

Universidade Federal de Santa Maria
Departamento de Informática
Prof. Cesar Tadeu Pozzer
Disciplina: Computação Gráfica Avançada
pozzer@inf.ufsm.br
Período: 2007/2

Agentes

We want to build intelligent actors, not just intelligent thinkers. -
Martha Pollack, from Computers and Thought Lecture, IJCAI-91.

Tópicos

<ul style="list-style-type: none">- Agentes<ul style="list-style-type: none">• Conceituação• Propriedades<ul style="list-style-type: none">- Autonomia- Habilidade social- Capacidade de reação- Capacidade pró-ativa- Categorias<ul style="list-style-type: none">• Reativos• Cognitivos• Híbridos<ul style="list-style-type: none">- Funcionais- Estados mentais	<ul style="list-style-type: none">- Multiagentes<ul style="list-style-type: none">• Aspectos fundamentais<ul style="list-style-type: none">- Estrutura- Organização- Coordenação- Interação• Aspectos arquiteturais<ul style="list-style-type: none">- Hierárquica- Nivelada• Aspectos do Ambiente<ul style="list-style-type: none">- Acessível x Inacessível- Agentes x Programas- Agentes x Objetos- Agentes em Jogos<ul style="list-style-type: none">• NPCs<ul style="list-style-type: none">- Visão- Audição- Autonomia- Objetivos
--	---

1. Conceituação

Existem diversas definições para agentes. Segundo Corrêa [COR 94], um agente pode ser definido como uma **entidade que funciona contínua e autonomamente** em um ambiente no qual existem outros processos e agentes. Para Amandi [AMA 97], um agente é uma entidade computacional com um **comportamento autônomo que lhe permite tomar suas decisões para agir**, levando em consideração as mudanças acontecidas no ambiente em que atua e o seu desejo de alcançar seus objetivos. Vendo agentes sobre outro enfoque, [SHO 93] afirma que um agente é uma **entidade à qual se atribuem estados**, denominados de estados mentais. Os estados mentais usuais são: crenças, decisões, capacidades, objetivos, intenções, compromissos e expectativas, conceitos análogos ou similares aos humanos. É em função destes estados que o agente age e toma suas decisões.

Segundo [D'AM 95], agentes, sob o ponto de vista prático, são **robôs que atuam em um ambiente**, interagindo com outros robôs ou com humanos em linguagem natural e utilizando sensores para captar informações do meio.

Como observado por [WOO 99], não há definição universalmente aceita do termo agente, mas há um consenso geral de que **autonomia é a idéia central**. Em termos de software, autonomia está relacionada a processos que operam sem a intervenção direta do homem, e que podem atingir seus próprios objetivos.

Uma explicação para esta divergência de opiniões surge do fato de existirem vários tipos de agentes, usados em diferentes domínios de aplicações. E vale também lembrar que agentes **são criados para atenderem propósitos específicos**, ou seja, são entidades que encapsulam conhecimentos sobre algum domínio [JUC 01]. Não existe um agente genérico que possa ser aplicado na resolução de qualquer tipo de problema.

Segundo [ZAM 01], o termo agente vem sendo utilizado para denotar desde simples processos de hardware e/ou software, até entidades sofisticadas capazes de realizar tarefas complexas. Segundo [RUS 95], agente é visto como um **sistema capaz de perceber através de sensores e agir em um dado ambiente através de atuadores**.

As seguintes propriedades são comumente aceitas, e tornam um agente diferente de um simples objeto de software:

- **Autonomia:** Permite ao agente executar a maior parte de suas ações sem interferência direta de agentes humanos ou de outros agentes computacionais, possuindo controle total sobre suas ações e estados internos (em jogos, a intervenção humana dificilmente ocorre, visto que agentes são usados para simular personagens autônomos [CUN 01]). Para ter autonomia, o agente deve ter um certo grau de inteligência, capacitando-o a sobreviver em um ambiente dinâmico e por vezes não benigno [COR 94]. A inteligência deve agir sobre seu conhecimento embutido, sua experiência, bem como em suas percepções do mundo [RUS 95]. Isso implica que, ao ser criado, um agente já seja provido de um certo grau de conhecimento, bem como capacidade de aprendizagem;
- **Habilidade social:** Esta propriedade permite que agentes, ou por impossibilidade de resolução de certos problemas, ou por algum tipo de conveniência, interajam com outros agentes (humanos ou computacionais), para completarem a resolução de seus problemas, ou ainda para auxiliarem outros agentes. Para que isso ocorra, é necessário que os agentes tenham capacidade de comunicação e mecanismos decisórios internos, que definam quando e quais interações são apropriadas;
- **Capacidade de reação:** Um agente deve, por meio de recursos de percepção, como visão e audição, captar e reagir a alterações no ambiente em que estiver inserido. O grau de complexidade da reação vai depender do tipo do agente (*Reativo ou cognitivo*);
- **Capacidade pró-ativa:** Agentes do tipo cognitivo, além de atuarem em resposta às alterações ocorridas em seu ambiente, apresentam um comportamento orientado a objetivos, tomando iniciativas quando julgarem apropriado.

Além destas propriedades, pode-se também destacar a temporalidade (o agente decide se permanece ou não em um ambiente), adaptabilidade (capacidade de se adaptar às modificações do ambiente), mobilidade (capacidade de locomoção entre ambientes), entre outras.

1.1. Categorias de Agentes

O ponto chave da criação de agentes se concentra na definição de habilidades que permitam aos agentes agirem de maneira autônoma. Estas habilidades podem incluir percepção e interpretação de mensagens, raciocínio baseado em crenças, tomada de decisão, planejamento e habilidade para executar planos incluindo passagem de mensagens. Os agentes podem ser categorizados quanto ao nível de capacidade de resolução de problemas:

- **Reativos:** Reagem à alterações no ambiente ou a mensagens de outros agentes. Não têm capacidade de raciocínio sobre suas intenções, reagindo tão-somente sobre regras e planos pré-definidos. Suas ações podem ser: atualizar a base de dados (fatos) e enviar mensagens para outros agentes ou para o ambiente;
- **Cognitivos:** Também conhecidos como Intencionais, Racionais ou Deliberativos, têm a habilidade de raciocínio sobre suas intenções e crenças, e podem, por um processo explícito, criar e escolher ações e planos a serem realizados. Planos podem ser revisados em caso de detecção de conflitos. Quando um agente possui modelos de outros agentes, sobre os quais raciocina para tomar decisões e criar planos, é considerado *Social*;
- **Híbridos:** A escolha das ações é realizada utilizando uma combinação de técnicas de arquiteturas reativas e cognitivas. Esta arquitetura foi proposta como uma alternativa para solucionar as deficiências cognitivas (incapacidade de reação rápida e adequada a situações não previstas) e reativas (incapacidade de descobrir alternativas para o seu comportamento quando a situação do mundo diverge dos seus objetivos). Segundo [NAR 03], os agentes híbridos utilizam planejamento de alto nível de abstração em uma fase de pré-processamento, enquanto decisões sobre alternativas de refinamento menos significativas são tratadas por sistemas reativos.

Segundo [OLI 96], as arquiteturas de agentes cognitivos podem ser divididas em duas categorias:

- **Funcionais:** O agente é composto por módulos responsáveis por cada uma de suas funcionalidades necessárias para sua operação. Além dos módulos, o agente possui conhecimento, objetivos, capacidade de percepção, comunicação, decisão e raciocínio;
- **Baseadas em estados mentais:** A estrutura do agente é definida sob uma perspectiva psicológica, que possui componentes (estados) mentais como crença, desejos, intenções, expectativas, entre outros.

Baseado na arquitetura de estados mentais, foi desenvolvida a arquitetura BDI, que considera apenas três estados mentais para a descrição do processamento interno de um agente: crença, desejo e intenção (*belief, desire, intention*). Além dos estados mentais, nesta arquitetura são definidos elementos de controle através dos quais o agente seleciona racionalmente o curso de suas ações. As crenças representam o conhecimento do agente acerca do ambiente onde está situado. Os desejos, segundo [RAO 95], representam o estado motivacional do sistema, ou seja, estados ou ações que o agente quer que se verifiquem. As intenções são os desejos a serem executados, e são usadas no processo de decisão do curso de ações que devem ser tomadas.

1.2. Multiagentes e Inteligência Artificial Distribuída (IAD)

Como já mencionado, um agente encapsula conhecimentos necessários para a resolução de um determinado problema. Desta forma, a interação de agentes com capacidades específicas pode ser usada na resolução de problemas que um único agente não poderia resolver. Segundo [ZAM 00], existem duas classes de sistemas com múltiplos agentes:

- **Resolução Distribuída de Problemas (DPS):** Divide a solução de um problema em particular entre um número de módulos que cooperam compartilhando conhecimento sobre o problema e sobre as soluções envolvidas. Estes módulos estão distribuídos em computadores interligados através de uma rede. Um controle central coordena tanto a cooperação quanto a competição entre os módulos autônomos;
- **Sistemas abertos:** Nesta classe, os agentes não são necessariamente projetados para atingirem um objetivo comum, podendo ingressar no sistema de maneira dinâmica. Desta forma, a chegada de agentes desconhecidos precisa ser levada em consideração, bem como a possível existência de comportamentos não benevolentes no curso das interações. Dentro desta classificação estão inseridos os Sistemas *Multiagente (SMA)*, que estudam o comportamento de um conjunto de agentes autônomos (possivelmente pré-existentes) cujo objetivo comum é a solução de um dado problema cuja resolução está além das capacidades de um único indivíduo [O'HA 96].

Para que sistemas multiagentes sejam viáveis, deve existir um sistema de coordenação dos agentes envolvidos [BOM 88], de forma que os agentes possam colaborar entre si para cumprir com a responsabilidade no sistema [AMA 97]. Esta coordenação deve permitir que estes agentes possam coordenar seus conhecimentos, objetivos, habilidades e planos individuais de uma forma conjunta, em favor da execução de uma ação ou da resolução de algum problema.

Sistemas multiagentes apresentam várias vantagens quando comparados com soluções convencionais [JEN 96]:

- Resolução mais rápida de problemas, pelo uso do paralelismo;
- Redução no fluxo de dados, pois somente soluções parciais em alto nível são transmitidas para outros agentes, ao invés de dados brutos para um lugar central;
- Maior flexibilidade, pois usam agentes de diferentes habilidades dinamicamente agrupados na resolução de problemas;
- Aumento da segurança, pois permite que agentes assumam responsabilidades de agentes que venham a falhar.

Uma característica importante de um SMA é permitir que agentes com diferentes habilidades possam interagir na resolução de um problema. O nível de heterogeneidade entre os agentes, segundo Jennings [JEN 96], pode assumir três níveis:

- **Baixa heterogeneidade:** Os agentes são idênticos ou diferem apenas pelos recursos disponíveis a eles;
- **Média heterogeneidade:** Os agentes diferem também pelos métodos de resolução de problemas;
- **Alta heterogeneidade:** Os agentes somente compartilham uma mesma linguagem de interação, sendo que suas outras características podem ser completamente diferentes.

A estrutura de um SMA permite que um determinado grupo de agentes possa cooperar para a solução de problemas que um único agente não poderia resolver. Esta abordagem se adapta ao desenvolvimento de sistemas complexos de software, pois quebra a complexidade do problema (decomposição) em problemas menores que podem ser solucionados por agentes cooperando e interagindo de modo organizado. Além da decomposição, a abstração da orientação a agentes é uma boa maneira de modelar sistemas complexos. Também é adequada para

identificar e gerenciar relacionamentos organizacionais na representação das dependências e interações que existem em um sistema complexo [JEN 99].

Para que um SMA seja eficaz, devem-se considerar critérios que viabilizem e garantam a coerência das ações dos agentes de modo a atingir os objetivos esperados. [MOU 96] classifica duas perspectivas para análise de agentes:

- **Perspectiva do agente:** Define características particulares (internas) do agente, que são de certa forma transparentes para os demais membros do grupo. Nesta perspectiva, pode-se ressaltar a categoria e arquitetura do agente, sua organização do conhecimento, adaptação, forma de aprendizado e raciocínio;
- **Perspectivas do grupo:** Uma vez que os agentes não estão mais agindo sozinhos, devem existir regras para garantir a correta integração entre os diversos membros do grupo: coordenação, organização, cooperação, comportamento coerente, planejamento, comunicação e interação. Estes itens provêem meios para que as ações dos agentes façam sentido em relação aos objetivos comuns do grupo [MOU 96].

A forma como cada agente realiza suas tarefas (papel), em um SMA, ou seja, a estrutura interna do agente, é irrelevante. O papel de um agente é aquilo que é esperado que ele faça dentro da organização, ou seja, um conjunto de responsabilidades bem definidas dentro do contexto global do sistema que o agente pode cumprir com um certo grau de autonomia [ZAM 00]. Cada agente tem controle próprio sobre seu processamento, sendo totalmente responsável em cumprir seu papel. Para isso, o agente deve encapsular funcionalidades que o habilitem a tal fim.

Para SMA, a perspectiva do grupo é de vital importância. O que deve ser garantido é que o grupo possa agir em sintonia sob diversos aspectos. Os aspectos que ditam a perspectiva do grupo na construção de um SMA podem ser divididos em 3 grandes grupos: fundamentais, arquiteturais e do ambiente. Cada um destes grupos é melhor descrito nas próximas seções:

1.2.1. Aspectos Fundamentais de SMA

Estes aspectos visam garantir a compatibilidade das ações de agentes individuais dentro dos objetivos do grupo.

Estrutura

A forma como os problemas são revolvidos pelo grupo e o papel que cada agente deve desempenhar deve estar bem definido em um SMA. A estrutura define as relações de informações e controle entre agentes para atender às seguintes condições [MOU 96]:

- **Cobertura:** Qualquer habilidade necessária para a resolução do problema deve estar inserida no rol de habilidades de pelo menos um agente;
- **Conectividade:** Agentes devem interagir de maneira a permitir que suas habilidades sejam integradas e desempenhadas no sentido de contribuírem para uma solução global;
- **Potencialidade:** Cobertura e conectividade devem ser atingíveis dentro de limitações computacionais e de comunicação, assim como as especificações de confiabilidade do grupo.

Organização

Para que agentes possam atingir objetivos pela cooperação mútua, compromissos globais, crenças mútuas, e intenções comuns aos agentes devem ser especificados [MOU 96]. Estas diretrizes são como leis que regem o comportamento de uma sociedade, para assegurar que ações individuais não venham a conflitar com o ideal do grupo. Elas devem sim facilitar ou mesmo viabilizar a realização dos objetivos globais da sociedade.

Coordenação

Para que um grupo de unidades autônomas e inteligentes possa interagir de maneira correta e eficiente, deve existir algum mecanismo de coordenação das ações desempenhadas por cada membro da equipe. A coordenação deve ocorrer a nível de conhecimento, objetivos, habilidades e planos individuais de uma forma conjunta, em favor da execução de uma ação ou da resolução de algum problema.

Sem um mecanismo de coordenação, os benefícios providos da resolução distribuída de problemas desaparecem e a comunidade pode degenerar em uma caótica coleção de indivíduos que agem de forma não coesa em relação ao sistema como um todo [MOU 96, JEN 96].

Segundo [JEN 96], a coordenação entre os agentes se faz necessária por uma série de fatores. Geralmente existem dependências entre as ações dos agentes, ou seja, a ação de um agente pode ser pré-requisito da ação de outro agente. Além disso, nenhum indivíduo tem competência, recursos ou informação suficientes para resolver um problema completo de forma independente, onde devem ser garantidos o respeito às restrições globais, à solução do problema e à viabilização dos procedimentos que garantam a harmonia quando da execução de uma tarefa de forma conjunta por mais de um agente.

A coordenação pode ocorrer de forma centralizada ou descentralizada. A coordenação centralizada pode ser desempenhada por um agente coordenador que, por reunir informações sobre toda a sociedade, é responsável por criar planos e atribuir tarefas aos membros desta sociedade [MOU 96, JEN 96]. A falha no funcionamento deste agente poderia comprometer o funcionamento de todo o sistema, caso não estejam implementados mecanismos de tolerância a falhas.

Umas das principais vantagens desta abordagem, que é um dos principais problemas apresentados em uma solução descentralizada, é a facilidade de se verificar o estado global do sistema, que está disperso através de seus componentes.

Interação

A interação entre agentes se faz necessária toda vez que um único agente não é capaz de, sozinho, realizar uma ação para atender aos objetivos do grupo. Para que uma interação ocorra em um momento determinado e conhecido, devem existir no mínimo dois agentes nesse dado momento. Deve ser conhecida a necessidade da interação para que o conteúdo da interação ou comunicação possa ser definido, dentro de uma gama de recursos disponíveis a serem utilizados para tal fim. Devem também estar definidos os mecanismos para que seja viabilizada a compreensão mútua entre os participantes da interação [BON 88].

A comunicação é um meio de realizar interação entre agentes. É composta por duas partes: o envio da mensagem (ação), realizada por um agente, e a recepção da mensagem (percepção), que pode ser endereçada a um número qualquer de agentes.

1.2.2. Aspectos Arquiteturais

O modelo arquitetural de um SMA define a disposição dos relacionamentos existentes entre os agentes. Segundo [SHE 98], a arquitetura de um SMA pode ser enquadrada em 4 categorias:

- **Hierárquica:** Tanto os agentes quanto sua comunicação são dispostos em uma estrutura hierárquica. Neste modelo, cada agente pode se comunicar apenas com os agentes supervisionados por ele ou por seu próprio supervisor. Esta restrição na comunicação reduz a quantidade de informação trocada entre os membros do sistema, porém a estrutura não permite que os agentes organizem-se dinamicamente para melhor atender às necessidades de uma tarefa específica onde os níveis mais baixos dependem dos níveis mais altos, e os níveis mais altos podem ter controle parcial ou total em relação aos níveis mais baixos;
- **Nivelada:** Sendo todos os agentes dispostos em um mesmo nível, a comunicação entre eles ocorre de forma direta, o que permite a criação de estruturas de maneira dinâmica para realizar uma tarefa específica;
- **Agentes compostos por agentes:** Pressupõe a existência de alguns agentes que são componentes de outros agentes;

Outra classificação arquitetural de um SMA pode ser feita em relação ao nível de abertura, que define a forma de ingresso ou saída de agentes em um grupo [SHE 98]:

- **Abertura Dinâmica:** Permite que agentes entrem ou saiam do sistema sem a necessidade de notificação aos outros agentes. Esta característica traz uma maior flexibilidade de adaptação do sistema em relação a mudanças no ambiente, porém exige serviços e computação adicionais para suportar o fato de que agentes podem aparecer e desaparecer de forma imprevisível. O sistema não precisa ser reiniciado quando ocorre alteração no quadro de agentes;
- **Abertura Estática:** Semelhante ao de abertura dinâmica, porém, na entrada de um novo agente, os demais devem ser notificados;

- **Abertura Off-Line:** Toda inclusão ou remoção de agentes somente pode ser realizada com o sistema desativado. Após a adição ou remoção, o sistema deve ser reiniciado.

1.2.3. Aspectos Ambientais

Independente dos aspectos fundamentais e arquiteturais, todo agente está situado em um ambiente. O ambiente constitui o contexto onde todas as interações entre os agentes ocorrem. Através do ambiente ocorre a dispersão do controle, dos dados e do conhecimento pela comunidade de agentes [D'AM 95].

O conhecimento, por parte do agente, do tipo de ambiente onde está situado é de grande importância para sua sobrevivência e adaptação frente às alterações do mundo, pois irá determinar como o agente deve atuar e perceber alterações do ambiente. As principais classificações a serem feitas em um ambiente são [RUS 95]:

- **Acessível x Inacessível:** Essa característica determina se o aparato sensorial do agente lhe fornece um estado completo do ambiente. Se isto ocorre, o ambiente é considerado acessível; caso contrário, é considerado inacessível. Um ambiente é considerado efetivamente acessível se os sensores detectam todos os aspectos relevantes para a escolha da ação. No caso de um ambiente ser acessível, não é necessário que o agente mantenha qualquer representação interna do mundo;

2. Agentes x Programas

Essa é uma questão que surgiu com o advento da tecnologia de agentes. O que um agente precisa ter para se diferenciar de um programa convencional? Nesta seção são extraídos fragmentos de textos de dois artigos que discutem o que realmente pode ser caracterizado como agente.

2.1. Primeira Referência

Stan Franklin and Art Graesser. *Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agent*. Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.

[Russell and Norvig 1995 - AIMA, page 33] "An agent is anything that can beviewed as perceiving its environment through sensors and acting upon that environment through effectors."

AIMA is an acronym for "Artificial Intelligence: a Modern Approach," a remarkably successful new AI text that was used in 200 colleges and universities in 1995. The authors were interested in software agents embodying AI techniques. Clearly, the AIMA definition depends heavily on what we take as the environment, and on what sensing and acting mean. If we define the environment as whatever provides input and receives output, and take receiving input to be sensing and producing output to be acting, every program is an agent. **Thus, if we want to arrive at a useful contrast between agent and program, we must restrict at least some of the notions of environment, sensing and acting.**

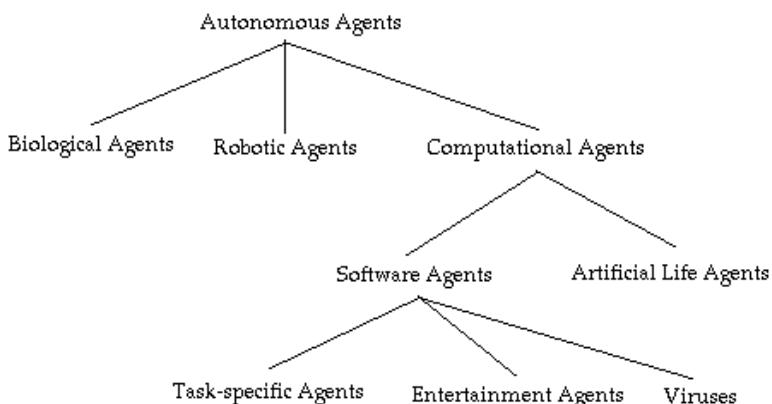
An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.

1. Autonomous agents are situated in some environment. Change the environment and we may no longer have an agent. **A robot with only visual sensors in an environment without light is not an agent.** Systems are agents or not with respect to some environment. The AIMA agent discussed above requires that an agent "can be viewed" as sensing and acting in an environment, that is, there must exist an environment in which it is an agent.
2. What about **ordinary programs?** A payroll program in a real world environment could be said to sense the world via its input and act on it via its output, **but is not an agent because its output would not normally effect what it senses later.** A payroll program also fails the "over time" test of temporal continuity. It runs once and then goes into a coma, waiting to be called again. Most ordinary programs are ruled out by one or both of these conditions, regardless of how we stretch to define a suitable environment. All software agents are programs, but not all programs are agents.

3. Nor are software agents defined by their tasks. A spell checker adjunct to a wordprocessor is typically not an agent for the reasons given in the preceding paragraph. However, **a spell checker that watched as I typed and corrected on the fly might well be an agent.** Tasks can be specified so as to require agents to fulfill them.

Our definition of an autonomous agents has succeeded in distinguishing between agents and programs. An agent need not be a program at all; it may be a robot or a school teacher. Software agents are, by definition, programs, but a program must measure up to several marks to be an agent. Every agent, by our definition, satisfies the first four properties:

Property	Other Names	Meaning
reactive	(sensing and acting)	responds in a timely fashion to changes in the environment
autonomous		exercises control over its own actions
goal-oriented	pro-active purposeful	does not simply act in response to the environment
temporally continuous		is a continuously running process
communicative	socially able	communicates with other agents, perhaps including people
learning	adaptive	changes its behavior based on its previous experience
Mobile		able to transport itself from one machine to another
flexible		actions are not scripted
character		believable "personality" and emotional state.



2.2. Segunda Referência

Leonard N. Foner. *What's An Agent, Anyway? A Sociological Case Study.* Agents Memo 93-01. Available from the Agents Group, MIT Media Lab. Copyright (C) May 1993.

The term "agent" has been picked up, widely appropriated, and in many cases misappropriated, by technical publications, lay publications, and many researchers in computer science.

There has been a flurry of references to "agents" in both the lay and technical press in recent months. While the idea of computational agents that do one's bidding goes back decades, the explosion of recent press has been astonishing. Pattie Maes and her group's work on learning interface agents has been cited in at least two dozen publications in the last year or so, many of them lay press (e.g., The Los Angeles Times, Mass High Tech or semi-technical press (e.g., MacWeek, Mac-World, etc). A symposium at MIT on agents and their programming, held in October of 1992, drew at least a thousand participants. Further, buzzword frenzy has hit the industry as well. We have, for example, a Macintosh program called "Magnet", recently released from No Hands Software, which bills itself as "the first intelligent agent for the Macintosh". In fact, the program is essentially a file-finder, which can locate files matching certain characteristics and perform some operation on the set, as well as other utilities such as synchronizing a PowerBook filesystem with that of some "home", stationary Macintosh.

We also have "At Your Service", from Bright Star, whose promotional literature starts with, "Remember the excitement you felt the first time you turned on a Mac? Now you can relive the magic and realize its full potential with At Your Service, the first animated Desk Accessory to give your Mac a 'human' personality.

At Your Service features Phil, your Personal Assistant...he talks...he listens...he works for you!" In fact, this program is pretty trivial: it utters a spoken greeting when the Mac is turned on, issues reminders for preset events, issues alerts when email arrives, reminds you to stop typing every few minutes to ease hand and eye strain, contains a screen saver, and so forth. At the Symposium on Agents at MIT in October 1992, Irene Greif of Lotus presented on a groupware version of Lotus 123 that made collaboration between multiple spreadsheet users easier. This is certainly respectable technology, but I would hardly call it an "agent", and her presentation had to work fairly hard to force it into the mold of an "agent-oriented" piece of work.

The popular imagination about agents that do our bidding has even extended to taking attributing human emotional and intentional states, out of context, to programs and shamelessly running with them. For example, consider Michael Shrage's article in The Boston Globe, March 7, 1993, imported by the Los Angeles Times Syndicate, entitled, "A New Dimension in Deception". In this article, Shrage picks up on the fact that a calendar-scheduling agent may have to decline a proposed meeting time and propose another to suit its user's tastes. However, a simple declination may not be sufficient; the agent might have to "lie" and "fabricate" a nonexistent appointment to get its peer agent to reschedule. Why this is a "new" dimension in deception, whereas using an answer machine for call-screening is not, is never explained. And perhaps predictably, the same loose talk about "thinking"⁵ that eventually disillusioned many about artificial intelligence seems to be cropping up again in the guise of agents that learn.

What is so irresistably attractive about this concept that it has exploded into faddism in this way? Why has the press picked up on agents and produces so many articles about them? Why have so many companies suddenly tried to position their products as based on agents, even when they are manifestly no such thing? An easy answer would be simple **fashion**. Fads sell. But rather than just leaving it at that, let us examine closely one of the best examples I know of which seems to be to be a true agent.

3. Agente x Objeto [Notas de aula prof. Geber]

Agente	Objeto
<ul style="list-style-type: none"> ☛ Intencionalidade: <ul style="list-style-type: none"> ☛ Encapsula <i>objetivos próprios</i> (mesmo que implicitamente) além de dados e métodos 	<ul style="list-style-type: none"> ☛ Sem objetivo próprio
<ul style="list-style-type: none"> ☛ Autonomia de decisão <ul style="list-style-type: none"> ☛ Pode iniciar ação da sua própria iniciativa para satisfazer seus objetivos (pro-atividade) ☛ Pode negar-se a um pedido de ação da parte de outro agente (negociação) 	<ul style="list-style-type: none"> ☛ Sem autonomia de decisão: <ul style="list-style-type: none"> ☛ Executa <i>apenas</i> quando invocado por outros objetos ☛ Executa <i>sempre</i> que invocado por outros objetos
<ul style="list-style-type: none"> ☛ Entrada e saída mais complexa: sensores e atuadores 	<ul style="list-style-type: none"> ☛ Entrada e saída: parâmetros e resultado de métodos
<ul style="list-style-type: none"> ☛ Continuidade temporal: sempre monitorando o ambiente 	<ul style="list-style-type: none"> ☛ Descontinuidade temporal: ativo apenas durante invocação dos seus métodos
<ul style="list-style-type: none"> ☛ Granularidade maior: <ul style="list-style-type: none"> ☛ Encapsula código do tamanho de um pacote ou componente ☛ Composto de vários objetos quando implementado no paradigma OO 	

4. Agentes e IA em Jogos de Computador

Para tornar jogos interativos mais reais e desafiadores, está se tornando muito comuns o uso de personagens autônomos controlados pelo computador, implementados por meio de agentes, também conhecidos como NPCs (*Non-player characters*).

NPCs são muito comuns em jogos de ação em primeira pessoa, como Quake, Doom, onde o jogador deve lutar contra inimigos virtuais, cujo controle é realizado exclusivamente pelo computador. Em outros gêneros de jogos, como no The Sims, pode haver uma alternância de controle autônomo e controle do jogador.

O nível de desafio apresentado pelos NPCs vai depender da tecnologia usada em sua implementação. Para criar comportamentos mais realistas, desenvolvedores de jogos utilizam, de alguma forma, técnicas de representação de Vida Artificial (*A-Life*), cuja origem baseia-se no estudo de organismos vivos reais.

As técnicas de *A-Life* visam emular o comportamento destes organismos por uma variedade de métodos que podem fazer uso de regras explícitas, algoritmos genéticos, algoritmos de agrupamento, entre outros. Ao invés de implementar explicitamente uma grande variedade de comportamentos complexos, os desenvolvedores devem quebrar o problema em problemas menores, por meio de implementações de comportamentos simples que, interligados por uma hierarquia de tomada de decisões, podem produzir comportamentos complexos de forma automática, ou seja, sem implementação direta [WOO 03].

Segundo [CUN 01], agentes em jogos de computador compartilham das mesmas características fundamentais de agentes (tais como as expostas no Capítulo 1). Neste contexto, uma boa definição é feita por [NAR 03], que define o agente como uma entidade que possui objetivos (*goals*), que é capaz de perceber certas propriedades do ambiente onde se encontra (*sensing*), podendo executar ações específicas neste ambiente (*acting*), sendo que algumas destas ações e/ou percepções podem/devem ser feitas através da cooperação (*communication*) com outros agentes, como mostrado na Figura 2.

Em jogos, é essencial que um NPC (agente) possa capturar informações do meio onde se encontra. Essa idéia é a mesma de agentes mecânicos (robôs), que necessitam interagir de diversas formas com o meio para poderem se locomover e tomar decisões. A coleta de dados do ambiente, entretanto, é sensivelmente diferente nos dois sistemas.

Em robôs reais, são usados dispositivos físicos analógicos como câmeras de vídeo, microfones, sensores de toque e calor, entre outros. A informação adquirida deve ser então decodificada antes de ser analisada. A interpretação dos dados das câmeras, por exemplo, é realizada por processos de visão computacional.

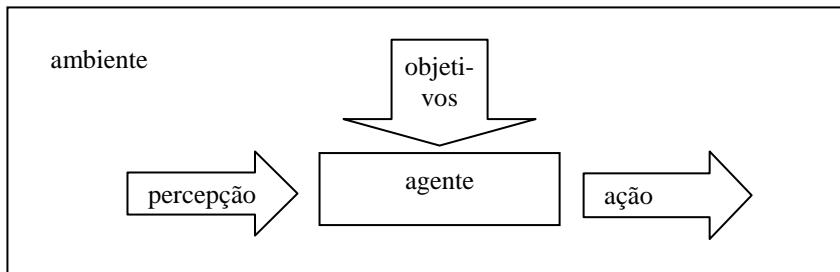


Figura 2: Estrutura de agentes em jogos

Quando o agente está imerso em um ambiente virtual (computacional), o processo de aquisição de dados é mais trivial. Uma vez que todo o universo de representação já está modelado na memória do computador, o acesso a tal informação pelo NPC pode ocorrer de forma direta. A informação que o NPC necessita para interagir e responder a estímulos do ambiente pode ser fornecida no momento de sua criação, ou à medida em que o agente interage com o ambiente.

A determinação de como e quanta informação será fornecida ao NPC vai determinar o grau derealismo da simulação. Além disso, um agente necessita de sensores inteligentes para focalizar o que precisa ver e ignorar o resto [MOF 95]. Outro fator a ser considerado são as propriedades do ambiente [RUS 95], apresentadas na Seção 2.2.3.

Em simulações computacionais, pode-se restringir o acesso à informação por meio de técnicas como a visão e audição:

- **Visão:** Faz uso de conceitos de óptica para determinar a visibilidade do NPC em função de sua direção e posição. Pelo uso de uma técnica de extração de seções (*frustum culling*) [MOL 02], pode-se determinar quais objetos estão localizados dentro do campo de visão. Deve-se observar que, em situações reais, nem tudo que está dentro do campo de visão pode realmente ser percebido. Uma vez visualizado o objeto, o agente pode receber toda ou alguma informação necessária sobre ele, que poderá ser armazenada na memória para futura utilização. A visão também é usada para fazer testes de colisão entre NPCs e obstáculos. Existem situações onde o processo de coleta de dados por meio de visão é desconsiderado. Em jogos de estratégia em terceira pessoa, como no Age of Empires, é muito comum o uso de algoritmos de busca (*pathfinding*), como o A*, para determinação da melhor rota a ser seguida pelo NPC entre dois pontos. Este processo faz uso de toda a informação disponível no ambiente para determinar a rota mais eficiente, desconsiderando a base de conhecimento do NPC;
- **Mensagens:** Uma das características fundamentais do agente é a comunicação com outros agentes pela troca de mensagens. As mensagens podem ser usadas para indicar a presença de inimigos, pedido por socorro, etc. A forma como as mensagens são trocadas caracteriza o grau de realismo do processo.

A consideração da distância e visibilidade entre agentes pode ser um bom fator atenuador e limitador de troca de mensagens. Em função destas restrições, um agente somente pode receber (escutar) uma mensagem de outro agente se estiver próximo (dentro de um limiar (*threshold*) pré-definido) e se não existirem objetos que “atenuem” a transmissão do “som” entre eles. De forma semelhante à visão, nem tudo que for ouvido pode ser percebido.

Em função dos dados capturados do ambiente, o personagem pode aprender e memorizar dados, utilizando técnicas de aprendizado mecânico (*Machine Learning*). Elas podem variar desde simples algoritmos que avaliam e armazenam as estratégias dos jogadores e seu efeito, para poder posteriormente determinar ataques mais eficientes, até técnicas mais formais e científicas, como o uso de Redes Neuronais e Algoritmos Genéticos [BUC 02]. Redes Neuronais procuram imitar, em vários aspectos, o funcionamento do cérebro humano, pela definição de células computacionais simples, chamadas neurônios, conectadas por ligações sinápticas, que possuem um peso sináptico associado; tais células são usadas para armazenar o conhecimento adquirido [HAY 01]. Algoritmos Genéticos são modelos computacionais baseados na evolução dos seres vivos. Nesses modelos, o conhecimento é representado por atributos booleanos.

Detalhes específicos da implementação de agentes em jogos com características de aprendizagem podem ser obtidos em [EVA 01, EVA 02], que descrevem técnicas usadas na implementação do jogo Black & White, considerado um clássico no aspecto do uso de agentes cognitivos. Nele, o jogador deve controlar e treinar uma população de criaturas. Este jogo incorpora uma variedade de diferentes esquemas de representação para criar os agentes, sendo algumas simbólicas e outras conexionistas, dada a inexistência de um único método representacional que possa ser usado para construir um agente completo [EVA 02]. Neste jogo, a geração dos planos é baseada em uma arquitetura BDI, juntamente com uma estrutura adicional, a Opinião, que é associada a cada desejo para expressar que tipos de objetos são mais adequados para satisfazer este desejo. Nesta arquitetura, crenças sobre objetos individuais são representadas simbolicamente, como uma lista de pares atributo-valor; opiniões sobre tipos de objetos são representadas como uma árvore de decisão; desejos são representados por meio de perceptrons; e intenções como planos.

Na implementação de agentes reativos, Máquinas de Estados Finitos (FSM) constituem o padrão de software mais utilizado [RAB 02], visto que esta categoria de agentes tem como principal característica a rápida resposta a estímulos externos. Uma FSM é composta por um conjunto finito de estados, um conjunto de entradas, um conjunto de saídas e uma função de transição. A função de transição, a partir das entradas e do estado atual, determina um novo estado e um conjunto de saídas [FUN 99]. As FSMs são geralmente representadas por diagramas de transição de estados.

Por fazerem uso de estruturas simples, como *if-else* e *switch* aninhadas, garantem implementações simplificadas, de fácil compreensão e depuração, e que são gerais para qualquer problema [RAB 02]. À medida que o controle de um NPC fica mais complexo, esta implementação torna-se inviável, tanto para depuração, como para modificação. Uma possível solução é o uso de estruturas hierárquicas, de forma que cada nodo da hierarquia possa expandir-se em uma nova hierarquia, e assim por diante, até o último nível, expandido em uma FSM [CUN 01]. Esta técnica é chamada *Hierarchical FSM* (HFSM). Mesmo com uma estrutura hierárquica, o NPC continua limitado às regras pré-estabelecidas.

5. Referências Bibliográficas

- [AMA 97] Amandi, A. A. Programação de agentes orientada a objetos. Porto Alegre: CPGCC da UFRGS, Tese de Doutorado, 1997.
- [BON 88] Bond, A. H., Gasser, L. Readings in distributed artificial intelligence. San Mateo: Morgan Kaufmann, 1988.
- [BUC 02] Buckland, M. AI techniques for game programming. Premier Press, 480p., 2002.
- [COR 94] CORREA, M. A. Filho. Arquitetura de diálogos entre agentes cognitivos distribuídos. Rio de Janeiro: COPPE da UFRJ, Tese de Doutorado, 1994.
- [CUN 01] Cunha, L. S., Giraffa, L. M. M. Um estudo sobre o uso de agentes em jogos computadorizados interativos. Technical Report Series, number 017, PUCRS, October 2001.
- [D'AM 95] D'Amico, C. et ali. Inteligência artificial: uma abordagem de agentes. Porto Alegre: CPGCC da UFRGS, 1995.
- [EVA 01] Evans, R. The future of ai in games: a personal view. Game Developers, August 2001.
- [EVA 02] Evans, R. Varieties of learning. In: AI Game Programming Wisdom. Ed. Charles River Media, 2002.
- [FUN 99] Funge, J. D. AI for computer games and animation – a cognitive modeling approach. A. K. Peters Ltd, 1999.

- [HAY 01] Hayking, S. Redes neurais: princípios e prática. Bookman, 2001.
- [JEN 96] Jennings, N. R. Coordination techniques for distributed artificial intelligence. In: O'HARE, G.M.P.; Jennings, N.R. (Eds.). Foundations of distributed artificial intelligence. New York: John Wiley & Sons. pp.187-210, 1996.
- [JEN 99] Jennings, N. R. Agent-oriented software engineering. In: Proceedings of the 9th European Workshop on Modeling Autonomous Agent In a Multi-agent System Engineering (MAAMAW-99), Heidelberg, Germany, Springer-Verlag, pp.1-7, 1999.
- [JUC 01] Juchem, M., Bastos, R. M. Engenharia de sistemas multiagentes: uma investigação sobre o estado da arte. Technical Report Series, number 014, April 2001, PUCRS.
- [MOF 95] Moffat, D., Frijda, N. H. Where there's a will there's an agent. Lecture Notes in Computer Science, Vol. 890, pp.245-??, 1995.
- [MOU 96] Moulin, B., Chaib-draa, B. An overview of distributed artificial intelligence. In: O'Hare, G., Jennings, N. R. (Eds.). Foundations of distributed artificial intelligence, John Wiley and Sons, 1996.
- [NAR 03] Nareyek, A. AI center homepage. (April 2003) Disponível em:
<http://www.ai-center.com/home/alex/publications.html>
- [OLI 96] Oliveira, F. M. Inteligência artificial distribuída. In: IV Escola Regional De Informática, Canoas, 3. Anais: Sociedade Brasileira de Computação, 239 p., 1996.
- [O'HA 96] O'Hare, G., Jennings, N. R. Foundations of distributed artificial intelligence. John Wiley and Sons, 592p.,1996.
- [RAB 02] Rabin, S. Implementing a state machine language. In: AI Game Programming Wisdom, Charles River Media, 2002.
- [RAO 95] Rao, A. S., Georgeff, M. P. BDI agents: from theory to practice. In: First International Conference On Multiagent Systems, pp. 312_319, 1995.
- [RUS 95] Russel, S. J., Norvig, P. Artificial intelligence: a modern approach. Prentice-Hall, New Jersey, 1995.
- [SHO 93] Shoham, Y. Agent-oriented programming. Artificial Intelligence, Amsterdam, Vol. 60, pp. 51-92, 1993.
- [SHE 98] Shehory, O. Architectural properties of multiagent systems. Pittsburgh, PA: [s.n.], 1998.
- [WOO 99] Wooldridge, M. Intelligent agents. In: WEISS, G. (Ed.) Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence, MIT Press, 1999.
- [WOO 03] Woodcock, S. Game ai: the state of the industry. (March 2003). Disponível em:
http://www.gamasutra.com/features/20001101/woodcock_01.htm
- [ZAM 00] Zambonelli, F., Jennings, N. R., Wooldridge, M. Organizational abstractions for the analysis and design of multi-agent systems. In: Proceedings of the 1st International Workshop on Agent-Oriented Software Engineering, 2000, Limerick, Ireland, pp.127-141, 2000.
- [ZAM 01] Zamberlam, A. O., Giraffa, L. M. M. Modelagem de agentes utilizando a arquitetura BDI. Technical Report Series, number 008, PUCRS, April 2001.
- [FRA 96] Franklin, S., Graesser, A. Is it an Agent, or just a Program?:A Taxonomy for Autonomous Agents. Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.
- [FON 93] Leonard N. Foner. *What's An Agent, Anyway? A Sociological Case Study*. Agents Memo 93-01. Available from the Agents Group, MIT Media Lab. Copyright (C) May 1993.
[\(http://foner.www.media.mit.edu/people/foner/Reports/Julia/Agents-Julia.pdf\)](http://foner.www.media.mit.edu/people/foner/Reports/Julia/Agents-Julia.pdf)