agentes Inteligentes

- Comportamento:  
"Falsa IA" versus "Comportamento inteligente"



Rotas pré-definidas  
Comportamento fixo  
Repetitivo  
Falta de interação  
com o ambiente



# Agentes Inteligentes



## ➤ Agentes Inteligentes

- Comportamento:  
"Falsa IA" versus "Comportamento inteligente"

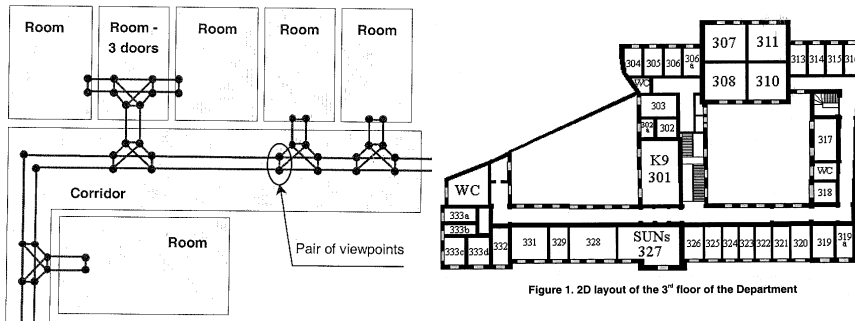


Figure 1. 2D layout of the 3<sup>rd</sup> floor of the Department

Figure 4. Pairs of viewpoints help to smoothly change the orientation of avatar during walk-through different places in a model.

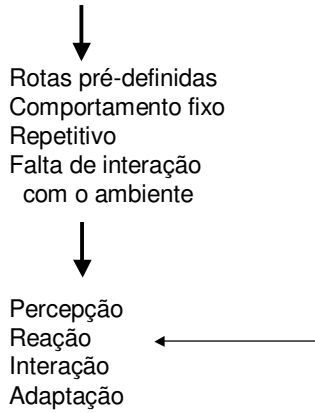
WRV200  
Zára & Cernohorský

# Agentes Inteligentes



## ➤ Agentes Inteligentes

- Comportamento:  
"Falsa IA" *versus* "Comportamento inteligente"

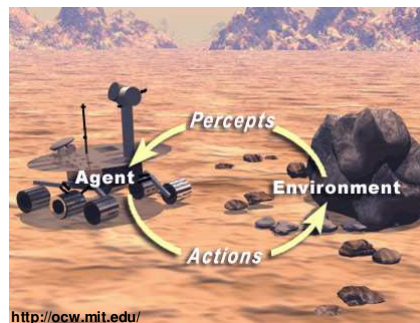
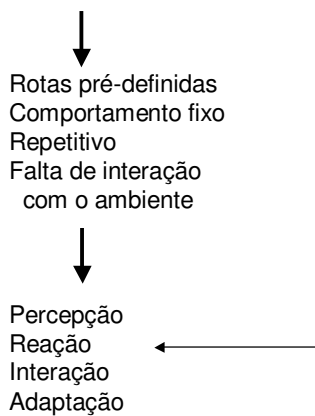


# Agentes Inteligentes



## ➤ Agentes Inteligentes

- Comportamento:  
"Falsa IA" *versus* "Comportamento inteligente"

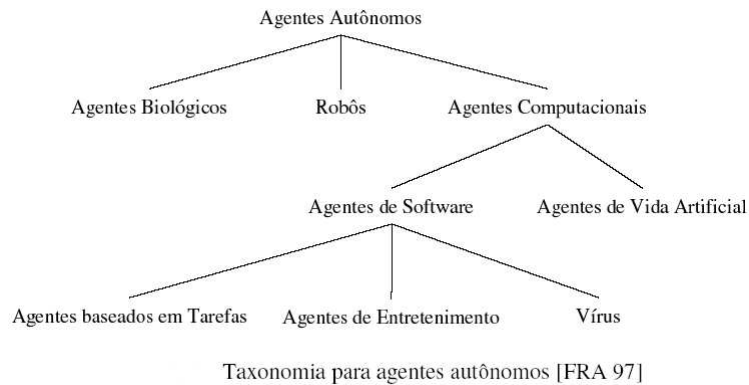


**Arquitetura de Controle**  
**Agentes Autônomos**  
**Agentes Inteligentes**

# Agentes Inteligentes



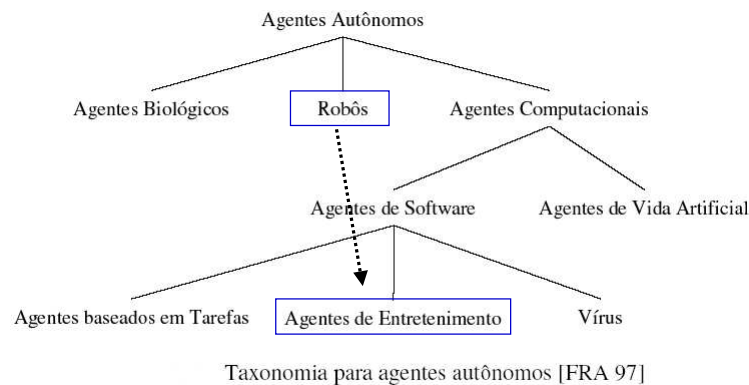
## ➤ Agentes Inteligentes - Taxonomia



# Agentes Inteligentes



## ➤ Agentes Inteligentes - Taxonomia



### **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, Coulored, TempORIZED)
  - Baseados em Regras:
    - RBS – Rule Based System  
(Facts, Rules and Inference)

# Agentes Inteligentes



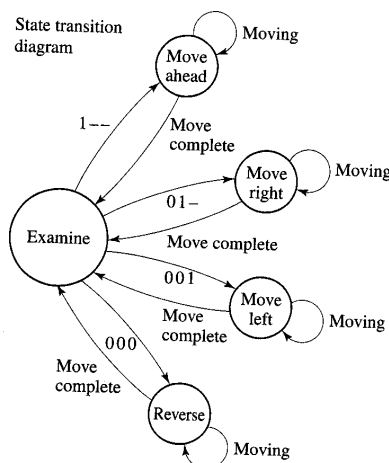
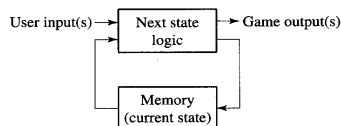
## ➤ Agentes Inteligentes – Ferramentas

- **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* ...



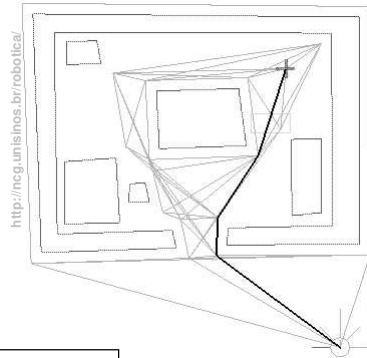


# Agentes Inteligentes



## ➤ Agentes Inteligentes – Ferramentas

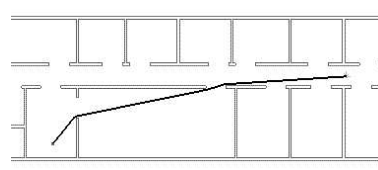
### • AgCog – Autômato



SimRob2D

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

**Execução:**  
\* Controle Deliberativo / Autômato

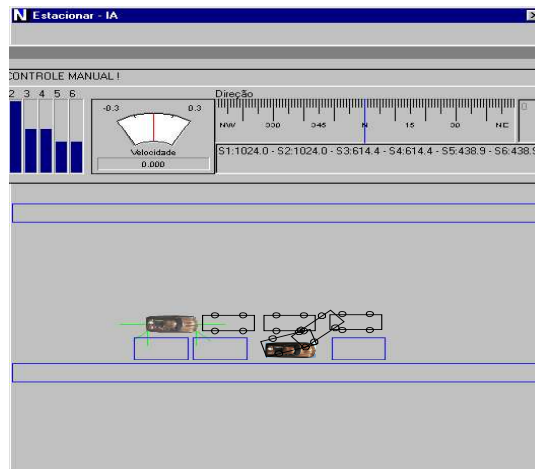


# Agentes Inteligentes



## ➤ 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*

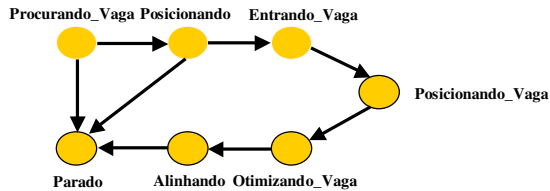
**SEVA-N**  
*Seva Neural*

# Agentes Inteligentes



## ➤ AgCog – Ferramentas: Autômato

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



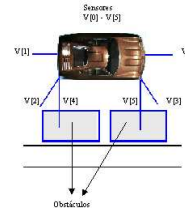
Se Estado\_Atual (Procurando\_Vaga) e Próximo\_ao\_Obstáculo (V[4]) e Próximo\_ao\_Obstáculo (V[5])  
Então Speed = Avanço\_Rápido e RotVel = Direção\_Reta;

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;

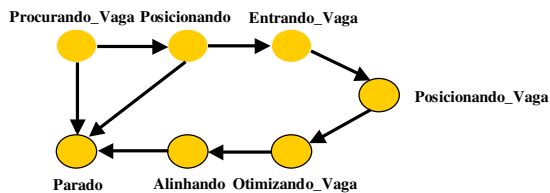


# Agentes Inteligentes



## ➤ 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;

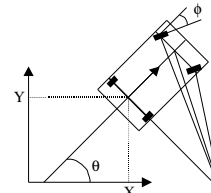
Se Estado\_Atual (Posicionando) e Longe\_do\_Obstáculo (V[4]) e Deslocamento\_Suficiente (Odômetro)  
Então Estado\_atual (Entrando\_Vaga) e Inicializa (Odômetro);

Se Estado\_Atual (Entrando\_Vaga)  
Então Speed = Ré\_Rápida e RotVel = Giro\_Esquerda\_Max;

$$X = V * \cos(\Phi) * \cos(\theta) \quad \text{Eq. 1}$$

$$Y = V * \cos(\Phi) * \sin(\theta) \quad \text{Eq. 2}$$

$$\theta = V / L * \sin(\Phi) \quad \text{Eq. 3}$$

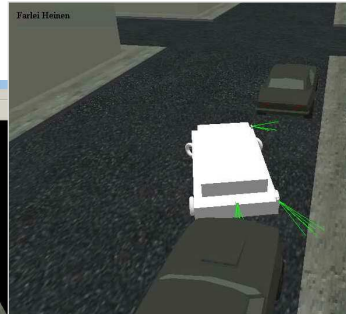
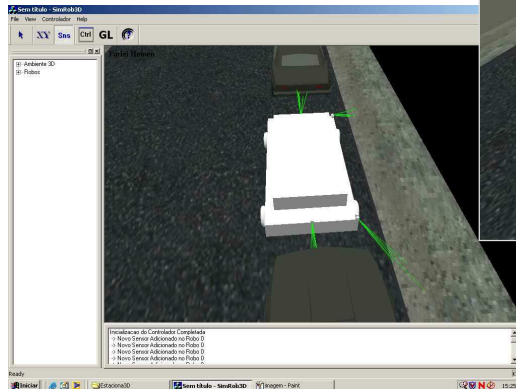


# Agentes Inteligentes



## ➤ 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

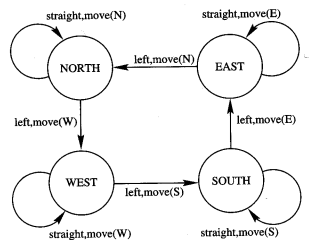


Figure 5.1. The WhichDir FSM.

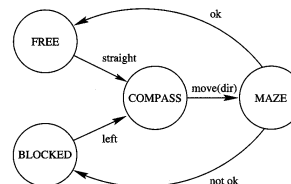


Figure 5.2. HFSM that uses the WhichDir FSM.

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."

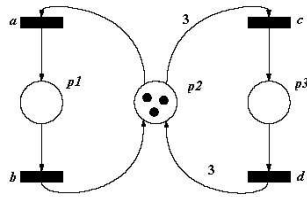
As the name implies, an HFSM is simply a hierarchy of FSMs. That is, each node of an HFSM may itself be an FSM. Just like functions and procedures in a regular programming language, this provides a convenient way to make the design of an FSM more modular. For example, if a character is at coordinates  $(x, y)$ , 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

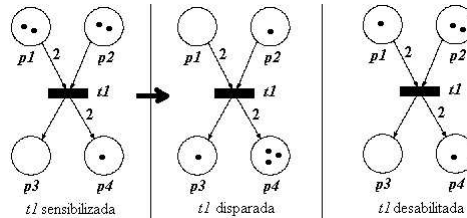
- AgCog – RdP (Rede de Petri)



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.robforge.com](http://www.robforge.com)



# Agentes Inteligentes

RBS (Rule Based System)  
RoboForge

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

The screenshot shows the RoboForge 'workshop' interface. At the top, there are navigation buttons for 'design', 'moves', 'a.i.', 'profile', and 'test'. The 'a.i.' button is selected. The main area is divided into several panels:

- Left Panel:** A flowchart of the rule-based system. It starts with 'Start', followed by 'Check Zones', 'Front is active?', 'Frontal Attack', 'Find Enemy', and a decision node 'angle in range 5.0° - 355.0°'. Below this are 'Orient Facing only' and another decision node 'The distance to my enemy > 6?', leading to 'Distance only'.
- Right Panel:** A 'decision' node for 'angle (°)'. It features a circular diagram with a central point and a radius. The diagram is labeled '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°?'.
- Bottom Right:** A 'simulation running' window showing a 3D view of a robot in a virtual environment.

At the bottom of the interface, there are status indicators: 'A.I. panel activated.', 'cost: 34,760 creds', 'weight: 3,941 kg', and 'health: 22,767 hp'.

# Agentes Inteligentes



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

This block contains three screenshots related to the Robocode game:

- Top Left:** A close-up of a robot's health and status, showing a score of 77.0 and the action 'Fire'.
- Top Right:** A top-down view of the Robocode arena, a green oval field with the word 'Robocode' in the center. Several robots are visible on the field.
- Bottom Left:** A screenshot of the Robocode game window running on a Windows operating system. It shows the arena, a robot's health, and a 'Fire' action.
- Bottom Right:** The 'About Robocode' dialog box. It contains 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.

# 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

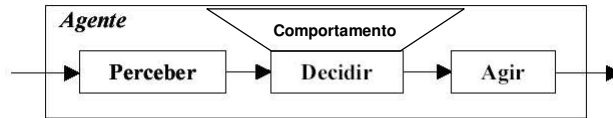
#*****
# Definicoes das Regras - Determinar o status do terreno
#*****
BLOCK_RULES
RL = IF Terreno_N IS Impossivel
    THEN Status_Terreno_N IS Impedido
RL = IF Terreno_N IS Alvo
    THEN Status_Terreno_N IS Impedido
...
RL = IF Direcao IS NO AND NOT Sonar_Dir_Atras IS Nenhum AND
    NOT Status_Terreno_L 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

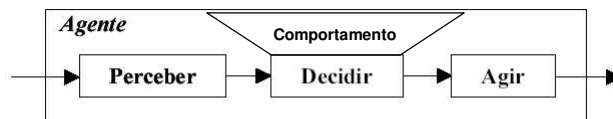
Assim como é difícil ter um agente puramente deliberativo e baseado em uma única ferramenta (FSA, RBS, etc), também é difícil de se ter um agente puramente reativo, sem memória ou estado interno...

# Agentes Inteligentes



## ➤ Agentes Inteligentes – Ferramentas

### • AgReact - Reativo



Arquitetura puramente reativa

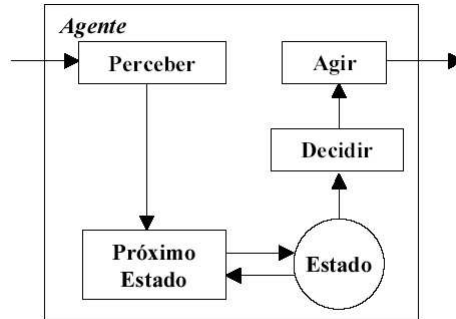
Assim como é difícil ter um agente puramente deliberativo e baseado em uma única ferramenta (FSA, RBS, etc), também é difícil de se ter um agente puramente reativo, sem memória ou estado interno...

[Agentes Hierárquicos e Híbridos](#)

# Agentes Inteligentes



- Agentes Inteligentes – Ferramentas
  - AgReact – Reativo... com estados

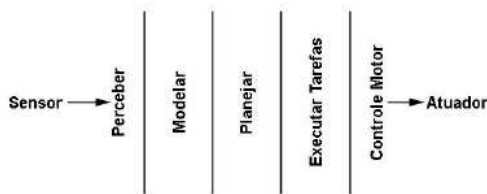


Arquitetura com Estado Interno  
sensação de déjà vu :)

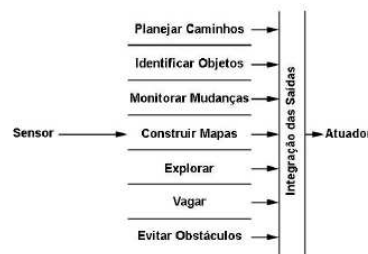
# Agentes Inteligentes



- Agentes Inteligentes – Ferramentas
  - AgReact – Reativo Hierárquico



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



Arquitetura Vertical:  
Subsumption - Brooks

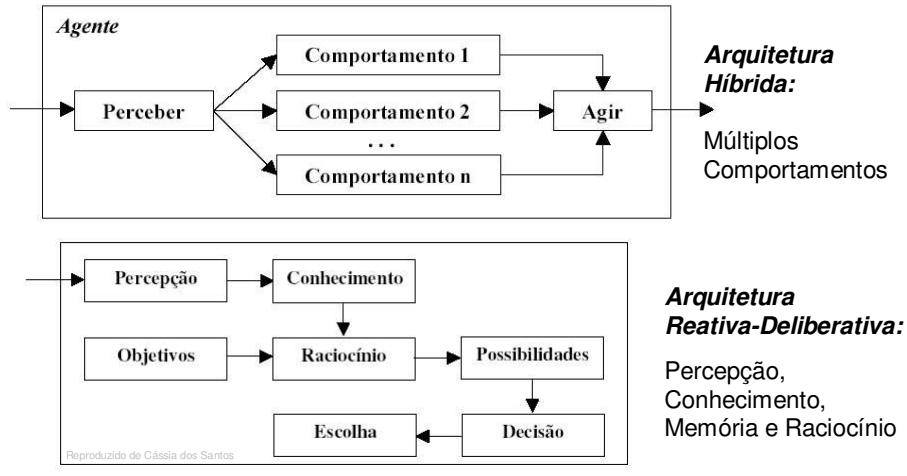


# Agentes Inteligentes



## ➤ Agentes Inteligentes – Ferramentas

- Agentes Cognitivos, Reativos e Híbridos

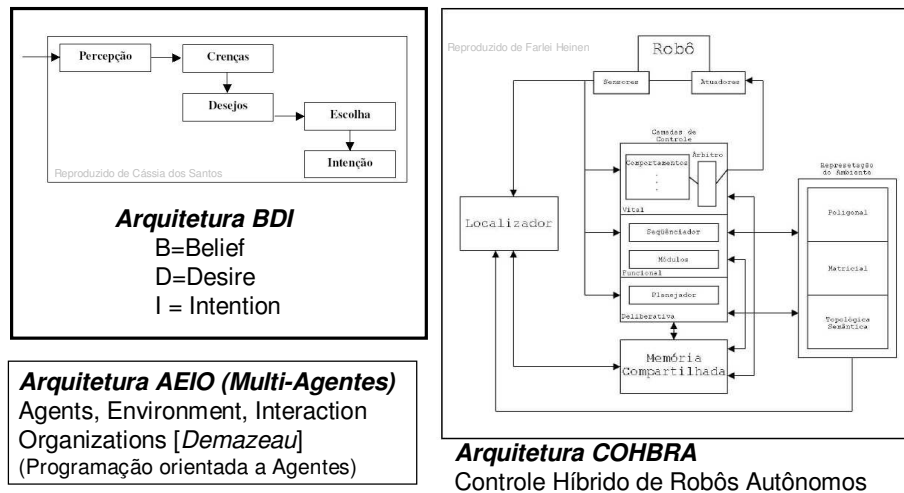


# Agentes Inteligentes



## ➤ Agentes Inteligentes – Ferramentas

- Agentes Cognitivos, Reativos e Híbridos

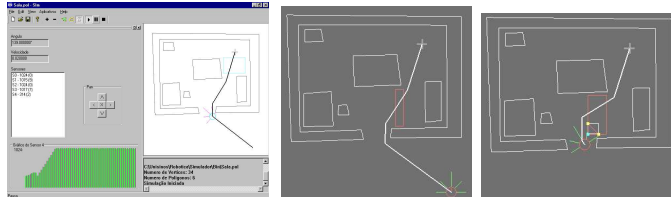


# Agentes Inteligentes

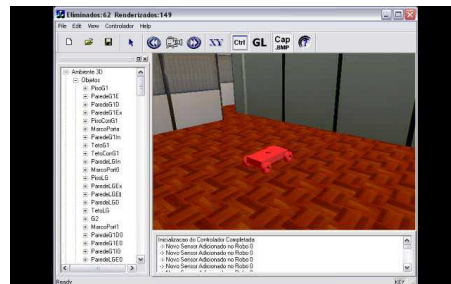
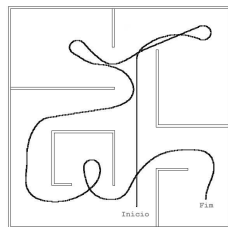


## ➤ Agentes Inteligentes – Ferramentas: Controle Híbrido

### SimRob2D



### SimRob3D / COHBRA



# Agentes Inteligentes



## Representação do Ambiente

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

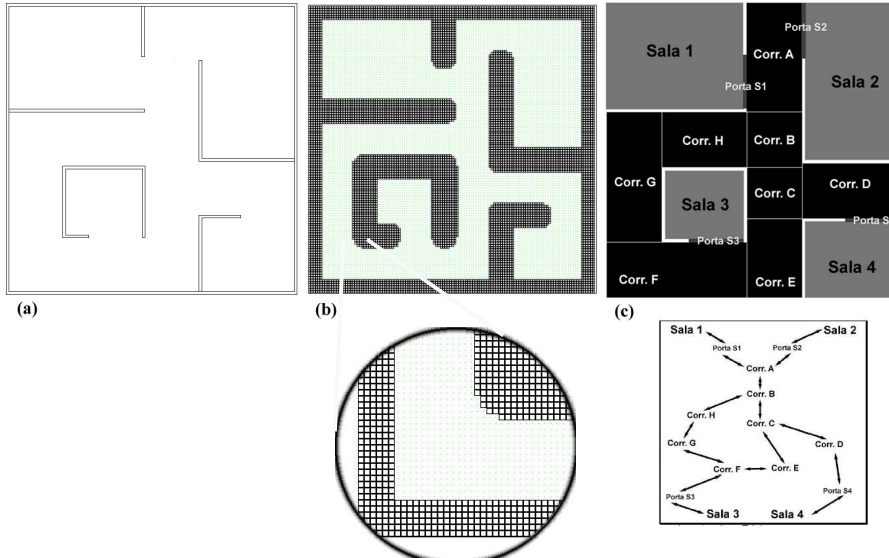
**Matricial:** Representa o ambiente através de uma matriz. Gerada a partir da representação poligonal. Utilizada principalmente para o planejamento de trajetória.

**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

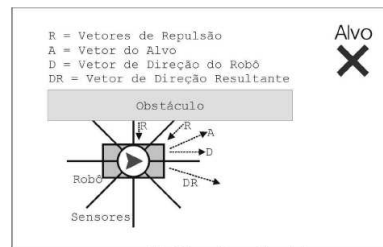


## Camada Vital

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



**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.

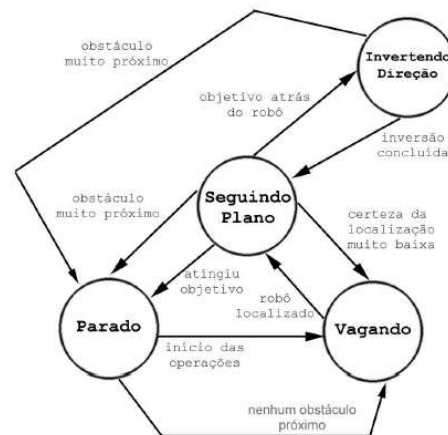


Fig. 6.4 Diagrama de Estados

# 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.

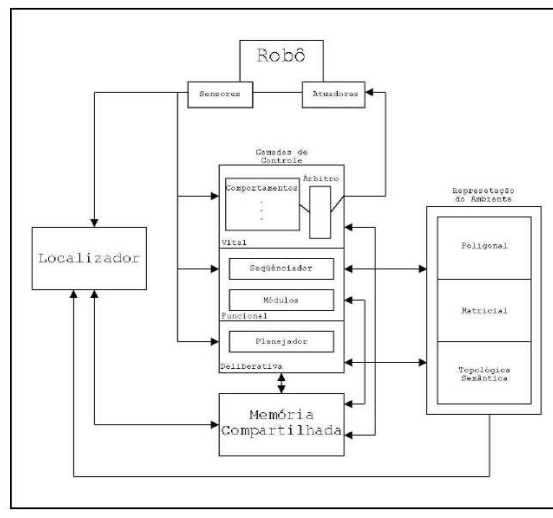
# Agentes Inteligentes



## Memória Compartilhada

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

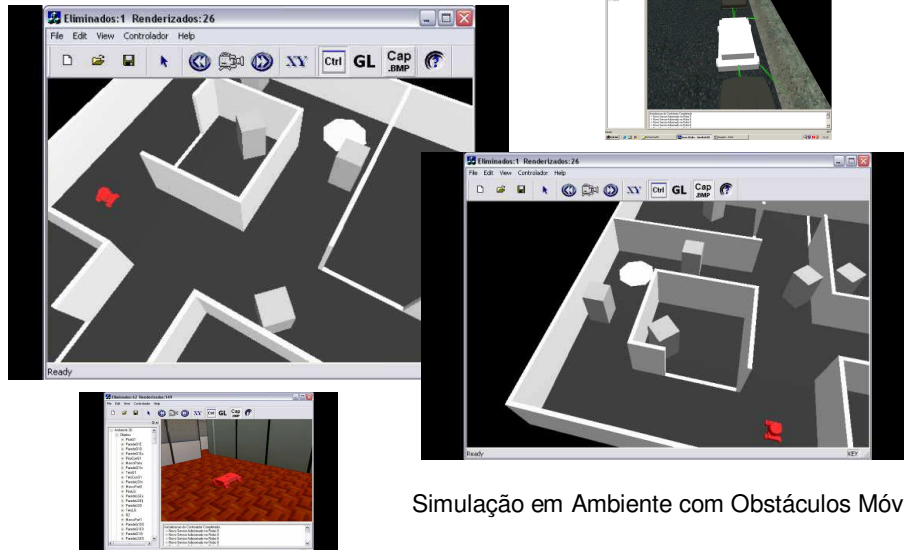
**Arquitetura COHBRA**  
Controle Híbrido de Robôs Autônomos



# 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!**

