

# Sistemas Multiagentes

Claudio Cesar de Sá  
[claudio.sa@udesc.br](mailto:claudio.sa@udesc.br)

Departamento de Ciência da Computação  
Centro de Ciências e Tecnologias  
Universidade do Estado de Santa Catarina

16 de março de 2017

# Sumário (1)

---

## O Curso

Ferramentas

Metodologia e avaliação

Dinâmica

Referências

## Introdução

Motivação aos SMAs

Os Elementos de SMAs

## Agentes Racionais

Tipos de Agentes

Arquiteturas de Agentes

## Construindo de Agentes Racionais

# Sumário (2)

---

## Teoria de Jogos, Coordenação e Planejamento

### Estratégias de Jogos

Teoria de Jogos Aplicado a SMA

### Coordenação

Jogos de Coordenação

Convenção Social

Papel Social

Grafos de Coordenação

Coordenação por Eliminação de Variáveis

Coordenação por Troca de Mensagens

### Planejamento

Abordagens ao Planejamento Multiagente – SMAs

Exemplos de Coordenação SMAs

# Sumário (3)

---

Projetos de SMAs

Implementação de SMAs  
Simulação de Ambientes

Conclusões

# Agradecimentos

---

- Patrícia Tedesco – UFPe
- Alexandre Gonçalves – UFSC
- Ao Google Images ... vários autores

# Disciplina

---

## Sistemas Multiagentes – OSIM001

- **Turma:**
- **Professor:** Claudio Cesar de Sá
  - claudio.sa@udesc.br
  - Sala 13 Bloco F
- **Carga horária:** 72 horas-aula • Teóricas: 36 • Práticas: 36
- **Curso:** BCC
- **Requisitos:** Vários – IA, LMA, TEC, SO, PRP, ...
- **Período:** 1º semestre de 2017
- **Horários:**
  - 6ª 10h10 (2 aulas) - F-104 – aula expositiva
  - 6ª 18h00 (2 aulas) - F-306 – lab

## Ementa

---

### Ementa

Motivação do paradigma. Agentes reativos e cognitivos. Teoria e arquitetura de agentes. Sistema multiagentes (SMA) reativo e cognitivo. Linguagens e protocolos de comunicação. Coordenação e negociação. Metodologias para desenvolvimento de SMAs. Ambientes de desenvolvimento.

## Objetivos (1)

---

- **Geral:** Apresentar o conceito de inteligência artificial distribuída: desenvolvimento de agentes e abordagens para coordenação de sistemas multiagentes, permitindo ao aluno ser capaz de modelar problemas de forma a modularizar sua solução de forma distribuída.

## Objetivos (2)

---

### ■ **Específicos:**

- Descrever o histórico e quadro atual da Inteligência Artificial – Moderna.
- Compreender a noção de Teoria de Problemas, computabilidade e complexidade na ótica de IA e IAD.
- Diferencia IAD (orientação a divisão de problemas) versus SMA (orientação a coordenação de agentes)
- Conhecer diferentes arquiteturas de agentes
- Modelar problemas computacionais através de aplicação de agentes.
- Descrever o processo de tomada de decisão e aprendizagem computacional baseado em sistemas multiagentes.
- Conceber, projetar e construir sistemas computacionais capazes de aplicar sistemas multiagentes como técnica de resolução.

## Conteúdo programático

---

- Conceitos de SMA (há muitos correlacionados há áreas diversas)
- Ferramentas: Netlogo e Picat
- Aplicação: vocees escolhem
- Um artigo ≡ projeto
- Um artigo OUTRO da área a ser apresentado: ficha técnica

## Ferramentas

---

- PICAT (com suporte)
- NETLOGO <http://ccl.northwestern.edu/netlogo/docs/>  
(escondido in WEB)

# Metodologia e avaliação (1)

---

## **Metodologia:**

*As aulas serão expositivas e práticas. A cada novo assunto tratado, exemplos são demonstrados utilizando ferramentas computacionais adequadas para consolidar os conceitos tratados. As aulas nas sextas-feiras a tarde poderão ser realizadas, também, na forma de estudo dirigido.*

# Metodologia e avaliação (2)

---

## Avaliação

- Duas provas (conceituais) –  $\approx 25\%$ 
  - $P_1$ : 25/mar
  - $P_2$ : 25/maio (provão: todo conteúdo)
- Exercícios de laboratório –  $\approx 10\%$
- Implementação de um protótipo –  $\approx 20\%$
- O artigo (resultados da implementação) –  $\approx 30\%$
- Para o artigo: muito material será fornecido em L<sup>A</sup>T<sub>E</sub>X...
- Apresentação de um artigo estudado sobre SMA –  $\approx 15\%$
- Presença e participação
- Média para aprovação: 5,0 (cinco)

## Dinâmica de Aula

---

- Teoria na parte da manhã – 10:00 hrs – F-104
- Ralação a tarde – LAB – estudar o NetLogo – vídeo-aulas

# Bibliografia (1)

---

## Básica:

- ALVARES, L. O., SICHMAN, J. *Introdução aos Sistemas Multiagentes*, Anais do EINE – Escola de Informática do Nordeste, Sociedade Brasileira de Computação – SBC, Brasil, 1997.
- FERBER, J. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Harlow, England, Addison-Wesley, 1999.
- WOOLDRIDGE, M.. *An introduction to MultiAgent Systems*, John Wiley, 2001
- [https://github.com/claudiosa/CCS/tree/master/https://github.com/claudiosa/CCS/tree/master/sistemas\\_multiagentes](https://github.com/claudiosa/CCS/tree/master/https://github.com/claudiosa/CCS/tree/master/sistemas_multiagentes)

## Complementar:

- Nikos Vlassis, A *Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence* Synthesis Lectures on Artificial Intelligence and Machine Learning 2007, 71 pages – guia dos tópicos destes slides
- O'HARE, G.; JENNINGS, N. (Editors) *Foundations of distributed artificial intelligence*, New York, NY: John Wiley, 1996.
- WEISS, G. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. London, MIT Press, 2001.
- Russell, S., Norvig, Peter; "Inteligência Artificial", Ed. Campus-Elsevier; Brasil, 2010 – em inglês.

## Bibliografia (2)

---

- Bittencourt, G.; "Inteligência Artificial, ferramentas e teorias"; 3. ed. UFSC; Florianópolis, SC; 2006.
- Barreto, J.M.; "Inteligência Artificial, uma abordagem híbrida"; 3a. ed.; RoRoRo; Florianópolis, SC; 2001
- Eberhart, R; Simpson, P.; Dobbins, R.; "Computational Intelligence PC Tools"; AP Professional; 1996; ISBN 0-12-228630-8.
- Fausett, Laurene; Fundamentals of Neural Networks; Prentice Hall Ind.; N. Jersey; 1994.
- Freeman, J. A.; Skapura, D. M.; "Neural networks – Algorithms, Applications and Programming Techniques"; Addison- Wesley Pub. Co.; New York; 1991.
- Luger, George F.; Inteligência Artificial; Artmed Ed. S.A.; P. Alegre; 2004.
- Mitchell, M.; "An introduction to genetic algorithms"; The MIT press; London; 1966.
- Rabuske, R. A.; Inteligência Artificial; UFSC; Florianópolis; 19??
- Resende, Solange O., Sistemas Inteligentes - Fundamentos e aplicações, Ed. Manole ([www.manole.com.br](http://www.manole.com.br)), 200?
- Rich, E.; "Artificial Intelligence"; McGraw-Hill Book Company; USA; 1983.
- Material didático disponível em: [www.inf.ufsc.br/~falqueto](http://www.inf.ufsc.br/~falqueto)

# Capítulo 1 – Introdução

(Contexto e Motivação aos SMAS)

## Rápido Histórico da IA ⇒ IAD ⇒ SMA

---

- IA cresceu muito nos anos 70 → 80... modelando a inteligência individual.
- Advento das redes de computadores modificou as necessidades!
- Inteligência como a integração dos processos de *raciocinar, decidir, aprender e planejar*.
- O *Modelo de Agente* aparece então como catalisador...

## Em verdade:

---

- Mundo onde informações e conhecimentos crescem (e mudam) rápido demais!
- O crescimento da Internet trás desafios constantes que incluem:
  - Acesso a informações relevantes
  - Identificação de oportunidades
  - Ação no momento preciso
  - Manipulação de grandes volumes de informação
- Ubiqüidade, Gerenciamento e Inteligência

## Encaminhando aos SMAs

---

- Vários problemas não podem mais serem tratados de modo **centralizado**, por exemplo:
  - Controle de linha de trens (Brasil) metrô (hum SP talvez)
  - Monitoramento de Redes de Computador
  - Diagnóstico Médico
  - Compra e Venda

# Encaminhando aos SMAs

---

- Vários problemas não podem mais serem tratados de modo **centralizado**, por exemplo:
  - Controle de linha de trens (Brasil) metrô (hum SP talvez)
  - Monitoramento de Redes de Computador
  - Diagnóstico Médico
  - Compra e Venda
- Como Resolvê-los?

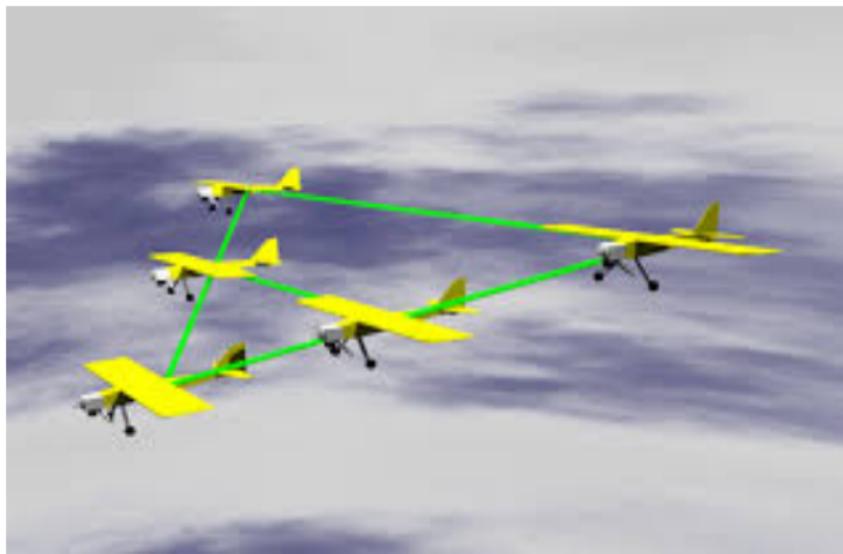
# Encaminhando aos SMAs

---

- Vários problemas não podem mais serem tratados de modo **centralizado**, por exemplo:
  - Controle de linha de trens (Brasil) metrô (hum SP talvez)
  - Monitoramento de Redes de Computador
  - Diagnóstico Médico
  - Compra e Venda
- Como Resolvê-los?  
Inteligência Coletiva ⇒ IA Distribuída ⇒
  - Resolução Distribuída de Problemas (RDP)
  - Sistemas Multiagentes (SMA) ⇐ **foco deste curso**

## Motivando aos SMAs

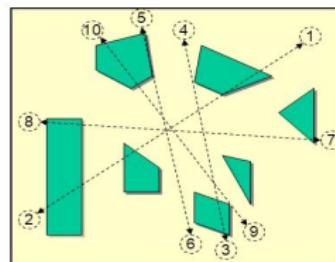
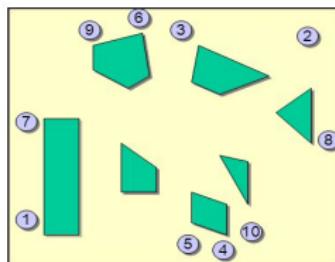
---



**Figura:** Observe o sentido das flechas – e o foco da missão

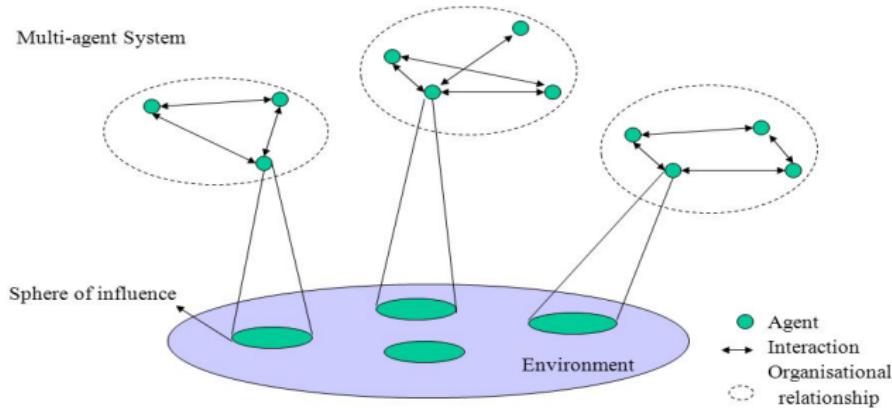
# Motivando aos SMAs

## Path Planning for Multiple Robots



# Motivando aos SMAs

## Multi-agent Systems (MAS)



8

Figura: Visão clássica de SMAs – comunidade de agentes  $\equiv$  SMA

# Motivação I

---

Projetar e construir sistemas multiagentes é uma tarefa difícil, pois:

# Motivação I

---

Projetar e construir sistemas multiagentes é uma tarefa difícil, pois:

- Apresenta todos os problemas já conhecidos dos sistemas distribuídos e concorrentes.

# Motivação I

---

Projetar e construir sistemas multiagentes é uma tarefa difícil, pois:

- Apresenta todos os problemas já conhecidos dos sistemas distribuídos e concorrentes.
- Dificuldades adicionais surgem da flexibilidade e complexidade das interações

# Problemas de tabuleiro são simples?

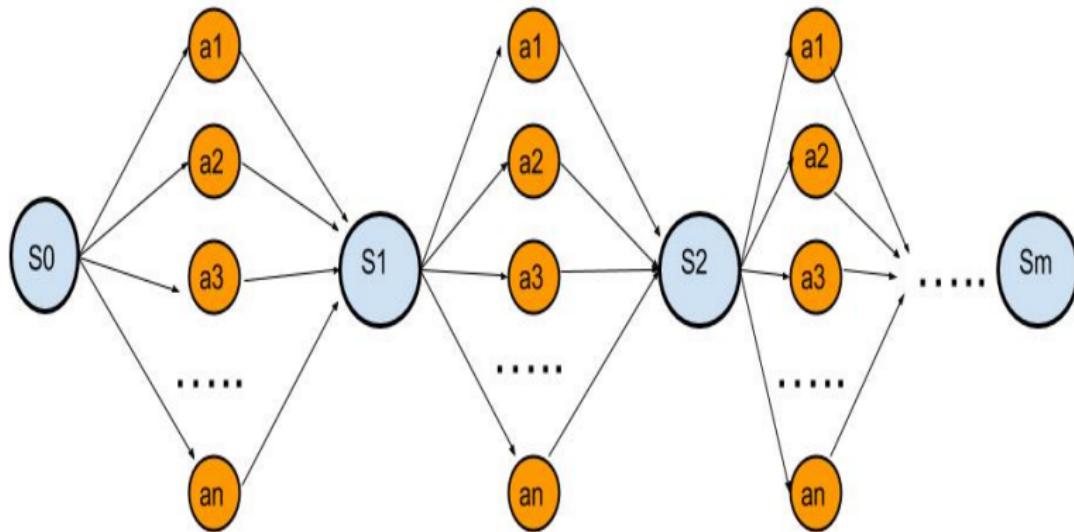
---



Figura: Admita um robo indo de uma posição inicial ( $S_0$ ) há uma final ( $S_m$ )

Exemplificando a complexidade por um DFD com **um agente**  $\times$  **n-ações**:

---



**Figura:** Complexidade via DFD de um agente  $\times$  ações  $\equiv$  um estado inicial ( $S_0$ ) há um estado final ( $S_m$ )

Exemplificando a complexidade por um DFD por  
**agentes** × **ações**:

---

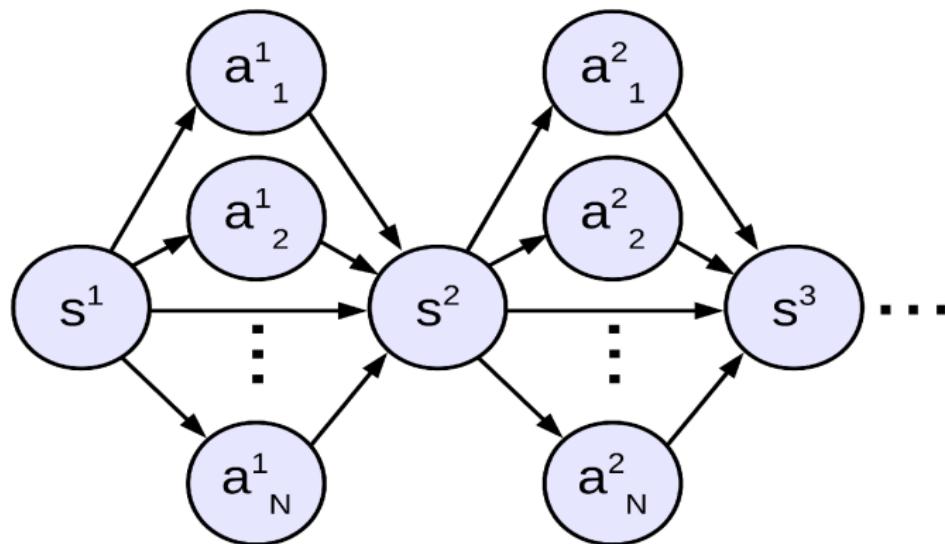


Figura: Complexidade via DFD de um SMA (agentes) × ações ≡ um único estado

## Exercício – papel e caneta mesmo

---

- Considere que as ações sejam do tipo um deslocamento de uma célula. 8 direções ou 4?

## Exercício – papel e caneta mesmo

---

- Considere que as ações sejam do tipo um deslocamento de uma célula. 8 direções ou 4?
- Considere que o robô não vai ficar em ciclos – passar em estados que já passou – fácil isto? Como resolver?

## Exercício – papel e caneta mesmo

---

- Considere que as ações sejam do tipo um deslocamento de uma célula. 8 direções ou 4?
- Considere que o robô não vai ficar em ciclos – passar em estados que já passou – fácil isto? Como resolver?
- Para as duas versões acima estime a complexidade dos mesmos em número de combinações possíveis de um  $S_0$  há um  $S_m$ .

## Exercício – papel e caneta mesmo

---

- Considere que as ações sejam do tipo um deslocamento de uma célula. 8 direções ou 4?
- Considere que o robô não vai ficar em ciclos – passar em estados que já passou – fácil isto? Como resolver?
- Para as duas versões acima estime a complexidade dos mesmos em número de combinações possíveis de um  $S_0$  há um  $S_m$ .
- Introduza as dimensões do tabuleiro em seus cálculos e refaça-os

## Motivação II – retomando ...

---

Dois principais impedimentos técnicos, pois:

## Motivação II – retomando ...

---

Dois principais impedimentos técnicos, pois:

- Inexistência de uma metodologia sistemática para claramente especificar e estruturar aplicações SMA.

## Motivação II – retomando ...

---

Dois principais impedimentos técnicos, pois:

- Inexistência de uma metodologia sistemática para claramente especificar e estruturar aplicações SMA.
- Inexistência de ferramentas e ambientes de desenvolvimento de SMA com qualidade industrial.

# Os Elementos de SMAs

---

O que abordaremos neste curso:

Projeto de Agente: ... **ao longo do curso**

Ambiente: ... **ao longo do curso**

Percepção: ... **ao longo do curso**

Controle: ... **ao longo do curso**

Conhecimento: ... **ao longo do curso**

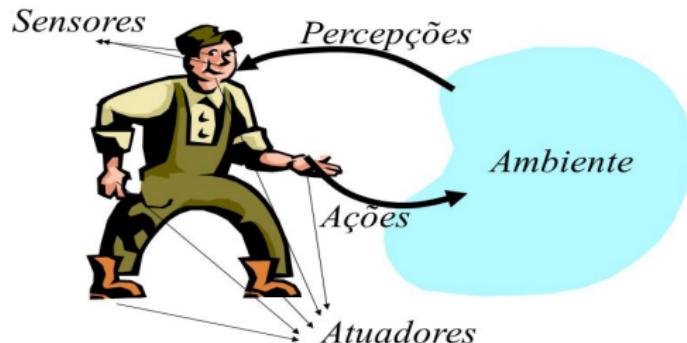
Comunicação: ... **ao longo do curso**

# Exercício

---

## Exemplo: agente humano

---



IA 13/10/10

6

**Figura:** Exercício: enumere domínios para este agente identificando os itens de SMAs

# Capítulo 2 – Agentes Racionais

# O que é um Agente?

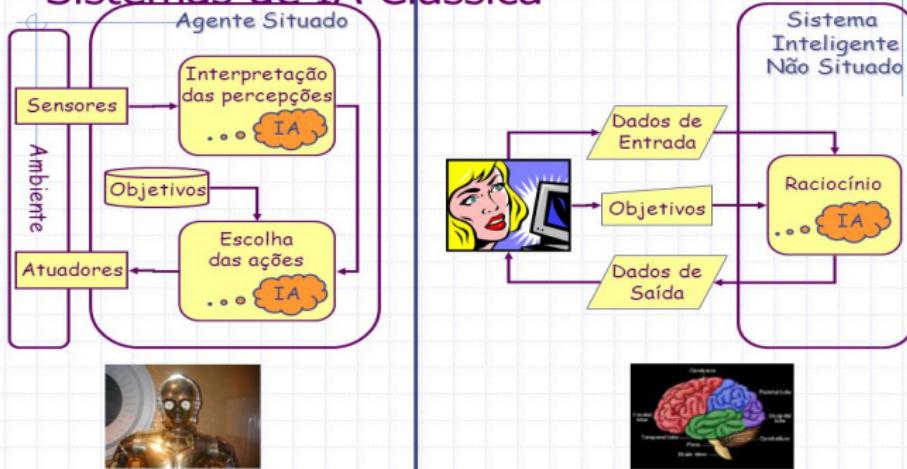
---

Qualquer entidade (humana ou artificial) que:

- está **imersa** ou **situada** em um ambiente (físico, virtual/simulado)
- **percebe** ou **sente** seu ambiente através de sensores (ex. câmeras, microfone, teclado, finger, ...)
- **age** sobre ele através de atuadores (ex. vídeo, auto-falante, impressora, braços, ftp, ...)
- **possui objetivos** próprios: explícitos ou implícitos
- **escolhe** suas ações em função das suas percepções para atingir seus objetivos

# Agente Situado x Não-Situado

## Agentes Inteligentes x Sistemas de IA Clássica



**Figura:** Agente situado versus a visão clássica de sistemas inteligentes

# O que é um Agente Racional?

---

- Agente Racional
  - faz a melhor ação possível dado um conjunto de percepções
  - segue o princípio da racionalidade:  
dada uma seqüência perceptiva, o agente escolhe, segundo seus conhecimentos, as ações que melhor satisfazem seu objetivo
- Limitações de:  
sensores  
atuadores  
raciocinador (conhecimento, tempo, etc.)

## Outras propriedades freqüentemente associadas aos Agentes (1)

---

- Autonomia:  
raciocínio, comportamento guiado por objetivos ou *reatividade*
  - Requer máquina de inferência e base de conhecimento
  - Essencial em sistemas especialistas, controle, robótica, jogos, agentes na internet ...
- Adaptabilidade & aprendizagem
  - Capacidade de adaptação a situações novas, para as quais não foi fornecido todo o conhecimento necessário com antecedência
  - Duas implementações: sistema com aprendizagem e/ou programação declarativa
  - Essencial em agentes na internet, interfaces amigáveis ...

## Outras propriedades freqüentemente associadas aos Agentes (2)

---

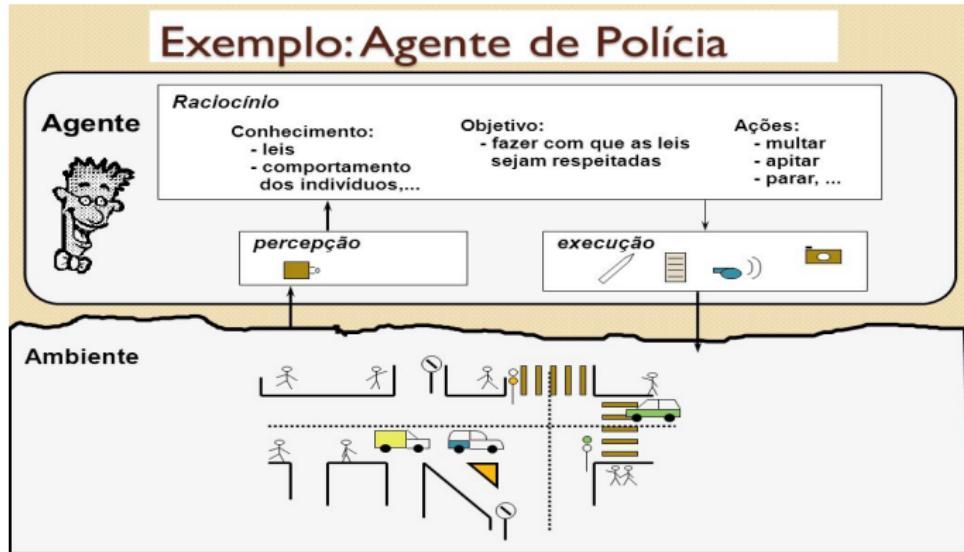
- Comunicação & Cooperação (sociabilidade)
  - Protocolos padrões de comunicação, cooperação, negociação
  - Raciocínio autônomo sobre crenças e confiabilidade
  - Arquiteturas de interação social entre agentes
- Personalidade
  - IA + modelagem de atitudes e emoções (computação afetiva)
  - Essencial em entretenimento digital, realidade virtual, interfaces amigáveis
  - ...
- Continuidade temporal (persistência)
  - Requer interface com sistema operacional e banco de dados
  - Essencial em filtragem, monitoramento, controle, ...
- Mobilidade (caso internet)
  - Requer itens como:

## Outras propriedades freqüentemente associadas aos Agentes (3)

---

1. Interface com rede
  2. Protocolos de segurança
  3. Suporte a código móvel
- Essencial em agentes de exploração da internet, ...

# Exercício



**Figura:** Exercício: enumere as características citadas a este exemplo – ao final do capítulo este exercício deve ser rediscutido

# Tipos de Agentes (1)

---

Em geral os agentes encontram-se em dois grupos (2 classes):

**Agentes Reflexivos:** geralmente são agentes simples, escolhem suas ações baseados **exclusivamente** nas percepções que têm do ambiente. Normalmente possuem uma representação do conhecimento implícita no código, por não possuirem memória, não tem histórico dos fatos e das ações que executou.

- Nota: nestes slides, ora são chamados de *reativos*
- Na 3a. versão do livro do Russel e Norvig – utiliza-se o termo *reflexivo*
- Ver fundamentação biológica para este termo – está correta
- Nos primórdios da área: o termo é *reativo*

## Tipos de Agentes (2)

---

**Agentes Cognitivos:** têm uma representação simbólica explícita do seu ambiente, no qual eles podem argumentar e predizer eventos futuros. Estes são dirigidos por intenções, isto é, por metas explícitas que conduzem seu comportamento e os tornam capazes de escolher entre possíveis ações. Engloba as características: *percepção, ação, comunicação, representação, motivação, deliberação, raciocínio e aprendizagem*.

# Arquitetura clássica de um agente reflexivo

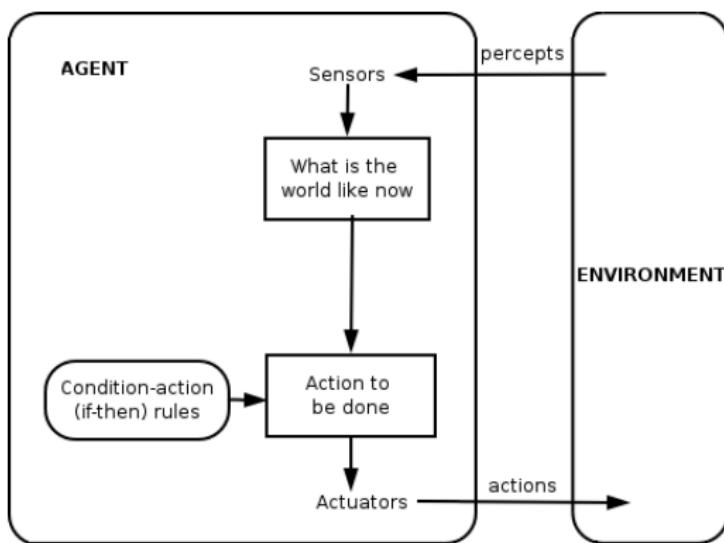


Figura: Arquitetura clássica – reflexivo

# Arquitetura clássica de um agente cognitivo (*que aprende algo!*)

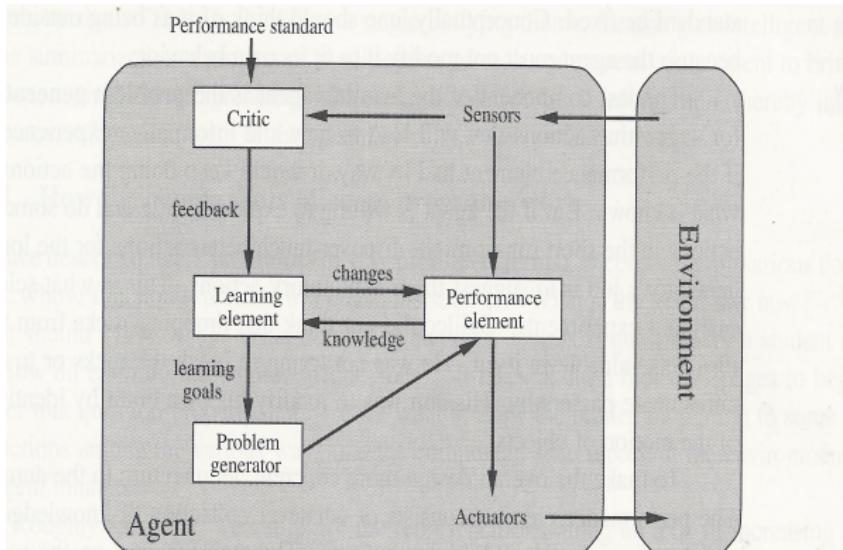


Figura: Arquitetura clássica – agente cognitivo com aprendizagem

# Arquiteturas ou Modelos de Agentes

---

Destes 2 grupos (reflexivo e cognitivo), delineia-se algumas arquiteturas:

- Agente tabela (menos complexo)
- Agente reativo
- Agente reativo com estado interno
- Agente baseado em objetivos (com *alguma* cognição)
- Agente otimizador
- Agente adaptativo (mais complexo)

Genericamente todos seguem algo como:

Um agente pode ser visto como um **mapeamento**: **seqüência e/ou fusão de percepções ⇒ ação**

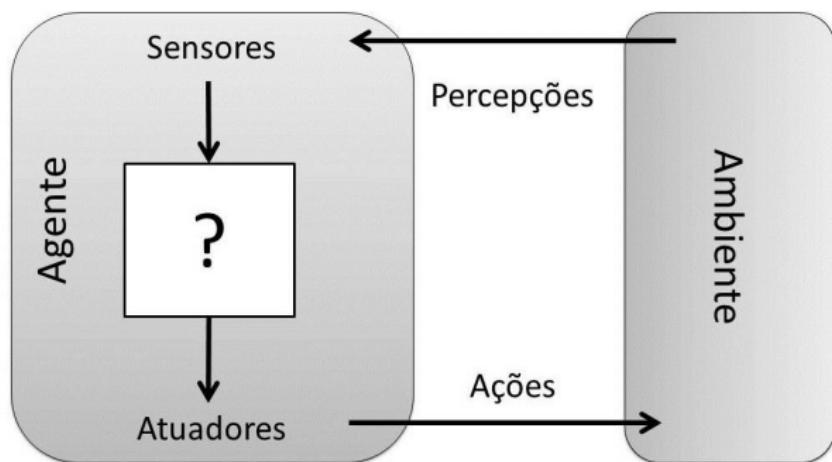


Figura: Agente Genérico

# Agente Tabela – é mesmo um agente racional?

## FredBF.com® (3.b) Arquiteturas: agente tabela

- Ex: Agente aspirador de pó

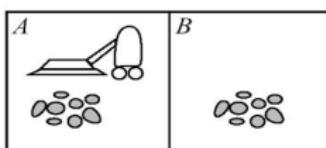


Tabela	
Percepção	Ação
[A,limpo]	Direita
[A,sujo]	Aspirar
[B,limpo]	Esquerda
[B,sujo]	Aspirar

```
Função AGENTE-ASPIRADOR-TABELA ([posição, estado]): ação {  
    ação = procura(Tabela, [posição,estado])  
    retorna ação  
}
```

**Problema:**  
Construir a tabela é muito  
trabalhoso!

Figura: Agente Tabela – seguem os *copyrights*

# Agente Tabela

---

## ■ Limitações

- Mesmo problemas simples requerem tabelas muito grandes Ex. jogo de xadrez  $30^{100}$
- Nem sempre é possível, por ignorância ou questão de tempo, construir a tabela
- Não há autonomia nem flexibilidade
- Este *infeliz* entra em pane se o conhecimento não estiver descrito na tabela (inferior há uma regra *if-then*)

## ■ Ambiente

- do tipo acessível: determinista, episódico, estático, discreto e minúsculo!

# Agente Reativo

---

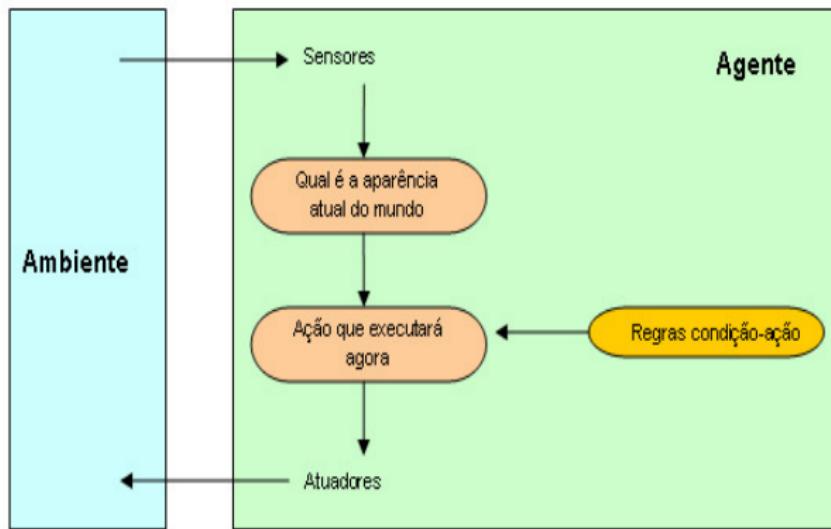


Figura: Agente Reativo – seguem os *copyrights*

# Agente Reativo

---

- Vantagens e desvantagens
  - Regras condição-ação: representação inteligível, modular e eficiente  
Ex: Se velocidade > 60 então multar
  - Não pode armazenar uma seqüência perceptiva, pouca autonomia
- Ambientes
  - Reflexo imprescindível em ambientes dinâmicos
  - Acessível, episódico, pequeno

# Agente reativo com estado interno

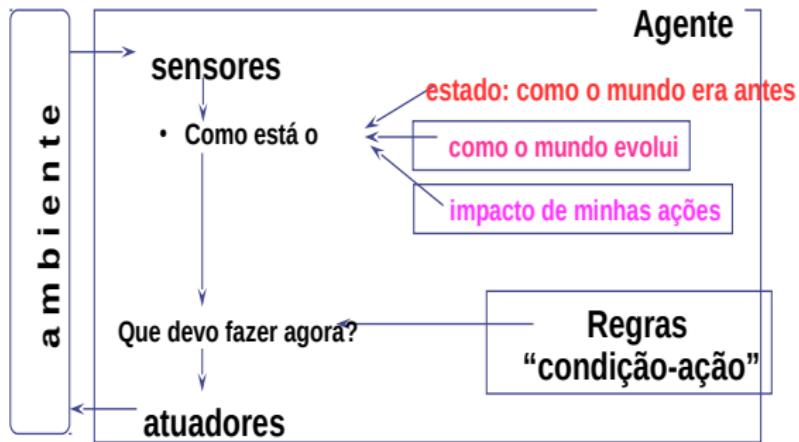


Figura: Agente reativo com estado interno – seguem os *copyrights*

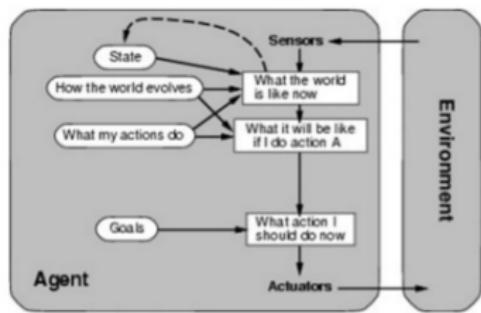
# Agente reativo com estado interno

---

- Desvantagens
  - pouca autonomia
  - não tem objetivo, não encadeia regras
  - melhorar a figura .... *como está o mundo agora?*
- Ambientes
  - determinista e pequeno
  - Ex. Tamagotchi – sucesso

# Agente baseado em objetivo

## Agente baseado em objetivo



### Características

- Possui um estado desejável (objetivo)
- Considera o que pode acontecer no futuro
- Pode selecionar um plano ao invés de uma ação

Figura: Agente baseado (orientado) em objetivo – seguem os *copyrights*

# Agente baseado em objetivo

---

- Vantagens e desvantagens
  - Mais complicado e ineficiente, porém mais flexível, autônomo
  - Não trata objetivos conflitantes
- Ambientes
  - determinista
  - Ex.ex.: xeque-mate no xadrez

# Agente otimizador (*utility based*)

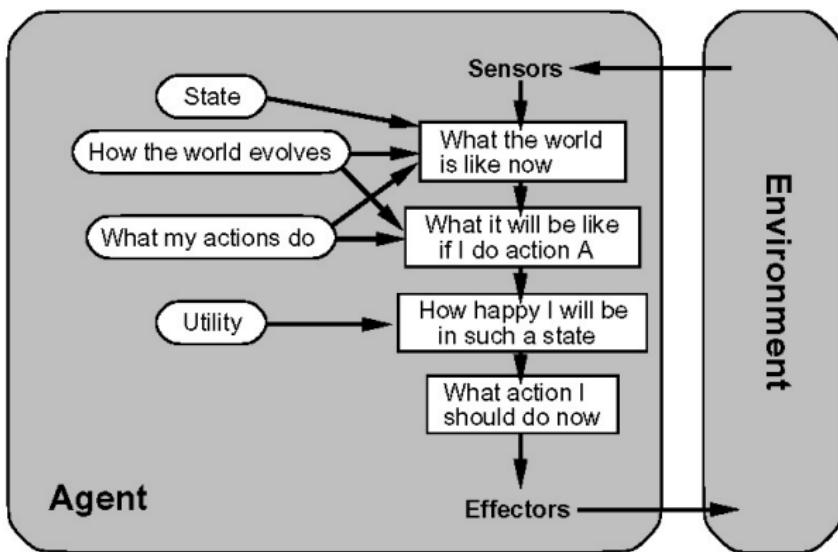


Figura: Agente otimizador – seguem os *copyrights*

## Agente otimizador

---

- Ambiente: sem restrição
- Desvantagem: não tem adaptabilidade
- Ex. alguns motoristas do Brasil  
Segurança e velocidade – conflito!

# Agente que aprende (*learning agent*)

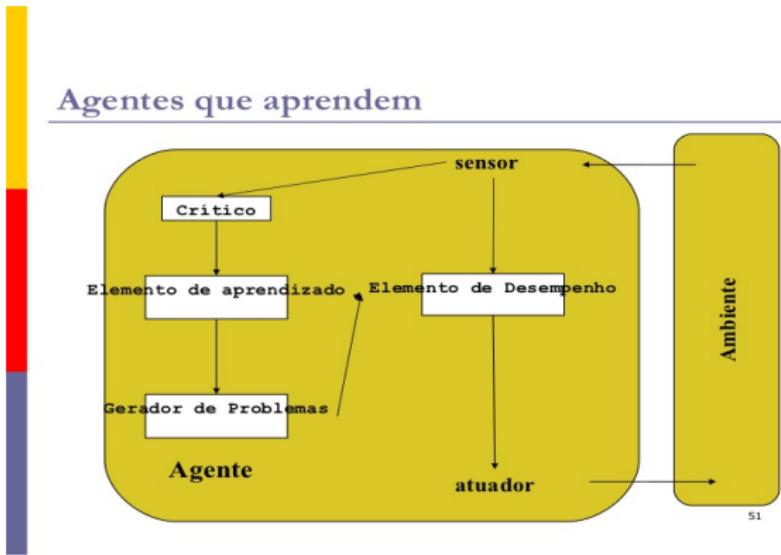


Figura: Agente que aprende – seguem os *copyrights*

## Agente que aprende

---

- Ambiente: sem restrição
- Vantagem: tem adaptabilidade (aprende)
- Ex. motoristas em um GPS

# Metodologia de desenvolvimento destes agentes racionais

---

**PEAS** = Peformance, Ambiente (*Enviroment*), Ação e Sensores

**Peformance:** como ter um indicativo de sucesso

**Ambiente:** cuidado deve ser sempre especificar o ambiente

**Ação:** o que vai fazer?

**Sensores:** o que vai perceber?

**Outros:** agentes da comunidade (falta isto na construção de agentes), ou seja um SMA!

# Exemplo ao avaliar (projetar ...) SMAs

---

Agent	Performance Measure	Environment	Actuators	Sensors
Playing Soccer	Score, Injuries, Team work.	Players, Referees, Field, Crowd, Goals, Ball.	Strength, Stamina, Coordination.	Eyes, Ears, Mouth, Ears, Touch.
Exploring Titan	Underwater mobility, Safety, Data, Navigation.	Shuttle, Rover, Atmosphere, Surface, Ocean.	Communication, Sustainability, Reliability.	Camera, GPS, Temperature, Pressure.
AI Book Shopping	Prices, Ease of site, Shipping time.	Websites, Internet, PC.	Correct Information, User.	Pictures, Information, Eyes.
Playing Tennis	Scoring, Stamina, Team work, Strategy.	Players, Referees, Crowd, Net, Court, Ball.	Strength, Stamina, Coordination.	Eyes, Skill, Footwork.
Practicing Tennis	Stamina, Lowering missed balls.	Player, Wall, Racket, Ball.	Stance, Racket Placement, Speed.	Eyes, Skill, Footwork.
High Jump	Form, Height, Landing.	Height bar, Padded Mat, Judge, Field.	Speed, Form, Leg Strength, Flexibility.	Eyes, Touch.
Knitting a Sweater	Correct Dimension, Reducing mistakes.	Yarn, Needles, Instructions, Room.	Speed, Yarn type, Sweater size, Precision.	Eyes, Hands.
Auction Bidding	Winning, Paying lowest price.	Opponents, Item, Auctioneer.	Budget, Item Value, Eagerness.	Eyes, Ears, Mouth, Knowledge of item.

Figura: Um bom exercício para reflexão – seguem os *copyrights*

# Exemplo ao avaliar (projetar ...) SMAs

## Outros exemplos

Tipo de Agente	Medida de desempenho	Ambiente	Atuadores	Sensores
Sistema de diagnóstico médico	Paciente saudável, minimizar custo, processos judiciais	Paciente, hospital, equipe	Exibir perguntas, testes, diagnósticos, tratamento, indicações	Entrada pelo teclado para sintomas, descobertas, respostas do paciente
Robô de seleção de peças	Percentagem de peças em bandejas corretas	Correia transportadora com peças; bandejas	Braço e mão articulados	Câmera, sensores angulares articulados

Figura: Um bom exercício para reflexão – seguem os *copyrights*

## Volte ao exemplo do agente policial – ver figura 8

---

- Quem é seu ambiente?
- Quem são seus sensores?
- Quem são seus atuadores?
- Qual é o seu mecanismo de raciocínio?
- Há outros agentes?
- Como é o seu ambiente (discreto, episódico..... – ver características do livro do Norvig–Russel)?
- Sua interação com o ambiente?
- Há outros agentes? Quais? Comunidades?

# Resumo do capítulo

---

1. Vocabulário
2. Tipos de agentes
3. Como iniciar um desenvolvimento de SMAs
4. Falta: porque soluções de SMAs são atrativas
5. Falta: diferença de SMAs com RDPs
6. Falta: Coordenação etc
7. Falta: Planejamento etc

Capítulo 3 – IAD  $\supseteq$  RDP  $\times$  SMA

## O que vai ter neste capítulo

---

- Mais conceitos sobre SMA dentro da IAD ... vantagens etc
- Quando a RDP é mais interessante (e quando não é)
- Vantagens da SMA (e desvantagens com relação a RDP)

# Porque distribuir?

---

- Sistemas em geral são distribuídos **funcionalmente**

# Porque distribuir?

---

- Sistemas em geral são distribuídos **funcionalmente**
  - Devido uma especificação (a necessidade ou requisitos)
  - Devido uma especialização
  - Dividir e diminuir a complexidade → decompor o problema

# Porque distribuir?

---

- Sistemas em geral são distribuídos **funcionalmente**
  - Devido uma especificação (a necessidade ou requisitos)
  - Devido uma especialização
  - Dividir e diminuir a complexidade → decompor o problema
- Sistemas em geral são distribuídos **fisicamente**

# Porque distribuir?

---

- Sistemas em geral são distribuídos **funcionalmente**
  - Devido uma especificação (a necessidade ou requisitos)
  - Devido uma especialização
  - Dividir e diminuir a complexidade → decompor o problema
- Sistemas em geral são distribuídos **fisicamente**
- Finalmente, na **resolução de problemas** (em geral): há alguns cuja solução é inherentemente distribuída ou fica mais fácil distribuindo!

# Motivando o *distribuído*

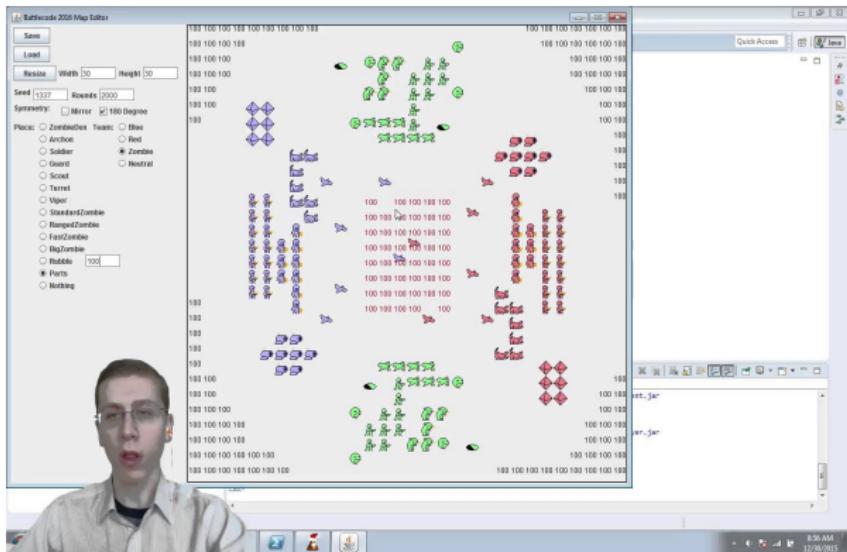


Figura: Ações paralelas, distribuídas, concorrentes ... e tudo coordenado?

# Motivando o distribuído

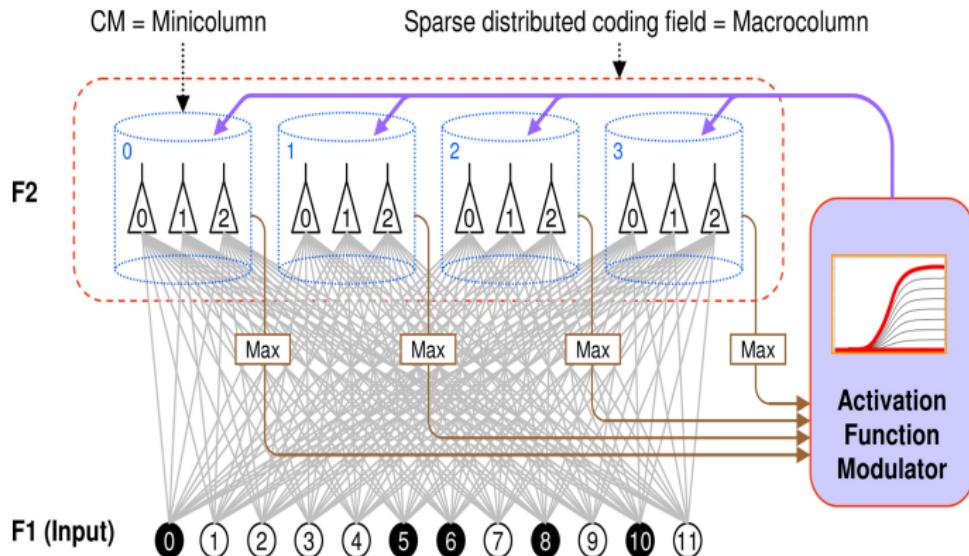
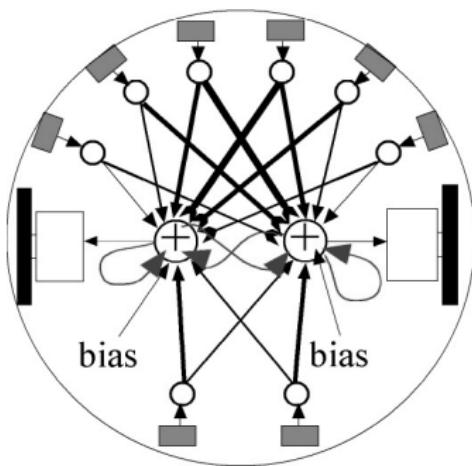


Figura: Mapeando funções via RN – representações funcionais – cérebro

## Motivando o *distribuído*

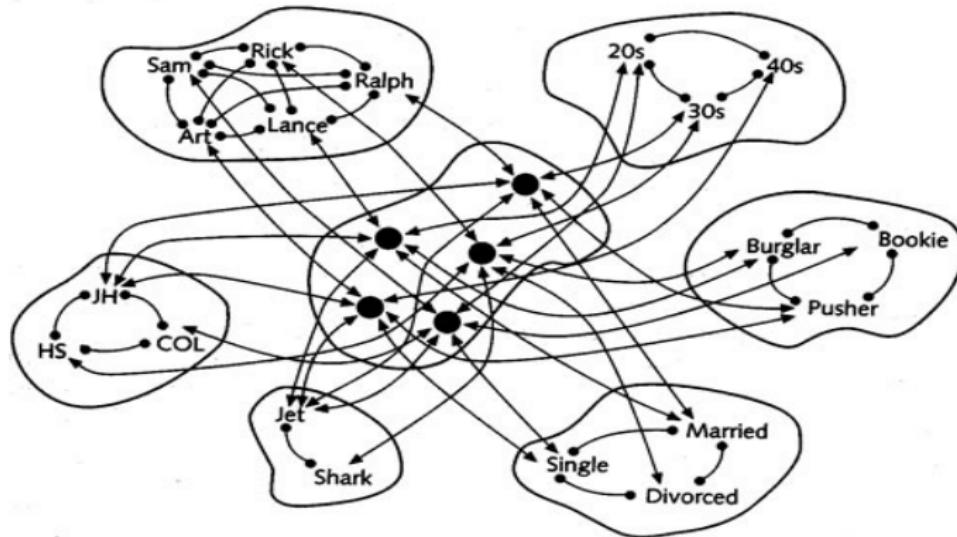
---



**Figura:** Robô: mapeando sensores funcionais em atividades motoras ⇒ Inspiração 100 % na entomologia – voltaremos a esta figura

Há muito cálculo *distribuído*, como os processos mentais:

---



## Motivando a distribuição:

---

- Porque o problema é fisicamente distribuído.
- Porque o problema é heterogêneo.
- Porque o problema só pode ser resolvido pela integração de pontos de vista locais.
- Porque precisamos de adaptação a mudanças estruturais...

## As vantagens da distribuição:

---

- Maior rapidez na solução dos problemas
- Diminuição do *overhead* de comunicação
- Maior flexibilidade
- Aumento da segurança – tolerância a falhas

# Motivando o *distribuído*

---



**Figura:** Ações paralelas, distribuídas, concorrentes ... e tudo planejado!

# IA Distribuída

---

- Entidades (ou várias) que interagem sob uma:
  - Organização (há uma conexão entre as partes)
  - Ação
  - Interação
- Metáfora usada de inteligência: **comportamento social** (sim, os dos seres animais, incluindo o *homo-sapiens!*)

## Resumindo a IAD

---

- Não é IA paralela (esta é voltada em parallelizar computacionalmente as implementações em IA), nem Sistemas Distribuídos.
- Um resolução grupal de problemas, através de *cooperação* (diferente de *colaboração*).
- Grande interatividade e capacidade de comunicação.
- Organização - meios que garantam a *convergência*: estruturas de autoridade e controle divididos.
- Divisão de conhecimento (nota: *o que é conhecimento?*) e recursos

# IA Distribuída: dois tipos de sistemas

---

- Resolução Distribuída de Problemas (RDP)
  - consciência do objetivo global e divisão clara de tarefas
  - Exemplos: robótica clássica, busca na Web, gerência de sistemas distribuídos, ...
- Sistemas Multiagentes (SMA)
  - não consciência do objetivo global e nem divisão clara de tarefas
  - Exemplos: n-puzzle, futebol de robôs, balanceamento de carga, robótica, ...

## Porque usar a metáfora de agentes?

---

- Fornece metodologias de desenvolvimento de sistemas inteligentes estendendo as de engenharia de software
- Fornece visão unificadora das várias sub-áreas da IA
- Ajuda a embutir a IA em sistemas computacionais tradicionais
- Permite tratar melhor a interação com ambiente
- Permite tratamento natural da IA distribuída (distribuir!!!)

## Ainda RDP × SMA

---

- RDP:
- Um grupo de especialistas
  - Habilidades Complementares
  - Organização Fixa

- SMA:
- Agentes podem preexistir
  - Organização varia em tempo de execução

## Fechando esta relação ... RDP × SMA

---

- RDP:
  - RDP é um subconjunto de SMA
  - Agentes benevolentes, concebidos em conjunto
- SMA:
  - SMA é base para RDP
  - Implementação descentralizada de várias propriedades

# Um Sistema Multiagente (SMA) formal

---

Um SMA é um sistema que possui os seguintes elementos:

- Um ambiente:  $E$
- Um conjunto de objetos:  $O$
- Um conjunto de Agentes:  $A$  ( $A \subseteq O$ )
- Um conjunto de relações  $R$ , a qual estabelece conexões entre os objetos
- Um conjunto de operações:  $O_p$
- Operadores que representam os resultados das operações em  $O_p$  e as reações do ambiente a eles.

Construa uma tupla para esta formalização

Isto é:

---

Um SMA:

- Consiste de uma coleção de componentes autônomos, com objetivos particulares
- Que se interrelacionam
  - De acordo com uma Organização
  - Interagindo, negociando e coordenando esforços para resolver tarefas

## Exemplos de uso de SMAs:

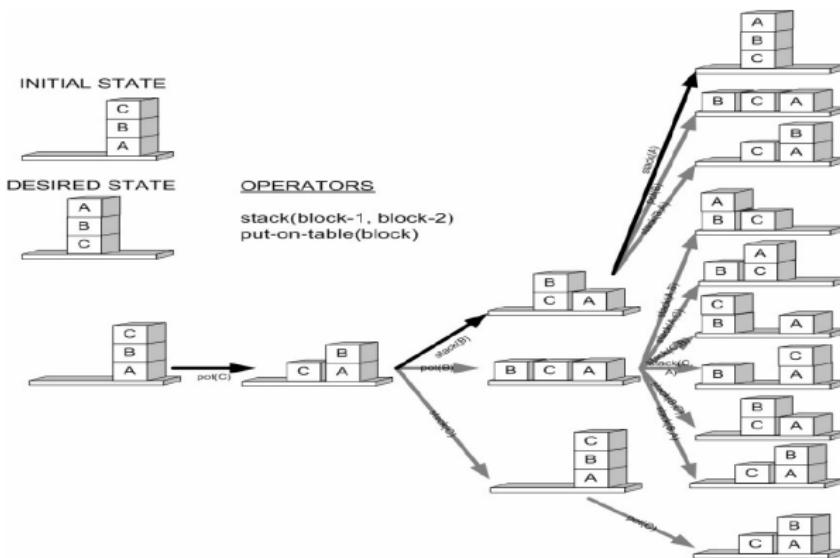


Figura: *Mundo dos blocos* – um agente neste caso é previsível

## Exemplos de uso de SMAs:

---



Figura: *Mundo dos blocos* com vários robôs

## Exemplos de uso de SMAs:

---

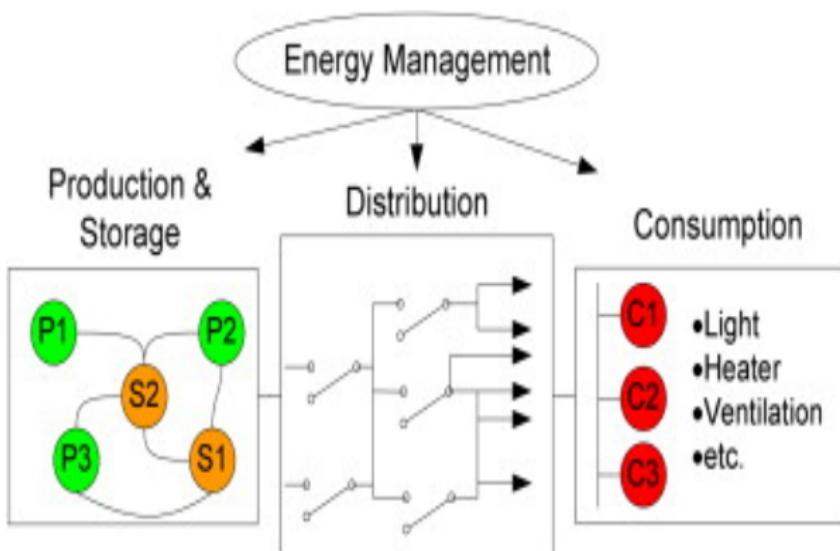


Figura: As chaves comutadoras são agentes?

## Exemplos de uso de SMAs:

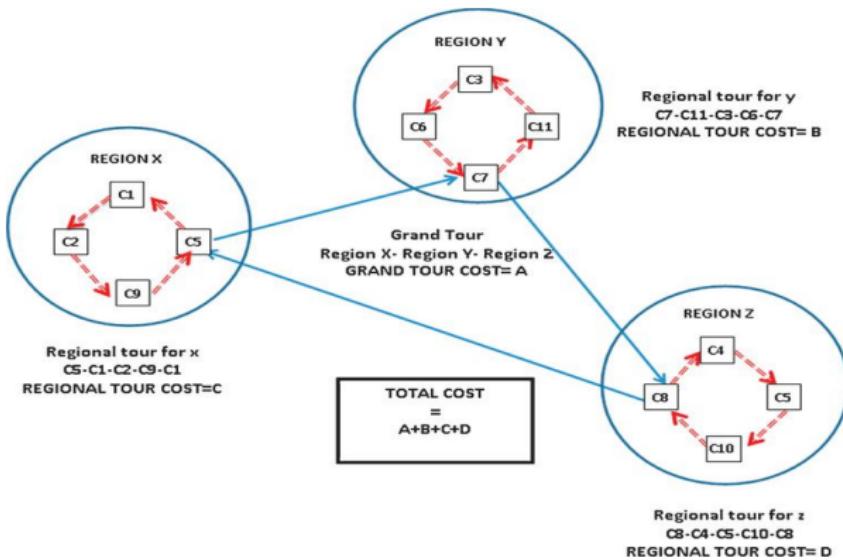


Figura: Removendo a dinâmica das regiões este é um exemplo de RDP

Em resumo... é uma boa idéia quando...:

---

- Precisamos manter a autonomia das sub-partes
- As interações são complexas
- Não é possível descrever o problema *a priori*.

## E as vantagens...:

---

1. Maior rapidez na solução dos problemas
  2. Diminuição do *overhead* de comunicação
  3. Maior flexibilidade
  4. Aumento da segurança – tolerância a falhas
- Uma vez definidos e motivados, agora faltam os detalhes de como implementar tudo isto!
  - Ver seção de projetos de SMAs!

Epílogo: um agente já tem uma realidade *complexa*:

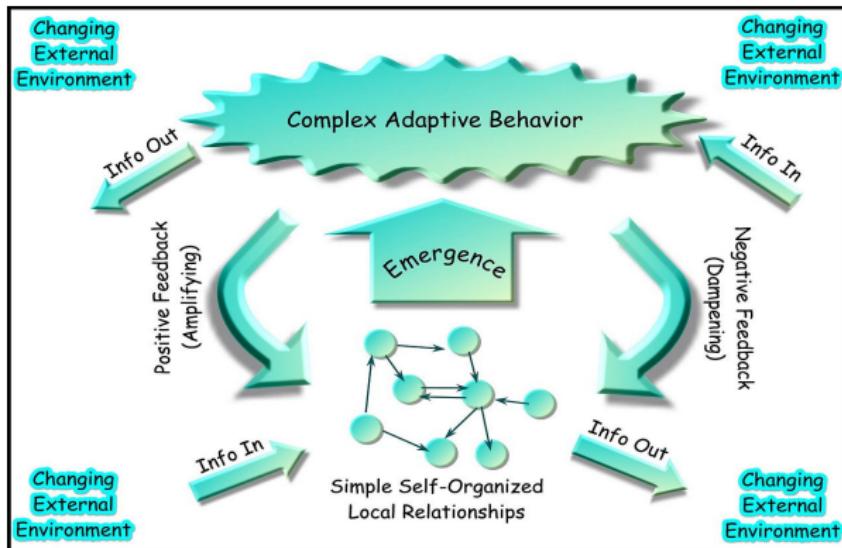


Figura: Um agente e sua *sociedade de agentes* ...

# Resumo do capítulo

---

1. Vocabulário
2. Diferença de SMAs com RDPs na IAD
3. Soluções de SMAs são atrativas porquê
4. Falta: Coordenação etc
5. Falta: Planejamento etc
6. Falta: projetar SMAs

# Capítulo 4 – Teoria de Jogos e Coordenação

3 partes fundamentais:

1. Teoria de Jogos
2. Coordenação
3. Planejamento

Nesta ordem

# Teoria de Jogos

---

# Teoria de Jogos

---

# Teoria de Jogos

---

- 
-

# Teoria de Jogos Aplicado a SMA

---

# Teoria de Jogos Aplicado a SMA

---

- $\prod_{x=1}^n \neq \prod_{x=1}^{n+1}$
- <https://www.codecogs.com/latex/eqneditor.php>
- <http://www.hostmath.com/>

# Teoria de Jogos Aplicado a SMA

---

- $\prod_{x=1}^n \neq \prod_{x=1}^{n+1}$
- <https://www.codecogs.com/latex/eqneditor.php>
- <http://www.hostmath.com/>
-

# Coordenação

---

# Jogos de Coordenação

---

pag 23

# Convenção Social

---

pag 24

# Papel Social

---

pag 25

# Grafos de Coordenação

---

pag 26

## Coordenação por Eliminação de Variáveis

---

pag 28

## Coordenação por Troca de Mensagens

---

pag 28

# Fundamentos de Planejamento

---

# Abordagens ao Planejamento de SMAs

---

- Coordenação central: controla todos os subplanos
- Esquemas de controle distribuído
  - Conhecimento parcial dos planos de outros agentes
- Planejamento Global Negociado
  - Compartilhamento de todos os planos
  - Ajuste local para a realização de objetivos comuns
- Modelagem Explícita da Equipe de Agentes
  - Compromissos conjuntos
  - Crenças, desejos e intenções comuns

## Exemplo de Coordenação SMAs

---

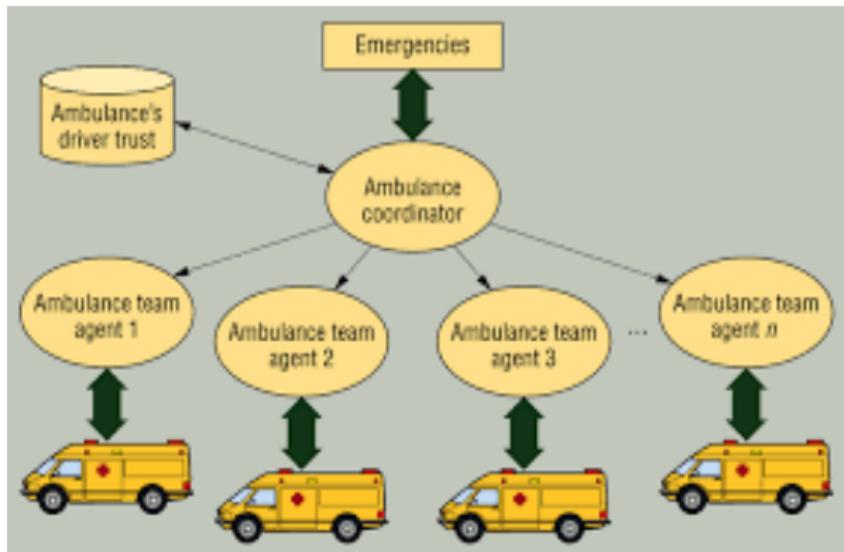


Figura: Coordenação de agentes  $\equiv$  SMA

# Exemplo de Coordenação SMAs

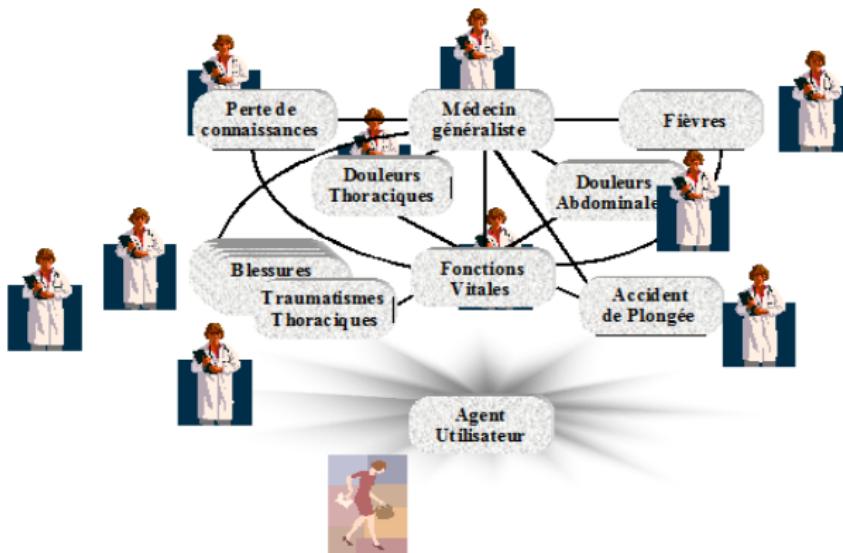


Figura: Coordenação de agentes  $\equiv$  SMA

# Capítulo 5 – Projetos de SMAS

# O que vai ter neste capítulo

---

- Alguma metodologia?
- Ambiente simulado: SIM (implementar)
- O que considerar na construção de SMAs
- Perspectivas: reais e visionárias

## Agentes: Metodologia de desenvolvimento

---

## Agentes: Metodologia de desenvolvimento

---

- Decompõem o problema em:  
percepções, ações, objetivos, ambiente e outros agentes

# Agentes: Metodologia de desenvolvimento

---

- Decompõem o problema em:  
percepções, ações, objetivos, ambiente e outros agentes
- Decompõem tipo de conhecimento em:
  - Quais são as propriedades relevantes do mundo?
  - Como o mundo evolui?
  - Como identificar os estados desejáveis do mundo?
  - Como interpretar suas percepções?
  - Quais as consequências de suas ações no mundo?
  - Como medir o sucesso de suas ações?
  - Como avaliar seus próprios conhecimentos?
- O resultado dessa decomposição indica a arquitetura e o método de resolução de problema (raciocínio)

Em geral a construção do agent segue um programa tal como:

```
funcao simulaAmbiente (estado, funcaoAtualizacao, agentes, final) repita
para cada agente em agentes faça Percept[agente] :=
    pegaPercepcao(agente,estado)
para cada agente em agentes faça
    Action[agente] := Programa[agente] (Percept[agente])
    estado := funcaoAtualizacao(acos, agentes, estado)
    scores := avaliaDesempenho(scores,agente,estado)
    opcional atehe_final
```

Cuidado para não cair em tentação e "roubar" do ambiente a descrição do que aconteceu. Usar a memória do agente!

# Desenvolvendo um agente inteligente:

---

- Projeto**
- Modelar tarefa em termos de ambiente, percepções, ações, objetivos e utilidade
  - Identificar o tipo de ambiente
  - Identificar a arquitetura de agente adequada ao ambiente e tarefa

- Implementação**
- O simulador de ambientes
  - Componentes do agente
  - Testar o desempenho com diferentes instâncias do ambiente

# Simulação de Ambientes

---

ESTRUTURAS ... ver NORVIG – pg 34 da T

# Simulação de Ambientes

---

ESTRUTURAS ... ver NORVIG – pg 34 da T

# Capítulo 6 – Conclusões