



EA 072 Inteligência Artificial em Aplicações Industriais

2-Agentes Inteligentes

2.1 Agentes Inteligentes

- Agentes são sistemas que:
 - aumentam capacidade de resolver problemas
 - funcionam de forma autônoma em ambientes complexos
- Complexidade: complexidade da informação

■ Aplicações

- gerenciamento de tráfego e serviços de transporte
- controle de redes de comunicação e de computadores
- sistemas de potência
- gerenciamento de transações de bancos, comércio
- monitoração, supervisão e controle de processos industriais
- sistemas de manufatura
- atendimento e serviços médicos, etc.

Engenharia e Agentes Inteligentes

- Natureza da inteligência, como descrever comportamentos inteligentes
- Objetivos múltiplos, possivelmente conflitantes e dinâmicos
- Crença do agente sobre o passado, presente e futuro
- Aquisição de informação sobre tarefas antes/durante escolha plano/ação

- Conhecimento incompleto/impreciso do ambiente
- Hipóteses sobre ocorrências de eventos e comportamento de agentes
- Dinâmica do ambiente e dos outros agentes.
- Modelos, etc.....

Questões

- 1–Como encontrar a informação que é relevante para o problema, como utilizar esta informação para resolver o problema ?
- 2–Classes de inferências e mecanismos de acesso ?
- 3–Garimpar dados, bases de dados de grande porte, como ?
- 4–Interação entre agentes (sistemas multiagentes) ?

- 5–Representações compartilhadas de informação e relações em um domínio ?
- 6–Linguagens e mecanismos de comunicação sobre a própria informação e não somente estruturas de dados e seus conteúdos.
- 7–Coordenação: protocolos utilizados pelos agentes para obter um acordo sobre o comportamento que o sistema deverá exibir ?

Definição de Agente

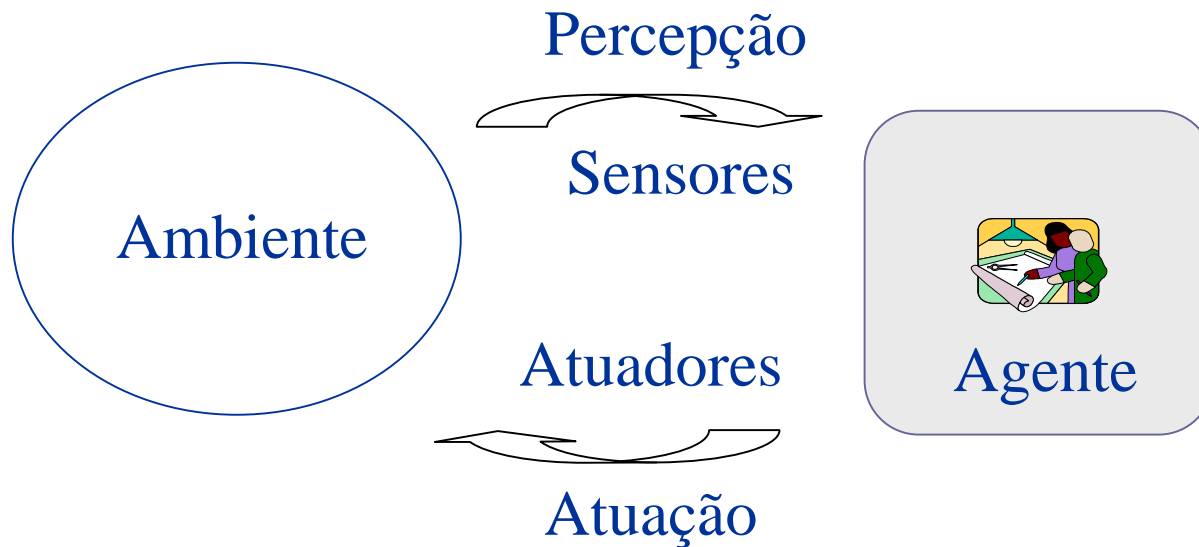
■ Agente

qualquer entidade que perceba um ambiente através de sensores, executem tarefas de processamento de informações e de conhecimento, e atue sobre o ambiente através de atuadores (Russel & Norvig 2003).

■ Exemplos

- robôs
- *softbots*
- veículos autônomos

Agente = Arquitetura + Programa



■ Características

1–Direção

- capaz de receber diretrizes de usuários e outros agentes

2–Autonomia

- comportamento determinado pela própria experiência

3–Persistência

- capaz de operar por longos períodos sem atenção

4–Confiabilidade

- desempenho de acordo com expectativas e objetivos

5–Antecipação

- antecipar necessidades, resultados, informações, aprendizagem

6–Pró-atividade

- iniciativa para resolver problemas, agregar informação.

7–Comunicação e interação

- comunicação e interação com outros sistemas e agentes;
- resolução de conflitos; competição, cooperação.

8–Flexibilidade

- tratar estruturas heterogêneas de agentes e fontes de informação

9–Adaptação

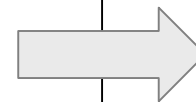
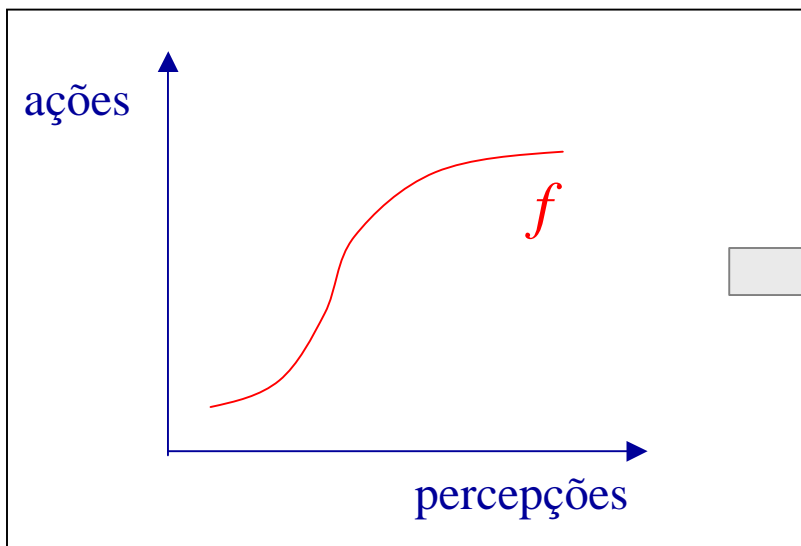
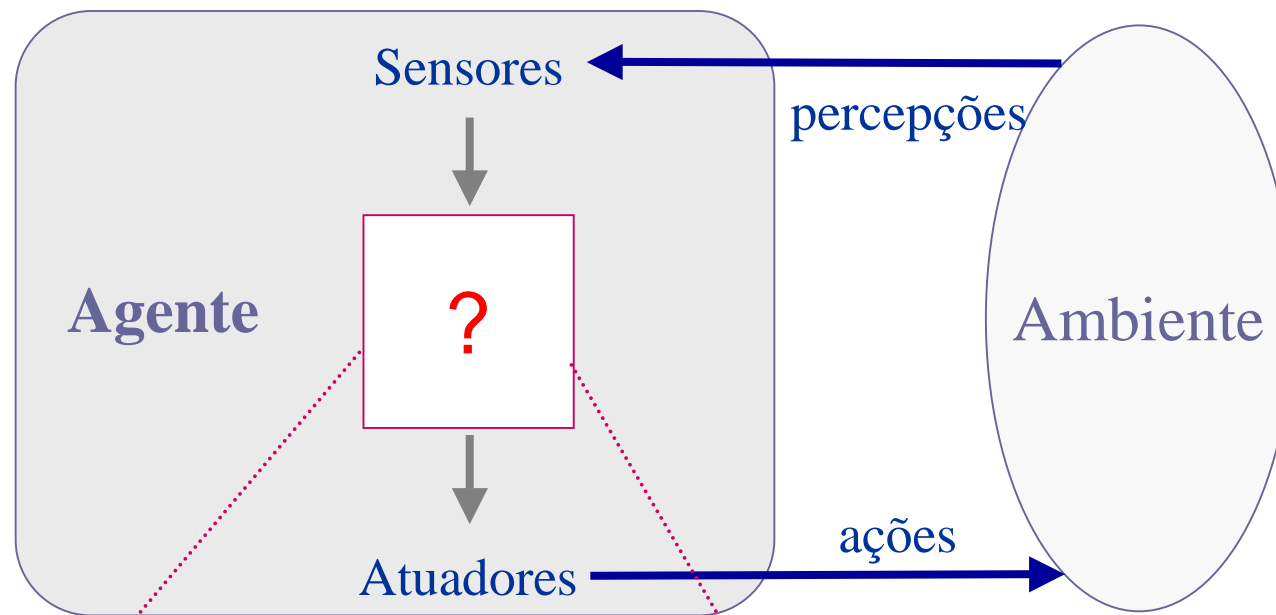
- acomodação de modificações de objetivos, tarefas, aprendizagem.

- Agente racional é um conceito central em IA:
 - conceito aplicado a variedade de aplicações/ambientes
 - útil para desenvolver um conjunto de princípios de projeto
 - construir agentes com chances de sucesso
 - agentes (razoavelmente) inteligentes

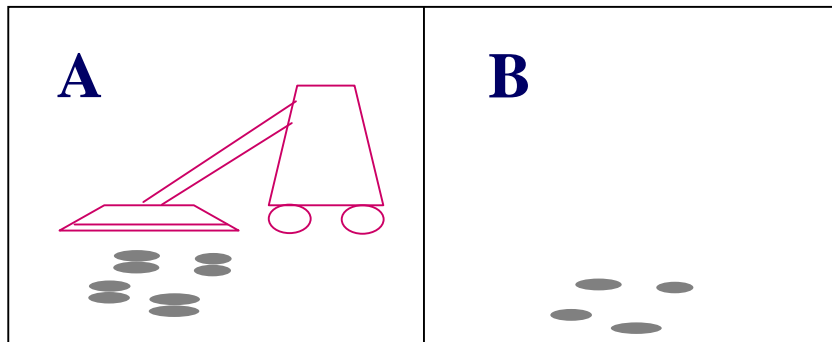
2.2 Agente Racional

- Racionalidade depende:

- 1 - Critério (medida) de desempenho que define o grau de sucesso
- 2 - Tudo que o agente percebeu: história (memória) perceptiva
- 3 - Conhecimento do ambiente (o que o agente sabe sobre o ambiente)
- 4 - Ações que o agente pode executar



Função de Agente
↓
Programa de Agente



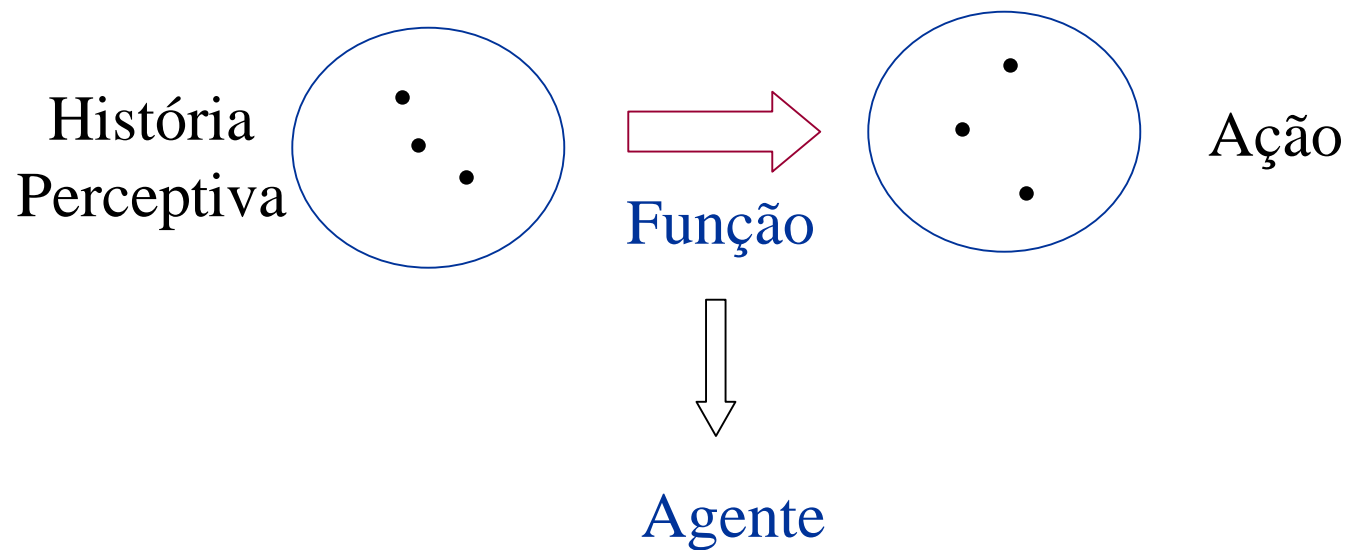
Sequência de <i>percepts</i>	Ação
[A, <i>Clean</i>]	<i>Right</i>
[A, <i>Dirty</i>]	<i>Suck</i>
[B, <i>Clean</i>]	<i>Left</i>
[B, <i>Dirty</i>]	<i>Suck</i>
[A, <i>Clean</i>], [A, <i>Clean</i>]	<i>Righ</i>
[A, <i>Clean</i>], [A, <i>Dirty</i>]	<i>Suck</i>
.....
[A, <i>Clean</i>], [A, <i>Clean</i>], [A, <i>Clean</i>]	<i>Righ</i>
[A, <i>Clean</i>], [A, <i>Clean</i>], [A, <i>Dirty</i>]	<i>Such</i>
.....

Agente Racional

Para cada sequência em sua história perceptiva, um agente racional deve realizar qualquer ação que maximize seu critério de desempenho, levando em consideração as evidências proporcionadas pela sequência perceptiva e qualquer outro conhecimento que o agente tenha disponível.

- **Mapeamentos definem agentes racionais:** a especificação das ações a serem realizadas pelos agentes em resposta a uma sequência perceptiva define um *projeto* de um agente racional.
- A racionalidade maximiza o desempenho esperado enquanto que a perfeição maximiza o desempenho real (Russel & Norvig, 2003)
- Onisciência, aprendizagem e autonomia
- Racional \neq onisciente
- Racional: autonomia + aprendizagem

Projeto de Agentes



2.3 Projetar Agentes

- Demanda uma boa idéia (do que é possível) a respeito de:
 - percepções e ações
 - critérios de desempenho e/ou metas a serem atingidas
 - ambiente
- **PEAS**: *Performance, Environment, Actuators, Sensors*,
- Real \times Artificial : robôs \times *softbots* (agentes de software)

Exemplos de Agentes (PEAS)

TIPO AGENTE	PERFORMANCE	ENVIRONMENT	ACTUATORS	SENSORS
Diagnóstico Médico	Paciente Saudável, Custos	Paciente, Hospital, pessoal	Perguntas, Testes, Exames Tratamentos	Sintomas, respostas do paciente, etc.
Análise Imagens de Satélites	Categorização correta	Downlink do satélite em órbita	Display de uma categorização da cena	Pixels de intensidade e cor variada
Robô de Manuseio	Percentagem de peças no lugar correto	Correia transportadora com peças	Juntas com braço e mão	Câmera, sensores de posição angular
Controle de Processos	Maximizar pureza, produção, segurança	Refinaria, operadores	Válvulas, bombas, aquecedores	Temperatura, pressão, vazão, químicos
Tutor de Linguas	Maximizar nota dos testes	Conjunto de estudantes em teste	Display de exercícios, sugestões, correções	Entrada por teclado

Formato Geral de um Agente

```
function SKELETON_AGENT(percept) returns an action  
  persistent: memory, a memória que o agente tem do mundo  
  
  memory  $\leftarrow$  UPDATE_MEMORY (memory, percept)  
  action  $\leftarrow$  CHOOSE_BEST_ACTION(memory)  
  
return action
```

- em cada instante o agente recebe um único *percept* como entrada
- projeto do agente decide sobre a história perceptiva (*memória*)
- critério de desempenho não faz parte do formato geral

Exemplo simples

function TABLE_DRIVEN__AGENT(*percept*) **returns** an action
 persistent: *percepts*, sequencia, inicialmente vazia
 table, tabela de ações indexada por percepts, inicia cheia

 append *percept* to the end of *percepts*
 action \leftarrow LOOKUP(*percepts*, *table*)

return *action*

2.4 Estruturas de Agentes

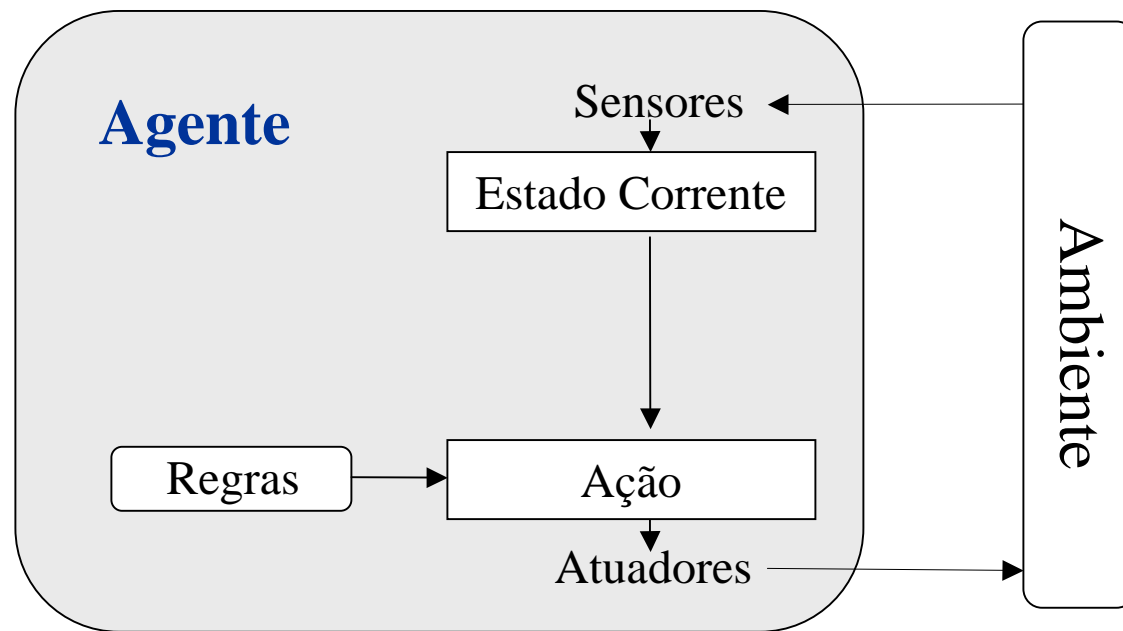
- Centralizadas

- Agente Reativo
- Agente Dinâmico
- Agente Orientado por Metas
- Agente Orientado por Utilidade

■ Distribuídas

- Agentes Interface
- Agentes Tarefa
- Agentes Informação

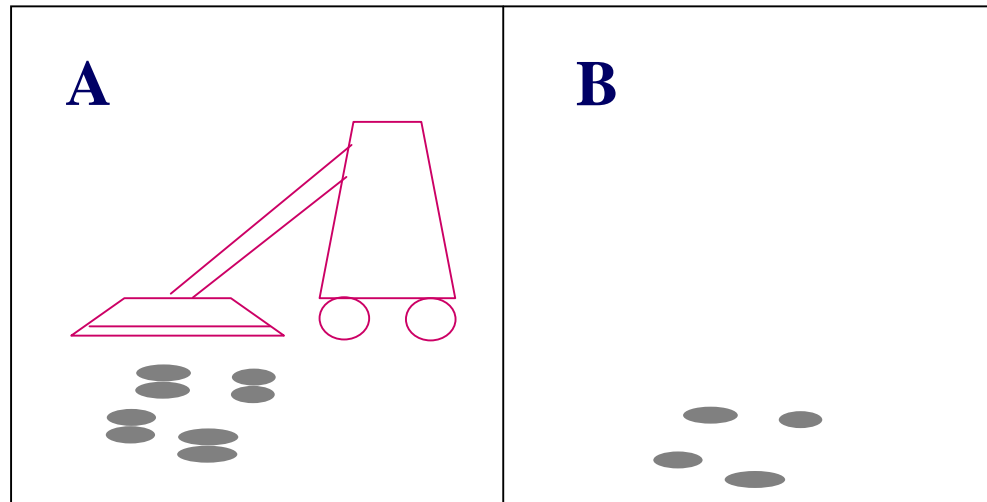
Agente Reativo



Programa de Agente (Reativo)

function SIMPLE_REFLEX_AGENT (*percept*) **returns** an action
persistent: *rules*, conjunto de regras condição-ação

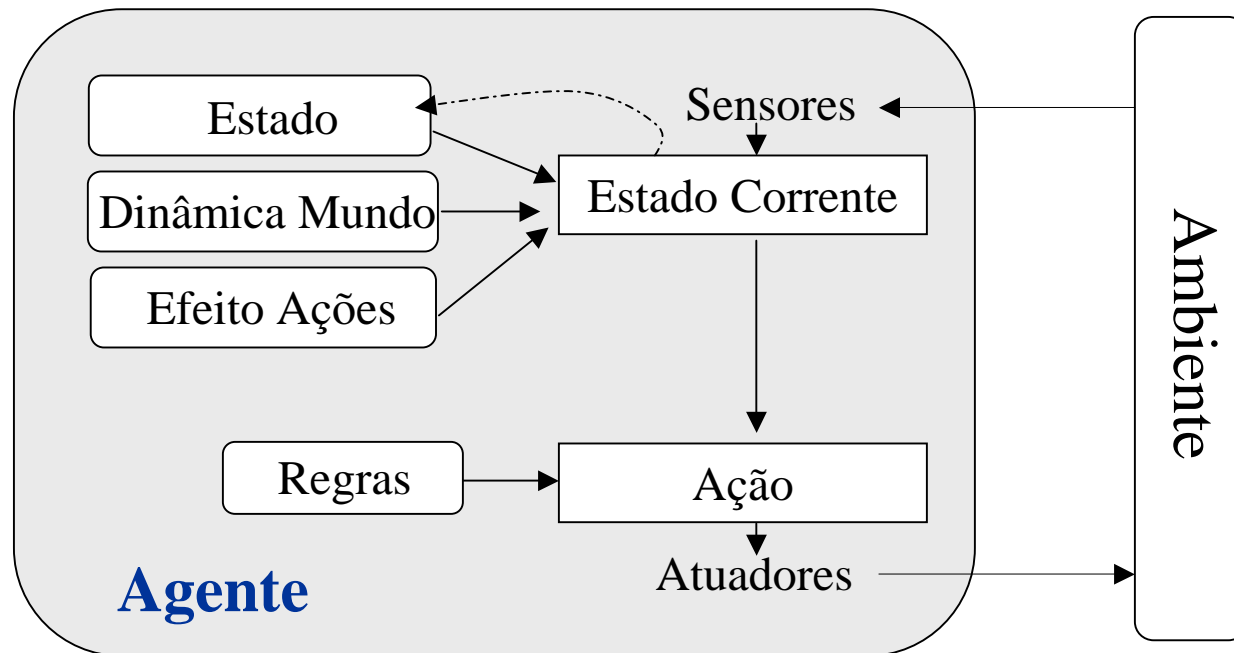
```
state ← INTERPRET_INPUT(percept)  
rule ← RULE_MATCH(state, rules)  
action ← RULE_ACTION [rule]  
return action
```



function REFLEX-VACUUM-AGENT (*[location, status]*) **returns** an action

if *status* = *Dirty* **then return** *Suck*
else if *location* = A **then return** *Right*
else if *location* = B **then return** *Left*

Agente Dinâmico (Estado Interno)

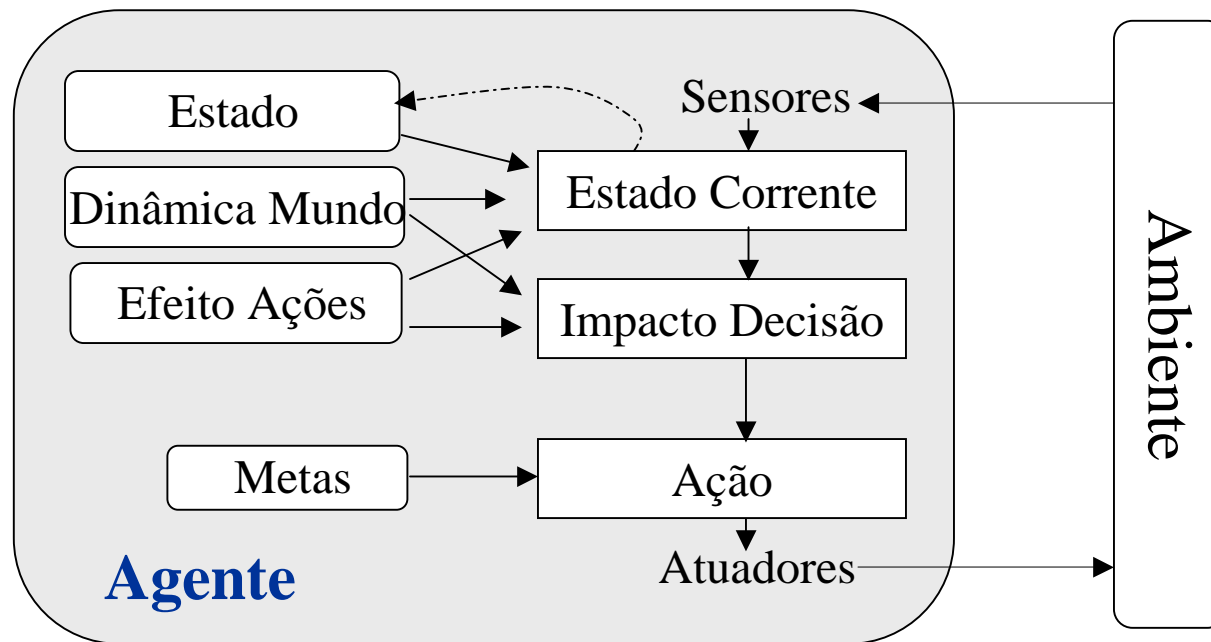


Programa de Agente (Dinâmico)

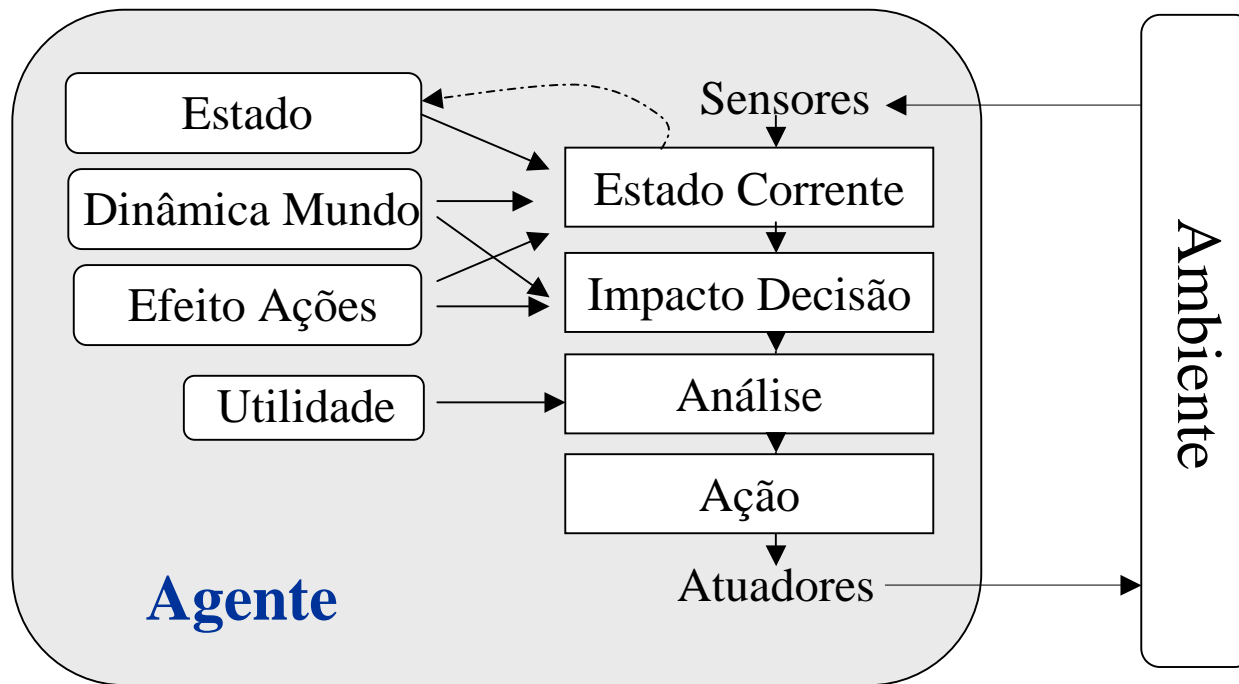
function MODEL_BASED_REFLEX_AGENT (*percept*) **returns** an action
persistent: *rules*, conjunto de regras condição-ação
state, descrição do estado corrente do mundo
action, ação mais recente, inicialmente vazia

state ← UPDATE_STATE(*state*, *action*, *percept*, *model*)
rule ← RULE_MATCH(*state*, *rules*)
action ← RULE_ACTION[*rule*]
return *action*

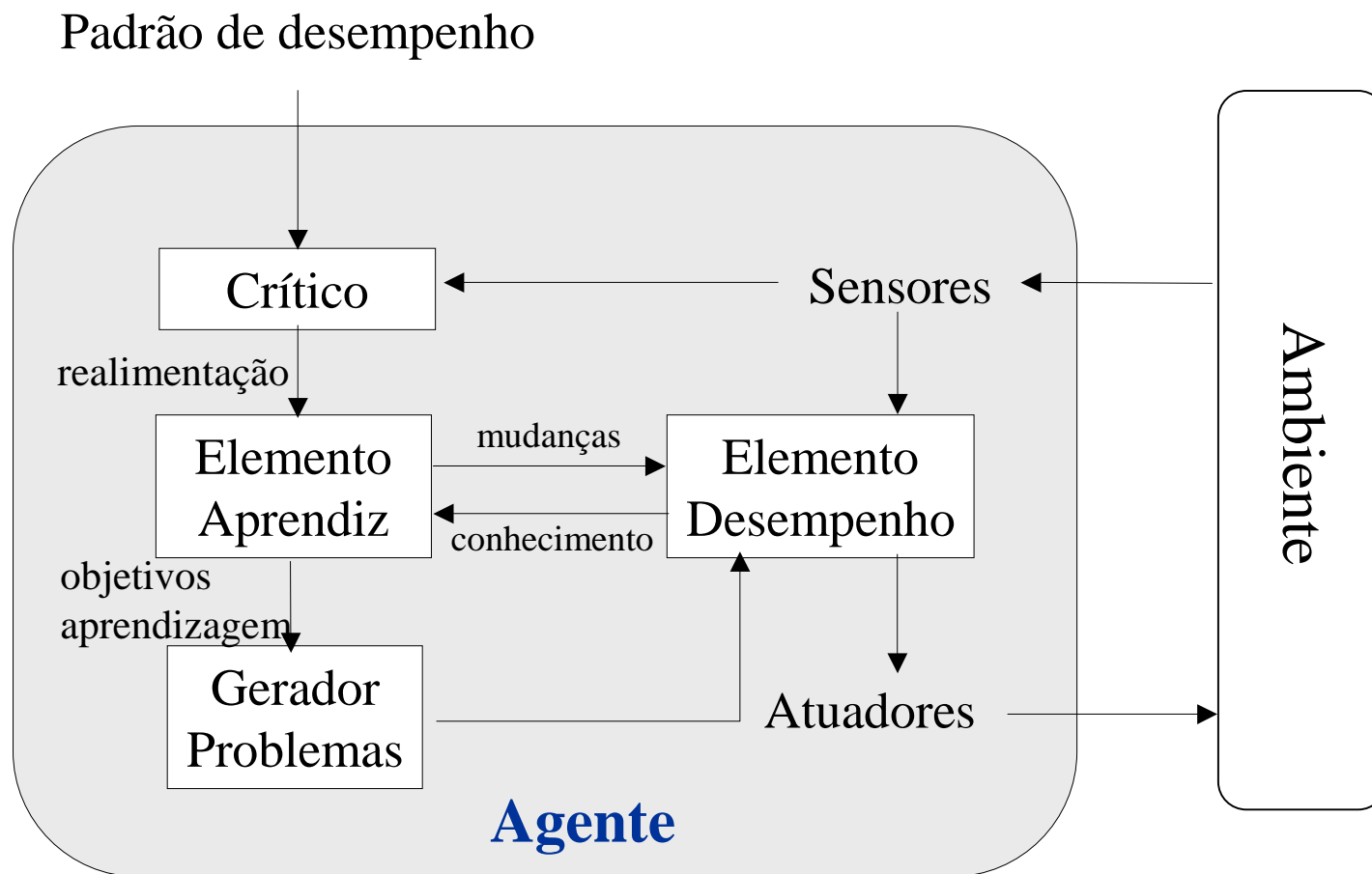
Agente Orientado por Metas



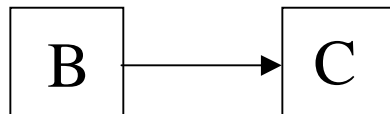
Agente Orientado por Utilidade



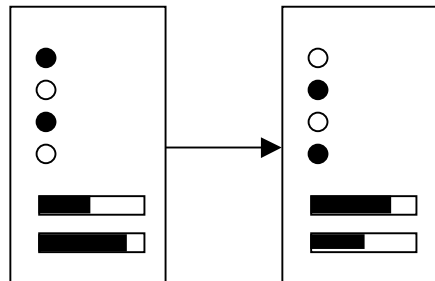
Agente com Aprendizagem



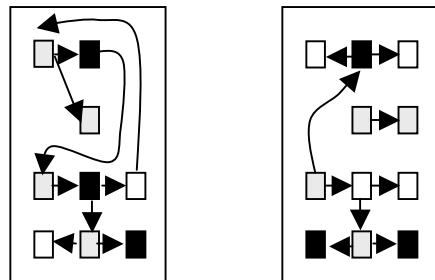
Representação de componentes de agentes



Atômica: sem estrutura interna
e.g. busca, jogos, HHM

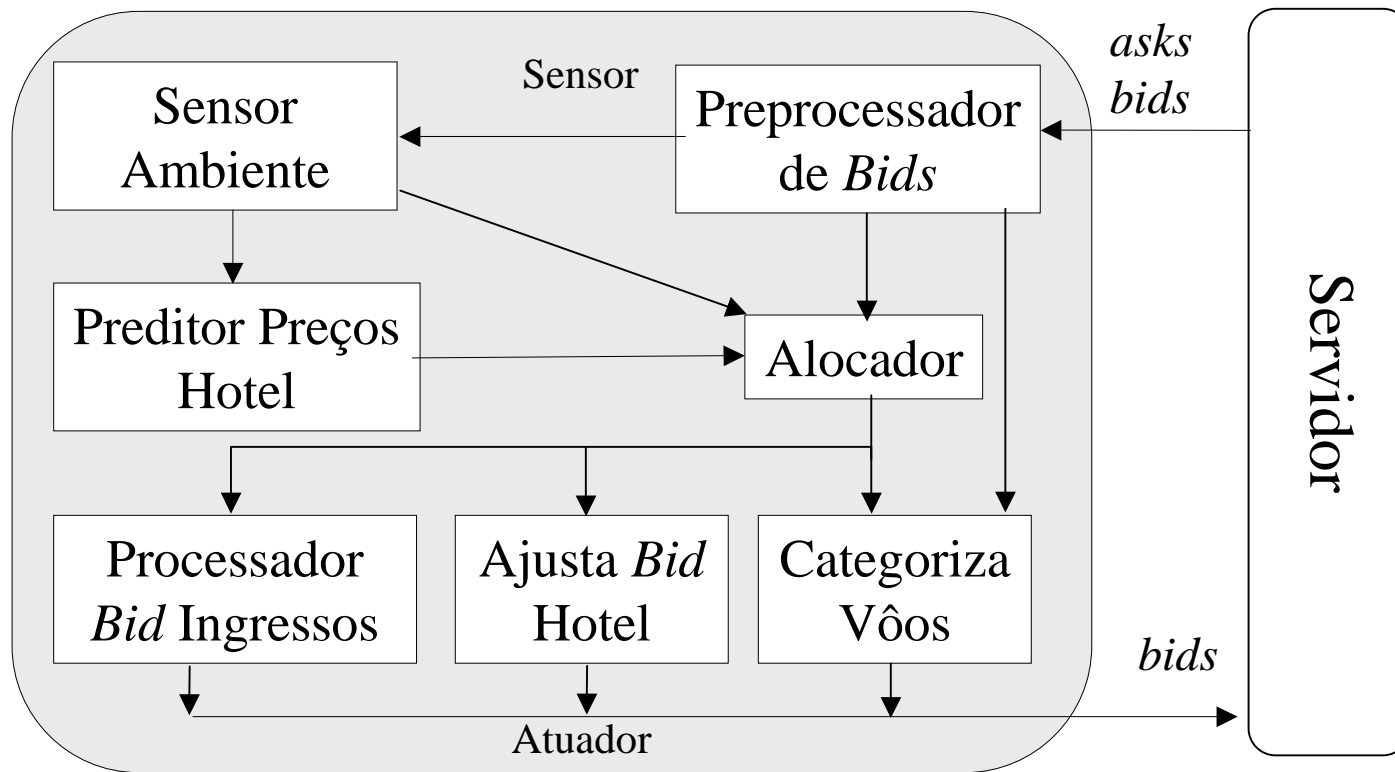


Fatorada: variável, atributo, valor
e.g. Bayes net, CS, LP, ML

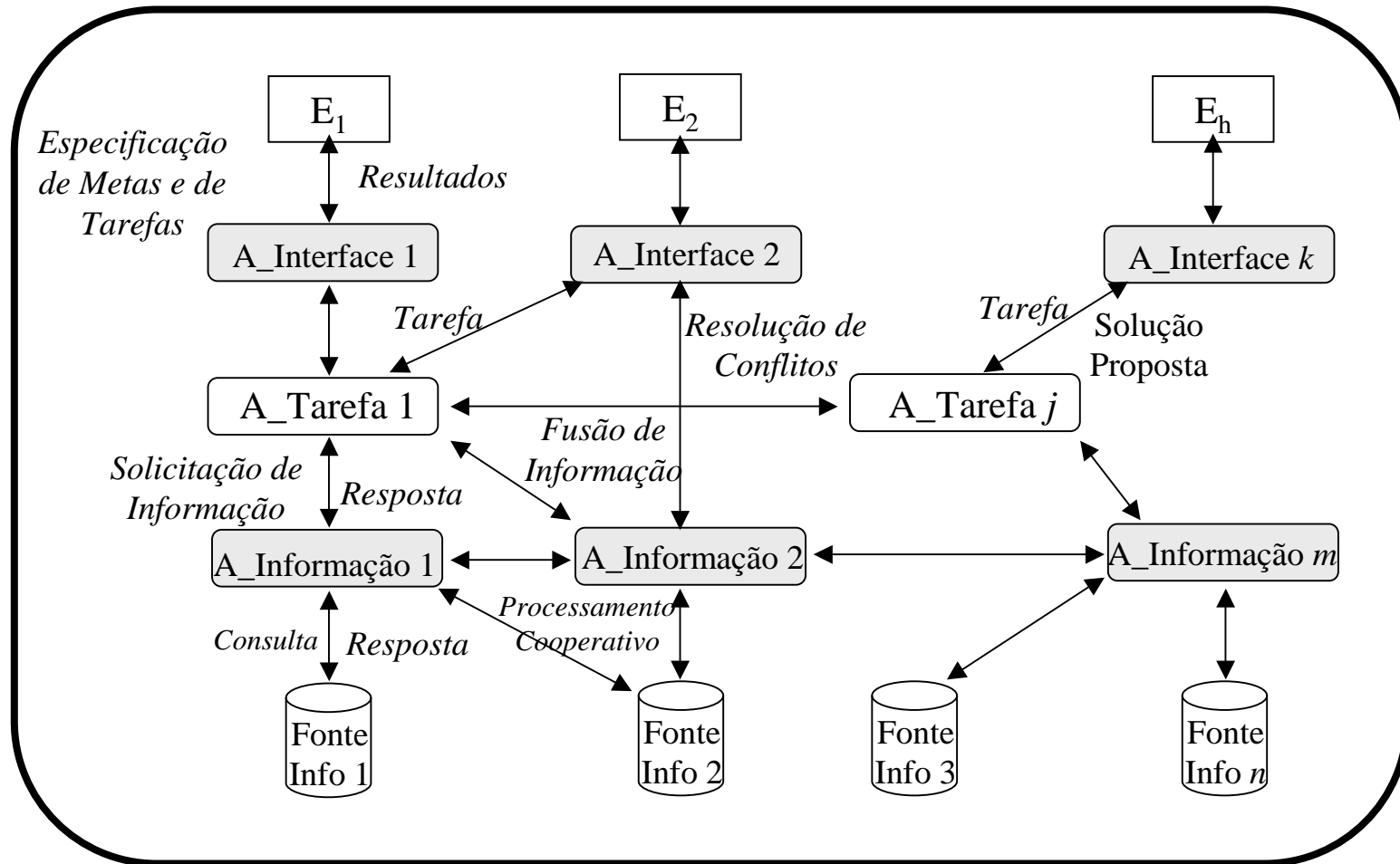


Estruturada: objetos, relações
e.g. LPO,DB, KBL, NL

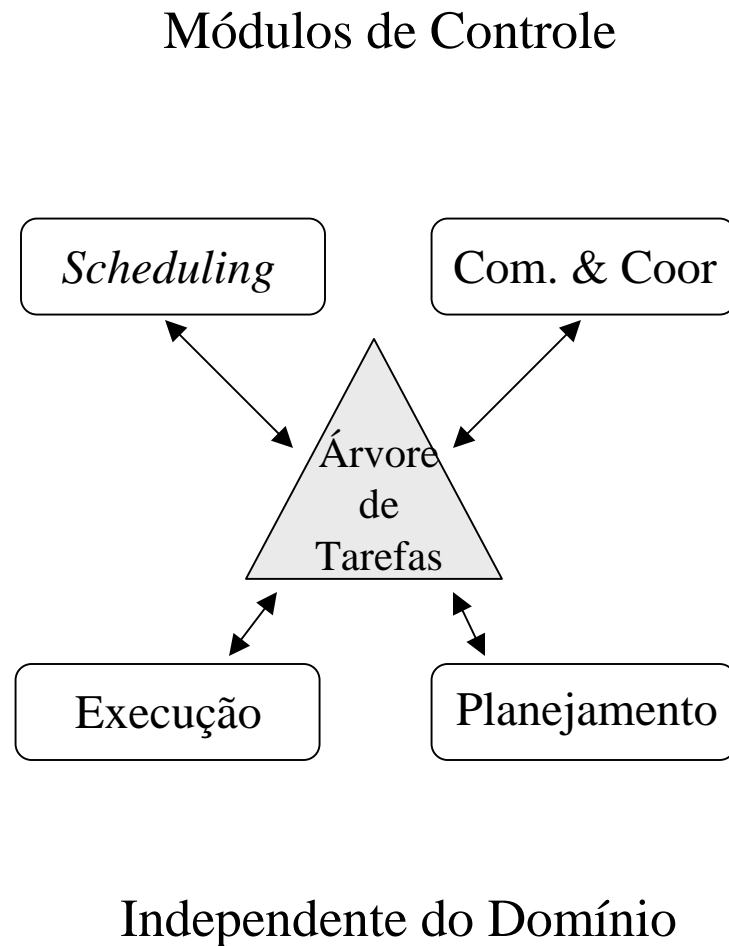
Exemplo: Fuzzy Trading Agent



Exemplo de Estrutura Distribuída



Estrutura Distribuída: Arquitetura Funcional



Conhecimento do Agente

Fragmentos de Planos
Independente do Domínio

Fragmentos de Planos
Dependente do Domínio

Beliefs
e
Base de Fatos

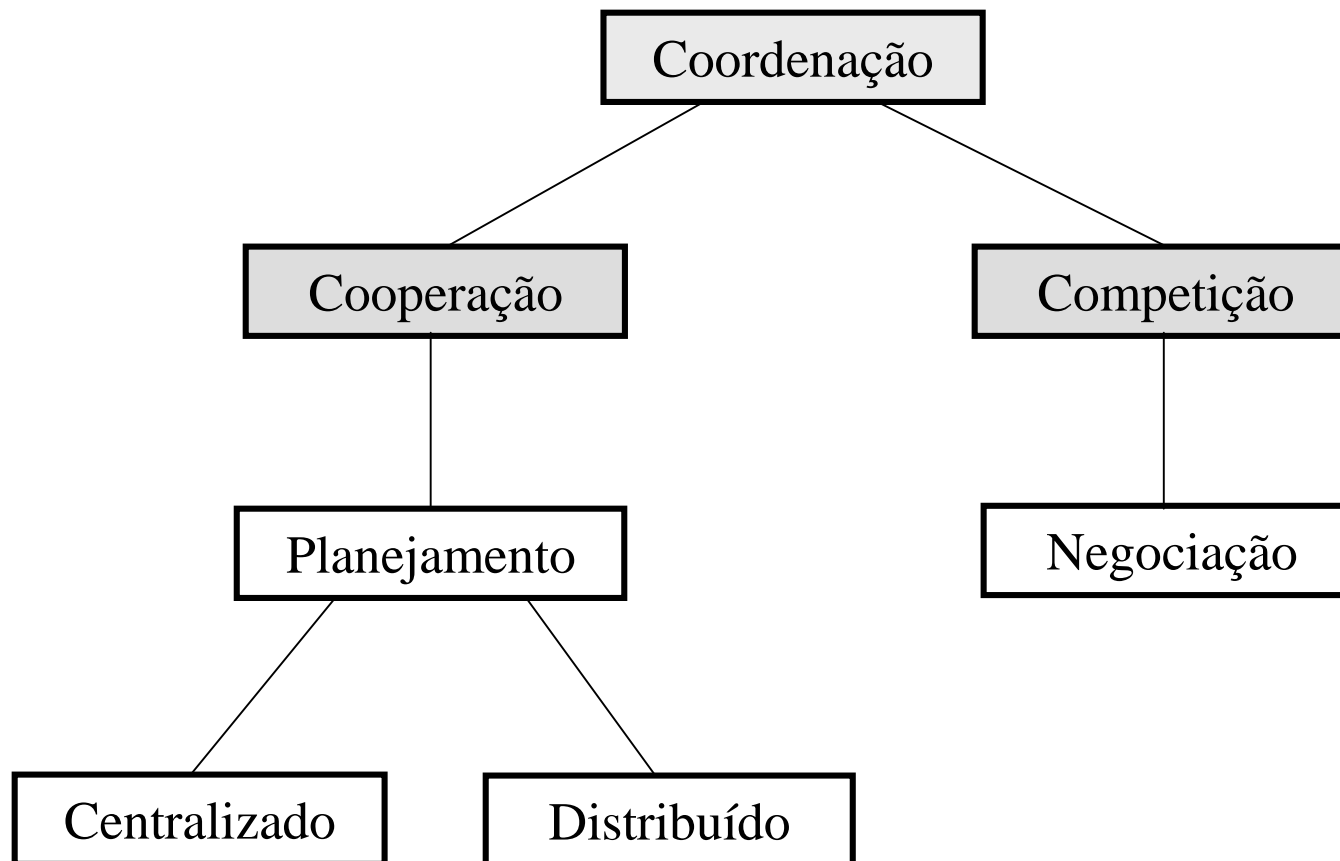
2. 5 Agentes e Sistemas Multiagentes

- Coleção de agentes
 - sociedade de agentes
 - interação entre agentes
- Protocolos de comunicação
 - mecanismo para trocar e entender mensagens
- Protocolos de interação/coordenação
 - mecanismo para negociar e cooperar

■ Características de ambientes multiagentes

- infraestrutura especificando protocolos de comunicação e interação
- tipicamente aberto, sem elemento central
- agentes autonomos
 - competitivos (*self-interested*)
 - cooperativos

Coordenação de Comportamento e Atividades



- Cooperação pressupõe sociabilidade
 - cada agente deve manter um modelo dos outros agentes
 - desenvolver um modelo de interações entre agentes
- Coerência global sem controle global explícito

Comunicação e Coordenação Sistemas Multiagentes

- Construção de sistemas multiagentes requer
 - escolha da arquitetura correta
 - distribuição de tarefas entre agentes
 - definição do mecanismo interação
- Sistema multiagente deve considerar
 - características da infraestrutura de comunicação
 - estratégias de coordenação das atividades dos agentes
- Componentes fundamentais
 - abstrações de baixo nível para comunicação
 - abstrações de alto nível dos mecanismos de controle e coordenação

Protocolos de Comunicação

- *Message_Passing*: mensagens de um agente para um ou mais agentes
- *Remote_Procedure_Calls*: RPC troca de informações via chamada de funções associadas aos agentes destino pelo agente origem (fornecendo, e.g., os argumentos da função): o agente destino fornece o valor da função
- *Tuple_Space ou Generativo*: depósito de dados central acessível por todos agentes. Comunicação colocando e retirando itens (*tuplas*) do depósito central

Protocolos de Controle

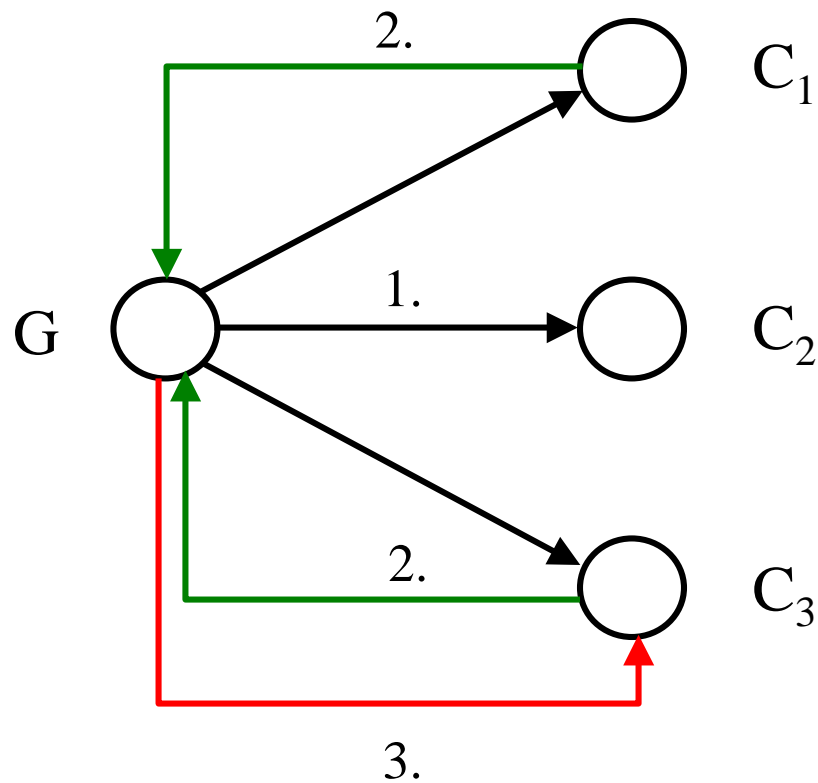
- *Pipeline*: agentes são produtores e consumidores de informação; recebem informação de um agente anterior, a processa, e envia informação para agente posterior
- *Request_Response*: esquema hierárquico onde níveis mais altos executam tarefas mais complexas, e.g. cliente-servidor e mestre-escravo

- *Subsumption*: agentes são conectados de forma topologicamente fixa: cada agente pode obter o controle global do sistema, e.g., via supressão da comunicação de agentes com prioridade mais baixa que a própria
- *Peer_to_Peer*: estratégia dinâmica onde agentes determinam qual deles irá obter o controle global do sistema (e.g., via voto)

Protocolos de Coordenação

- Voting Protocols
- Market_Oriented Protocols
- Contract_Nets Protocols
- Game_Theory Protocols

Rede de Contratos



1. anuncia tarefa

2. faz proposta

3. contrata

- Em geral: protocolo em comunicação SMA significa
 - protocolo de transporte (HTTP, FTP, etc.)
 - *framework* de alto nível para interação (*game, contract, negotiation, etc.*)
 - restrições sobre as trocas possíveis de primitivas de comunicação

2. 6 Natureza de Ambientes

- Completamente observável × parcialmente observável
- Determinístico × estocástico
- Episódico × sequencial
- Estático × dinâmico
- Discreto × contínuo
- Conhecido × desconhecido

Referências

- 1- Sycara, K. et. al. Distributed Intelligent Agents, *IEEE Expert*, vol. 11, no. 6, pp. 36-46, December 1996.
- 2- Letjer, M., Dean, T. A Framework for the Development of Multiagent Systems, *IEEE Expert*, vol. 11, no. 6, pp. 47-59, December 1996.
- 3- Bradshaw, J. *Software Agents*. MIT Press, Massachusetts, 1997.
- 4- Leonard, A. *Bots: The Origin of New Species*. Hardwired Press, California, 1997.
- 5- Weiss, G. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.
- 6- Wooldridge, M. *An Introduction to Multiagent Systems*, Wiley 2002

- 7- Ferber, J. Multi-Agent Systems: Na Introduction to Distributed Artificial Intelligence Addison-Wesley, 1999
- 8-Jennings, N., Sycara, K., Wooldridge, M. A Roadmap of Agent Research and Development, *Autonomous Agents and Multi-Agent Systems*, Vol. 1, pp. 7-38, 1998.
- 9- Ferber, J. Multi-Agent Systems: Na Introduction to Distributed Artificial Intelligence Addison-Wesley, 1999
- 10-Jennings, N., Bussmann, S. Agent-Based Control Systems. *IEEE Control Systems Magazine*, June 2003, pp. 61-73, 2003.
- 11- He, M., Jennings, R. Designing a Successful Trading Agent: A Fuzzy Set Approach, *IEEE Trans. Fuzzy Systems*, vol.12, no. 3, pp. 389-410, 2004.

Observação

Este material refere-se às notas de aula do curso EA 072 Inteligência Artificial em Aplicações Industriais da Faculdade de Engenharia Elétrica e de Computação da Unicamp. Não substitui o livro texto, as referências recomendadas e nem as aulas expositivas. Este material não pode ser reproduzido sem autorização prévia dos autores. Quando autorizado, seu uso é exclusivo para atividades de ensino e pesquisa em instituições sem fins lucrativos.