

5

Agentes em Histórias Interativas

Segundo Mateas [97], com o interesse recente por pesquisa em agentes autônomos, têm surgido muitos estudos na interseção entre agentes e narrativas, dentre os quais podem-se destacar o uso de agentes como personagens em histórias e agentes que podem contar histórias. Essa visão se aplica diretamente na abordagem deste trabalho, tanto a nível de geração como de visualização das histórias.

A tecnologia de agentes tem sua origem em IA distribuída [15], em trabalho cooperativo suportado por computador [7], em animação comportamental [119][145][136], em personagens emocionais [9][10][11][91][118] e em robótica [18]. Apesar de programação orientada a agentes ter sido proposta em 1993 como uma paradigma pós-objeto [124], somente em 2003-2004 é que agentes passaram a ser tratados com rigor e abrangência da Engenharia de Software [59]. Uma referência clássica sobre teorias, arquiteturas e linguagens de agentes, sob o enfoque da comunidade de IA, pode ser encontrada em [146]. Outras referências mais recentes são [2] e [85].

Uma definição precisa de agentes só é possível de ser dada em função do propósito pelo qual o agente é construído, uma vez que agentes são entidades que encapsulam conhecimento sobre algum domínio [85]. Entretanto, agentes podem ser genericamente definidos como entidades de software que têm as seguintes características: autonomia, habilidade social, reatividade e pro-atividade. Racionalidade, habilidade de aprender e mobilidade são características adicionais que não são nem necessárias nem suficientes para caracterizar agentes.

Para o contexto deste trabalho, em especial tratando a geração de histórias, a idéia principal do uso de agentes é a especificação de uma modelagem que os permita alcançarem seus objetivos em circunstâncias que podem envolver competição. Deseja-se que o comportamento dos personagens baseie-se na busca de seus objetivos, os quais surgem quando condições particulares aparecem nas histórias [33]. Neste trabalho, o

objetivo do agente é um conjunto de fatos que este agente deseja tornar verdadeiros em um certo instante.

Em se tratando da visualização gráfica das histórias, o agente é visto como uma entidade (personagem autônomo) 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 com outros agentes [103]. A Figura 5.1 ilustra esta definição.

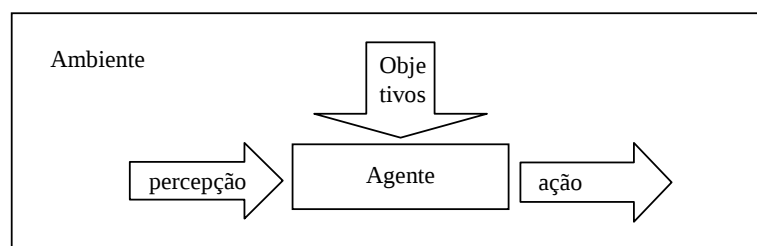


Figura 5.1: Agentes na representação gráfica de histórias.

Este capítulo destina-se a fazer um estudo das tecnologias de agentes, voltadas à implementação do sistema de *storytelling* que está sendo proposto. Na Seção 5.1 são apresentados conceitos gerais da teoria de agentes e sistemas multi-agentes, bem como discussões das técnicas mais adequadas à implementação do trabalho proposto. Na Seção 5.2 faz-se uma apresentação sobre comportamentos de agentes e a proposição de um modelo comportamental que é usado para o gerenciamento das ações dos agentes na etapa de visualização gráfica. Na Seção 5.3 apresentam-se aspectos de implementação de câmeras autônomas, bem como especificações das estratégias usadas neste trabalho. Por fim, na Seção 5.4, faz-se um estudo de drives e emoções, bem como discussões de como estas funcionalidades podem ser incorporadas no sistema que está sendo proposto. Drives são necessidades básicas como fome, sede, interação social, entre outros.

5.1

Modelos de Agente para Storytelling

Autonomia, habilidade social, capacidade pró-ativa e reatividade são propriedades comumente aceitas que tornam um agente diferente de um simples objeto de software. A autonomia permite ao agente executar a maior parte de suas ações sem interferência direta de agentes humanos ou de outros agentes computacionais. A habilidade social é a propriedade

que permite que agentes interajam com outros agentes (humanos ou computacionais) para completarem a resolução de seus problemas ou para auxiliarem outros agentes usando alguma linguagem de comunicação de agente¹. Esta interação deve-se à impossibilidade de resolução de certos problemas ou a algum tipo de conveniência. A capacidade pró-ativa refere-se ao comportamento orientado a objetivos, que leva o agente a tomar iniciativas quando julgar apropriado. Por último, reatividade é a capacidade do agente reagir à percepção do ambiente, através de seus sentidos, tais como visão e audição.

Quando se cria um agente para uma aplicação específica, deve-se ter em mente a seleção das 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 a passagem de mensagens.

Os agentes podem ser categorizados quanto à arquitetura que suporta a resolução de problemas: cognitivos, reativos e híbridos. Agentes 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. Neste grupo, existe uma categoria baseada em estados mentais, chamada BDI, que considera três estados mentais para a descrição do processamento interno de um agente: crença, desejo e intenção (*belief, desire, intention*). As crenças representam o conhecimento do agente acerca do ambiente onde está situado. Os desejos, segundo Rao [117], 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.

Muitos destes conceitos se aplicam na modelagem do IPG, visto que tem-se claramente especificado o estado motivacional que vai levar o agente a agir para atingir seus objetivos. As crenças podem representar os atributos especificados na base de dados que são previamente modelados de forma a direcionar a geração dos planos que conduzirão a história.

A arquitetura reativa é uma abordagem que rompe com o paradigma de IA simbólica tradicional. Esta arquitetura tem raízes em alguns trabalhos pioneiros em animação [119], mas surge mais formalmente nos trabalhos de R. Brooks [17]. Brooks propõe que entidades reativas devem responder

¹KQML (*Knowledge Query and Manipulation Language*) [152] ou FIPA ACL (*Agent Communication Language*) [1], por exemplo.

dinamicamente a mudanças no ambiente. Ademais, Brooks lança os primeiros princípios de emergência e *situatedness* para as entidades reativas.

O princípio de emergência declara que a inteligência de um sistema de agentes emerge da interação de agentes com eles mesmos e com seus ambientes. Cada agente isoladamente é tão simples quanto os agentes sem mente propostos por Minsky [101]. Neste princípio, a inteligência surge da interação. O princípio de *situatedness* afirma que a inteligência de um agente está situada no mundo e não em algum modelo formal do mundo construído no agente. Portanto, um agente usa a sua percepção do mundo, ao invés de usar deduções baseadas em alguma representação simbólica deste mundo.

Nesta arquitetura, o agente não tem capacidade de raciocínio sobre suas intenções, reagindo tão-somente sobre regras e planos pré-definidos. Esta é uma abordagem que se adequa perfeitamente às histórias baseadas no enredo (*plot-based*), visto que os planos já foram previamente definidos na fase de geração das histórias. Não é desejado que os personagens raciocinem e nem tomem decisões inteligentes, para evitar que executem ações espontâneas que mudem o rumo da história já simulada (Os agentes devem agir como atores guiados [132][23]). Os agentes devem ter habilidades suficientes para cumprir ordens, que neste caso são eventos da história, bem como apresentar características reativas mediante a interação com outros agentes. Aspectos de aprendizagem também não se fazem necessários.

Tratando-se o gerador e o visualizador das histórias como um sistema único e integrado, pode-se dizer também que este faz uso de uma arquitetura híbrida, onde a escolha das ações é realizada utilizando-se uma combinação de técnicas de arquiteturas reativas e cognitivas. A arquitetura híbrida 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 Nareyek [103], 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.

No sistema proposto, também existem diversos aspectos de Sistemas Multi-agentes (SMA) [148]. Estes sistemas estudam o comportamento de um conjunto de agentes autônomos cujo objetivo comum é a solução de um dado problema cuja resolução está além das capacidades de um único indivíduo [67]. Para o trabalho em questão, deve-se estudar como se dá a interação entre personagens dentro de enredos e no ambiente de representação gráfica. Os personagens interagem com outros personagens, positiva ou

negativamente, na busca de objetivos e, para que sejam consistentes, devem usar uma lógica que possibilite inferir objetivos e adotar planos para atingi-los [33]. Como o objetivo geral é a criação de enredos e não a satisfação dos objetivos específicos de cada personagem, Ciarlini [33] faz uso de um único planejador que é usado para inserir operações de todos os agentes na busca de seus objetivos.

Quando se trata da representação gráfica das histórias, as características de sistemas multi-agentes se mostram mais visíveis ainda. Cada personagem é tratado como uma unidade autônoma e deve dispor de mecanismos de coordenação, interação e comunicação de modo que seus conhecimentos, objetivos, habilidades e planos individuais sejam usados de modo organizado, em favor da execução de uma seqüência ordenada de eventos gerados na etapa de criação da história.

Segundo Jennings [83], 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 deve ser garantido o respeito às restrições globais, à solução do problema e à viabilização dos procedimentos que garantem a harmonia quando da execução de uma tarefa de forma conjunta por mais de um agente.

No presente trabalho, a coordenação é implementada de forma centralizada, ou seja, existe um módulo responsável pelo controle da ordem de execução das operações especificadas no enredo. Como diversas operações requerem mais de um agente, mecanismos de interação também fazem-se presentes.

Para que uma interação ocorra, deve ser conhecida a necessidade da interação para que o conteúdo da interação (mensagem) possa ser definido, dentro de uma gama de recursos disponíveis a serem utilizados para tal fim. A comunicação é um meio de realizar a interação entre agentes. É composta por duas partes: o envio da mensagem (ação) e a recepção da mensagem (percepção). Faz-se uso de uma arquitetura nivelada, ou seja, a comunicação entre eles ocorre de forma direta. Para finalizar, tem-se o ambiente, que constitui o contexto onde todas as interações entre os agentes ocorrem. Através do ambiente, ocorre a disseminação do controle, dos dados e do conhecimento pela comunidade de agentes [3]. O nível de acessibilidade do ambiente determina a quantidade de detalhes que o aparato sensorial do agente pode detectar. Neste trabalho, o ambiente é considerado efetivamente

acessível, ou seja, os sensores detectam todos os aspectos relevantes para a escolha da ação. Neste caso, não é necessário que o agente mantenha qualquer representação interna do mundo, visto que qualquer informação necessária pode ser imediatamente obtida.

5.2

Modelagem comportamental

O comportamento de um agente engloba as possíveis ações que este agente pode desempenhar. Essas ações são resultado da combinação de um elenco de regras que foram associadas ao agente. A especificação destas regras é resultado da avaliação das principais características do sistema onde o agente está inserido e, conseqüentemente, do que é esperado que o agente seja capaz de realizar. Para um melhor entendimento do comportamento de um agente, Reynolds [120] sugere quebrá-lo em várias camadas, como apresentado na Figura 5.2, que retrata o comportamento de movimentação em uma hierarquia de 3 camadas.

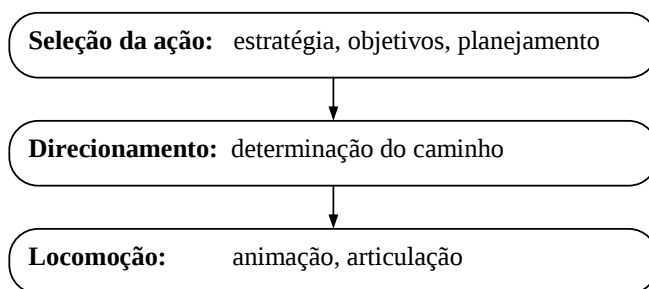


Figura 5.2: Uma hierarquia do comportamento movimentação [120].

Para esta pesquisa de tese, este é o principal comportamento desempenhado pelos agentes, uma vez que a locomoção é pré-requisito para diversas outras operações dentro do elenco de eventos das histórias. A idéia de separação em hierarquia se aplica perfeitamente no escopo desta tese. Pela análise da Figura 5.2, no topo da hierarquia está a seleção da ação. Como já apresentado no Capítulo 4, este processo é definido ou pelo IPG ou como resultado do comportamento reativo resultante da interação entre os agentes. Além disso, novas ações podem surgir da interação do agente com o cenário. A camada de direcionamento (*steering*), refere-se à forma como o agente adquire informação do ambiente para que possa chegar ao local determinado, evitando possíveis obstáculos. Por último, tem-se a animação gráfica da ação que está sendo realizada.

Operações relacionadas à movimentação de personagens referem-se sempre a um único indivíduo. Desta forma, comportamentos de movimentação em grupo não são tratados. Também não é objetivo desta tese fazer um estudo aprofundado de comportamentos de movimentação/manobras. Reynolds [120] apresenta uma descrição mais detalhada destes tópicos.

Na Seção 5.2.1 é apresentado um modelo comportamental simples e ao mesmo tempo eficiente para gerenciar seqüências de ações estáticas ou dinâmicas necessárias à dramatização dos eventos estipulados dentro dos enredos.

5.2.1 Modelo Comportamental

Máquinas de Estados Finitos (*Finite State Machines* - FSM) são provavelmente o padrão de software mais utilizado em jogos para controlar o comportamento de agentes reativos. As FSMs são geralmente representadas por diagramas de transição de estados. Elas são fáceis de implementar e depurar [116], além de apresentarem alta eficiência, uma característica essencial da implementação de agentes reativos, cuja característica marcante é a resposta rápida a estímulos externos.

Entretanto, à medida que a complexidade do comportamento dos agentes aumenta, as FSMs tendem a crescer de forma descontrolada, o que torna a implementação impraticável. Uma solução possível para este problema é o uso de máquinas hierárquicas (HFSM), também chamadas *Behavioral Transition Networks* [72], como proposto por vários pesquisadores [122][54][107]. Nesta abordagem, níveis mais altos lidam com ações mais genéricas, enquanto níveis mais baixos lidam com ações mais específicas. Entretanto, a hierarquia nem adiciona mais poder ao modelo, nem reduz o número de estados. Ela pode somente reduzir significativamente o número de transições e tornar a FSM mais intuitiva e simples de compreender [61]. É importante observar que qualquer HFSM pode ser reescrita como uma FSM sem hierarquia [66].

Uma característica observada durante o desenvolvimento do modelo de comportamento do agente foi que geralmente as ações a serem realizadas, em muitos casos, possuem pré-condições. Essas pré-condições são expressas por meio de ações auxiliares que devem ser realizadas para garantir a integridade e veracidade da ação previamente estabelecida (ação terminal). Quando se está trabalhando com a representação de ações complexas e abstratas,

onde a execução de uma ação depende da execução de um conjunto de ações precedentes, HFSMs apresentam as mesmas limitações que as FSMs. Como exemplo, suponha que um agente (personagem) tem como objetivo a libertação de outro personagem que está raptado. Para que isso ocorra pode ser necessário, por exemplo, que o personagem deva executar um conjunto de ações auxiliares, como andar, lutar, conversar, dentre outras. Muitas destas ações podem envolver outros agentes que habitam o mesmo ambiente. Quando o personagem cumprir todos os pré-requisitos necessários, o agente pode então realizar a ação à qual está designado.

Para lidar com estes tipos de situações, a arquitetura proposta incorpora uma estrutura de pilha a cada agente como memória auxiliar à FSM, de modo a evitar a criação de nós adicionais que seriam necessários para tratar situações onde ações possuem como pré-condições a execução de outras ações. A pilha possui apenas duas posições. Na base (*slot 0*) é guardada a ação que representa o objetivo final (*goal*) do agente, ou seja, o evento que representa o fim da missão, que, para o exemplo anterior, é a libertação do prisioneiro. No topo da pilha (*slot 1*) é armazenada a ação intermediária corrente. A meta do agente é sempre executar a ação que está no topo da pilha, que é sempre a última inserida. Uma ação permanece na pilha enquanto sua execução estiver ocorrendo ou até que outra ação mais importante (geralmente pré-condição) faça-se necessária.

Na Figura 5.3 é apresentado a arquitetura, composta basicamente por 5 módulos, usada no controle dos comportamentos dos agentes:

Perceptor: Continuamente adquire informação do mundo. O sistema visual do agente realiza testes de colisão com objetos na cena. O sistema de mensagens processa mensagens enviadas por outros agentes. Estas mensagens podem mudar o estado interno de modo que o agente possa participar de uma ação coletiva;

Pilha de duas posições: A pilha é usada como um repositório de ações que devem ser executadas;

Fila de história: Mantém um histórico das últimas N ações executadas. Com este repositório, no caso da inexistência de ações específicas, o Gerenciador de Ações pode selecionar ações aleatórias não repetitivas para o agente. A pilha, juntamente com a fila, representam a memória de trabalho do agente;

Gerenciador de Comportamentos (FSM): É o elemento central da arquitetura. Ele processa entradas, determina novas ações auxiliares

e designa ações ao effector que melhor refletem o estado corrente do agente;

Effector: É o meio pelo qual o agente muda o ambiente. Mensagens podem invocar outros agentes e, conseqüentemente, mudar seus atributos. Além disso, este módulo é o responsável pela movimentação do agente.

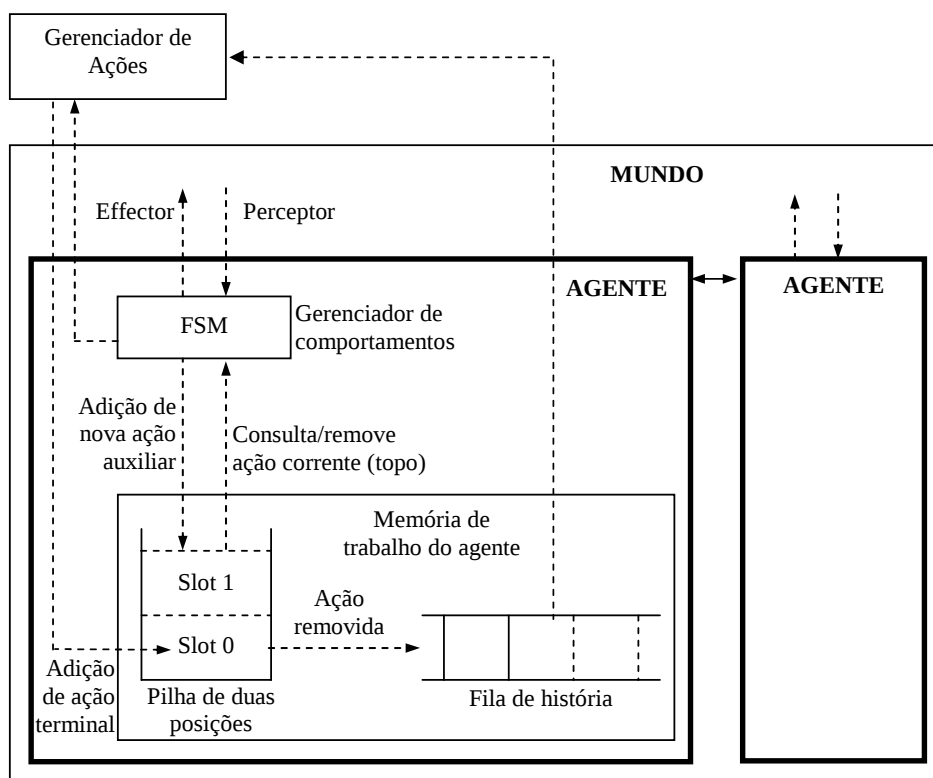


Figura 5.3: Modelo de gerenciamento de tarefas do agente.

A memória de trabalho do agente é representada pela pilha e fila de história. O estado corrente do agente é representado unicamente pela pilha. Tanto o Gerenciador de Ações, como o próprio agente, podem inserir operações na pilha. O Gerenciador sempre insere uma nova ação terminal na base toda vez que a pilha estiver vazia. O próprio agente deve comunicar ao gerenciador que não tem mais ações a executar. Garantindo que as ações terminais são sempre inseridas na base, a pilha ainda tem uma posição livre para tratar ações auxiliares que se fizerem necessárias. Essas ações auxiliares são sempre inseridas pelo próprio agente, mediante a impossibilidade do cumprimento da ação terminal, segundo regras implementadas em cada nó da FSM. Como exemplo, se o agente estiver se locomovendo, ações candidatas correspondem a outras ações de deslocamento, no caso, para evitar colisão com possíveis objetos. Se o agente deve atacar um determinado

local, ações de deslocamento e luta podem ser continuamente inseridas até que a ação de ataque seja finalizada.

Na Figura 5.4 é apresentado um comportamento simplificado da pilha para o exemplo da libertação da vítima. No estado inicial, o herói recebe como ação **terminal** a libertação da vítima. Esta ação é então armazenada na base da pilha do herói, e somente será removida com o final de sua execução. No estado **Free** da FSM, duas pré-condições estão associadas: o herói deve estar próximo da vítima e deve derrotar o vilão (raptor da vítima). Quando a FSM do herói é executada, as pré-condições são avaliadas. Supondo que ele já esteja no mesmo local da vítima, este cria uma ação auxiliar para lutar com o vilão, que é colocada no topo da pilha. A ação de lutar, por sua vez, tem como pré-condições estar próximo e ser mais forte que o oponente. Caso os personagens estejam próximos e o vilão seja mais forte que o herói, uma ação para tornar o herói mais forte é adicionada em sua pilha pelo estado **Fight** da FSM. Neste momento, a pilha já está com as duas posições ocupadas (pilha cheia). Neste caso, a ação do topo (**Fight**) é removida para a inserção da nova (**Get_stronger**), mesmo que esta ainda não tenha sido finalizada. Quando a operação **Get_stronger** finalizar, a execução volta à base da pilha (**Free**), o que faz com que a operação **Fight**, por ainda não ter sido executada e por ser pré-condição da libertação, seja novamente reinserida no topo da pilha, desta vez com suas pré-condições já satisfeitas. A ação **Fight** é então executada até que o herói derrote o vilão. Quando o **Fight** é finalizado, a ação **Free** pode então ser executada. Quando a operação **Free** é finalizada, o personagem não tem nenhuma ação a realizar. Neste momento, ele solicita ao Gerenciador o envio de uma nova ação.

Em termos gerais, o agente deve ter a habilidade de parar a execução da ação corrente em favor da execução de uma ação de mais alta prioridade. Isso não trás nenhum problema, porque, se for necessário, a ação removida pode ser reinserida na pilha pela FSM tantas vezes quanto forem necessárias.

Fazendo-se uma analogia à máquinas hierárquicas, cada posição da pilha age como uma FSM, que armazena a configuração local do contexto sendo executado. A pilha não necessita ter mais do que duas posições, pois com um slot variável é possível armazenar uma ação auxiliar. Ações auxiliares, resultantes de outras ações auxiliares não necessitam de espaço dedicado, visto que a partir da ação da base é possível reconstruir a hierarquia de eventos necessários a sua execução. Devido a esta estratégia, pode haver um pequeno overhead na reinserção de ações auxiliares que foram removidas em função de outras ações auxiliares mais atuais.

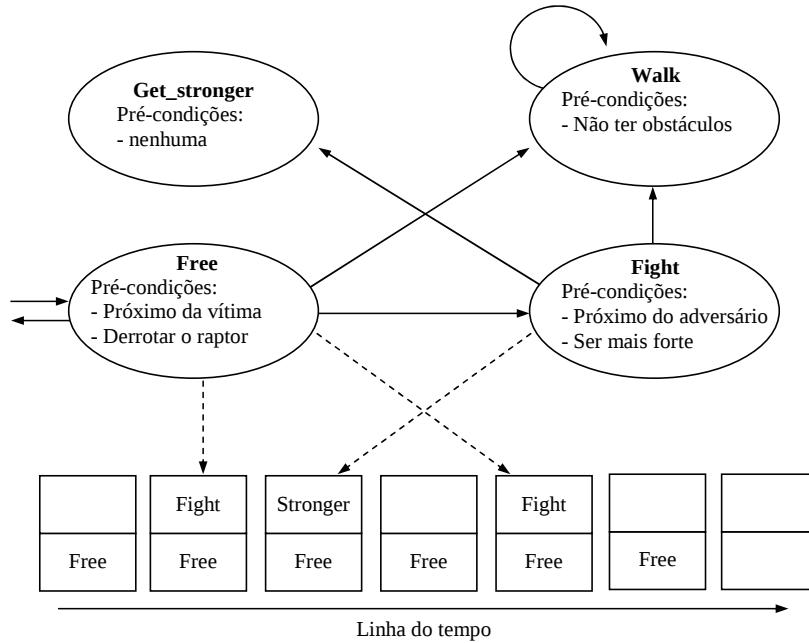


Figura 5.4: Gerenciamento da pilha pela FSM. Pré-condições de cada ação estão associados aos nós do grafo.

O uso de mais de duas posições pode levar a situações onde ações auxiliares, mesmo não mais válidas e necessárias, sejam executadas. Como exemplo, suponha que em um dado momento, o agente está executando a ação A_1 que consiste em ir a um determinado *waypoint* para desviar de um obstáculo. Neste mesmo tempo, ele detecta a presença de um inimigo que o faz fugir para uma direção perpendicular. Após a fuga, uma nova rota ao destino final deve ser criada. Se A_1 ainda estivesse na pilha, estaria no topo da pilha e deveria ser executada. Entretanto, após a fuga, a ação A_1 pode estar totalmente obsoleta, o que poderia resultar em uma ação sem lógica do ponto de vista do usuário.

Este mecanismo de pilha permite também que após o término da ação terminal, novas ações terminais sejam inseridas pelo próprio agente. Isso pode ocorrer em várias circunstâncias. Pode-se especificar que, após uma libertação, por exemplo, ambos os personagens devem ir para suas respectivas moradias.

Uma abordagem que também faz uso de uma pilha como forma de expandir as potencialidades das FSM foi proposta por [135], com a diferença que a pilha tem tamanho ilimitado. Neste presente trabalho, todos os exemplos testados indicaram que uma pilha com apenas duas posições é a melhor solução. De fato, na arquitetura proposta, ações complexas como “perseguir um inimigo, desviando de obstáculos, para iniciar um combate”,

podem ser adequadamente tratadas.

Mais exemplos que mostram o gerenciamento de situações complexas, que envolvem comportamentos de manobra, são apresentados na próxima seção.

5.2.2

Comportamentos de manobra

O modelo proposto incorpora explicitamente um único tipo de manobra: atingir uma posição específica (*Seek*). Apesar de parecer um tanto simplificado e limitado, este comportamento é utilizado implicitamente como base para execução de outros comportamentos como: perseguir, seguir caminho e evitar colisão.

Para melhor compreender como estes comportamentos são implementados, deve-se observar como está estruturado o ambiente onde os agentes estão inseridos. A estratégia proposta de movimentação do mundo é baseada em *waypoints*² e análise de terreno (*Terrain Reasoning*). *Waypoints*, de modo geral, são sinalizações de locais que podem ajudar o agente a navegar no mundo, uma vez que eles agem como nós de um grafo que representam os caminhos pelos quais o agente pode navegar [87]. Análise de terreno é uma técnica de IA para jogos que auxilia o raciocínio sobre informações do terreno, de modo que planejamento, tomada de decisões, controle de ações, dentre outros, podem ser processados [130].

Todas as ações são associadas com *waypoints*, que podem ser estáticos, no caso de objetos dispostos no cenário (como moradias e construções), ou dinâmicos, no caso de ações que envolvem outros personagens. No presente trabalho, a cada objeto do cenário estão associados 5 *waypoints*, como mostrado na Figura 5.5. Os quatro presentes nos cantos representam um caixa envolvente (*bounding box*) estendida do objeto e o *waypoint* central representa a porta, que é usado para guiar o personagem para dentro e para fora do objeto. O volume envolvente é usado para testes de colisão com o caminho corrente do agente. Em frente ao local de entrada, é definida uma região proporcional ao tamanho do objeto que é considerada área externa do objeto. Tanto a área externa como a interna podem ser usadas para designar alvos em ações do tipo *Walk*.

O motivo principal para o uso da rede de *waypoints* como estratégia de locomoção, em oposição a algoritmos de planejamento de caminho do tipo A* [21], é a possibilidade de se utilizar o modelo comportamental,

²Esta tese sugere a tradução de *waypoint* para pontos de caminho.

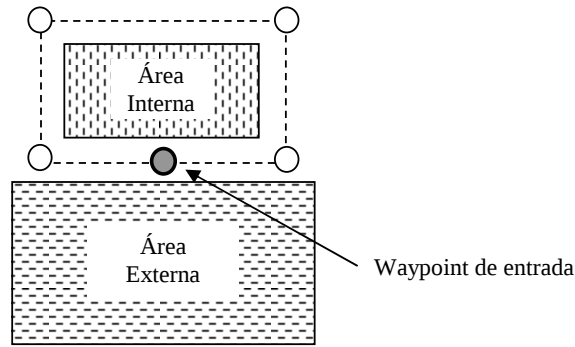


Figura 5.5: *Waypoints* associados aos objetos.

apresentado na seção anterior, como suporte à movimentação e desvio de obstáculos, bem como para o tratamento de situações que envolvam na locomoção do agente para o interior de um objeto.

A idéia é quebrar o caminho completo em pequenos segmentos, como mostrado na Figura 5.6. A criação dos segmentos ocorre sempre que o caminho selecionado (corrente) fizer colisão com algum objeto da cena (que inclusive pode ser dinâmico). O novo segmento adicionado deve momentaneamente mudar a rota de modo a evitar a colisão. Para a escolha do novo segmento, faz-se uso da visão do agente. Se o caminho tiver interseção com a caixa envolvente do objeto, procura-se pelo *waypoint* mais próximo do destino que seja visível pelo agente. A nova rota geralmente representa a menor distância ao destino. A Figura 5.7 mostra um exemplo deste algoritmo.

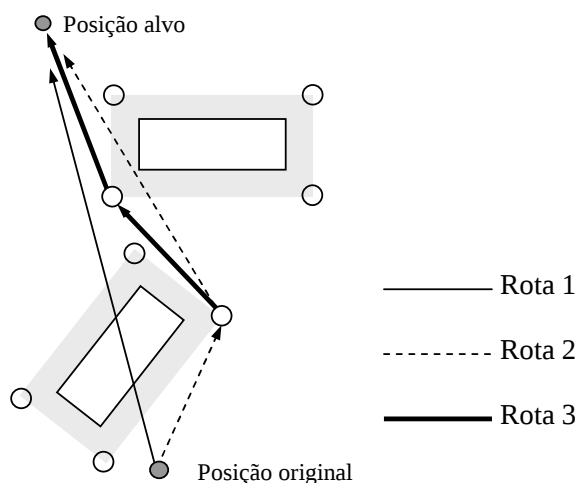


Figura 5.6: Composição incremental da rota.

Cada segmento, à medida que o agente se move no ambiente, é convertido em uma ação do tipo *Walk*, cujo destino imediato é a extremidade

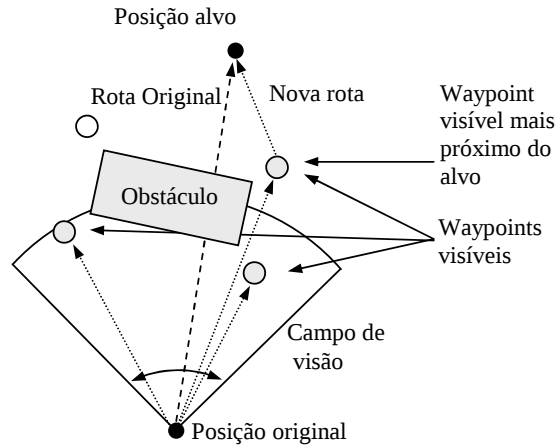


Figura 5.7: Algoritmo para seleção do melhor *waypoint*.

final do segmento. Uma vez que o caminho é organizado como um conjunto de pequenos caminhos individuais, essas ações de **Walk** podem ser facilmente inseridas e removidas na pilha do agente, sempre que se fizerem necessárias para assegurar a finalização da ação terminal. Desta forma, a FSM trabalha transparentemente com qualquer tipo de ação, incluindo planejamento de caminho e tratamento de colisão com objetos fixos e móveis.

Para situações mais complexas, como fazer o agente entrar em uma casa, por exemplo, faz-se uso da mesma estratégia. O agente inicialmente encontra o *waypoint* visível mais próximo do destino final. Quando ele chega a esta posição, tenta novamente ir ao destino final. Se uma nova colisão for detectada, um novo *waypoint* (diferente do corrente) é selecionado. Este processo continua até que o agente atinja o *waypoint* de entrada. Neste momento, os testes de colisão são desabilitados e o agente pode então entrar em casa. Na figura 5.8 é apresentado um exemplo do funcionamento do algoritmo proposto.

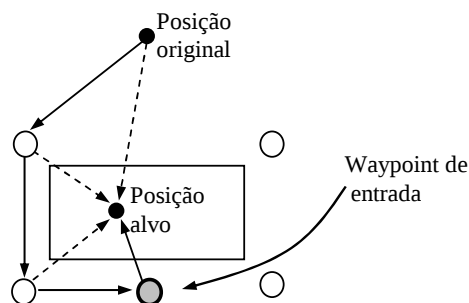


Figura 5.8: Exemplo de uma rota para entrar em um objeto do cenário.

Esta estratégia de tratamento de colisão falha quando as caixas envolventes de objetos têm interseção. Na figura 5.9 é apresentado um

exemplo desta configuração. Pode-se observar em (b) que o trajeto selecionado conduz o agente a um *waypoint* não alcançável, impedindo-o de atingir o alvo.

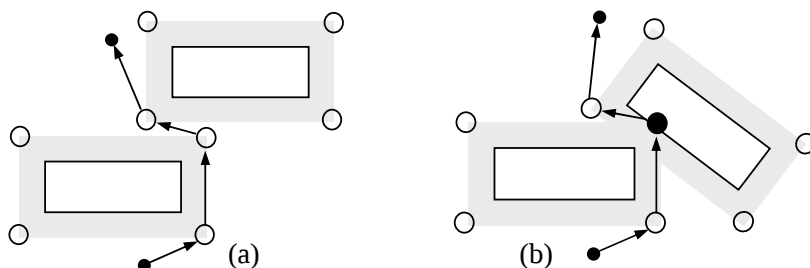


Figura 5.9: Exemplo de caminhos atingível (a) e não atingível (b).

5.3

Câmera Virtual

Conceitos de agentes também fazem-se presentes na estipulação de comportamentos que a câmera virtual deve apresentar. A câmera é um dos principais módulos de visualização, pois é responsável por determinar qual cena, 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 que estão ocorrendo. Ela deve ser autônoma o suficiente para selecionar o conteúdo a exibir, se posicionar, se locomover e fazer transições entre tomadas sucessivas.

Existem diversos trabalhos com o intuito de criar sistemas que procuram determinar o melhor posicionamento da câmera em cenas interativas 3D [63][111]. Para situações de tomadas individuais, diversas estratégias e restrições de tomadas podem ser encontradas em Drucker et al [43][45]. Uma tomada é o intervalo de tempo que a câmera grava continuamente. Aspectos matemáticos para definição de parâmetros de câmera, dada a geometria da cena e posição do personagem desejado, podem ser vistos em [14][68][44]. Diversas heurísticas e restrições referentes a tomadas sucessivas na criação das cenas são propostas [69][32]. Para definição da ordem destas tomadas, existem fórmulas específicas que melhor capturam a essência da cena. Uma longa discussão destas fórmulas, com situações práticas de uso, pode ser encontrada em [5].

Dois trabalhos merecem atenção especial, pois têm como foco a definição de sistemas automáticos de cinematografia para serem usados em ambientes virtuais interativos. Tomlinson [134] apresenta um sistema que é

responsável pelo controle de uma câmera e luzes em um mundo virtual 3D habitado por personagens autônomos e controlados pelo usuário, de forma que o conteúdo emocional possa ser evidenciado. A câmera é implementada por um agente chamado *CameraCreature*. Ele encapsula sensores para extrair informações sobre os estados emocionais e motivacionais, além das ações que estão sendo realizadas pelos agentes do ambiente. Além dos sensores, esta câmera autônoma também possui emoções, motivações e ações. O modelo de emoção da câmera é usado na escolha da tomada a ser realizada e dos parâmetros relacionados. Cada estado emocional causa um efeito visual característico. O estado emocional da câmera é calculado por uma função que leva em consideração o seu temperamento, a taxa de variação da emoção no tempo e fatores internos e externos que afetam o estado emocional. Uma vez que o estado emocional dos agentes da cena tem influência no estado emocional da câmera, os estilos de tomadas refletem o estado emocional dos personagens. Para ressaltar este estado emocional do grupo, são usados tanto estratégias de posicionamento de câmeras como recursos de iluminação

Em He [69], é descrito o *Virtual Cinematographer* (VC), uma arquitetura que automaticamente gera especificações de câmera para captura de eventos em ambientes virtuais 3D, em tempo real, a partir de informações e eventos passados pela aplicação. A estrutura geral é apresentada na Figura 5.10.

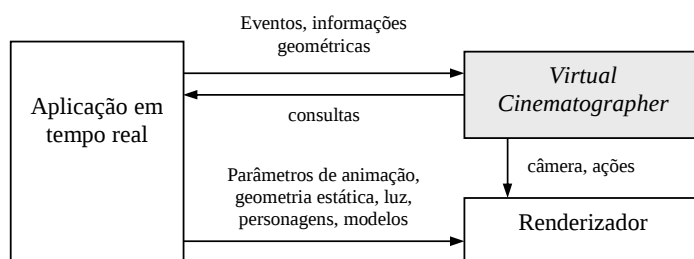


Figura 5.10: Arquitetura de uso do Virtual Cinematographer [69].

O conhecimento necessário para determinação das seqüências de tomadas para cada cena, bem como as condições necessárias para fazer a transição de tomadas, são encapsulados em *idioms*. Os *idioms* são implementados por máquinas de estados finitas hierárquicas (HFSM), onde cada estado está associado com um módulo de câmera. Cada *idiom* tem a capacidade de capturar um tipo de situação particular, como o diálogo entre dois personagens ou o deslocamento de um personagem de um lugar para outro.

Nesta tese, faz-se uso de uma arquitetura semelhante a He [69]. Entretanto existem vários aspectos que diferenciam os dois modelos. Um deles está no tipo de cena a ser visualizada. Em He [69], a câmera deve monitorar ações de um personagem específico, que geralmente é o que está sob o controle do usuário. No sistema proposto, a câmera deve decidir, entre um elenco de personagens que compõem a história que está sendo visualizada, qual deve ser exibido e de que forma. Estes personagens podem estar realizando ações individuais ou em dupla. Além de selecionar o(s) personagem(s), a câmera deve também se posicionar e se locomover para acompanhar personagens em movimento.

Para a seleção do(s) personagem(s), a câmera consulta o Gerenciador de Ações para saber qual a operação corrente. Com esta informação, descobrem-se quais são os personagens e, desta forma, a ação que está sendo processada. Maiores detalhes sobre a câmera são apresentados no Capítulo 7, que trata da implementação.

5.4

Drives e emoções

A representação de agentes com comportamentos realistas é fundamental em aplicações de entretenimento, como cinema, iTV [42], jogos de computador e realidade virtual [39]. Segundo Brezeal [16], o comportamento, juntamente com expressões faciais, são o meio pelo qual o agente pode expressar sua motivação, que é determinada por drives e emoções.

Os drives influenciam na seleção de comportamento e no estado emotivo [57]. Uma vez que a expressão do agente reflete seu estado emotivo, o grau de satisfação dos drives influencia indiretamente nas expressões faciais [16]. As expressões faciais, em aplicações de entretenimento, podem 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 [142]. As expressões faciais podem ser representadas, por exemplo, pelas sobrancelhas, olhos, pálpebras, orelhas e boca.

Drives não podem se satisfazer sozinhos. Eles se tornam satisfeitos quando o agente é habilitado a ativar um comportamento adequado para supri-los. 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 [16]. Um drive ou emoção é considerado ativo quando sua intensidade for superior a um dado limiar (*threshold*).

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 está 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) [142]. Segundo Izard [81], a ativação de emoções, assim como o aumento ou redução de sua intensidade, pode ocorrer por estímulos de sensores cognitivos e não cognitivos.

Segundo Velazquez [142], as emoções também têm influência no comportamento de agentes autônomos. Esta constatação resulta do fato das 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 [37]:

- Como as emoções são mantidas e como é a 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 agentes;
- Como elas são influenciadas pelas características internas do agente.

Segundo modelos teóricos sobre emoções [49][80], 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. Estas emoções também apresentam características que podem ser facilmente mapeadas em traços distintos por meio de expressões faciais [41]. São elas: raiva,

tristeza, alegria, medo, repulsa e surpresa (*distress/sadness, anger, enjoyment/happiness, fear, disgust, surprise*). Elas possuem traços característicos, principalmente nos olhos e boca, e em locais onde há formação de rugas, como mostrado na Figura 5.11.

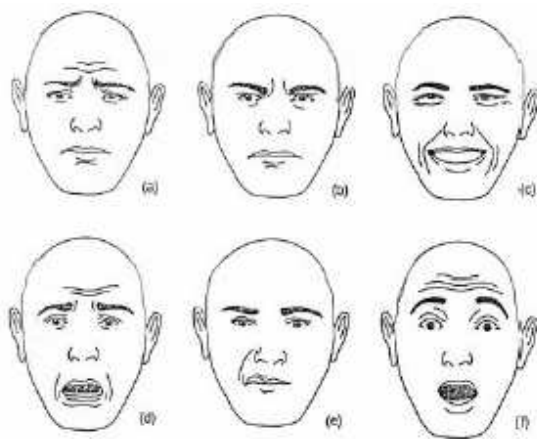


Figura 5.11: Exemplos das expressões faciais universais [109].

Durante o desenvolvimento deste trabalho, investigou-se a possibilidade da incorporação de drives e emoções, bem como expressões faciais aos agentes. 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 agentes e o efeito sobre o direcionamento da história. As expressões faciais estão relacionadas somente com a representação gráfica.

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 momento³. Em se tratando da geração da história, podem ser facilmente associados a pré- e pós-condições das operações, bem como podem ser usados nas regras de inferência de objetivos. Suas intensidades podem ser explicitamente definidas, da mesma forma como ocorre com os demais atributos, em função da operação que está sendo processada e de características inerentes de cada personagem.

A nível de representação gráfica, o agente está imerso em um ambiente real e, por isso, propenso a interações com o meio e com outros agentes, o que pode influenciar fortemente na alteração de drives e emoções e, conseqüentemente, no seu comportamento. Como nesta etapa

³Uma representação mais robusta pode ser implementada usando lógica nebulosa (*fuzzy logic*) e variáveis lingüísticas.

são considerados aspectos do cenário, como distâncias e obstáculos, bem como personagens figurantes, pode-se determinar a intensidade de drives e emoções para cada ação que os personagens realizam. Desta forma, eles podem ter um papel fundamental na geração de novos objetivos, podendo assim influenciar no direcionamento da história sendo criada pelo IPG.

Alterar o rumo da história previamente simulada e certificada, pela inclusão de novas ações que visem suprir algum drive, por exemplo, é uma questão que ainda deve ser estudada e é sugerida como trabalho futuro. Em um momento inicial, drives e emoções podem ter apenas um efeito passageiro, servindo assim como enriquecedores da apresentação gráfica da história. Pode-se considerar a geração de ações que levem o drive ao regime normal, mas após a finalização destes eventos, as ações que foram interrompidas devem voltar a ser executadas, sem comprometer de forma alguma o direcionamento da história.

Referente às emoções, a afeição é a única que está atualmente implementada no IPG. Na etapa de visualização, as emoções podem ser usadas apenas para efeito da própria representação, ou seja, fazer uso de expressões faciais para descrever o estado emocional do agente associado ao comportamento que está sendo apresentado a cada momento. Nesta abordagem, as emoções são resultado da execução de uma tarefa concreta (*task-oriented emotions*) [64].

Sob o aspecto de implementação, a incorporação dos drives é um processo relativamente simples. Uma vez que ações para suprir drives são geradas pelo próprio agente, este pode usar a estrutura da pilha para gerenciar as ações auxiliares. Desta forma, a ação terminal continua na base da pilha e será executada assim que drives forem supridos e pré-condições satisfeitas, observando-se sempre qual tem maior prioridade. No caso das emoções, uma alternativa simples seria o uso de diferentes texturas, que seriam mapeadas sobre a face do personagem. Representações mais complexas geralmente utilizam-se estruturas que tratam tanto a configuração dos ossos [109] como dos músculos presentes na face [110]. Os músculos faciais podem tanto ser usados para representação das expressões bem como da fala do personagem [92]. Neste caso, deve-se dispor de modelos complexos de personagens, o que está fora do escopo desta tese. Além disso, para ressaltar a expressão facial, o módulo de câmera, em comunicação com os agentes, deve fazer um enquadramento que possa evidenciar esta característica.

Na Figura 5.12 é apresentado um esquema conceitual para a incorporação de drives e emoções no sistema proposto. Maiores detalhes

sobre o **Gerenciador de Ações** podem ser vistos no Capítulo 6.

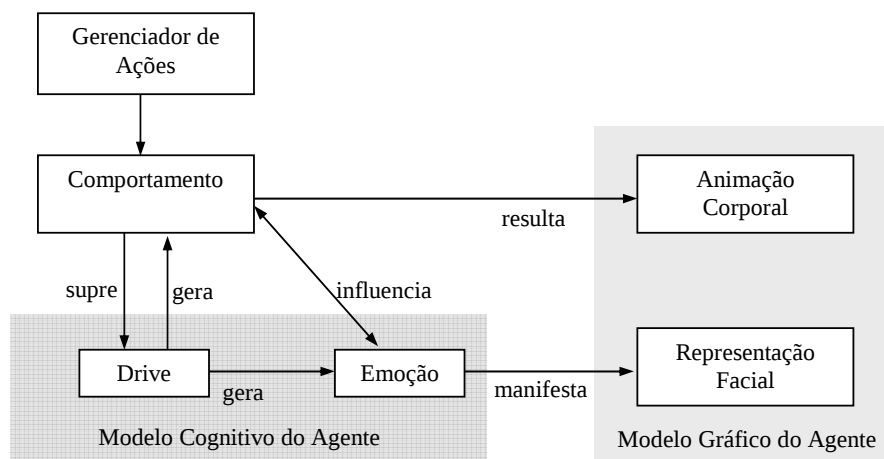


Figura 5.12: Esquema conceitual para tratamento de drives e emoções.

5.5 Conclusões

Neste capítulo apresenta-se uma abordagem geral sobre teorias de agentes e sistemas multi-agentes 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 alternativa para atender à complexidade do sistema.

Para o tratamento de comportamentos, o uso da pilha, associada à máquina de estados finitos (FSM), tem se apresentado como uma ótima solução ao tratamento de ações complexas. A limitação em duas posições mostra-se como melhor alternativa para o tratamento de ações intermediárias que surgem como forma de suprir pré-requisitos das ações terminais. O mesmo vale para ações resultantes do efeito que drives e emoções têm sobre os agentes.

Apesar de não terem sido incorporados na implementação, drives e emoções mostram um forte potencial para aumentar o grau de realismo das histórias, tanto a nível de geração como de visualização. Atualmente, o grupo de pesquisa do ICAD está estudando a possibilidade do tratamento de drives e emoções para a definição de comportamentos mais realistas. Este grupo pretende brevemente fazer testes reais e definir qual o nível de influência adequado que drives e emoções 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.