

Agentes Inteligentes

- **Introdução**

Criar sistemas que possam aumentar nossa capacidade de resolver problemas, ou que possam funcionar de forma autônoma em ambientes complexos.

Complexidade: sob o ponto de vista da 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 em bancos, comércio; monitoração, supervisão e controle de processos industriais e sistemas de manufatura; atendimento e serviços médicos, etc.;

- **Engenharia e Agentes Inteligentes:**

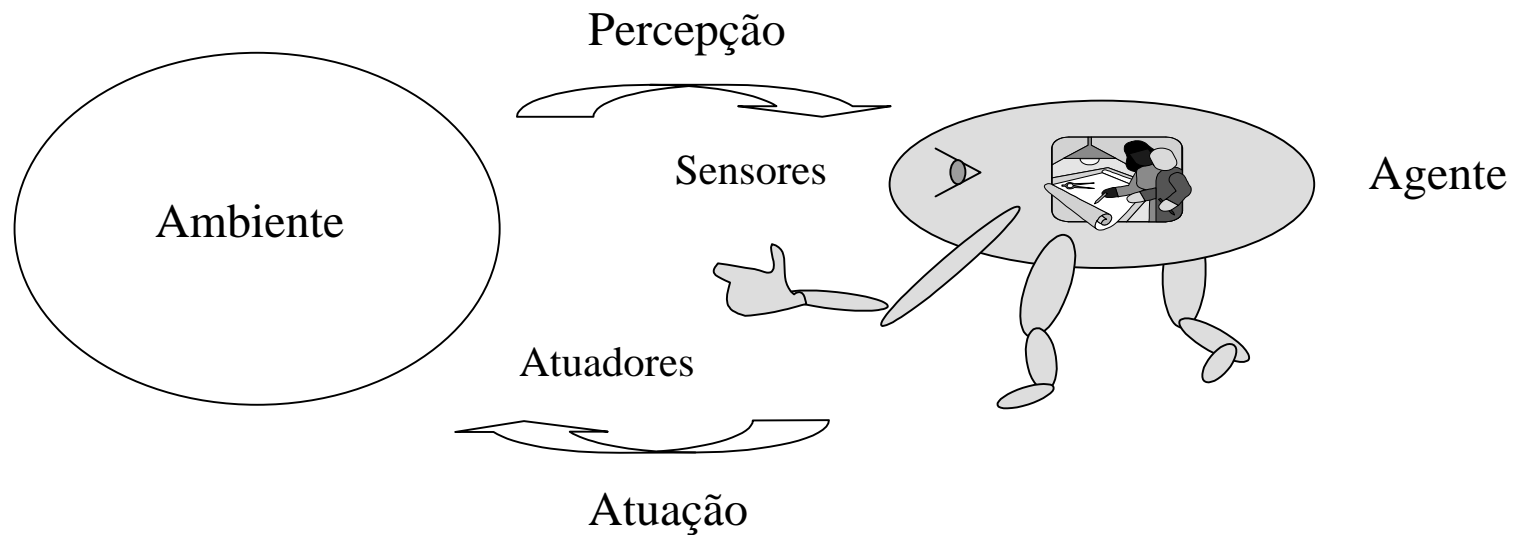
- 1 - Natureza da inteligência,
Como descrever comportamentos inteligentes.
- 2 - Objetivos múltiplos, possivelmente conflitantes e dinâmicos,
- 3 - Crença do agente sobre o passado, presente e futuro,
- 4 - Aquisição de informação sobre tarefas antes/durante escolha plano/ação,
- 5 - Conhecimento incompleto/impreciso do ambiente,
Hipóteses razoáveis sobre ocorrências de eventos e comportamntne de agentes,
- 6 - Dinâmica do ambiente e de outros agentes.
- 7 - Modelos, etc.....

- **Algumas questões essenciais:**

- 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 ?
- 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 e características**

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 1995).



Agente = Arquitetura + Programa

- Características desejáveis:

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,

5 - Antecipação: antecipar necessidades, resultados, informações, aprendizagem,

6 - Atividade: capaz de iniciar solução de problemas, fusão de informação, etc.,

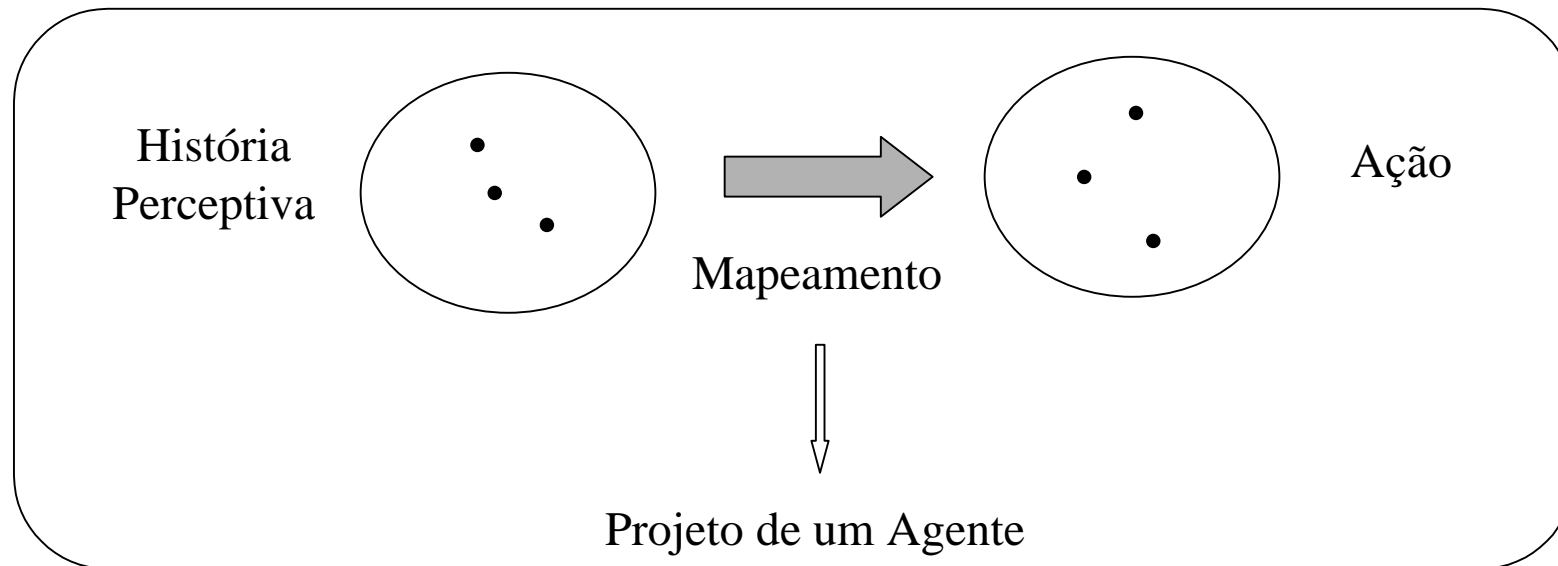
7 - Cooperação: entre sistemas e outros agentes; resolução de conflitos; conflitos, etc.

8 - Flexibilidade: estruturas heterogêneas de agentes e fontes de informação,

9 - Adaptação: acomodação de modificações de objetivos, tarefas, aprendizagem.

Agente racional - racionalidade depende de, pelo menos, quatro itens:

- 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.



• Estruturas de Agentes

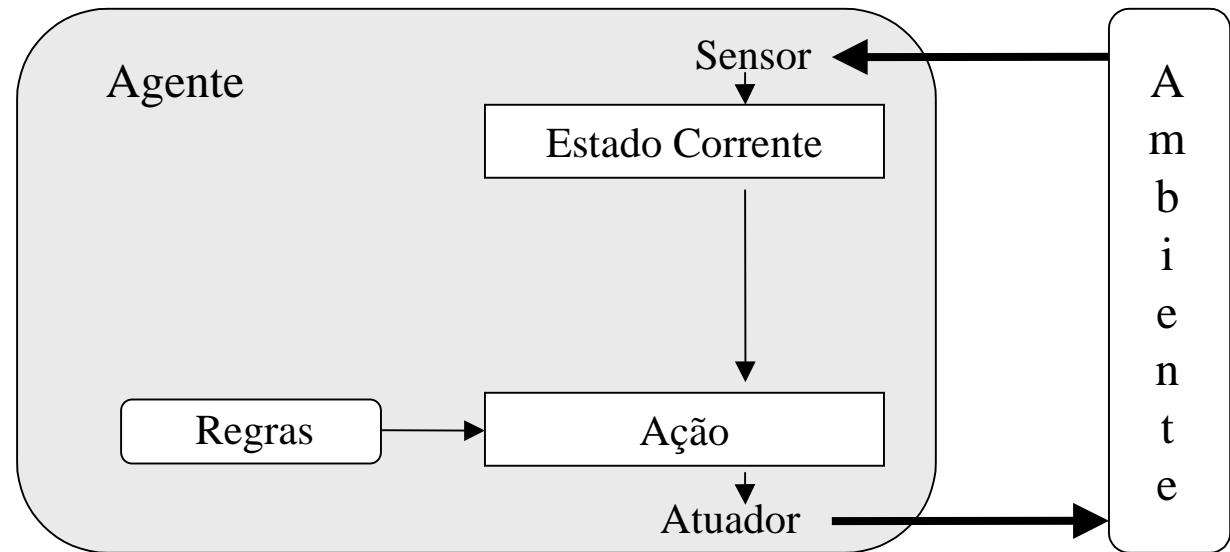
Centralizadas

- Agentes Reativos
- Agentes Dinâmicos
- Agentes Orientados por Metas
- Agentes Orientados por Utilidade

Distribuídas

- Agentes Interface
- Agentes Tarefa
- Agentes Informação

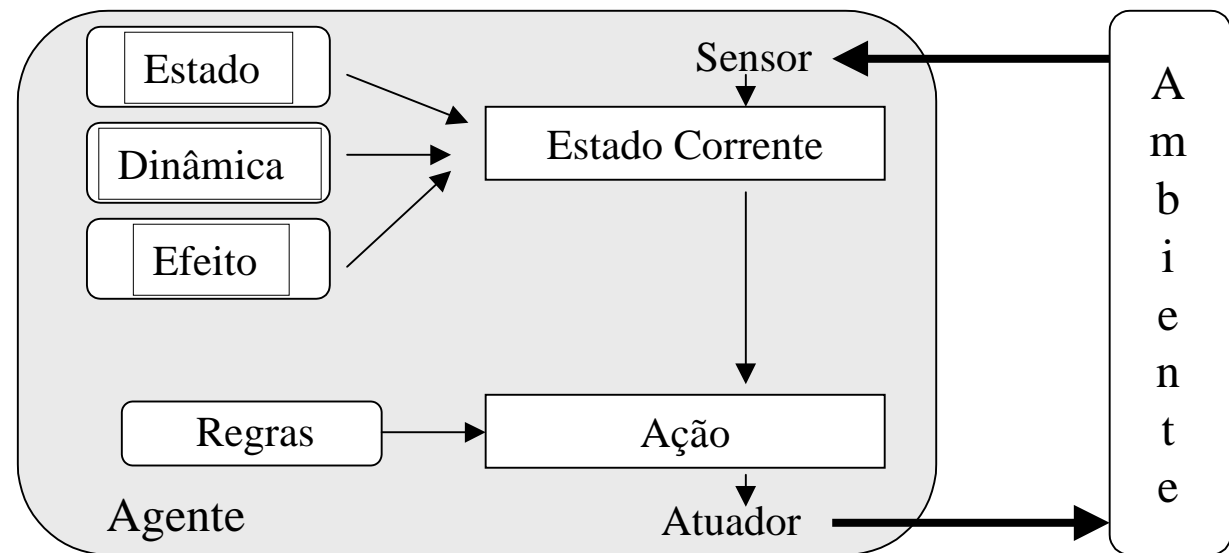
Agentes Reativos



function Simple_Reflex_Agent (*percept*) **returns** *action*
static: *rules*, conjunto de pares condição-ação

```
state ← Interpret_Input(percept)  
rule ← Rule_Match(state, rules)  
action ← Rule_Action [rule]  
return action
```

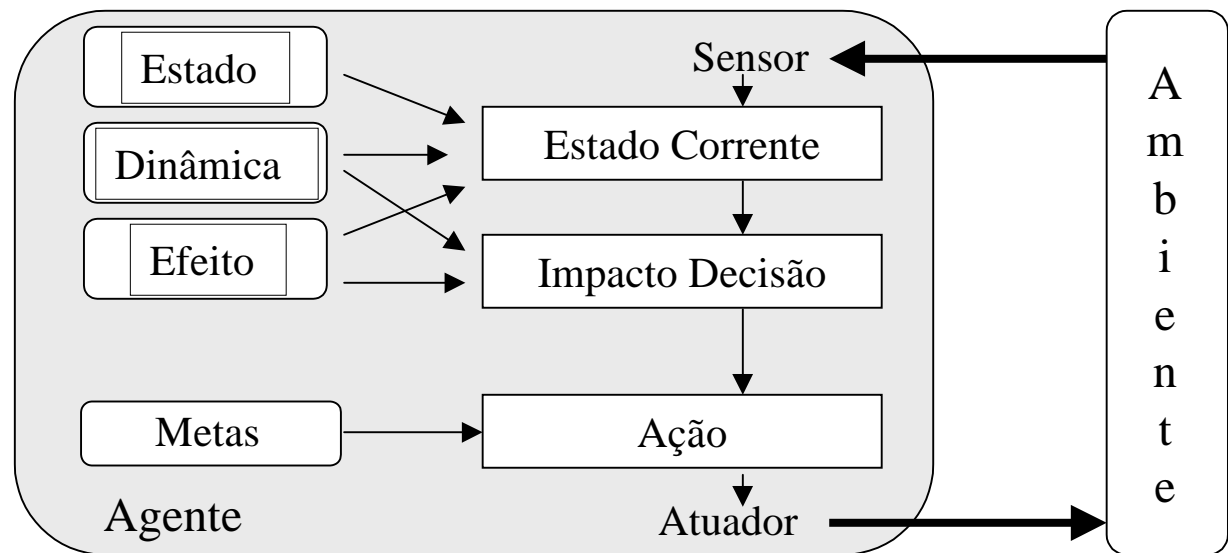

Agente Dinâmico (Estado Interno)



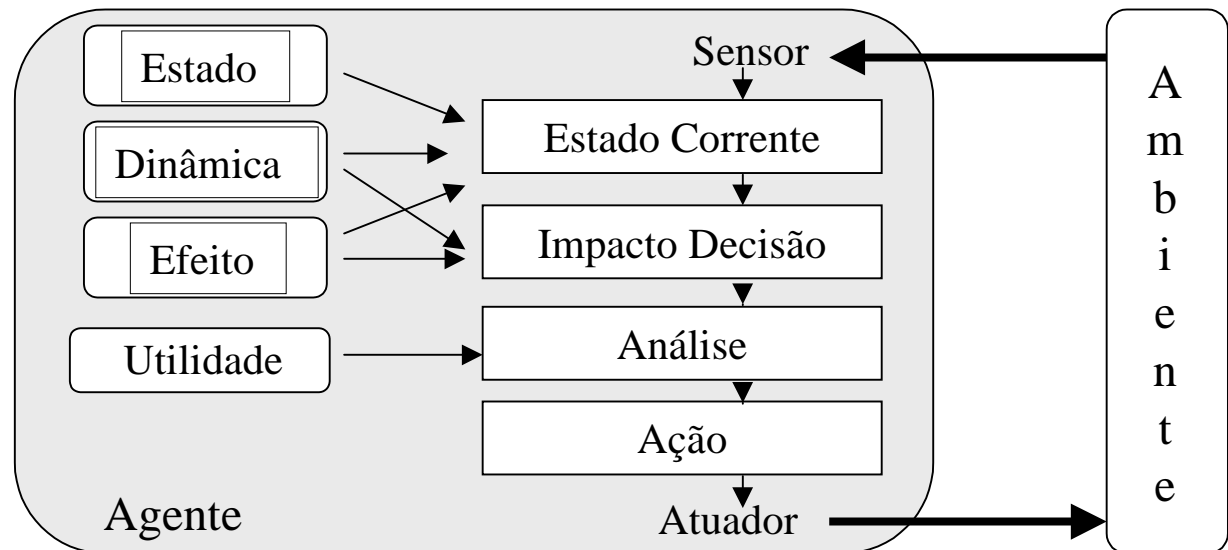
function Reflex_Agent_With_State (*percept*) **returns** *action*
static: *rules*, conjunto de pares condição-ação
state, descrição do estado corrente do mundo

```
state ← Update_State(state, percept)  
rule ← Rule_Match(state, rules)  
action ← Rule_Action [rule]  
state ← Update_State(state, action)  
return action
```

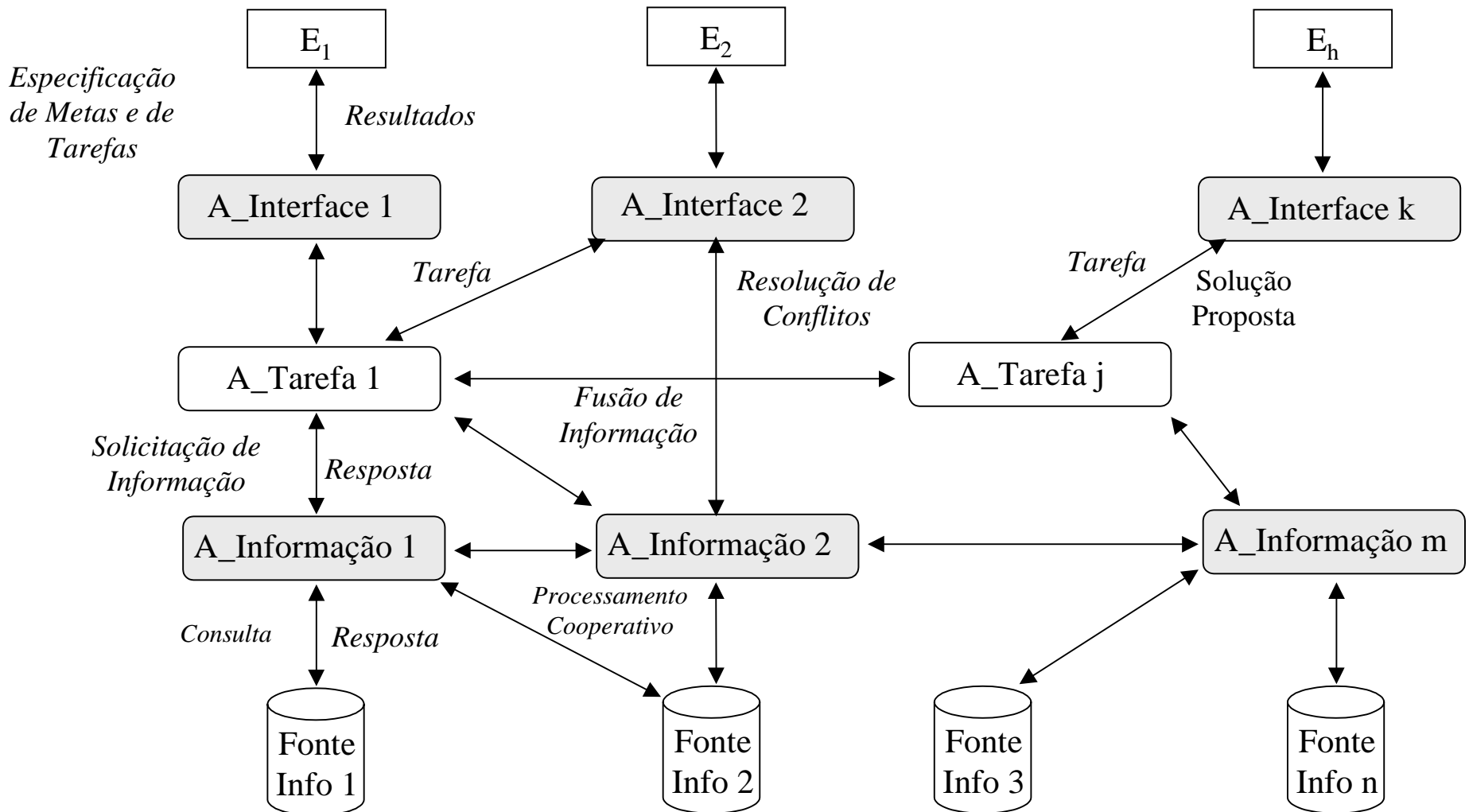
Agentes Orientados por Metas



Agentes Orientados por Utilidade

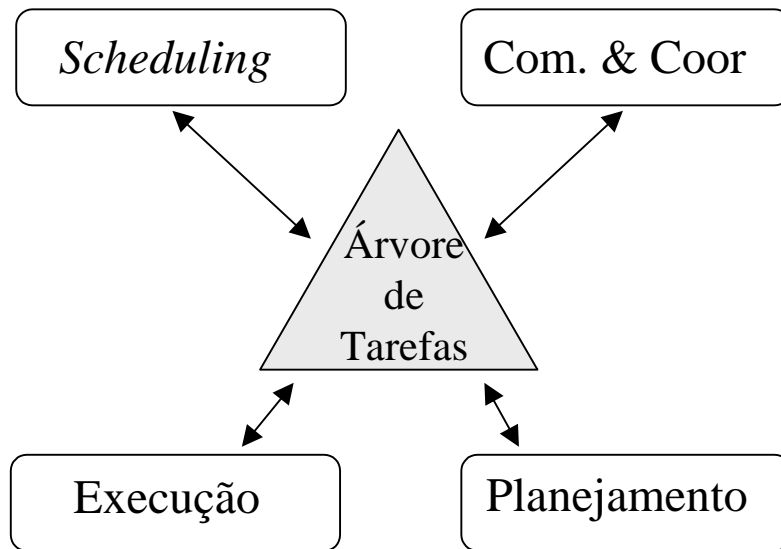


Estruturas Distribuídas: Exemplo de Organização



Estruturas Distribuídas: Exemplo de Arquitetura Funcional

Módulos de Controle



Independente do Domínio

Conhecimento do Agente

Fragments de Planos
Independente do Domínio

Fragments de Planos
Dependente do Domínio

Beliefs
e
Base de Fatos

- **Comunicação e Coordenação em Sistemas Multi-Agentes**

- Construção de sistemas multi-agentes requer a escolha da arquitetura correta. Isto inclui a questão de como distribuir as responsabilidades do sistema entre os seus agentes (componentes) e como estes agentes devem interagir para cumprir estas responsabilidades.
- Um sistema multi-agente sempre deve considerar dois pontos:
 - . As características da infraestrutura de comunicação a ser usada,
 - . Estratégias de coordenação das atividades dos agentes.
- Duas componentes fundamentais:
 - . Abstrações de baixo nível para proporcionar comunicação,
 - . Abstrações de alto nível para representar mecanismos de controle e coordenação.

• Protocolos de Comunicação

- *Message_Passing*: comunicação entre agentes na forma de mensagens roteadas de um agente particular para um ou mais agentes.
- *Remote_Procedure_Calls*: RPC envolve a troca mútua de informações via chamada de funções associadas com os agentes destino pelo agente de origem (o qual fornece, e.g., os argumentos da função). O agente no destino fornece o valor da função.
- *Tuple_Space* ou Generativo: assume um depósito de dados central acessível por todos os agentes. Agentes se comunicam colocando e retirando itens (*tuplas*) do depósito central

• Protocolos de Controle

- *Pipeline*: agentes atuam como produtores e consumidores de informação. Eles recebem informação de um agente anterior, processa esta informação, e envia informação para um agente posterior.
- *Request_Response*: esquema hierárquico onde níveis mais altos executam tarefas mais complexas. Exemplos incluem 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 outros agentes com prioridade mais baixa que a própria.
- *Peer_to_Peer*: estratégia dinâmica onde os 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*

- **Características do Ambiente**

- Acessível ou não acessível
- Determinístico ou não determinístico
- Episódico ou não episódico
- Estático ou dinâmico
- Discreto ou contínuo

• Referências

- Russel, S., Norvig, P. *Artificial Intelligence-A Modern Approach*, Prentice Hall, New Jersey, USA, 1995
- Sycara, K. et. al. Distributed Intelligent Agents, *IEEE Expert*, vol. 11, no. 6, pp. 36-46, December 1996.
- Letjer, M., Dean, T. A Framework for the Development of Multiagent Systems, *IEEE Expert*, vol. 11, no. 6, pp. 47-59, December 1996.