

Inteligência Artificial

Sistemas Multiagente

Inteligência Artificial Distribuída

- Distributed Problem Solving(DPS): interessa-se pela decomposição de problemas entre um número de módulos que cooperam entre si e dividem conhecimento, desenhados especificamente para um problema em particular.
- Multiagent Systems (MAS): caracteriza-se pela existência de um certo número de agentes autônomos, heterogêneos e potencialmente independentes, trabalhando juntos para resolver um problema. Estes agentes são aptos a se adaptarem ao meio ambiente, reagir a ele e provocar mudanças neste meio
- Parallel Artificial Intelligence (PAI): interessa-se mais por problemas de performance do que por avanços conceituais, preocupando-se principalmente em desenvolver linguagens e algoritmos de computação paralela



Agentes Inteligentes

- Um agente é uma entidade de software que funciona de forma contínua e autônoma em um ambiente em particular, geralmente habitado por outros agentes, e que seja capaz de intervir no seu ambiente de forma flexível e inteligente, sem requerer intervenção ou orientação humana constante. De um modo ideal, que funcione continuamente por longos períodos de tempo.

Bradshaw, J. M. An introduction to software agents In
Bradshaw, J. M. Ed. Software Agents. Massachusetts: MIT
Press, 1997.

Agentes Inteligentes

- ♦ **O que é um agente???**
 - Um agente é algo que percebe seu ambiente através de sensores e atua no ambiente através de atuadores.
 - Agente Humano, Agente Animal, Agente Robótico, Agente em Software, Termostatos, etc...
 - A Função Agente mapeia dados da percepção para ações.

Agentes Inteligentes

- ♦ **O que é um agente racional???**
 - O objetivo da IA, segundo Russel & Norvig é projetar agentes que façam um bom trabalho agindo no seu ambiente. O princípio básico da utilização de agentes é que eles devem "saber das coisas" (know things).
 - Um agente racional ideal é aquele que, para cada possível sequência de percepção, realiza uma ação que maximiza seu desempenho (mapeamento ideal), tendo como base as evidências fornecidas pela sequência de percepções e pelos conhecimentos previamente existentes no agente.

Agentes Inteligentes

- ♦ O que é um Agente Autônomo?
 - Agentes Autônomos são sistemas computacionais que operam em ambientes dinâmicos e imprevisíveis. Eles interpretam dados obtidos pelos sensores que refletem eventos ocorridos no ambiente e executam comandos em atuadores que produzem efeitos no ambiente.
 - O grau de "autonomia" de um agente está relacionado à capacidade de decidir por si só como relacionar os dados dos sensores com os comandos aos atuadores em seus esforços para atingir seus objetivos, satisfazer motivações, etc...

Um programa não é um agente simplesmente porque:

- podem ser agendados previamente para realizar alguma tarefa;
- realizam tarefas de computação de baixo nível instruídos por linguagens de alto nível;
- abstraem ou encapsulam detalhes;
- agregam funções cognitivas;
- manifestam características de inteligência distribuída;
- realizam um papel mediador entre pessoas e programas;
- desempenham um papel de assistente inteligente;
- podem migrar de um computador para outro;
- podem apresentar-se ao usuário como um personagem confiável;
- comunicam-se através de uma linguagem de agentes;
- manifestam um comportamento que denota um estado mental

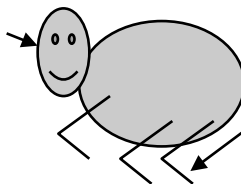
Agentes Inteligentes

- ♦ O projeto de um agente consiste em especificar que ações um agente deve tomar em resposta a qualquer seqüência de percepções.
 - Medida de Desempenho
 - Segurança, velocidade, destino, conforto, etc...
 - Ambiente
 - Observável, determinístico, episódico, estático,...
 - Atuadores
 - Rodas, pés, vídeos, mensagens, etc...
 - Sensores
 - Imagens, gps, teclados, mensagens, encoders, ultra-som,...

Agentes Inteligentes

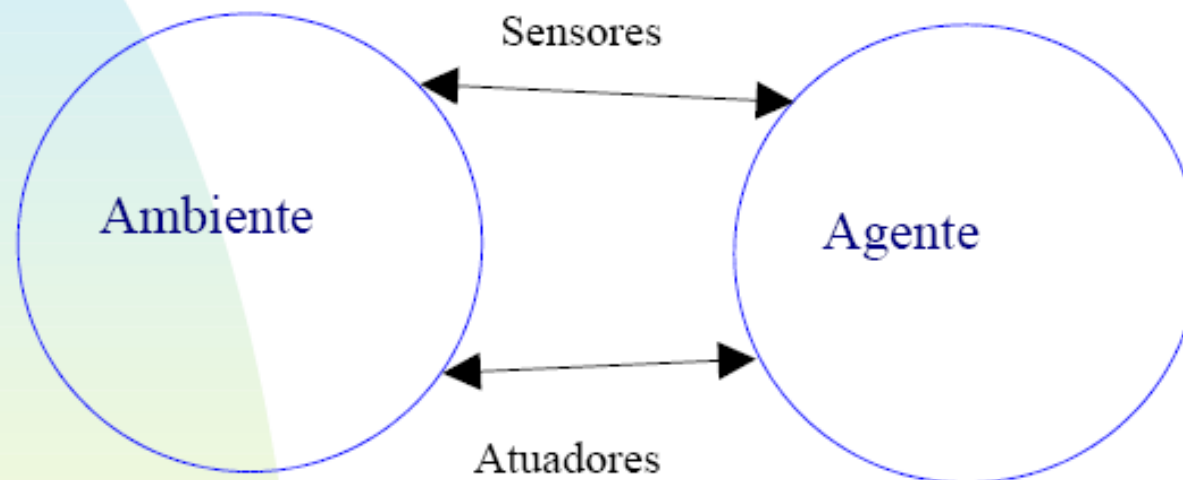
- ♦ Tipos de Agentes
 - Software Agents
 - Agentes são considerados entidades computacionais baseadas na idéia de que os usuários necessitam apenas especificar um objetivo em alto nível ao invés de utilizar instruções explícitas, deixando as questões de como e quando agir a cargo do agente.
 - Aplicações: Interfaces Amigáveis, Cartografia, Auxílio ao Ensino, Auxílio ao Diagnóstico Médico.
 - Hardware Agents
 - Agentes que operam em ambientes físicos (AGVs, Robôs, Embedded Systems, etc.)
 - Agentes Físicos capazes de detectar mudanças ambientais e, através da reavaliação de seus objetivos encontrar uma nova seqüência de ações capazes de persegui-los, sem que esta seqüência tivesse sido prevista.

sensores



atuadores

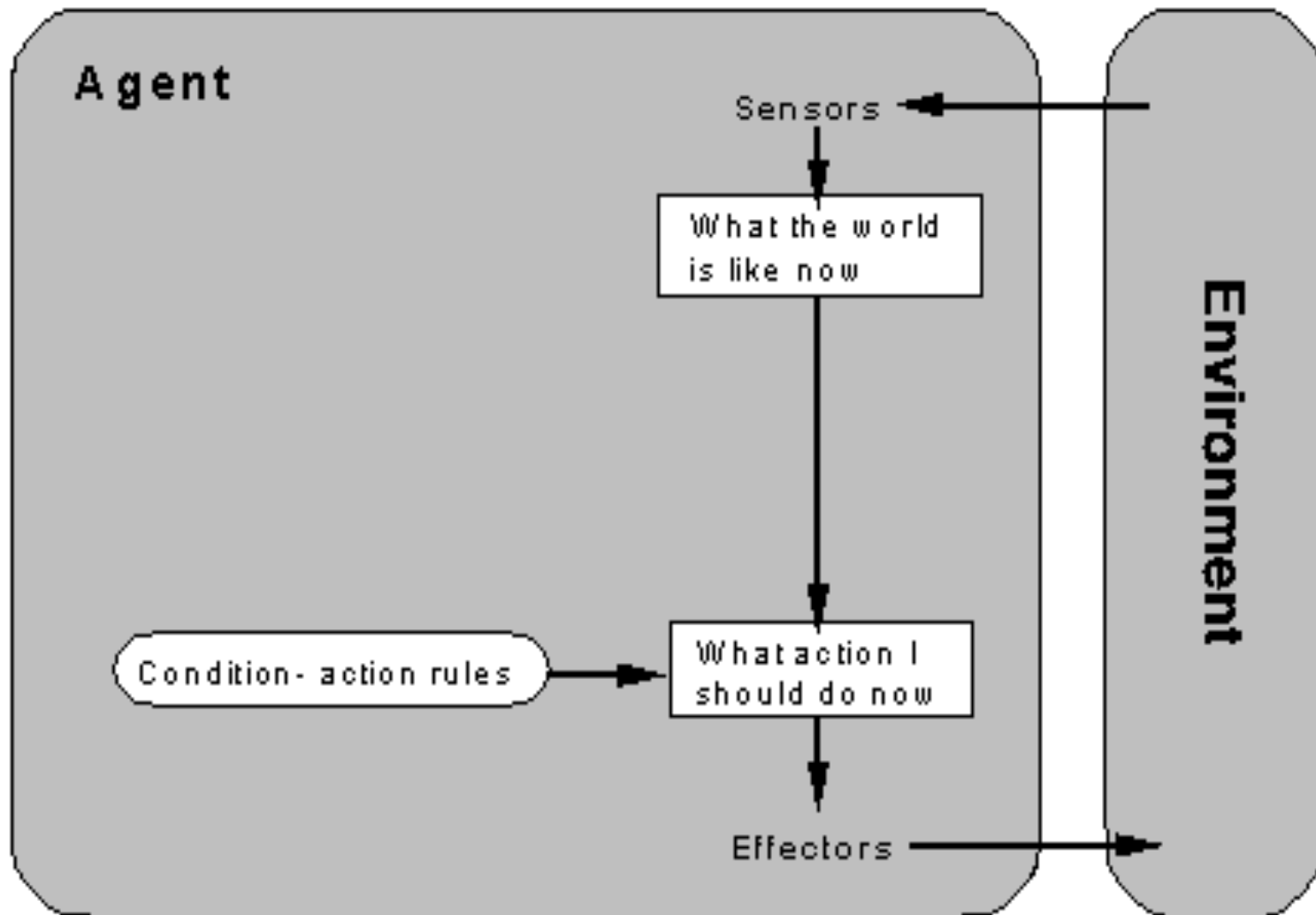
Modelo de agente



Agentes Reflexivos

- ♦ Não tem memória.
- ♦ Quando cessa a percepção, cessa a ação.
 - If car-in-front-is-braking (brake-light on)
 - then initiate-braking

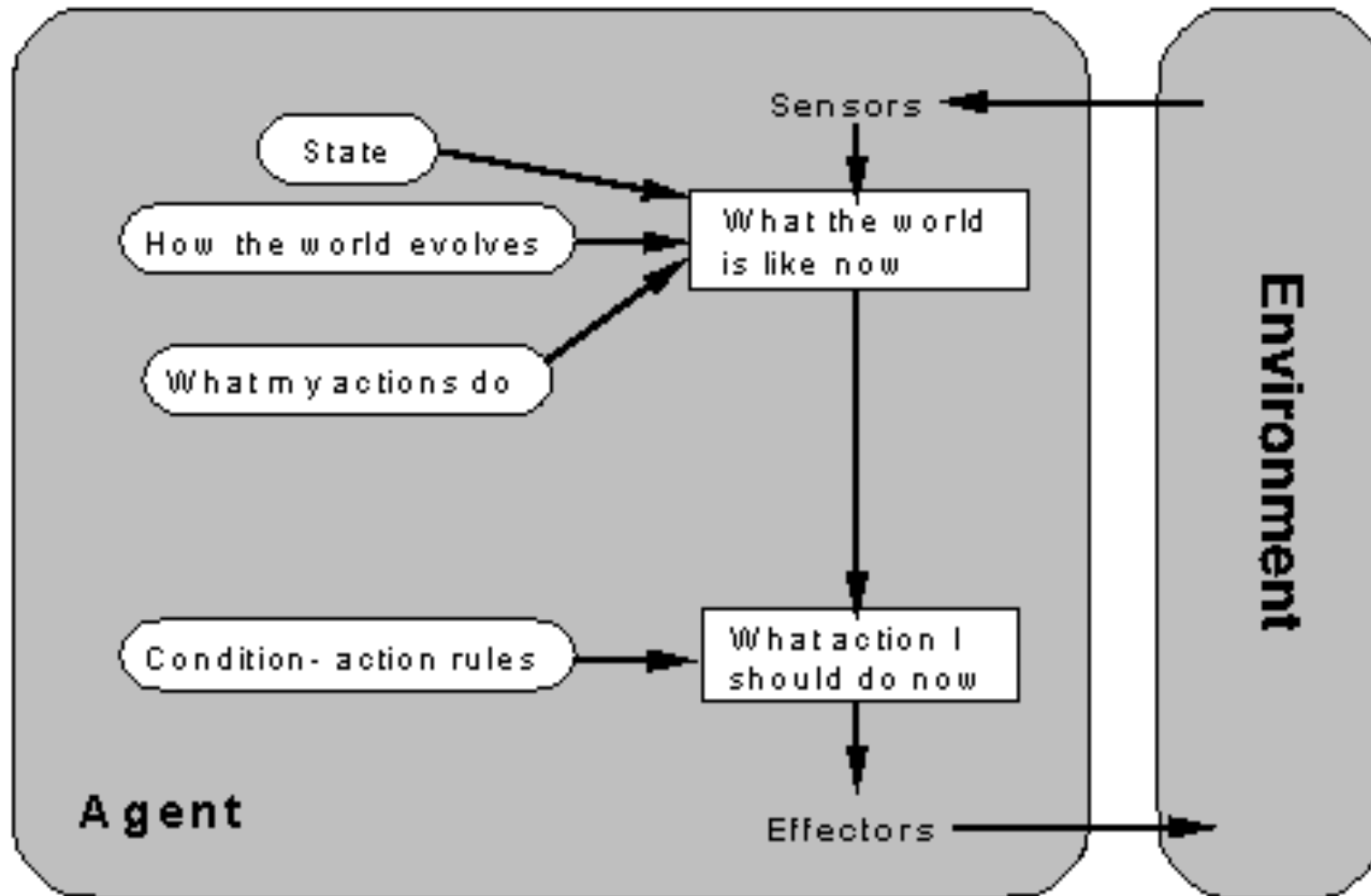
Agentes Reflexivos



Agentes com Estados Internos

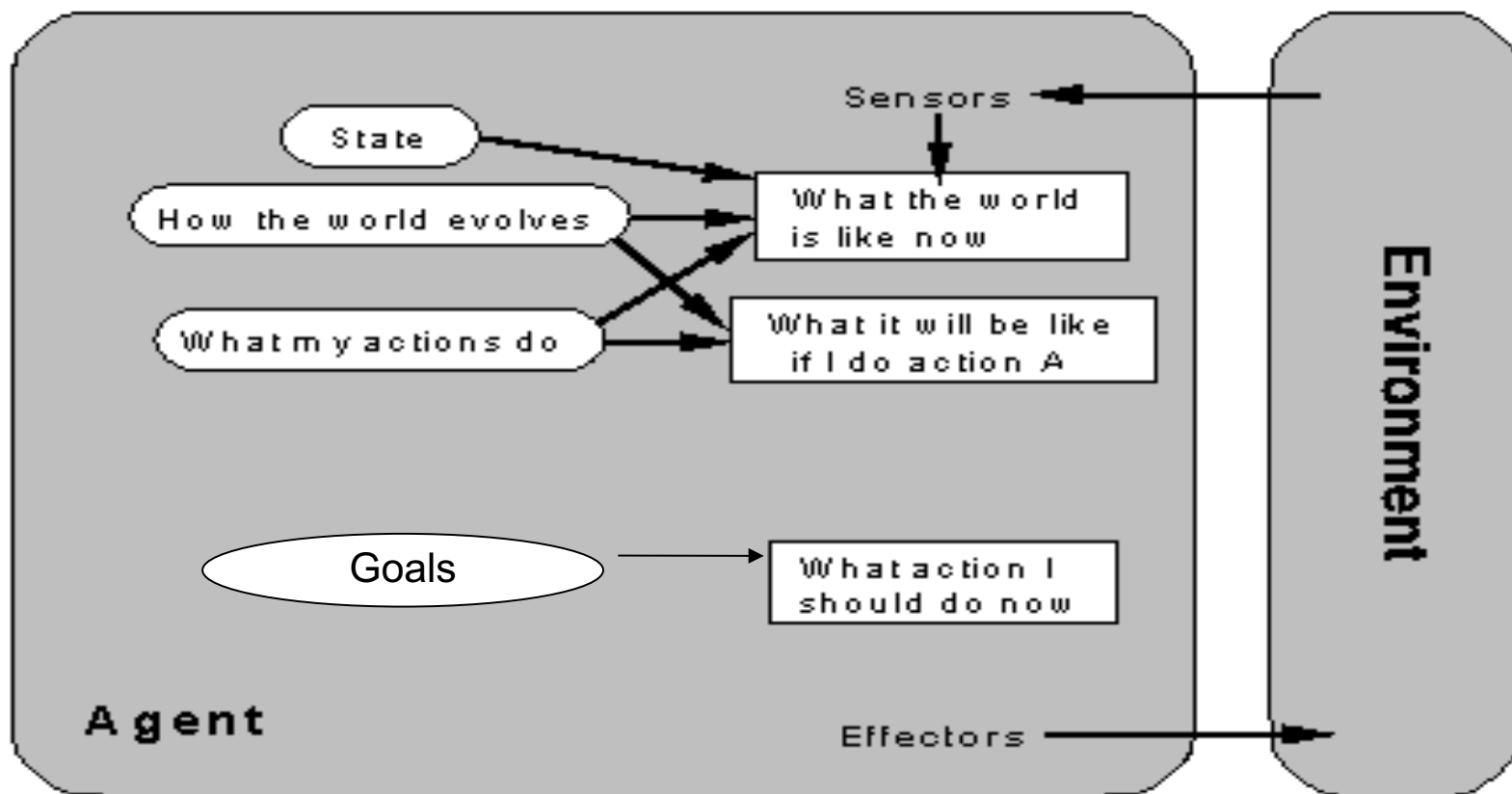
- Guarda informações que não são percebidas no momento
 - Como o mundo evolui (modelo do mundo)
 - O que as ações provocam no mundo

Agentes com Estados Internos



Agentes com Metas

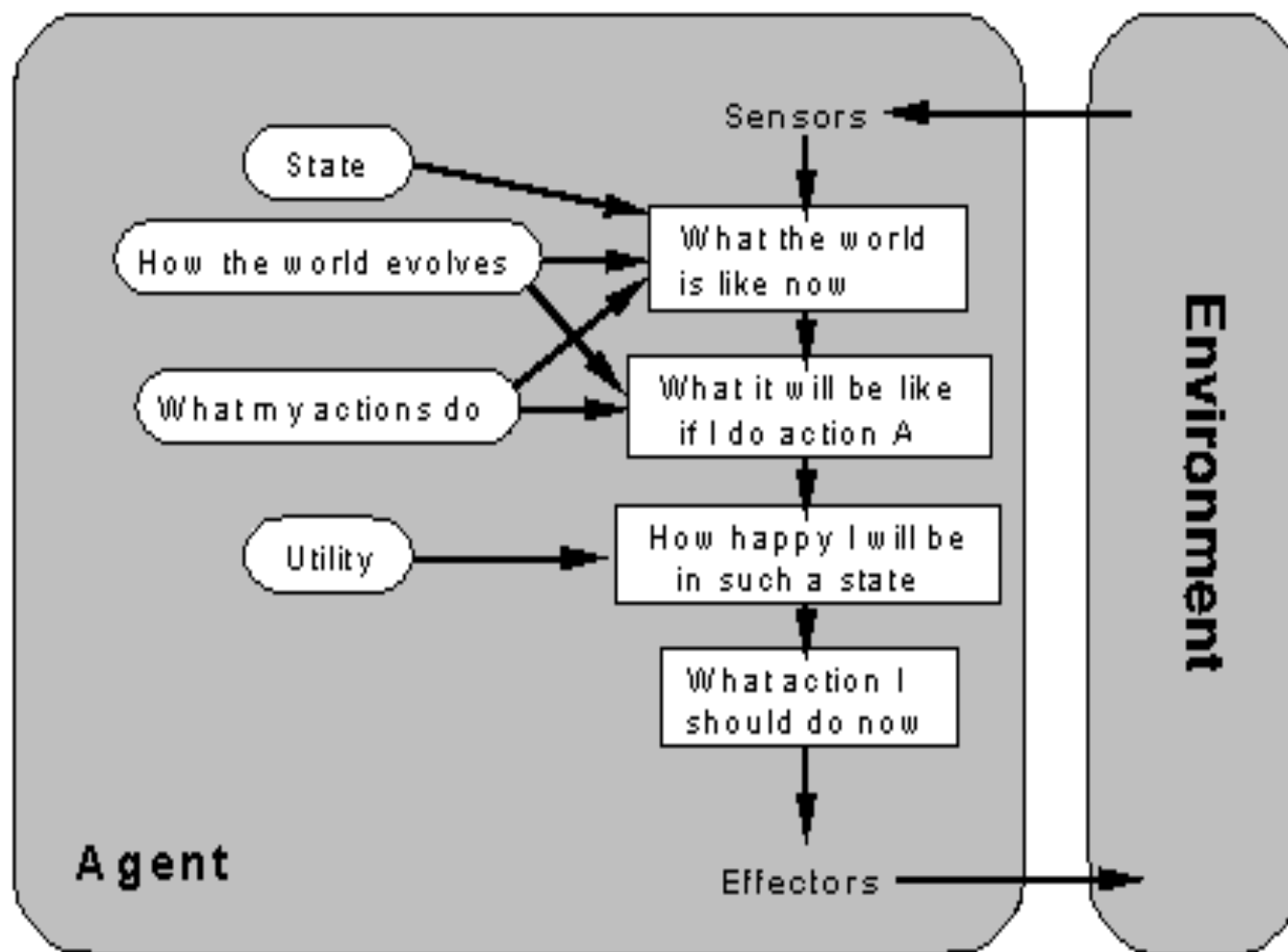
- Metas
- Busca e Planejamento são subcampos da IA cujo objetivo é achar seqüências de ações que conduzam ao objetivo do agente.



Agentes baseados em Utilidade

- Utilidade é uma função que mapeia um estado em um número real que descreve o grau de "felicidade" associado ao estado.
- Permite decisões racionais em casos em que o objetivo tem algum "problema";
- Quando existem objetivos conflitantes (p.ex.: velocidade e segurança);
- Quando existem vários objetivos, a utilidade "diz" qual tentar alcançar primeiro.

Agentes baseados em Utilidade



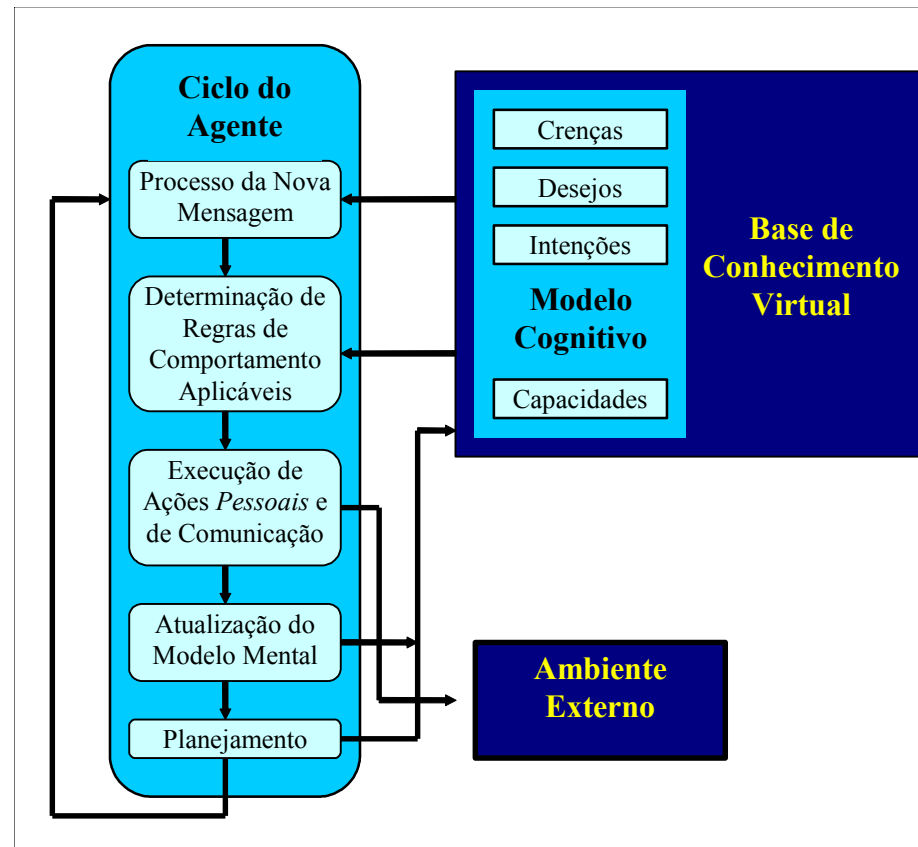
Modelo ideal de agente

- Reatividade
- Autonomia
- Comportamento cooperativo
- Habilidade de comunicação ao nível de conhecimento
- Capacidade de inferência
- Continuidade temporal
- Personalidade
- Adaptabilidade
- Mobilidade

Estado da arte



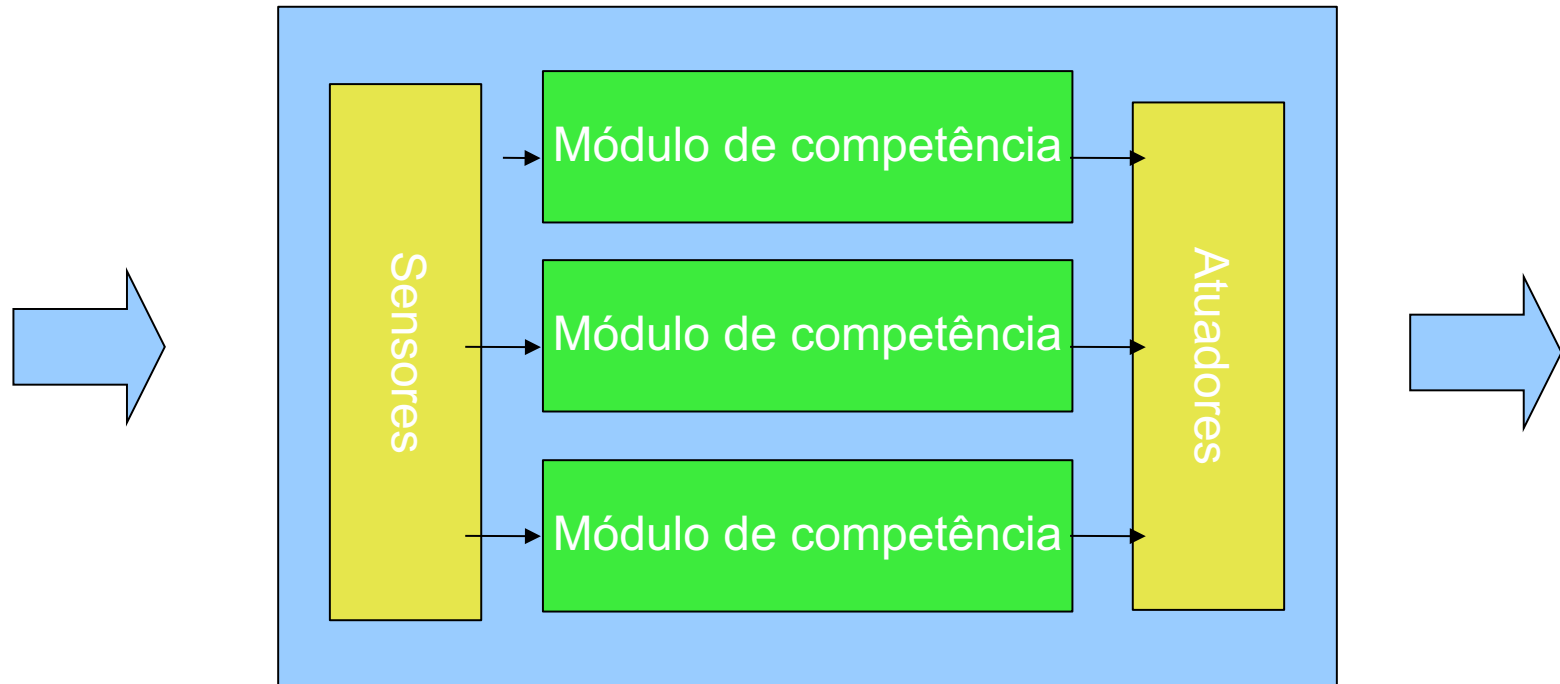
Processo de execução dos agentes



Tipos de arquiteturas

- Duas abordagens principais:
 - Sistemas Multiagentes Cognitivos ou Deliberativos
 - existência de uma forma explícita de representação de conhecimento
 - representação formal explícita do ambiente e dos outros agentes
 - comunicam-se entre si e planejam sua ação futura
 - capacidade de raciocínio sobre o conhecimento que dispõe.
 - Sistemas Multiagentes Reativos
 - ênfase principal é no comportamento
 - a atividade de um agente é produzida pela interação entre o agente e seu meio ambiente, e não pelo processo de raciocínio que ocorre internamente

Agentes reativos



Arquiteturas Cognitivas

- Funcionais
 - Composta por módulos que representam suas funcionalidades
- Baseadas em Estados Mentais
 - Perspectiva de inspiração psicológica
 - BDI (*Belief-Desire-Intention*)
 - Componente Filosófico
 - Componente de Arquitetura de *Software*
 - Componente Lógico

Estados Mentais

- Estados mentais de informação
- Estados mentais pró-ativos
- Crença
 - É um estado mental intencional fundamental para as interações do agente, com noção idêntica a de conhecimento (Corrêa)
- Desejo
 - Estados do mundo que o agente quer provocar
- Intenção
 - Estados do mundo que o agente se comprometeu em provocar (subconjunto dos desejos)
 - Devem ser consistentes

Estrutura BDI (Rao/Georgeff)

- Beliefs (crenças)
 - Contém a visão do agente a respeito do ambiente
- Desires (desejos)
 - Julgamentos sobre situações futuras (pode ser irrealista)
- Goals (objetivos)
 - Subconjunto de desejos sobre o qual o agente age (realístico e sem conflitos)
- Intentions (intenções)
 - Subconjunto de objetivos que o agente decide seguir por ter os recursos necessários
- Plans (planos)
 - Combinação de intenções do agente em unidades consistentes

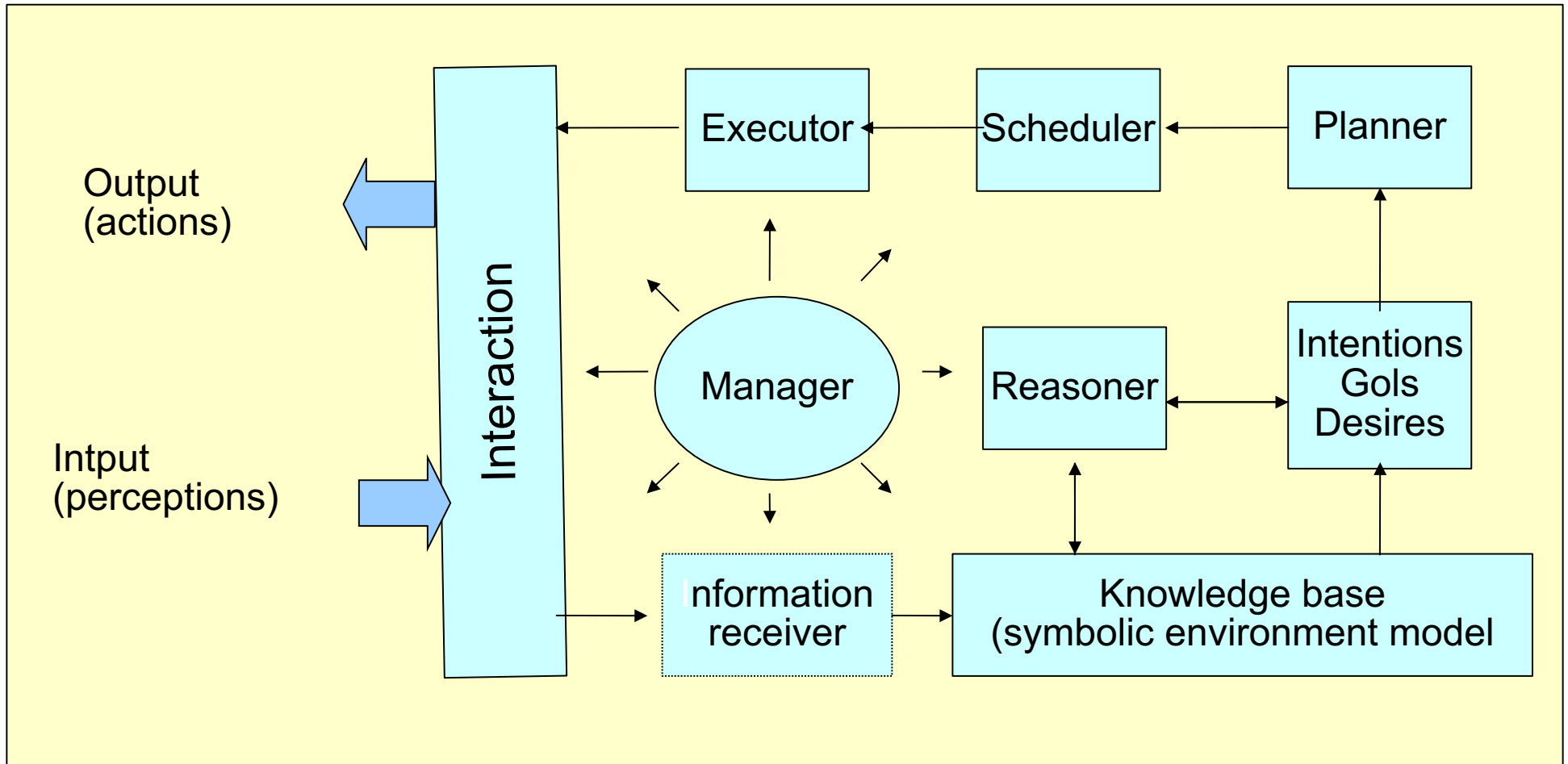
Propriedades das Intenções

- Direcionam raciocínio meio-e-fim
- Persistem
- Restringem deliberações futuras
- Influenciam crenças sobre as quais raciocínios práticos futuros são realizados

Agentes BDI

- Agente BDI:
 - Processo Deliberativo
 - Geração de Opções
 - Processo de Filtragem
 - Raciocínio Meio-e-Fim
- Estratégias de Comprometimento
 - Cega
 - Obcecada
 - Liberal
- Reconsideração de Intenções
 - Ousada
 - Cauteloso

Arquitetura de agente deliberativo



Sistemas Multiagente

- É uma rede fracamente acoplada de solucionadores de problemas que trabalham em conjunto para resolver problemas que vão além da sua capacidade individual. Estes solucionadores de problemas são essencialmente autônomos, distribuídos e, muitas vezes heterogêneos em sua natureza.

Baker, Albert. JAFMAS - A java based agent framework for multiagents systems. Cincinnati: Department of electrical & computer Engineering and Computer Science University of Cincinnati, 1997. Doctoral thesis.

Sistemas Multiagente

- Constituída por elementos que desempenham as funções:
 - Cooperação;
 - Resolução de conflitos;
 - Negociação;
 - Comprometimentos;
 - Comunicação;
 - Interação.

Cooperação

- Tipologia de cooperação entre agentes



Coordenação

- Envolve a seleção, ordenamento, comunicação de resultados das atividades dos agentes.
- É realizada em tempo real, isto é, não é possível modelar e especificar, a priori, a subdivisão de um problema.
- Leva em conta:
 - Caráter quantitativo das interações;
 - Performance e demanda de recursos;
 - Prováveis atividades futuras dos agentes;
 - Sobrecarga;
 - Prazos para a execução de tarefas;
 - Outras informações simbólicas (crenças, desejos e intenções);
 - Estados conjuntos desejados ou indesejados.

Necessidade de coordenação

- Existe independência na ação dos agentes;
- Existe interdependência na ação dos agentes;
- Contenção de recursos para a solução de um problema;
- Existem métodos alternativos para a solução de um problema por diferentes agentes;
- Os agentes dispõem de informações diferentes ou contraditórias para a solução de problemas;
- Agentes precisam operar concorrentemente ou seqüencialmente para solucionar um problema;
- Problemas com características de explosão combinatória.

Negociação

- Processo de concordância mútua em atingir um estado conjunto;
- Usado como parte do processo de coordenação para estabelecer compromissos entre os agentes;
- Estabelece restrições no comportamento individual dos agentes;
- Intercâmbio de informações para transmissão de resultados intermediários.

Comunicação

Cooperação	Estratégias			
	Protocolos			
Comunicação	Linguagens de comunicação			
	Quadro de avisos	Difusão de mensagens	Federação	Mensagens diretas
	Transporte			

Comunicação X Coordenação

- Comunicação (KQML, FIPA, etc):
 - Quando enviar mensagens, o que comunicar, e em que ordem executar tarefas é decidido em tempo de projeto do sistema.
- Coordenação:
 - Camada superior de abstração sobre a função de comunicação;
 - Decisões tomadas em tempo de execução;
 - Maior independência de domínio.

Importância de Padrões em SMA

- *Agentes não interagem somente na sua plataforma original*
 - *Interoperabilidade*
- Promover maior adoção da tecnologia
- Foco na utilização de infra-estrutura e não no seu desenvolvimento
 - Redução de riscos

Padrões para SMA

- OMG (Object Management Group)
 - MASIF (Mobile Agent System Interoperability Facility)
- DARPA
 - KSE (Knowledge Sharing Effort)
 - KQML (Knowledge Query Manipulation Language)
- FIPA (Foundation for Intelligent Physical Agents)

FIPA – A Organização

- *Foundation for Intelligent Physical Agents*
 - *Foundation for Intelligent Interoperable Agents*
 - *Organização sem fins lucrativos*
 - *Estabelecida em dezembro de 1996*
 - *Visa produzir padrões de software para agentes heterogêneos e interativos*
 - *WebSite: <http://www.fipa.org>*
-
-

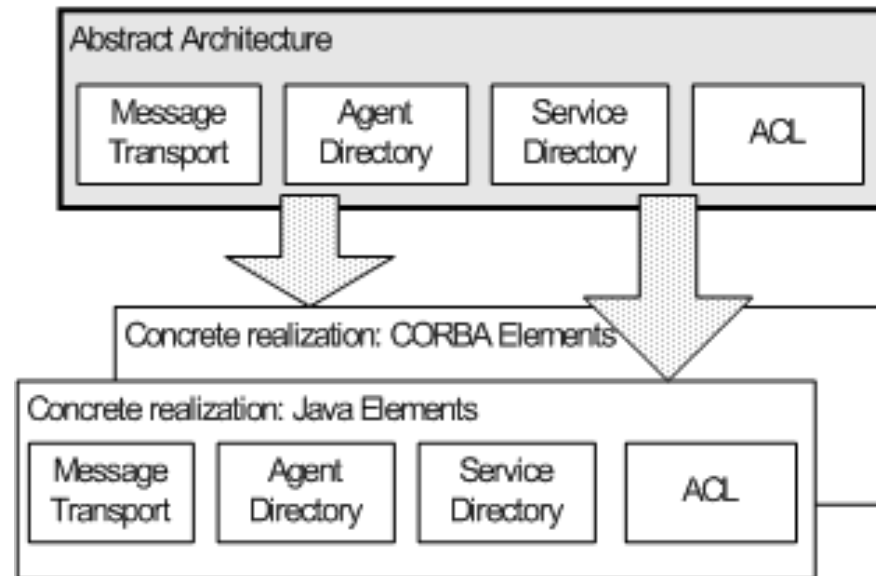
FIPA – Alguns Números

- + de 15 implementações
 - No mínimo 6 possuem código aberto
- + de 70 membros
 - Universidades
 - Indústrias
- 127 plataformas FIPA registradas no AgentCities
 - 87% em pleno funcionamento

FIPA: A Arquitetura Abstrata

- Busca promover interoperabilidade e reutilização
- Permite que arquiteturas concretas diferentes possam interoperar devido ao projeto abstrato em comum das arquiteturas
- Na arquitetura abstrata proposta pela FIPA é definido:
 - Um modelo de serviços e descoberta de serviços
 - Um meio de interoperabilidade no transporte de mensagens
 - Várias formas de linguagens de comunicação de agentes (ACL), conteúdo e representações de serviços de diretórios
- Especificação: SC00001L

Arquitetura Abstrata

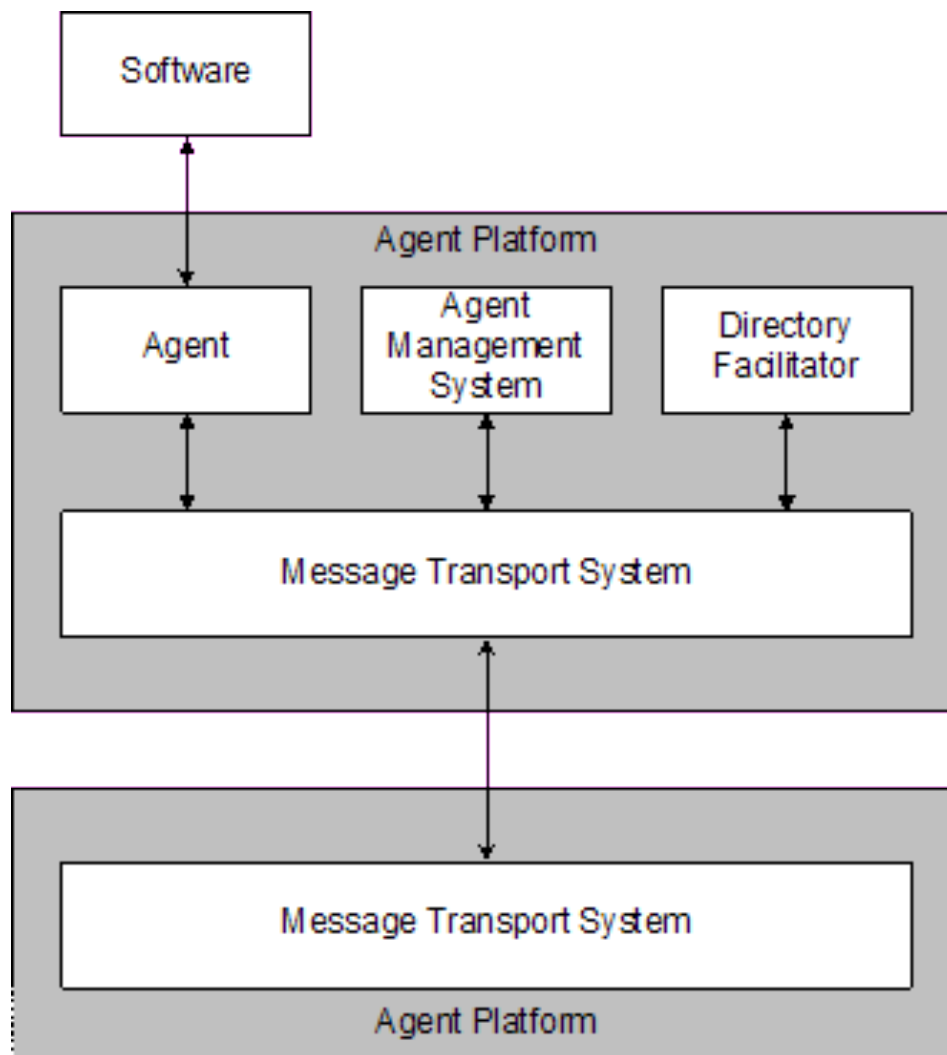


Especificação: SC00001L

Gerenciamento de Agentes

- Fornece a estrutura onde os agentes existem e operam
- Modelo de referência para:
 - criação, registro, localização, comunicação, migração e remoção de agentes
- As entidades definidas são serviços (seguindo as definições da AA) e não dependem de configurações específicas
- Especificação: SC00023J

Gerenciamento de Agentes



Gerenciamento de Agentes

Componentes Lógicos:

- *Agente*
 - *Directory Facilitator - DF (opcional)*
 - *Agent Management System – AMS*
 - *Message Transport Service – MTS*
 - *Agent Platform – AP*
 - *Software*
-
-

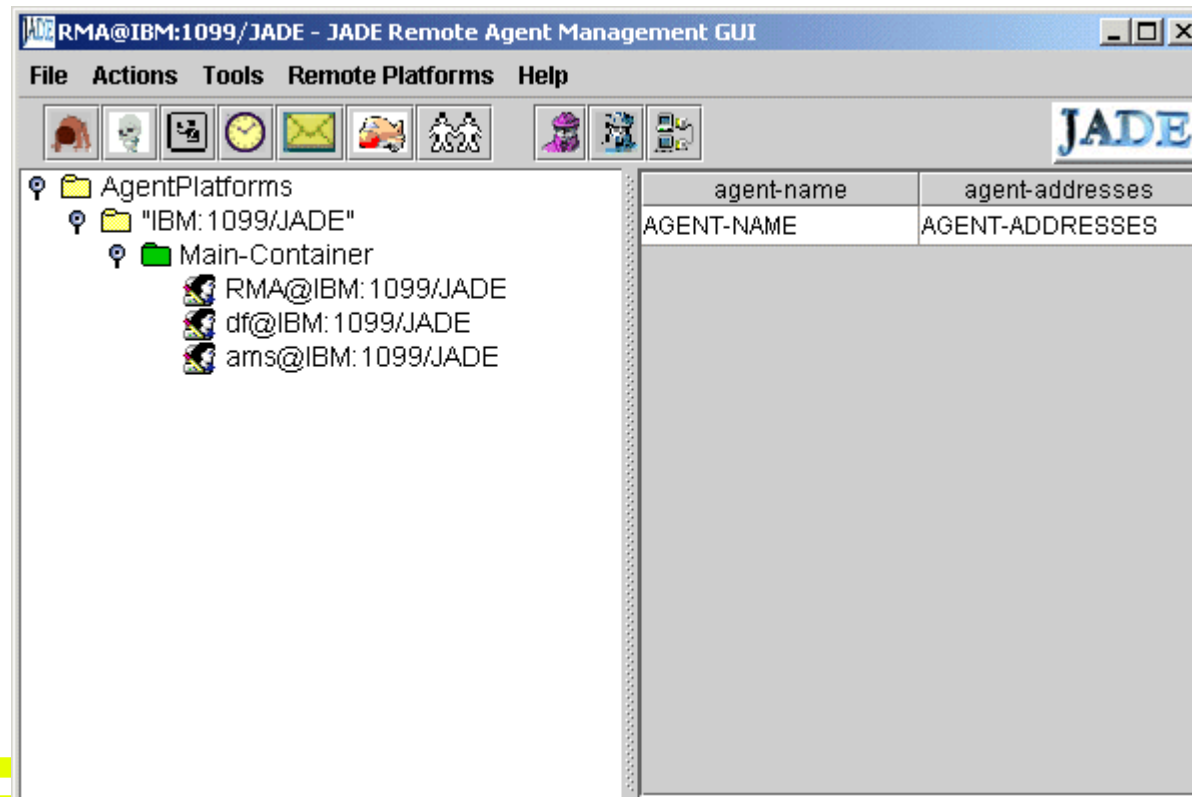
Plataformas de desenvolvimento de sistemas multiagente

- Jade
- Netlogo
- Jason
- Jadex
- Etc...
 - <http://jadex-agents.informatik.uni-hamburg.de/xwiki/bin/view/Links/Agent+Platforms>



JADE (Java Agent DEvelopment Framework)

- *Implementa a arquitetura abstrata FIPA*
- <http://jade.tilab.com/>
- http://www.cs.uta.fi/kurssit/AgO/harj/jade_harkat/doc/tutorials/JADEAdmin/startJade.html



Instalando o JADE

1. Descompactar o arquivo JADE-bin-3[1].4.zip
2. Copiar a pasta jade para a raiz do C:\
3. Incluir no classpath os seguintes caminhos:
c:\jade\lib\jade.jar;
c:\jade\lib\jadeTools.jar;
c:\jade\lib\iiop.jar.jar;
c:\jade\lib\http.jar

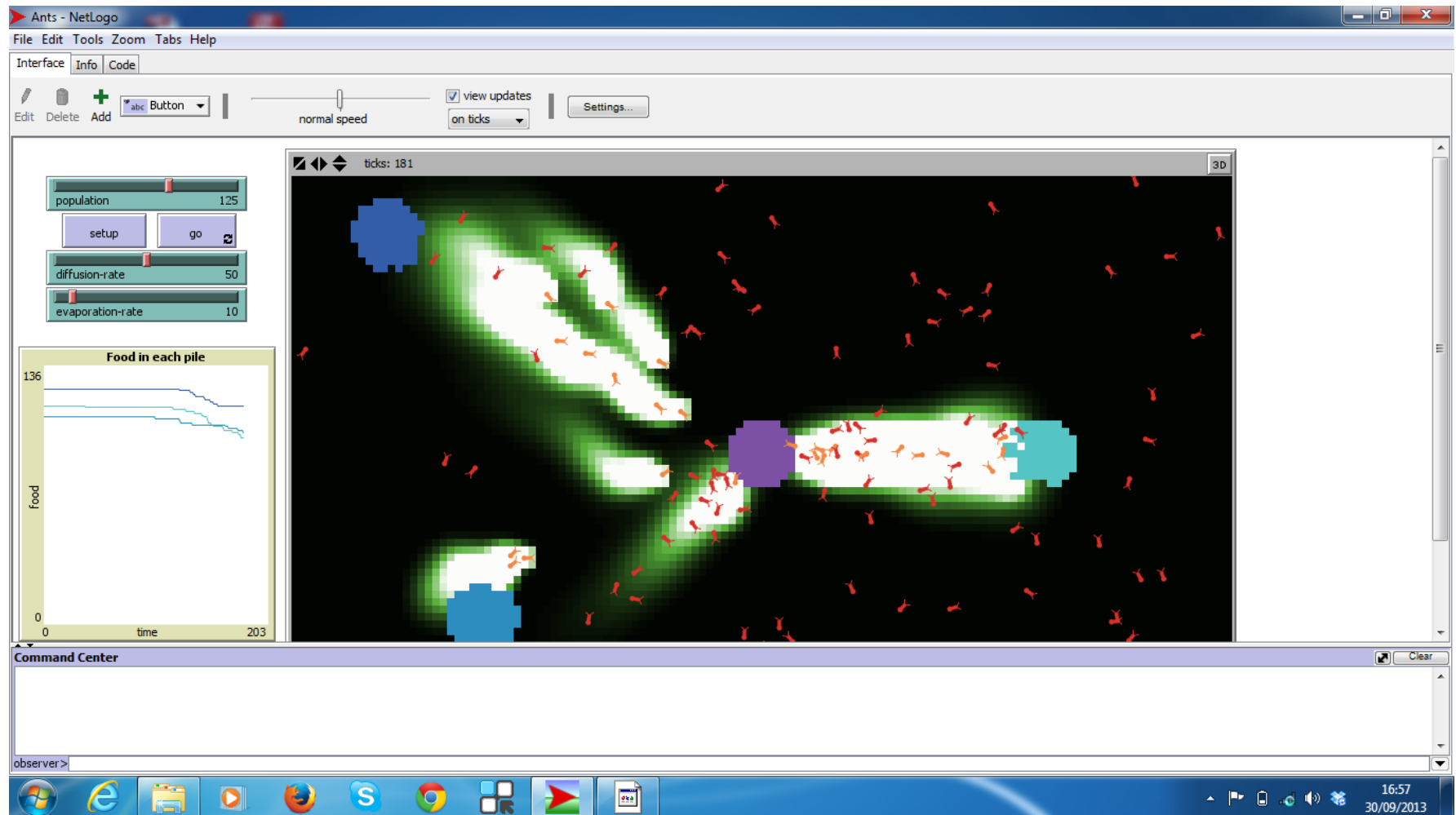
Testando o JADE

1. Descompactar o arquivo JADE-examples-3[1].4.zip
2. Copiar a pasta examples, que se encontra dentro de \jade\src para a pasta do C:\jade, resultando em C:\jade\examples
3. Selecionamos o exemplo HalloWorldAgent, que se encontra na pasta c:\jade\examples\hallo
4. Clique no menu Iniciar -> Executar. Digite o comando: command e clique em OK
5. Execute o comando: cd c:\jade\examples\hallo
6. Compile o arquivo: javac HalloWorldAgent.java
7. Execute o agente através do comando: java jade.Boot
NomeDoAgente:HalloWorldAgent
8. A tela com o agente será apresentada e o agente estará sendo executado

Netlogo

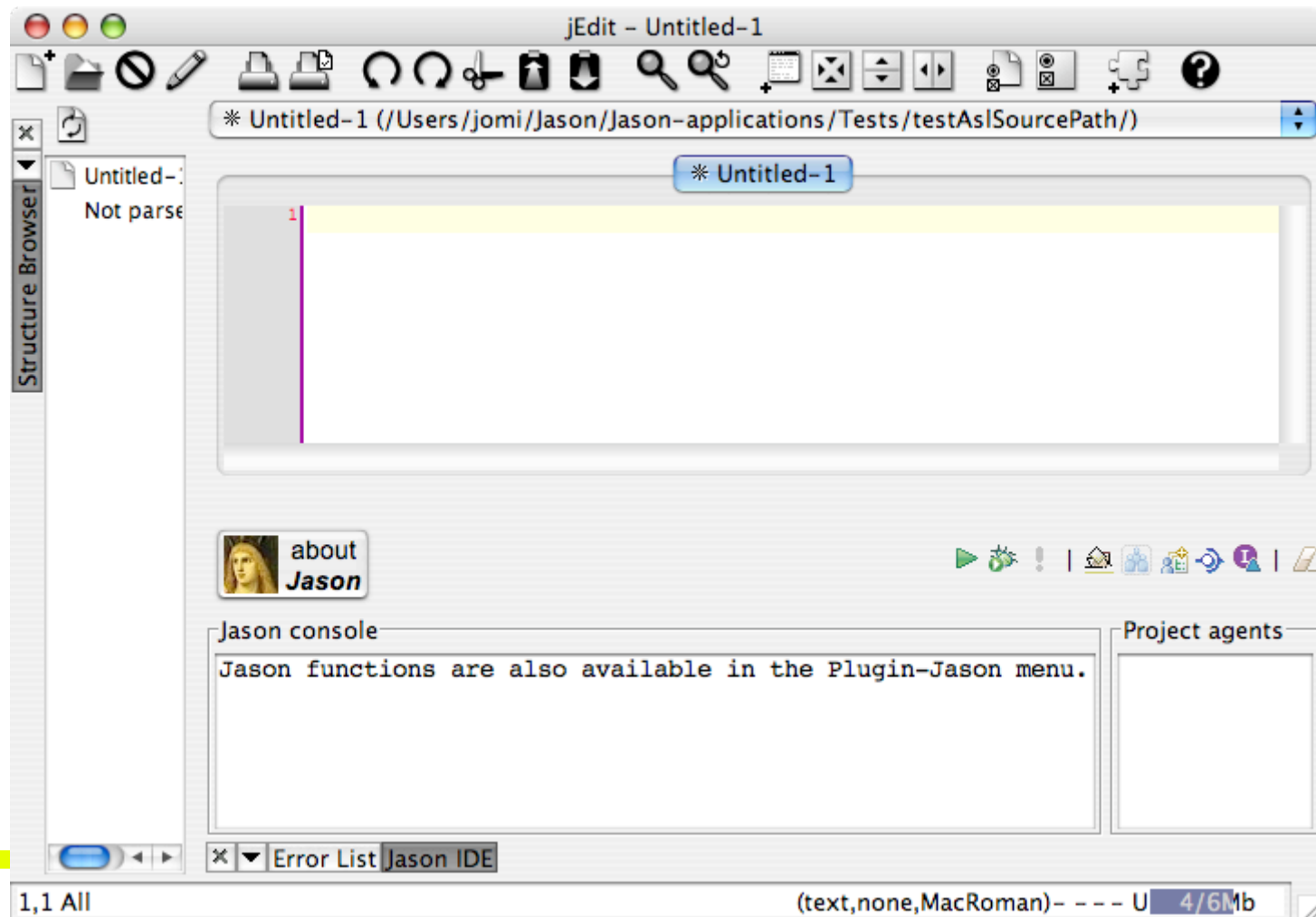
- Ambiente de modelagem programável para simulação de fenômenos naturais e sociais, escrito por Uri Wilensky em 1999 . <http://ccl.northwestern.edu/netlogo/>
 - Adequado para modelar sistemas complexos com centenas ou milhares de agentes operando de forma independente, tornando possível explorar a conexão entre o comportamento micro-nível dos indivíduos e os padrões de nível macro que surgem a partir de sua interação.
 - Explora o comportamento dos agentes sob diferentes condições. É também um ambiente de autoria que permite criar modelos. NetLogo é simples de usar mas avançado o suficiente para servir como uma ferramenta poderosa para os pesquisadores em muitos campos.
 - Possui extensa documentação e tutoriais. Ele também vem com a Biblioteca Models, uma grande coleção de simulações pré-escritas que podem ser utilizados e modificados. Simulam áreas de conteúdo nas ciências naturais e sociais, incluindo a biologia, medicina, física e química, matemática e ciência da computação e economia e psicologia social..
-
-

Netlogo



Jason

<http://jason.sourceforge.net/wp/>



Jason

Jason é um interpretador para uma versão estendida da linguagem AgentSpeak. Ele implementa a semântica operacional da linguagem, e fornece uma plataforma para o desenvolvimento de sistemas multi-agente, com muitos recursos personalizáveis pelo usuário. Jason está disponível Open Source, e é distribuído sob a GNU LGPL

Tutorial: <http://jason.sourceforge.net/mini-tutorial/getting-started/>

Engenharia de Software Orientada a Agentes - AOSE

- Seis desafios dos sistemas multiagente:
 - Decompor problema e alocar tarefas a agentes individualmente,
 - Coordenação, controle e comunicação,
 - Fazer os agentes atuar de forma coerente,
 - Raciocinar sobre os demais agentes e manter a coordenação,
 - Conciliar objetivos conflitantes entre os agentes e
 - Engenharia prática do sistema multagente.

Engenharia de Software Orientada a Agentes - AOSE

- MaSE Multiagent Software Engineering (DeLoach 96) utiliza um número de modelos gráficos que descrevem os tipos de agentes em um sistema e sua interface com os demais agentes bem como uma arquitetura do design interno dos agentes.

Engenharia de Software Orientada a Agentes - AOSE

- Define os sistemas multiagente em termos de classes de agentes e sua organização, em termos o que os agentes comunicam usando conversações
- Duas fases
 - Analise: inclui tres etapas:
 - capturing goals
 - applying use cases
 - refining roles.
 - Design: quatro etapas:
 - creating agent
 - classes, constructing conversations, assembling agent
 - classes, and system design

Multiagent Systems Engineering

