

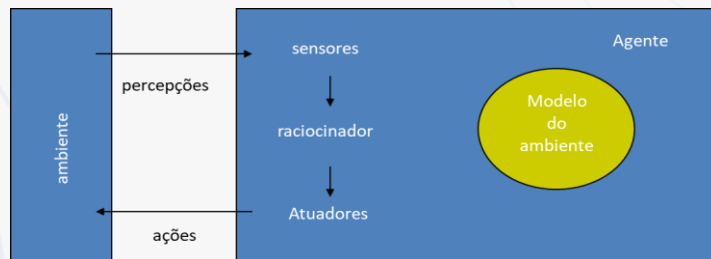


Inteligência Artificial  
Prof. Luiz Antonio Ferraro Mathias

## Agentes

Um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores (câmeras, microfone, teclado, finger etc.) e de agir sobre esse ambiente por intermédio de atuadores (vídeo, alto-falante, impressora, braços, ftp etc.).

Um agente humano tem olhos, ouvidos e outros órgãos como sensores, e tem mãos, pernas, boca e outras partes do corpo que servem como atuadores. Um agente robótico poderia ter câmeras e detectores da faixa de infravermelho funcionando como sensores e vários motores como atuadores.



## Agentes

Wooldrige & Jennings afirmam que agentes são sistemas que apresentam um comportamento determinado por um processo de raciocínio baseado na representação de suas atitudes, tais como crenças, comprometimentos e desejos.

Segundo Adja F. Andrade, um sistema de agentes é semelhante a uma sociedade de seres, que resolvem problemas e trocam informações. Trata-se de um sistema aberto, onde qualquer agente pode entrar ou deixar a sociedade.

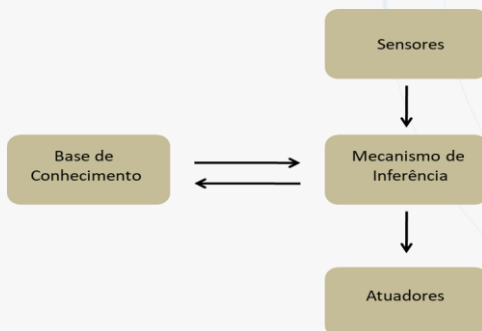
Para GIESE (1998), as partes componentes, assim como o funcionamento da arquitetura básica de um agente autônomo é assim descrita:

- **Setas:** simbolizam o fluxo de dados;
- **Sensores:** recebem informações do ambiente e providenciam dados para o mecanismo de inferência;
- **Mecanismo de inferência:** é o cérebro do agente inteligente. Quando notificado de algum evento, o mecanismo de inferência opera sobre os conjuntos de regras e execução de raciocínio simbólico complexo para determinar como reagir ao evento e qual a ação executar;
- **Base de conhecimento:** é o local onde o agente armazena seu conhecimento;



## Agentes

- **Atuadores:** são responsáveis pela execução das ações do agente sobre o ambiente.



O ambiente dos agentes pode ser: físico: robôs; software: softbots (robôs de software). Realidade virtual (simulação do ambiente físico); softbots e avatares.



## Propriedade dos agentes

**Acessível:** quando os sensores do agente podem obter o estado completo do ambiente. Também dizemos que o ambiente é efetivamente acessível se todas as informações necessárias para se tomar uma decisão ótima estão disponíveis aos sensores do agente;

**Determinista:** o próximo estado do ambiente pode ser completamente determinado pelo estado atual e as ações selecionadas pelo agente;

**Estático:** o ambiente não muda enquanto o agente está definindo a próxima ação a ser realizada;

**Episódico:** a experiência do agente é dividida em episódios. Cada episódio consiste na percepção e ação por parte do agente. Cada episódio não depende das ações que ocorreram em episódios prévios;

**Discreto:** quando existe um número distinto e claramente definido de percepções e ações em cada episódio;

**Contínuo:** quando não existe um número distinto e claramente definido de percepções e ações em cada episódio.

**Autonomia:** capacidade de agir sem a necessidade de intervenção humana, balizados em seus próprios princípios;



## Característica dos agentes

**Mobilidade:** capacidade de se mover através de uma rede de computadores, manter-se estáticos ou residentes na máquina do cliente temporariamente;

**Cooperação:** capacidade dos agentes de trabalharem de forma colaborativa e em conjunto com o objetivo de concluir tarefas em comum (ambiente multiagente);

**Comunicabilidade (habilidade social):** capacidade do agente de se comunicar com outros agentes ou pessoas;

**Aprendizagem:** capacidade de aprender, avaliando as condições ambientes e tomando decisões quando a ação mais correta;

**Reatividade:** habilidade de reação a mudanças no ambiente do agente;

**Pró-atividade:** capacidade de apresentar um comportamento independente, ou seja, mostrar iniciativa em relação ao cumprimento de determinados objetivos;



## Nível de inteligência dos agentes

De acordo com River (apud SOUZA, 1996), as aplicações com agentes apresentam diferentes níveis de inteligência e podem ser classificados como:

- **Baixo:** neste nível, os softwares agentes desempenham tarefas rotineiras, disparadas por eventos externos. Estes agentes não executam redes de regras complexas, não se adaptam a mudanças, ou seja, não alteram suas regras;
- **Médio:** estes agentes utilizam uma base de conhecimento para desenvolver raciocínio em eventos monitorados. Podem adaptar-se a mudanças de condições na base de conhecimento e manipular as novas condições, porém, normalmente não desenvolvem novas regras para tratar as mudanças;
- **Alto:** neste nível de inteligência, os softwares de agentes utilizam tanto aprendizado quando raciocínio na base de conhecimento. Aprendem com o comportamento do usuário, adquirem novos conhecimentos e desenvolvem novas regras, com o tempo podem adaptar-se a mudanças de condições.



## Tarefas executadas pelos agentes

Para Wooldrige & Jennings (apud SOUZA, 1998), existem três classes distintas de agentes segundo a tarefa que executam, são elas:

- **Gopher:** são considerados agentes que executam as tarefas mais simples, baseando-se em suposição e regras pré-especificadas. Por exemplo, o agente pode avisar o usuário que ele possui uma reunião marcada para sexta-feira às 14h;
- **Prestadores de serviço:** são agentes que executam tarefas bem definidas, quando requisitas pelo usuário. Estes agentes podem organizar uma reunião (negociar datas e horários com os participantes);
- **Pró-ativo/preditivo:** são agentes que desempenham as tarefas mais complexas, apresentando serviços ou informações voluntariamente para o usuário, sem ser explicitamente solicitado, sempre que julgar apropriado. Como por exemplo, um agente pode monitorar novos grupos na Internet e retornar discussões que ele acredita serem de interesse do usuário.



## Aquisição da inteligência

De acordo com Nwama (apud GIESE, 1998), os agentes podem ser classificados quanto à aquisição de inteligência em:

- **Reflexivos/reativos:** são agentes simples, possuem um mapeamento de situações e respostas associadas. Quando um estado ambiente ocorre, o agente executa a ação correspondente. Eles atuam utilizando um tipo de comportamento estímulo/resposta, respondendo para o estado presente do ambiente no qual eles estão inseridos.
- **Deliberativos:** derivam do paradigma do pensamento, onde os agentes possuem um modelo simbólico e de raciocínio interno e dedicam-se a planejar e negociar tarefas para alcançar coordenação com outros agentes. Eles preveem que ações serão necessárias para alcançar o objetivo, interpretando o modelo para, então, executá-las.



## Tipologia dos agentes

Com base nas classes de agentes (Gopher, Prestadores de Serviço e Preditivos) podemos identificar alguns tipos de agentes:

- **Colaborativos:** enfatizam a autonomia e cooperação (com outros agentes) para executar suas próprias tarefas. Eles podem aprender, mas não é dada muita ênfase a esta característica em suas operações. Geralmente, agentes colaborativos podem negociar;
- **De interface:** enfatizam a autonomia e aprendizado para executar suas tarefas. Interagem com o usuário, recebendo especificações e fornecendo resultados;
- **Móveis:** são programas escritos especificamente em uma linguagem *script*, o qual pode ser despachado de um computador cliente e transportado para um computador servidor remoto para execução;



## Tipologia dos agentes

- **De Informação:** exercitam o papel de gerenciar, manipular e colecionar informações de diversas fontes distribuídas. Não existe um padrão para suas operações, podendo ter características variadas: estáticos ou móveis, não-cooperativos ou sociais e aprender ou não;
- **Híbridos:** são constituídos pela combinação de duas ou mais filosofias de agentes. Adotados para tentar maximizar a potencialidade e minimizar as deficiências das várias filosofias;
- **Heterogêneos:** conjunto de dois ou mais agentes, os quais pertencem a duas ou mais classes de agentes inteligentes diferentes. Um sistema de agente heterogêneo pode também conter um ou mais agentes híbridos;
- **Inteligentes:** aspiração dos pesquisadores ao invés da realidade.



## Sistemas multiagentes

Um sistema multiagente (SMA) é composto por um conjunto de agentes que interagem em um ambiente comum. Exemplos: Jogos de computador, Interpretação de linguagem natural, Simulação, Assistentes pessoais, Comércio eletrônico, Tutores inteligentes, Controle de redes de energia elétrica.

De acordo com a sua complexidade a arquitetura de um SMA pode ser classificada em três grupos:

- a) Arquitetura simples: quando é composta por um único e simples agente;
- b) Arquitetura moderada: quando é composta por agentes que realizam as mesmas tarefas, mas possuem diferentes usuários e podem residir em máquinas diferentes (Fig. 1);
- c) Arquitetura complexa: quando é composta por diferentes tipos de agentes (Fig. 2), cada um com certa autonomia, podendo cooperar e estar em diferentes plataformas (KNAPIK; JOHNSON, 1998).



## Sistemas multiagentes

Em um SMA uma política de cooperação é necessária, uma vez que Agentes expressam as suas necessidades a outros Agentes a fim de realizar uma determinada tarefa. O processo de cooperação pode ocorrer de duas formas: partilhada de tarefas, Agentes auxiliam outros Agentes em uma determinada tarefa; partilha de resultados, os Agentes disponibilizam suas informações para a comunidade (GIRARDI, 2004). De acordo com a sua cooperação a arquitetura de um SMA por ser classificada em três grupos:

- a. Arquitetura Quadro-Negro: os Agentes não se comunicam de maneira direta, mas através de um quadro-negro (Fig. 3). O quadro-negro é uma estrutura de dados dividida em regiões ou níveis para facilitar a busca de informações. Em SMA funciona como um repositório de mensagens de perguntas e respostas, onde um Agente deposita uma mensagem de pergunta e espera que outro Agente colete esta mensagem, processe e deposite uma mensagem de resposta. Estas mensagens podem ser interpretadas como mensagens em que Agentes solicitam recursos de outros Agentes. Não é uma arquitetura viável para sistemas de tempo real;



## Sistemas multiagentes

- b. Arquitetura de troca de mensagens: os Agentes se comunicam diretamente por troca de mensagens assíncronas. Nesta arquitetura os Agentes precisam saber todos os nomes e endereços de todos os Agentes presentes no sistema. Estas mensagens precisam ser protocoladas para que possam ser encaminhadas e compreendidas pelos Agentes;
- c. Arquitetura Federativa: quando o número de Agentes no sistema é grande, mensagens de broadcasting levarão muito tempo para serem processadas. Por isto, os Agentes são estruturados em grupos ou federações onde existem Agentes facilitadores em cada grupo para receber e encaminhar mensagens entre Agentes remetentes e destinatários.





## Sistemas multiagentes

A Coordenação entre Agentes é referente à maneira como estarão organizados, para alcançar seus objetivos de sistema. Existem dois mecanismos de coordenação:

- **Mecanismo mestre-escravo:** neste mecanismo existem duas classes de Agentes. Agentes mestres, ou gerentes, que distribuem tarefas para Agentes escravos, ou trabalhadores, e ficam à espera de resultados. Os Agentes escravos executam tarefas. Podem existir Agentes facilitadores se, os demais Agentes estiverem distribuídos em grupos;
- **Mecanismo de mercado:** neste mecanismo os Agentes estão organizados em um mesmo nível e todos conhecem as tarefas que todos podem desempenhar. A vantagem é a diminuição da quantidade de mensagens trocadas, pois todos os Agentes se conhecem (PARAISO, 1997).



## Sistemas multiagentes

A comunicação é uma importante característica que os Agentes inteligentes possuem para atingir seus objetivos. Assim, foi definida uma Linguagem de Comunicação entre Agentes (ACL - Agent Communication Language) (FININ; LABROU; MAYFIELD, 1993) que foi dividida em:

- a. Vocabulário: consiste em um dicionário de todos os conceitos utilizados no domínio dos Agentes. Também conhecido como ontologia;
- b. Linguagem Interna: trata-se de linguagem de programação baseada em lógica de primeira ordem, para codificação de dados simples, regras, restrições e expressões. Também conhecida como linguagem KIF (*Knowledge Interchange Format*);
- c. Linguagem Externa: é uma camada linguística que encapsula estruturas KIF, para comunicação mais eficiente. Também conhecida como KQML (*Knowledge Query and Manipulation Language*).



## Limitação dos agentes

De acordo com Wooldrige & Jennings (apud YEPES, 2002), o enfoque em agentes cria uma quantidade de problemas comuns a todas as aplicações baseadas em agentes:

- **Nenhum controlador total do sistema:** uma solução baseada em agentes pode não ser apropriada para os domínios em que os vínculos globais têm que ser mantidos, nos domínios onde uma resposta tempo real deve ser garantida, ou nos domínios em que os becos sem saída ou os travamentos devem ser evitados;
- **Nenhuma perspectiva global:** as ações dos agentes, por definição, são determinadas pelo estado local dos agentes. Entretanto, desde que em quase nenhum sistema realista do agente, o conhecimento global completo é uma possibilidade, isto pode significar que os agentes fazem decisões globais sub-ótimas;
- **Confiança e delegação:** para que os usuários estejam de acordo com a ideia de delegar tarefas aos agentes, devem primeiramente confiar neles. Os usuários têm que adquirir confiança nos agentes que trabalham em seu interesse, e este processo pode levar bastante tempo. Deve-se atentar para que o agente não exceda a sua autoridade com relação ao usuário ou organização; um agente deve saber suas limitações



## Aplicações

Diversas são as aplicações para agentes conforme quadro abaixo:

Tipo de agente	Medida de Desempenho	Ambiente	Sensores	Atuadores
Sistemas de diagnóstico médico	Paciente saudável, minimizar custos, processos judiciais	Paciente, hospital, equipe	Entrada pelo teclado para sintomas, descobertas, respostas do paciente	Exibir perguntas, testes, diagnósticos, tratamentos, indicações
Sistema de análise de imagens de satélite	Definição correta da categoria da imagem	Link de transmissão de satélite em órbita	Arrays de pixels em cores	Exibir a categorização da cena
Robô de seleção de peças	Porcentagem de peças em bandejas corretas	Correia transportadora com peças; bandejas	Câmera, sensores angulares articulados	Braço e mão articulados
Controlador de refinaria	Maximizar pureza, rendimento, segurança	Refinaria, operadores	Sensores de temperatura, pressão, produtos químicos	Válvulas, bombas, aquecedores, mostradores
Instrutor de inglês interativo	Maximizar nota de aluno em teste	Conjunto de alunos, testes de agência	Entrada pelo teclado	Exibir exercícios, sugestões, correções



## Aplicações

As aplicações com agentes podem envolver ainda:

- **Sistemas de Administração de Rede:** podem ajudar a filtrar e a executar ações automáticas a um nível mais alto de abstração, e podem até mesmo serem usados para descobrir e reagir a padrões de comportamento de sistema;
- **Acesso/Administração móvel:** podem processar dados pela sua fonte e enviar somente respostas compactadas para o usuário, em vez de sobrecarregar a rede com grandes quantias de dados não processados ;
- **Mail e Messaging:** podem facilitar funções permitindo que regras de manuseio de correio sejam especificadas previamente, e deixando para os agentes operarem em nome do usuário de acordo com essas regras;



## Estudo de caso

Todo ambiente de Datawarehouse enfrenta o obstáculo da sobrecarga de informações. Portanto, a essência do sucesso no ambiente de DW consiste na capacidade de monitorar e gerenciar o fluxo de dados. Segundo A. L. Gonçalves, citando W. H. Inmon, o monitoramento dos dados no ambiente de um DW envolve:

- a. Número de registro que existem no DW;
- b. Calcular e armazenar os padrões de crescimento dos registros;
- c. Mensurar perfis de dados;
- d. Calcular e armazenar os padrões de crescimento desses perfis de dados;
- e. Verificar a validade dos dados que são carregados, garantindo muita qualidade nos dados do DW.

Para efetuar eficientemente o monitoramento e o gerenciamento do fluxo de dados no DW, podem ser utilizados agentes autônomos responsáveis pela carga, sintetização, envelhecimento e particionamento dos dados.



## Linguagem de programação para Agentes

Várias são as linguagens de programação que podem ser empregadas para desenvolvimento de agentes de software:

- **Java:** linguagem de programação similar em sintaxe ao C++ (W3C, 1996). Conta com o Java Agent Template para a construção de agentes de software que podem se comunicar em uma rede P2P distribuídos pela Internet;
- **LALO (*Langage d'Agents Logiciel Objet*):** linguagem de programação orientada a agentes e um ambiente para desenvolvimento de sistemas multiagentes. Um programa escrito em LALO é traduzido em código fonte C++, para depois ser compilado (FININ, 1997);
- **Obliq:** é uma linguagem interpretada e orientada a objetos com suporte para computação móvel, tal como agente móvel (ITA, 1997a);
- **Phantom:** é uma linguagem interpretada projetada para aplicações distribuídas, interativas e em larga escala, tais como sistemas de conferência distribuída, jogos em rede, e ferramentas de trabalho colaborativo (FININ, 1997);
- **3APL: (*Artificial Autonomous Agents Programming Language*):** Ferramenta experimental de linguagem de programação para o desenvolvimento, implementação, monitoramento e depuração de agentes.



## JADE (Java Agent Development Framework)

JADE (Java Agent Development Framework) é uma estrutura de software totalmente implementada na linguagem Java. Ele simplifica a implementação de sistemas multiagentes por meio de um middleware que atende às especificações do FIPA (*Foundation for Intelligent Physical Agent*) e por meio de um conjunto de ferramentas gráficas que suportam as fases de depuração e implantação. Um sistema baseado em JADE pode ser distribuído entre máquinas (que nem precisam compartilhar o mesmo sistema operacional) e a configuração pode ser controlada por meio de uma GUI remota. A configuração pode ser alterada até mesmo em tempo de execução, movendo os agentes de uma máquina para outra, como e quando necessário. O JADE é totalmente implementado na linguagem Java e o requisito mínimo do sistema é a versão 5 do JAVA (o ambiente de tempo de execução ou JDK).

Além da abstração do agente, JADE fornece um modelo de composição e execução de tarefa simples, mas poderoso, comunicação de um agente ponto a ponto com base no paradigma de passagem de mensagem assíncrona, um serviço de páginas amarelas com suporte para mecanismo de descoberta de assinatura de publicação e muitos outros recursos avançados que facilitam o desenvolvimento de um sistema distribuído.

A documentação técnica desta plataforma pode ser obtida em <https://jade.tilab.com/>.

