

Agentes e Emoções em Histórias Interativas

Cesar Tadeu Pozzer
Antônio L. Furtado
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente 225
Rio de Janeiro, RJ, 22453-900, Brasil
Email: {pozzer, furtado}@inf.puc-rio.br

Angelo Ernani Maia Ciarlini
Departamento de Informática Aplicada
UNIRIO - Universidade do Rio de Janeiro
e-mail: angelo.ciarlini@uniriotec.br

PUC-RioInf.MCC39/03 Outubro, 2003

Abstract: This work presents a general survey on agent and multi-agent technologies, with special attention to autonomous characters, frequently used in intelligent electronic games. All characters should encapsulate resources that allow them to represent graphically typical events of stories dynamically created by means of a simulation process. The agents can interact with each other and with the scene to which they belong. They should also reflect users' interactions. The resources are implemented over an architecture that integrates modules for generation, representation and exhibition of interactive stories.

Keywords: Interactive TV, Computer Graphics, Artificial Intelligence, Agents.

Resumo: Neste trabalho, apresenta-se uma revisão geral da tecnologia de agentes e sistemas multiagentes, sob o foco da construção de elementos autônomos, semelhantes a personagens inteligentes de jogos eletrônicos. Todos os personagens deverão ser providos de recursos que os tornem capazes de fazer a representação gráfica de eventos típicos em histórias criadas dinamicamente, por um processo de simulação. Os agentes podem interagir entre si e com o cenário onde estiverem inseridos, bem como sofrer influência de ações do usuário. Todos estes processos são fundamentados sobre uma arquitetura que integra módulos de geração, representação e exibição de histórias interativas.

Palavras-chave: Geração de Enredos, Inteligência Artificial, Agentes.

1 Introdução

A geração de personagens virtuais com comportamentos inteligentes é um dos desafios mais almeçados e mais difíceis de serem alcançados em simulações computacionais. Com o surgimento da área de Inteligência Artificial (IA), pesquisadores acreditavam que em um futuro próximo, máquinas poderiam simular e se igualar a seres humanos na realização das mais diversas tarefas. Com o passar do tempo, constatou-se que tais ambições não eram tão triviais como esperado.

Passando por vários períodos de altos e baixos, a IA atualmente encontrou um novo campo de aplicação muito promissor, que está diretamente ligado com entretenimento digital. Jogos de computador, que a cada dia estão mais avançados e que diariamente causam espanto e surpresas pelas inovações tecnológicas, estão fazendo intenso uso de técnicas de IA para dar a seus personagens comportamentos que procuram imitar seres humanos nos mais diversos aspectos, além de recursos gráficos e técnicas de animação que já não permitem diferenciar o real do computacional (virtual). Diversas técnicas de IA são usadas em jogos para dar aos personagens características de aprendizado, tomada de decisões, inferência de objetivos, comportamentos autônomos e em grupo, entre outros.

Qualquer jogo que permita que o usuário jogue contra a máquina deve ter algum nível de IA, que pode ser muito primitivo, como no caso dos primeiros jogos criados na década de 80, como o “Pac-man”, até o uso de técnicas sofisticadas, como o “The Sims”, um jogo de simulação social, onde personagens virtuais interagem entre si e convivem em um ambiente. Cada personagem possui emoções e necessidades básicas como fome, interação social, entre outras, que devem ser supridas para manter a harmonia do grupo.

Mesmo com comportamentos que parecem ser reais, o grau de inteligência de personagens em jogos atuais, em sua grande maioria, é limitado a um conjunto de respostas associadas a ações predefinidas em scripts. Os personagens não podem reagir de forma apropriada a situações que não estão definidas nos scripts ou que não foram modeladas e incorporadas nos personagens [NAR 03].

Das possíveis aplicações de IA para jogos, algumas já estão bem desenvolvidas, como algoritmos de busca, onde se destaca o A* [MAT 02], que é um algoritmo que usa uma função heurística para determinar uma estimativa de custo da melhor rota entre dois pontos em um mapa. Porém, existem outros aspectos que não possuem soluções definidas, como no caso de aprendizagem e comportamentos, bem como tomada de decisões.

Em jogos eletrônicos, geralmente a modelagem de comportamento e a tomada de decisões são implementadas por meio de agentes. Os personagens devem tomar decisões adequadas para atingir seus objetivos, adaptando seu comportamento às características de novos ambientes [CUN 01]. Isso pode ser realizado pela percepção do ambiente que os cerca, por meio do tato, audição, visão, e pela realização de ações, como andar, lutar, etc.

A definição dos objetivos dos personagens pode ser muito variada. Em jogos de guerra, por exemplo, podem-se evidenciar dois objetivos claros: permanecer vivo e eliminar os inimigos. Além disso, pode-se também adicionar os objetivos de procurar pelo inimigo ou fugir dele. Estes objetivos permanecem inalterados durante todo o jogo e são definidos pelo programador durante a fase de desenvolvimento. No caso do “The Sims”, os objetivos dos personagens são definidos pelo jogador, em tempo real.

No presente trabalho é apresentada uma abordagem de agentes na definição de personagens autônomos para construção de uma ferramenta integrada de geração e exibição de histórias interativas [FUR 96], cujo principal propósito é ser usada em TV interativa [DRI 00]. Esta ferramenta seria composta por três grandes módulos, como mostrado na Figura 1. Cada módulo possui uma relação muito forte com os demais, e com isso, surgem pontos de interseção.

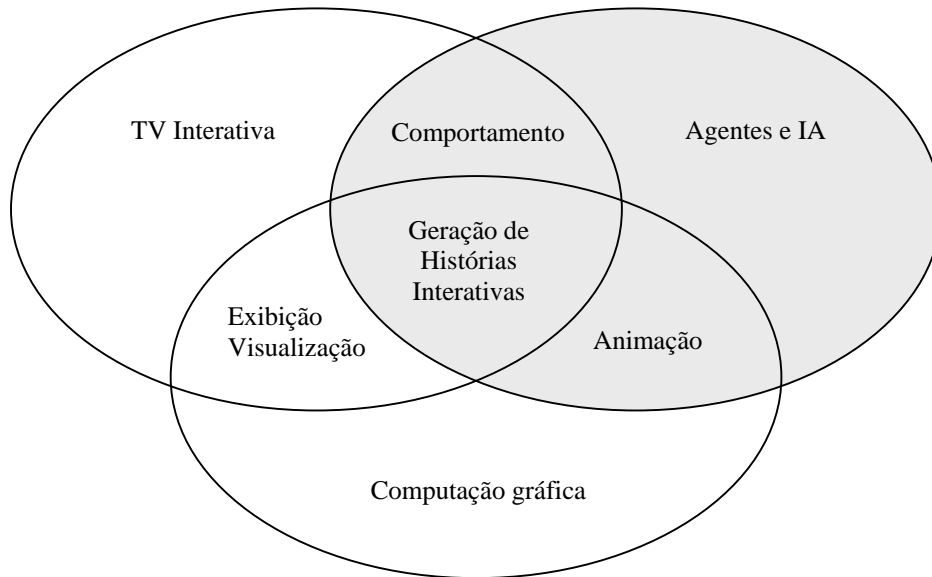


Figura 1: Módulos que compõem o sistema de geração e visualização de histórias interativas. A região sombreada representa o foco deste trabalho.

O elemento central da arquitetura consiste na geração da história, que é realizada por um processo de simulação, fazendo uso da linguagem Prolog estendida com Constraint Logic Programming. Este processo gera um conjunto de operações (que são definidas por pré e pós-condições), necessárias para a realização da história. Estas operações descrevem as ações que devem ser realizadas pelos personagens, segundo uma ordem cronológica predefinida. Uma descrição mais detalhada do processo de geração de histórias pode ser vista em [POZ 03a]

Para tornar este conjunto de operações plausível de ser representado como um conteúdo interativo em iTV, definem-se outros dois módulos, que são responsáveis por traduzir seqüências de operações executadas em informação gráfica animada, suficientemente realista para o propósito em questão. Estes outros dois módulos são o de Inteligência Artificial (IA) e o de Computação Gráfica, que são as bases para dar vida aos personagens da história. A IA se faz necessária para assegurar que cada personagem, aqui representado como um agente de software, seja capaz de expressar um comportamento inteligente e realista, por meio de animações gráficas condizentes com as ações sendo realizadas frente ao telespectador (usuário).

O resultado final de todo o processo é a exibição gráfica da representação da história sendo simulada. O módulo de Computação Gráfica é o responsável por este

processo. Para isso, deve fazer uso de algoritmos de renderização, iluminação e técnicas de modelagem de personagens, bem como do cenário [POZ 03b].

Este trabalho está organizado em 7 seções. Na Seção 2 são apresentados conceitos gerais referentes à arquitetura de agentes, tipos e classificações. Também é feita uma rápida discussão de sistemas multiagentes. A aplicação de agentes e técnicas gerais de IA em jogos é apresentada e discutida na Seção 3. Na Seção 4 é apresentado o processo de geração automático de histórias interativas. Na Seção 5 são discutidas e apresentadas técnicas para implementação de agentes que sejam capazes de realizar os eventos das histórias geradas. Uma abordagem de agentes, emoções e comportamentos é apresentada na Seção 6. Na Seção 7 são apresentadas as conclusões finais.

2 Agentes

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.

2.1 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.

2.2 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-existent) 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 (papéis), 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:

2.2.1 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 contribuir 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.

Um 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.

2.2.2 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;
- **Nívelada:** 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;
- **Sistemas Multiagentes compostos por Sistemas Multiagentes:** Considera um SMA como sendo um agrupamento de vários módulos que, por sua vez, são vistos isoladamente como SMAs.

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.

2.2.3 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;
- **Determinístico x Não-determinístico:** É determinístico se o estado do ambiente é determinado somente por seu estado atual e as atuações dos agentes;
- **Episódico x Não-Episódico:** Em um ambiente episódico a experiência do agente é dividida em episódios. Cada um deles consiste em percepções e ações dos agentes, e a qualidade de cada ação depende somente do episódio em si;
- **Estático x Dinâmico:** Um ambiente é dinâmico se pode mudar enquanto o agente está deliberando; caso contrário é estático. Um ambiente é dito *semi-estático* quando não muda com a passagem do tempo, apenas com as ações desempenhadas por agentes;

- **Discreto x Contínuo:** É chamado discreto o ambiente que tem um número limitado de percepções e ações distintas e claramente definidas.

3 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 na seção 2 deste trabalho). 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.

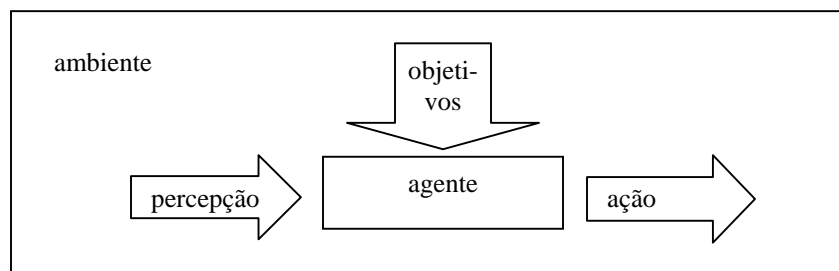


Figura 2: Estrutura de agentes em jogos

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.

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 de realismo 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;
- **Audição:** 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 Neurais e Algoritmos Genéticos [BUC 02]. Redes Neurais 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 nó 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.

4 Agentes e Geração de Histórias Interativas

Uma característica comum de agentes em jogos é a definição estática de objetivos durante a fase de desenvolvimento, em função de suas características. Isso tende a fazer com que o comportamento dos agentes resulte em situações previsíveis.

Neste trabalho faz-se uso de uma abordagem mais geral da teoria de agentes. Em vez de delegar ações específicas e estáticas a cada personagem, pretende-se definir agentes cujas operações (eventos) sejam dinâmicos, de modo que seja criado um SMA capaz de suportar a exibição de histórias genéricas, criadas dinamicamente, dentro de padrões pré-estabelecidos.

A geração automática de histórias teve início com os estudos do pesquisador russo Vladimir Propp [PRO 68], que observou que, em textos literários de gêneros específicos (por exemplo: contos de fadas), é muito comum a ocorrência de eventos típicos e de certos padrões de encadeamento entre os eventos.

Propp sugeriu a caracterização dos textos de um determinado gênero pela associação de *funções* a pequenos trechos das narrativas. A ocorrência das funções ao longo de uma narrativa obedeceria sempre a determinadas seqüências, que poderiam envolver todas ou apenas parte das funções típicas do gênero. As funções proppianas seriam, então, os eventos típicos básicos da narrativa, descrevendo-se os enredos como seqüências de ocorrências das funções. O encadeamento entre os eventos é um processo que tem de respeitar uma lógica, segundo a qual um evento só ocorre na presença de determinadas condições. Cada evento, por sua vez, pode também criar as condições necessárias para a execução de outros eventos [CIA 99].

A criação de um enredo de um certo gênero é realizada por um processo de *simulação*, que tem como ponto de partida a descrição da situação inicial dos personagens, o modelo de comportamento atribuído a eles (especificado em termos de objetivos a serem perseguidos em situações previstas), e as alternativas que cada personagem tem para atingir seus objetivos, que são especificadas em termos das operações (sujeitas a padrões de ordenação) que caracterizam o gênero.

Para a geração semi-automática de narrativas de um certo gênero, [FUR 99] faz uso de um kernel chamado IPG (*Interactive Plot Generator*). Este kernel gera uma narrativa baseada em um contexto que define os seguintes itens, sendo que os dois últimos caracterizam o gênero da narrativa:

- Uma configuração inicial do personagem e sua situação inicial, usadas como ponto de partida para criação do enredo;
- Uma biblioteca de operações, estabelecendo os eventos e enredos típicos do gênero, através da definição das operações e restrições de integridade que podem ser executadas;
- Um conjunto de regras lógicas, para inferir objetivos a serem almejados (perseguidos) pelos personagens, na medida da ocorrência de determinadas situações.

A Figura 3 mostra o esquema de geração de enredos proposto por [CIA 02].

O processo de geração dos enredos faz parte de um ciclo com múltiplos estágios, onde alternam-se fases de inferência de objetivos e de planejamento, e no qual, após cada

fase de planejamento, o usuário pode intervir. Esta solução é implementada com o uso de técnicas de Inteligência Artificial e Bancos de Dados, por meio da linguagem Prolog.

Como o algoritmo de geração de planos do IPG é não linear, ele gera planos compostos por um conjunto de operações apenas parcialmente ordenadas. O exemplo apresentado na Figura 4, baseado nas funções de Propp, ilustra esta ordenação. Para este exemplo, a operação 2 somente pode ser executada após a 1. A ordenação total é definida pelo usuário, e, como indicado (em mais detalhe) na Figura 5, pode admitir diferentes seqüências.

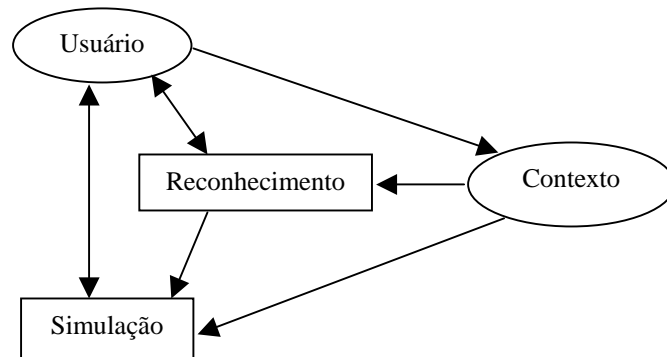


Figura 3: Esquema de geração de enredos

0: Init()
1: Absence_of_younger_people (princess)
2: Kidnapping_of_a_person (princess, dragon)
3: Call_for_help (hero, tsar)
4: Departure_of_seeker_hero (hero)
5: Fight_in_an_open_field (hero, dragon)
6: Out_of_earth_receipt (hero)
7: Victory_in_open_battle (hero, dragon)
8: Return (hero)
9: Reward (hero)

1 < [2]
5 < [7, 8]
7 < [9]
4 < [5]
6 < [7]
8 < [9]

Figura 4: Ordem parcial das operações

O IPG é baseado no paradigma de geração e reconhecimento de planos e permite realizar simulações sobre uma base de dados especificada. Além de criação de aplicativos de entretenimento, nas quais histórias são dinamicamente criadas pela modelagem do

comportamento de personagens, explorando suas possíveis interações, tais simulações podem ser usadas em pelo menos outras duas formas [FUR 02]:

- Para auxiliar no processo de especificação e validação de um sistema em desenvolvimento, de forma que desenvolvedores possam ajustar o comportamento dos agentes;
- Como suporte a tomada de decisão em tempo real, de forma que pessoas ou organizações possam tomar decisões baseadas em comportamentos previstos de outras partes do sistema.

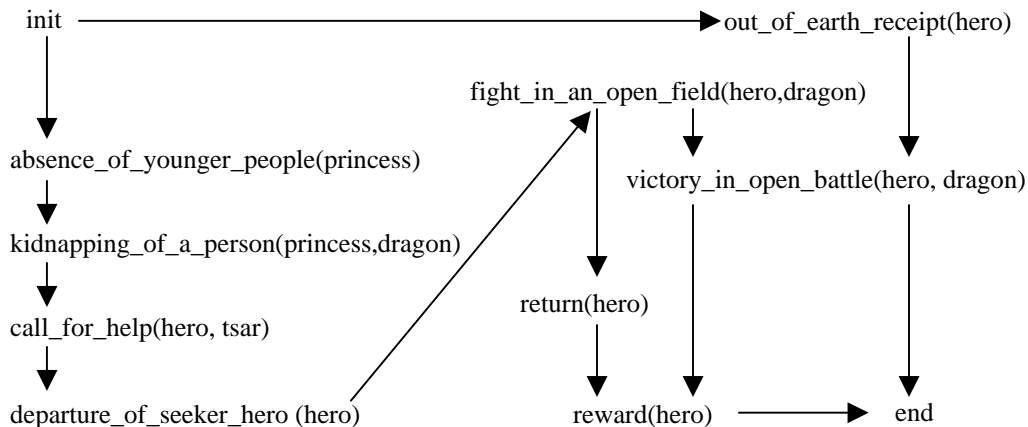


Figura 5: Exemplo de um plano. Setas indicam possíveis seqüências

O planejamento é necessário para obter conjuntos de eventos combinados que são necessários para atingir os objetivos dos agentes. A execução de um plano pode trazer à tona novos objetivos, e assim por diante. Estas interações prosseguem até que não existam mais objetivos a serem inferidos ou até quando o usuário decide parar o processo de simulação.

A interação entre os agentes (personagens), em situações que os levam a inferir objetivos e adotar planos para atingi-los, com mútuas interferências positivas ou negativas [WIL 83], apresenta diversas características que se assemelham com Sistemas Multiagentes [CIA 99]. Por outro lado, nos Sistemas Multiagentes não se configura de forma tão flagrante a questão de conflito ou competição entre objetivos, que é tratada neste planejador, podendo conduzir à eventual desistência da busca de objetivos dos personagens. Outro aspecto divergente refere-se ao nível de autonomia dos agentes. Existe um único planejador que define as operações de todos os agentes na busca de seus objetivos, o que é incompatível com a propriedade de autonomia dos agentes.

5 Modelo de Execução de Histórias

O resultado do processo de simulação é um conjunto de operações (Figura 4), associadas a personagens, com pré e pós-condições, bem como possíveis valores de atributos de cada personagem.

Para permitir a representação gráfica destas operações, em um cenário virtual 3D, cada personagem é representado por um agente. Aos agentes são fornecidos recursos de IA para que possam realizar ações, de forma que o fluxo da história seja corretamente representado.

O resultado do processo de execução da história é um conjunto de informações geométricas que representam a estrutura pontual da cena em um dado momento. Estes dados devem ser enviados a um dispositivo (*engine*) gráfico, que faz a renderização da cena com uso de técnicas de Computação Gráfica.

Para que ocorra a integração dos módulos de geração, execução e visualização das histórias, propõe-se uma estrutura modular, cuja representação simplificada é apresentada na Figura 6.

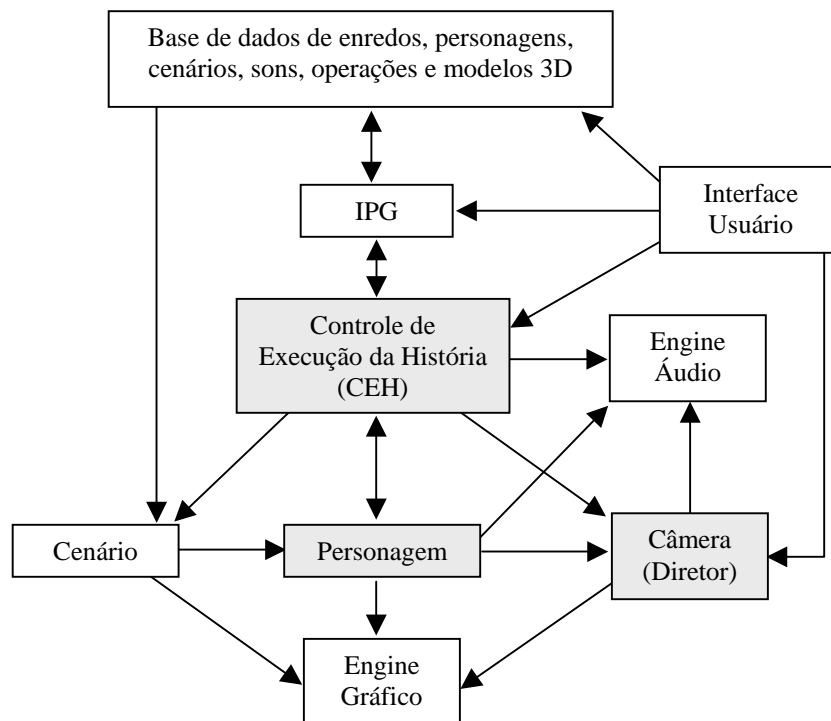


Figura 6: Estrutura de módulos de controle e execução de operações.

Neste trabalho, são focados exclusivamente aspectos referentes a técnicas de IA e de agentes (módulos sombreados Figura 6). Não são focados aspectos gráficos, exceto quando técnicas de IA fizerem uso explícito de tais recursos.

5.1 Controle de Execução da História

O módulo CEH é o responsável pela atribuição das operações que definem a história aos respectivos agentes, bem como pelo acompanhamento do desempenho deles no cumprimento destas operações. À medida que as operações são concluídas, é realizada

também a atualização do escalonamento temporal (*timeline*) da história, que situa cada operação dentro de uma ordem previamente definida pelo IPG e pelo usuário.

O gerenciamento do *timeline* é um processo muito complexo, pois ao mesmo tempo sofre influência e controla o andamento da realização das operações. O *timeline* pode ser alterado pelo usuário, para acelerar ou retardar o tempo médio da execução das operações.

Existe uma contínua comunicação bidirecional entre o módulo de controle com cada personagem, que informa, a cada passo, sobre o estado do cumprimento da última operação fornecida. Quando uma operação for finalizada, uma nova é delegada. Uma operação é finalizada quando o agente obteve sucesso ou não no seu cumprimento. A escolha da operação a ser delegada depende do agente e do andamento do fluxo da história (*timeline*). Caso o personagem não faça parte do fluxo principal, como é o caso de um ator coadjuvante, ações genéricas como descansar, trabalhar, dormir e andar podem ser delegadas.

O módulo CEH também tem influência no posicionamento da câmera, que é a responsável pela seleção da cena a ser exibida, visto que, a cada momento, diversas interações, entre diversos personagens, podem estar ocorrendo. Em função do *timeline*, pode-se determinar quais agentes estão associados com as cenas principais. Estas informações são então passadas para a câmera que, juntamente com informações fornecidas por cada agente individualmente sobre ações sendo realizadas, escolhe a cena de maior importância. O *timeline*, de forma similar à câmera, também pode ser integrado com o módulo de áudio para associar sons com as operações sendo realizadas pelos agentes.

Em termos gerais, este módulo pode ser visto como o módulo de controle centralizado do SMA. Neste grupo, a comunicação entre os agentes é nivelada, ou seja, ocorre de forma direta, sem a necessidade de mediadores. Outra característica dos agentes é o nível de heterogeneidade, que é baixo. Este Sistema Multiagente possui uma abertura estática para entrada e saída de agentes, que pode ocorrer quando um determinado agente for derrotado por outro após uma operação de luta, por exemplo.

5.2 Estrutura de um Personagem

Cada personagem, aqui tratado como um agente reativo, implementado por meio de uma máquina de estados finitos (FSM), é responsável pelo cumprimento das operações fornecidas pelo módulo CEH, em função da história sendo exibida. Para este trabalho, não existe uma evidente necessidade do uso de agentes cognitivos, visto que o fluxo da história é definido no processo de simulação, pelo IPG, e os agentes se limitam à realização de ações que foram previamente determinadas. A estrutura de um personagem é apresentada na Figura 7.

Tanto as operações, bem como a informação do cenário e de outros agentes são armazenadas em uma memória local, que pode conter caminhos já percorridos e os que ainda não foram concluídos, posição de objetos da cena, de personagens, operações já realizadas, bem como atributos internos, como drives e emoções, que são descritos na Seção 6.

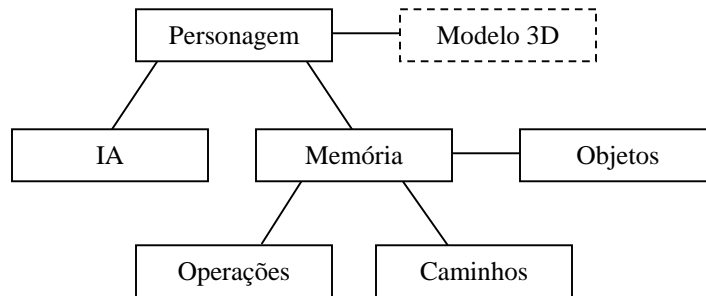


Figura 7: Modelo interno simplificado de um personagem

Existe um conjunto fixo de operações que um agente pode realizar (Tabela 1). Como cada operação descreve um evento muito genérico, elas são associadas a um conjunto de micro-operações (Tabela 2), que podem ser mais facilmente interpretadas e realizadas pelos agentes.

Tabela 1: Conjunto de operações (adaptadas de Propp)

Operações	Descrição	Micro-operações
Rapto	Rapto de um personagem por outro	Deslocar, observar, pegar
Libertação	Libertação de um personagem	Deslocar, soltar
Recompensa	Premiação por uma ação realizada	Dar, pegar
Partida	Ir a um determinado local	Deslocar
Ausência de pessoa mais jovem	Estado de vítima indefesa	Deslocar, esperar
Luta em campo aberto	Luta entre personagens	Deslocar, lutar
Recebimento emergindo do chão	Recebimento de poderes mágicos	Dar, pegar

Tabela 2: Conjunto de micro-operações

Micro-operação	Descrição	Especializações
Deslocar	Fazer o deslocamento de um personagem de um local para outro	Voar Caminhar Rastejar Nadar Teletransportar
Lutar	Duelo entre dois ou mais personagens	Espada Arma de fogo Fogo (dragão) Magia
Esperar	Representação de ação inexistente ou momento de descanso	Permanecer parado Dormir Descansar
Pegar/soltar	Capturar ou adquirir pessoa ou coisa	Personagens

		Objetos Poderes
Pedir/dar	Requisição de algo por meio de diálogo	Informação (fala/escuta) Poderes Objetos Ajuda
Observar	Captação dos dados do ambiente próximo ao personagem	Procurar Ver Ouvir
Trabalhar	Ato de trabalhar	Doméstico Agrícola

O agente deve ser autônomo, comunicar-se e interagir freqüentemente com outros agentes, com o módulo CEH, com a câmera, e com o cenário, como mostrado na Figura 8. Os dados enviados para a câmera dizem respeito às ações sendo realizadas e ao grau de importância associado, para que a câmera possa escolher, entre um elenco de operações sendo realizadas em paralelo, a de maior importância.

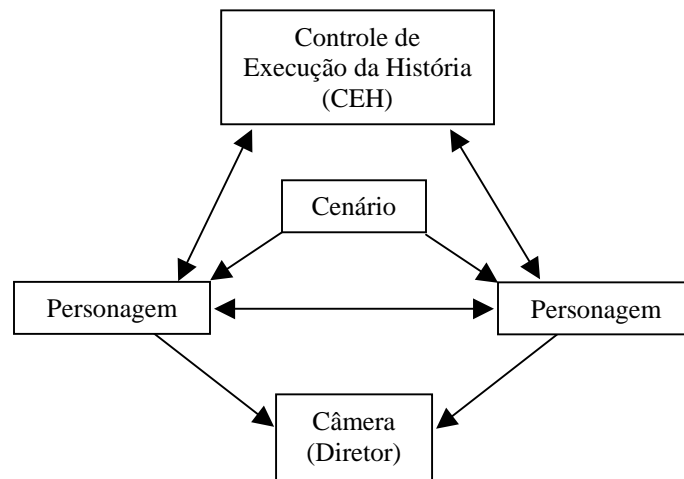


Figura 8: Estrutura de controle do fluxo da história

A captura de dados do cenário, por meio de visão se faz necessária para permitir que o agente se locomova de maneira correta e possa atingir seus objetivos, tais como perseguir um outro agente (cenas de rapto) ou ir até um determinado local.

Na comunicação entre os personagens, pode ocorrer a troca de diversas mensagens, como mostrado na Tabela 3. As mensagens estão associadas a ações, e são finalizadas com seu término. Para que esta comunicação ocorra, os personagens devem estar próximos.

Tabela 3: Descrição das mensagens trocadas entre agentes

Operação	Descrição da mensagem	Finalização
Luta	Redução do nível de vida	Morte de um indivíduo ou fuga
Pedir/dar	Troca de informação	Obtenção da informação desejada / tempo de comunicação máximo
Pegar/soltar	Indicar relação de subordinação (estar preso/solto)	Ato de pegar/soltar

De forma geral, cada agente possui um ciclo de execução de ações que é realizado a cada iteração. Em ordem cronológica, são realizados os seguintes processos:

- Verificação do término de ações
- Verificação do nível de vida
- Comunicação com o CEH
- Observação do cenário, por visão e audição
- Leitura de mensagens de outros personagens
- Verificação e determinação da ação corrente
- Determinação das micro-operações necessárias
- Realização das micro-operações
- Envio para câmera da ação e nível de importância
- Atualização da animação do modelo 3D associado
- Envio do modelo geométrico para o módulo de representação gráfica

5.3 Câmera Virtual

Um dos principais módulos de visualização é o da câmera, pois é responsável por determinar qual cena da história, a cada momento, deve ser exibida. A câmera deve desempenhar o papel de um diretor que, como em um filme, deve definir diversos parâmetros para ressaltar, da melhor forma possível, os principais fatos ocorrendo. Em cenas dinâmicas, a câmera pode também sofrer um deslocamento contínuo, para acompanhar os personagens.

A escolha da cena a ser exibida deve ser baseada no nível de importância das operações e micro-operações sendo realizadas pelos personagens. Por exemplo, cenas de luta terão maior peso que cenas de locomoção. O mesmo vale para diálogos. De forma geral, a câmera deve procurar cobrir o maior número de diferentes personagens, em tomadas curtas de exibição.

Além destes recursos automáticos, a câmera pode ser controlada pelo usuário de duas maneiras:

- **Navegação livre:** Neste modo de operação, o usuário pode “sobrevolar a cena” e fazer observações de quaisquer lugares sob qualquer ângulo;
- **Fixada a um personagem:** usada para acompanhar as tarefas de um determinado personagem, sob o foco de 1ª ou 3ª pessoa.

Uma vez determinada a cena a ser exibida, diversos parâmetros relacionados à Computação Gráfica, como posição, inclinação, movimentação, abertura focal, devem ser

determinados, de forma a impedir que a câmera fique localizada atrás de objetos e que possa exibir todos os elementos que fazem parte da cena.

Uma análise detalhada de processos da escolha de cenas 3D em tempo real, pelo uso de técnicas cinematográficas e de IA, pode ser encontrada em [HE 96, TOM 00]. Aspectos matemáticos para definição de parâmetros de câmera, dado a geometria da cena e posição do personagem desejado, podem ser vistos em [BLI 88, HAW 03].

6 Agentes e Emoções

A representação de agentes com comportamentos realistas é crucial em aplicações de entretenimento, como cinema, iTV [DRI 00], jogos de computador e realidade virtual [DAV 00]. Segundo [BRE 98], o comportamento, juntamente com expressões faciais, são o meio pelo qual o agente (ou robô) pode expressar sua motivação, que é determinada por *drives* e emoções. Drives são necessidades básicas como fome, sede, interação social, entre outros.

Os drives influenciam na seleção de comportamento e no estado emotivo [FUR 02]. Uma vez que a expressão do agente reflete seu estado emotivo, o grau de satisfação dos drives influenciam indiretamente nas expressões faciais [BRE 98] que, em aplicações de entretenimento, podem por sua vez ser usadas para ativar o aspecto emotivo do observador.

Além de expressões faciais, o agente também pode fazer uso de gestos físicos e expressões vocais como rir, chorar, bocejar (*non-language sounds*) para exteriorizar seu estado emocional [VEL 97]. As expressões faciais podem ser representadas, por exemplo, pela imagem das sobrancelhas, olhos, pálpebras, orelhas e boca.

Tanto drives como emoções podem ser facilmente representados internamente ao agente por meio de valores numéricos, dentro de um intervalo predefinido, que indicam sua intensidade no momento. Um drive ou emoção é considerado ativo quando sua intensidade for superior a um dado limiar (threshold).

Drives não podem se satisfazer sozinhos. Eles se tornam satisfeitos quando o agente é habilitado a ativar um comportamento adequado para supri-lo. Por exemplo, para suprir o drive de fome, o agente deve prover-se de recursos de alimentação, assim como deve dormir para suprir o drive de fadiga. Quando tal comportamento é ativado, a intensidade do drive é reduzida [BRE 98]. Os agentes podem usar recursos de aprendizagem para identificar habilidades, inclusive de comunicação com outros agentes, que contribuem para satisfazer cada um dos drives que os afetam.

Geralmente os drives têm um comportamento cíclico temporal, ou seja, sem estímulos um drive tende a aumentar em intensidade, a não ser que seja satisfeito. A cada drive corresponde um valor desejável, com limite inferior e superior aceitáveis, dentro dos quais é considerado permanecer em regime homeostático (drive satisfeito).

Os drives têm influência direta sobre as emoções. Emoções positivas, como a felicidade, ocorrem quando os drives são satisfeitos. Emoções negativas, como a tristeza, ocorrem quando um drive não é satisfeito, ou quando é super-satisfeito (excedendo o limite superior aceitável).

As emoções podem ser provenientes de estímulos externos (eventos do ambiente ou de outros agentes) e internos (intensidade dos drives) [VEL 97]. Segundo [IZA 93], a

ativação de emoções, assim como o aumento ou redução de sua intensidade, pode ocorrer por estímulos de sensores cognitivos e não cognitivos, que podem ser agrupados em quatro categorias: neuronais, sensoriais, motivacionais e cognitivos.

Segundo Costa [COS 96], em termos gerais, existem três circunstâncias para geração de emoções:

- Quando uma seqüência de comportamentos foi completada;
- Quando ocorre uma interrupção em uma determinada seqüência;
- Quando um personagem gera explicitamente uma emoção.

Segundo [VEL 97], as emoções também têm influência no comportamento de agentes autônomos. Esta constatação resulta do fato de as emoções não serem provenientes somente da ação de drives, mas também de estados internos do agente, bem como da interação com outros agentes.

A intensidade de uma emoção precisa ultrapassar um limiar para ser expressa externamente ao agente. Além do nível de ativação, diversas questões devem ser consideradas ao se trabalhar com emoções [COS 96]:

- Como as emoções são mantidas e como é sua alteração no tempo - Cada emoção deve possuir um limiar (*threshold*) de ativação e uma função de decaimento, que controla a duração da emoção;
- Como elas interferem em outras - Algumas emoções podem ativar ou inibir outras emoções (ex: o medo pode inibir a felicidade);
- Como elas surgem da interação com outros personagens (agentes);
- Como elas são influenciadas pelas características internas do agente.

Segundo modelos teóricos sobre emoções [EKM 92, IZA 91], existe um conjunto de emoções básicas ou primárias, que possuem como características determinados aspectos que as diferenciam das demais, e que estão relacionados com eventos fundamentais da vida cotidiana. Elas também apresentam características que podem ser facilmente mapeadas em traços distintos por meio de expressões faciais [DON 99]. São elas: raiva, tristeza, alegria, medo, repulsa e surpresa. (anger, distress/sadness, fear, enjoyment/ happiness, disgust, surprise).

Neste trabalho, assume-se que drives e emoções podem ser considerados, tanto na geração como na representação de histórias interativas, sob três aspectos: forma como são gerados, sua influência no comportamento dos personagens e sua representação gráfica, segundo a relação apresentada na Figura 9.

No processo de simulação, drives e emoções também podem estar associados aos personagens. Suas intensidades devem ser explicitamente definidas, da mesma forma como ocorre com os demais atributos, em função da operação sendo processada e de características inerentes de cada personagem.

A nível de representação, o agente está imerso em um ambiente real, e por isso propenso a interação com o meio e com outros agentes, o que pode influenciar fortemente na alteração de drives e emoções e, conseqüentemente, no comportamento e micro-operações a serem realizadas. Como nesta etapa são considerados aspectos do cenário, como distâncias e obstáculos, bem como personagens coadjuvantes, pode-se determinar a

intensidade de drives e emoções para cada micro-operação que os personagens devem realizar.

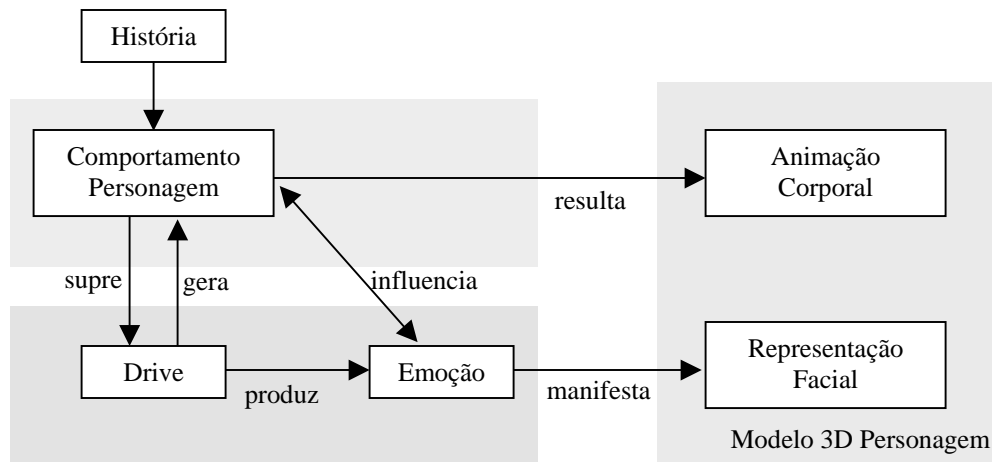


Figura 9: Estrutura interna do comportamento de um agente.

As emoções podem ter um papel fundamental na geração de novos objetivos, além de fazer parte das pré e pós-condições de certas operações, podendo assim influir no rumo da história sendo criada pela simulação. Por isso, deve-se ter cuidado para que no processo de visualização, inclusive em confronto com outros personagens, não ocorra, como consequência de se considerar emoções decorrentes desse processo, alguma alteração do fluxo principal da história, que já foi definido a nível de simulação, pelo IPG.

Por este motivo, na fase de exibição pretende-se inicialmente considerar emoções apenas para efeito da própria representação, ou seja, fazer uso de expressões faciais para descrever o estado emocional do agente associado ao comportamento (operação) sendo apresentado a cada momento. Desta forma, a geração das emoções tem lugar exclusivamente no processo de simulação da história.

Para o modelo de visualização, a representação gráfica das emoções, por meio de expressões faciais, tem-se revelado um recurso muito promissor e interessante, visto que vai dar ao usuário a sensação de vivenciar com mais intensidade as operações sendo realizadas pelos agentes (personagens), constituindo assim um meio adicional de comunicação com o telespectador.

Para o contexto de contos de fadas, as expressões faciais podem ser facilmente associadas com as operações e micro-operações disponibilizadas nos agentes. Nesta abordagem, as emoções são resultado da execução de uma tarefa concreta (*task-oriented emotions*) [GRA 01].

Representações gráficas de expressões faciais podem ser muito complexas dependendo do nível de realismo e técnicas empregadas. Como neste trabalho faz-se uso de modelos geométricos simplificados para representar a estrutura 3D do personagem, que é principalmente baseada em textura, essa mesma técnica foi adotada para a representação das emoções, visto que simulações geométricas de boca, olhos, e face do personagem exigem malhas e algoritmos de animação muito complexos e caros.

Desta forma, pretende-se, pelo menos em um momento inicial, mapear as expressões faciais sobre a textura, que é então aplicada sobre a superfície do personagem. Cada expressão facial deve, desta forma, ser desenhada a mão e armazenada em uma base de dados de texturas de expressões faciais, para cada personagem presente na história que deva exibir tal recurso. No caso de robôs, geralmente usam-se servomecanismos para controle de expressões faciais por meio de olhos, orelhas e boca [BRE 98].

Para ressaltar a expressão facial, o módulo de câmera, em comunicação com os agentes, deve fazer um enquadramento que possa evidenciar esta característica.

7 Conclusões

Neste trabalho apresentou-se uma abordagem geral sobre teorias de agentes e sistemas multiagentes, necessárias para dar suporte à exibição interativa de histórias criadas dinamicamente por um processo de simulação.

Estas teorias podem ser tanto aplicadas ao processo de geração da história, como, principalmente, à exibição da história gerada. O uso de agentes tem mostrado ser uma boa prática para atender à complexidade de um sistema, dividindo-o em unidades menores e autônomas. São essas unidades que, trabalhando cooperativamente, são responsáveis pela reprodução de cenas que transmitem os eventos presentes em uma história.

Neste sistema, que pode ser usado tanto para fins de entretenimento, como em aplicações comerciais, ou em atividades educacionais, os agentes exercem um papel fundamental na forma como a informação é transmitida ao usuário.

Até o momento, foi implementado apenas um protótipo inicial da arquitetura de agentes proposta neste trabalho, e que foi resultado de experimentos que procuraram mostrar as principais características necessárias para que a exibição de histórias interativas e dinâmicas pudesse ser realizada.

Nestes testes verificou-se a dificuldade inerente à representação de ações que, aparentemente, pareciam simples de serem realizadas por personagens autônomos, controlados por um núcleo central. A partir desta constatação, partiu-se para a proposição de arquiteturas mais complexas que inevitavelmente deveriam considerar a comunicação entre agentes, bem como uso de recursos de armazenamento de conhecimento. Outra constatação foi que os agentes devem ter um nível de autonomia muito elevado para que a representação das cenas possa ser ainda mais simplificada.

Atualmente, está-se estudando a possibilidade do tratamento de drives e emoções para a definição de comportamentos mais realistas. Pretende-se brevemente fazer testes reais e definir qual o nível de influência adequado que eles devem ter sobre as ações dos personagens. O mesmo vale para as expressões faciais, que, além de estarem associadas a drives e emoções, também possuem uma relação muito forte com o modelo geométrico usado na representação do personagem. O conjunto mínimo necessário de expressões somente vai ser definido mediante experimentos reais, em função das possíveis ações que cada agente poderá realizar, levando também em conta as potencialidades da câmera virtual, que desempenha um papel fundamental para a visualização das cenas.

Agradecimentos

Os autores agradecem o suporte financeiro do CNPq e da FINEP através das bolsas de doutorado e de pesquisa e através do Projeto VisionLab.

Referências Bibliográficas

- [AMA 97] Amandi, A. A. Programação de agentes orientada a objetos. Porto Alegre: CPGCC da UFRGS, Tese de Doutorado, 1997.
- [BLI 88] Blinn, James. Where I am? what am I looking at? IEEE Computer Graphics and Applications, pp. 76-81, 1988.
- [BON 88] Bond, A. H., Gasser, L. Readings in distributed artificial intelligence. San Mateo: Morgan Kaufmann, 1988.
- [BRE 98] Brezeal, C. A motivational system for regulating human-robot interaction. Proceedings of the Fifteenth National Conference on Artificial Intelligence, Madison, pp. 54-61, 1998.
- [BUC 02] Buckland, M. AI techniques for game programming. Premier Press, 480p., 2002.
- [CIA 99] Ciarlini, A. Geração interativa de enredos. Tese de Doutorado, Departamento de Informática, PUC-Rio, 1999.
- [CIA 02] Ciarlini, A., Feijó, B., Furtado, A. An integrated tool for modelling, generating and exhibiting narratives. AIS'2002 AI, Simulation and Planning in High Autonomy Systems, Lisboa, Portugal, pp. 150-154, April, 2002.
- [COR 94] CORREA, M. A. Filho. Arquitetura de diálogos entre agentes cognitivos distribuídos. Rio de Janeiro: COPPE da UFRJ, Tese de Doutorado, 1994.
- [COS 96] Costa, N., Feijó, B. Agents with emotions in behavioral animation. Computer & Graphics, Vol. 20 (3). pp. 377-384, 1996.
- [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.
- [DAV 00] Davenport, G., Agamanolis, S., Barry, B., Bradley, B., Brooks, K. Synergistic storyscapes and constructionist cinematic sharing. IBM System Journal, Vol. 39, Nos. 3&4, pp. 456-469, 2000.

- [DON 99] Donato, G., et al. Classifying facial actions. IEEE Trans. on Pattern Analysis and Machine Intelligence, 21 (10), pp. 974-989, 1999.
- [DRI 00] Driscoll, G. The essential guide to digital set-top boxes and interactive TV. Ed. Prentice Hall, 2000.
- [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.
- [EKM 92] Ekman, P. An argument for basic emotions. In Stein, N. L. and Oatley, K. eds. Basic Emotions. Hove, UK: Lawrence Erlbaum, pp.169-200, 1992.
- [FUN 99] Funge, J. D. AI for computer games and animation – a cognitive modeling approach. A. K. Peters Ltd, 1999.
- [FUR 96] Furht, B. Interactive television systems. Proceedings of the 1996 ACM symposium on Applied Computing, pp. 7-11, February 1996.
- [FUR 99] Furtado, A., Ciarlini, A. Operational characterization of genre in literary and real-life domains. In Proc. of the ER'99 Conceptual Modelling Conference, Paris, France, November 1999.
- [FUR 02] Furtado, A. L., Ciarlini, A. E. M. Cognitive and affective motivation in conceptual modelling. Revista Colombiana de Computación, Vol 3 (2), December 2002. Disponível em: <http://www.unab.edu.co/editorialunab/revistas/rcc/rev32.htm>
- [GRA 01] Gratch, J., Marsella, S. Tears and fears: modeling emotions and emotional behaviors in synthetic agents. In Proceedings of the 5th International Conference on Autonomous Agents, Montreal, Canada, June 2001.
- [HAY 01] Hayking, S. Redes neurais: princípios e prática. Bookman, 2001.
- [HAW 03] Hawkins, B. Creating an event-driven cinematic camera, part one. January 8, 2003. Disponível em: http://www.gamasutra.com/features/20030108/hawkins_01.htm.
- [HE 96] He, L., Cohen, M. F., Salesin, D. H. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (Proceedings of SIGGRAPH), pp. 217-224, August 1996.
- [IZA 91] Izard, C. E. The psychology of emotions. New York: Plenum Press, 1991.

- [IZA 93] Izard, C. E. Four systems for emotion activation: cognitive and noncognitive processes. *Psychological Review*, 100(1), pp. 68-90, 1993.
- [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.
- [MAT 02] Matthews, J. Basic a* pathfinding made simple. In: *AI Game Programming Wisdom*, Ed. Charles River Media, 2002.
- [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.
- [MOL 02] Moller, T., Haines, E. *Real-time rendering*, A.K.Peters Ltd, Natick, MA, 900 p., 2nd edition, July 2002.
- [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.
- [POZ 03a] Pozzer, C. T., Feijó, B., Ciarlini, A. Proposição de um novo paradigma de conteúdo para TV interativa. *Série Monografias em Ciência da Computação (MCC38/03)*, DI/PUC-Rio, Rio de Janeiro, 2003.
- [POZ 03b] Pozzer, C. T., Dreux, M., Feijó, B., Representação gráfica de histórias interativas. *Série Monografias em Ciência da Computação (MCC40/03)*, DI/PUC-Rio, Rio de Janeiro, 2003.

- [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.
- [PRO 68] Propp, V. Morphology of the folktale. Laurence Scott (trans.), Austin: University of Texas Press, 1968.
- [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.
- [TOM 00] Tomlinson, B., Blumberg, B. Nain, D. Expressive autonomous cinematography for interactive virtual environments. Proceedings of the fourth international conference on Autonomous agents, pp. 317-324, June 2000.
- [VEL 97] Velázquez, J. D. Modeling emotions and other motivations in synthetic agents. In Proceedings of the Fourteenth National Conference on Artificial Intelligence, pp. 10-15, Menlo Park, Calif.: AAAI Press, 1997.
- [WIL 83] Wilensky, R. Planning and understanding. Addison-Wesley, London, 1983.
- [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.