

Patrícia Rocha de Toro

Sistema de Controle Emocional Hedonista para Criaturas Artificiais

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientador: Ricardo Ribeiro Gudwin
Co-orientador: Mauro Sérgio Miskulin

Campinas, SP
Novembro / 2007

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

F149u Toro, Patrícia Rocha
Sistema de Controle Emocional Hedonista
para Criaturas Artificiais
Patrícia Rocha de Toro. – Campinas, SP:
[s.n.], 2007.

Orientadores: Ricardo Ribeiro Gudwin; Mauro Sérgio Miskulin.
Dissertação (mestrado) - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. Agente Emocional Hedonista. 2. Comportamento
Autônomo. 3. Emoções.
I. Gudwin, Ricardo Ribeiro. II. Miskulin, Mauro Sérgio.
III. Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. IV. Título

Título em Inglês: An Hedonistic Emotional Control System
for Artificial Creatures
Palavras-chave em Inglês: Hedonistic Emotional Agent, Autonomous Behavior,
Emotions
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca Examinadora: Ademar Ferreira, Fernando Antônio Campos Gomide,
Fernando José Von Zuben
Data da defesa: 12/11/2007

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidata: Patrícia Rocha de Toro

Data da Defesa: 12 de novembro de 2007

Título da Tese: "Sistema de Controle Emocional Hedonista para Criaturas Artificiais"

Prof. Dr. Ricardo Ribeiro Gudwin (Presidente):

Ricardo A. Gudwin

Prof. Dr. Ademar Ferreira:

Ademar Ferreira

Prof. Dr. Fernando Antônio Campos Gomide:

Fernando A. C. Gomide

Prof. Dr. Fernando José Von Zuben:

Fernando José Von Zuben

Patrícia Rocha de Toro

Sistema de Controle Emocional Hedonista para Criaturas Artificiais

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Banca Examinadora:

Prof. Dr. Ricardo Ribeiro Gudwin - UNICAMP

Prof. Dr. Ademar Ferreira - USP

Prof. Dr. Fernando Antônio Campos Gomide - UNICAMP

Prof. Dr. Fernando José Von Zuben - UNICAMP

Campinas, SP
Novembro / 2007

Resumo

O uso de emoções como metáfora computacional para o desenvolvimento de sistemas de controle para agentes autônomos tem recebido diversas contribuições na literatura recente. Nesta dissertação apresentamos uma arquitetura para um Agente Emocional Hedonista, inspirada parcialmente em diversas destas contribuições. Para tanto, desenvolvemos um controlador que confere comportamento autônomo a uma criatura artificial. Duas implementações foram realizadas: uma baseada em um algoritmo heurístico e a outra, em um algoritmo genético. Ambas as implementações consideram dois níveis de controle hierárquico: um nível de controle direto e um nível de decisão de metas. Desenvolveu-se um simulador para a criatura artificial onde diferentes controladores podem ser acoplados à criatura, por meio da rede, rodando como processos separados. Utilizou-se em ambos os casos “Medo”, “Fome” e “Curiosidade” como metáforas emocionais. O trabalho sinaliza, entre outros resultados, a adequação dessas metáforas como *design patterns* para a construção do comportamento do agente, sugerindo um novo paradigma de programação no desenvolvimento de agentes, que substitui uma programação orientada a casos de uso por uma programação orientada a propósitos.

Palavras-chave: Agente Emocional Hedonista, Comportamento Autônomo, Emoções.

Abstract

The use of emotions as a computational metaphor for the development of autonomous agents control systems has received many contributions in the recent literature. In this work, we present an architecture for an Hedonistic Emotional Agent, partially inspired in many of such contributions. For that, we developed a controller which confers autonomous behavior to an artificial creature. Two implementations were performed: the first based on the generation of an heuristic algorithm, and the second one using a genetic algorithm. Both of them considers two layers of control: a low level of direct control and a higher level of goal decision. We developed a simulator for the artificial creature where different controllers can be attached to the creature through the network, running as different processes. We used in both cases “Fear”, “Hunger” and “Curiosity” as emotional metaphors. The work signalizes, among others results, the adequacy of these metaphors as “design patterns” for the construction of agent behavior, suggesting a new programming paradigm in the development of agents, which exchanges a use-case oriented programming for a purpose-oriented new way of programming.

Keywords: Hedonistic Emotional Agent, Autonomous Behavior, Emotions.

Agradecimentos

Aos meus pais, Pedro e Doroti que ajudaram a construir os alicerces da minha formação como pessoa e como estudante, e que no período do mestrado apesar das dificuldades, não me permitiram desanimar e me apoiaram em todos os momentos.

Ao Rô, companheiro especial de minha vida.

Aos Professores Mauro Sérgio Miskulin, co-orientador deste trabalho, e Sérgio Lorenzato que muito me incentivaram nos passos iniciais desta jornada.

Aos meus tios, Edison e Doraci que carinhosamente jamais deixaram de acreditar em mim.

Aos meus colegas do departamento de Engenharia de Computação e Automação Industrial: Gustavo Pavani, Paulo Coelho, Rafael Pasquini, Fábio e Romarik, e em especial Luís Junqueira, Adler, Paulo Costa e Mábia que me ajudaram, apoiaram e colaboraram na elaboração deste trabalho.

Às minhas amigas Cynthia, Letícia, Jane, Monike, Adriane e Veruska pelo carinho e amizade que espero conservar para sempre.

À Maristela que além de facilitar a minha vinda para Campinas, ensinou-me novos sentidos para a palavra amizade.

Sinto-me grata também aos funcionários, Noêmia, Washington, Warley, Jerusa, Edson Vicente, Maria José, Soraia, Giane, Jaqueline e Carmen.

À CAPES, pelo apoio financeiro.

Aos Professores da banca de qualificação e defesa, Fernando Antônio Campos Gomide, Fernando José Von Zuben e Ademar Ferreira, por me ajudar a enxergar qualidades e limitações desse trabalho.

Agradeço especialmente ao Professor Ricardo Ribeiro Gudwin, orientador deste trabalho, a quem reconheço a seriedade, dedicação e competência, virtudes que espero carregar comigo em minha vida profissional.

E finalmente agradeço sobre tudo a Deus onde encontrei e encontro forças para superar as dificuldades e ânimo para sempre seguir adiante o meu caminho.

*Ao meu pai, homem guerreiro e vencedor.
A minha mãe, mulher iluminada.*

Sumário

| | |
|--|-------------|
| Lista de Figuras | xi |
| Lista de Tabelas | xiii |
| Lista de Acrônimos | xv |
| Trabalhos Publicados Pelo Autor | xvii |
| 1 Introdução | 1 |
| 1.1 Motivações | 3 |
| 1.2 Objetivos | 4 |
| 1.3 Organização da Dissertação | 5 |
| 2 Emoções em Sistemas Artificiais | 7 |
| 2.1 Prólogo | 7 |
| 2.2 Introdução | 7 |
| 2.3 Agentes Computacionais | 8 |
| 2.3.1 Agentes Reativos | 9 |
| 2.3.2 Agentes Deliberativos | 10 |
| 2.3.3 Agentes Emocionais | 10 |
| 2.3.4 Funções de Utilidade | 11 |
| 2.4 Agentes Emocionais na Literatura | 11 |
| 2.4.1 Modelo OCC - Ortony, Clore, Collins | 12 |
| 2.4.2 Computação Afetiva - Rosalind Picard | 13 |
| 2.4.3 Contribuição da Neurociência Cognitiva - Antonio Damásio | 14 |
| 2.4.4 Arquitetura baseada em Cognição e Emoção - Aaron Sloman | 16 |
| 2.4.5 Motivações e Emoções Fisiológicas - Lola Cañamero | 18 |
| 2.4.6 Mecanismos Neurais das Emoções - Juan Velásquez | 20 |
| 2.4.7 Agentes Emocionais: Reproduzindo Damásio - Rodrigo Ventura | 23 |
| 2.4.8 Agentes Emocionais em Ambiente de Tempo Real: Sarmiento | 24 |
| 2.4.9 Agentes Emocionais Reativos: Balkenius | 28 |
| 2.5 Emoções: Buscando um Consenso | 29 |
| 2.6 Agente Emocional Hedonista | 30 |
| 2.7 Criaturas Artificiais | 32 |

| | | |
|----------|---|-----------|
| 2.8 | Resumo do Capítulo | 33 |
| 3 | Simulação da Criatura Artificial | 35 |
| 3.1 | Prólogo | 35 |
| 3.2 | A Simulação de Criaturas e seus Objetivos | 35 |
| 3.3 | Descrição da Criatura | 39 |
| 3.3.1 | Sensor Visual | 39 |
| 3.3.2 | Sensores de Contato | 39 |
| 3.3.3 | Sensor de Energia | 40 |
| 3.3.4 | Atuadores de Posição do Sensor Visual | 40 |
| 3.3.5 | Atuadores de Movimentação da Criatura | 41 |
| 3.4 | Descrição do Ambiente | 41 |
| 3.4.1 | Orientação da Criatura Autônoma | 42 |
| 3.5 | Descrição do Simulador | 44 |
| 3.5.1 | Configuração do Ambiente | 46 |
| 3.5.2 | Configurações do Veículo | 49 |
| 3.5.3 | Acompanhamento dos Dados Sensório-Motores | 50 |
| 3.5.4 | Um Exemplo de Simulação | 51 |
| 3.6 | Sistema de Controle Autônomo da Criatura | 53 |
| 3.6.1 | Nível de Controle Direto | 53 |
| 3.6.2 | Nível Superior de Controle - Decisão de Metas | 56 |
| 3.7 | Resumo do Capítulo | 58 |
| 4 | Sistema de Controle Hedonista de Decisão de Metas | 59 |
| 4.1 | Prólogo | 59 |
| 4.2 | Controle Hedonista | 59 |
| 4.3 | Algoritmo Heurístico | 61 |
| 4.4 | Algoritmo Genético | 66 |
| 4.4.1 | Representação dos Indivíduos e Geração da População Inicial | 67 |
| 4.4.2 | Avaliação da População | 67 |
| 4.4.3 | Parada do Algoritmo | 70 |
| 4.4.4 | Operador genético: <i>Crossover</i> | 71 |
| 4.4.5 | Operador genético: <i>Mutação</i> | 71 |
| 4.4.6 | Seleção | 72 |
| 4.5 | Cenários de Aplicação dos AG's | 72 |
| 4.6 | Resultados | 75 |
| 4.6.1 | Algoritmo Heurístico | 75 |
| 4.6.2 | Algoritmo Genético | 77 |
| 4.6.3 | Comparação entre os Algoritmos | 79 |
| 4.7 | Discussão | 80 |
| 4.8 | Resumo do Capítulo | 87 |
| 5 | Conclusão e Trabalhos Futuros | 89 |

Referências bibliográficas

91

Lista de Figuras

| | | |
|------|---|----|
| 2.1 | Características típicas de agentes | 9 |
| 2.2 | Adaptada de Ortony et al. (1988) - modelo OCC | 13 |
| 2.3 | Adaptada de Sloman (2001) onde as setas representam fluxos de informações e interações entre os diversos elementos da arquitetura | 17 |
| 2.4 | Adaptada de Velásquez (1998) - <i>releasers: Basic Computational Units</i> (BCU) | 21 |
| 2.5 | Adaptada de Velásquez (1998) - <i>framework</i> de controle baseado em emoção | 21 |
| 2.6 | Adaptada de Ventura (2000) - a arquitetura proposta | 23 |
| 2.7 | Adaptada de Sarmento (2004) - ambiente <i>Pyrosim</i> e alguns agentes em ação | 24 |
| 2.8 | Elementos da arquitetura | 26 |
| 2.9 | Agente emocional hedonista | 31 |
| | | |
| 3.1 | Arquitetura da criatura artificial | 38 |
| 3.2 | Sensor visual | 39 |
| 3.3 | Sensores de contato | 40 |
| 3.4 | Sensor de energia em situação parcialmente carregado | 40 |
| 3.5 | Atuadores de posição do sensor visual | 41 |
| 3.6 | Sistema de coordenadas da criatura | 42 |
| 3.7 | Distância entre eixos e o ângulo das rodas | 43 |
| 3.8 | Caixa de diálogo do socket | 45 |
| 3.9 | Tela inicial do simulador | 46 |
| 3.10 | Caixa de diálogo para o tamanho do ambiente | 46 |
| 3.11 | Caixas de diálogo para a opção abrir e salvar a simulação | 47 |
| 3.12 | Add Object: escolha de materiais | 47 |
| 3.13 | Caixa de diálogo: tamanho do objeto | 48 |
| 3.14 | Definição das propriedades do objeto | 48 |
| 3.15 | Edit Car: edição da criatura | 49 |
| 3.16 | Drive Car | 50 |
| 3.17 | Estrutura sensório-motora da criatura | 51 |
| 3.18 | Um cenário possível visualizado na janela principal do simulador | 52 |
| 3.19 | O mesmo cenário, visualizado a partir do sistema de controle | 52 |
| 3.20 | Movimentação ilimitada da criatura | 53 |
| 3.21 | Exemplo de uma árvore de planos | 55 |
| | | |
| 4.1 | Visão do ambiente | 62 |

| | | |
|------|--|----|
| 4.2 | Representação do fator emocional curiosidade (em tom na cor cinza). Ocorre alto índice de curiosidade em lugares desconhecidos e baixo índice de medo. | 62 |
| 4.3 | Visão do ambiente | 63 |
| 4.4 | Fator emocional fome onde a prioridade é o abastecimento de energia | 63 |
| 4.5 | Algoritmo genético utilizado neste trabalho | 67 |
| 4.6 | Função de avaliação | 68 |
| 4.7 | Operador de crossover | 71 |
| 4.8 | Visão do ambiente | 73 |
| 4.9 | Representação do fator emocional curiosidade (em diferentes tons na cor cinza) . . . | 73 |
| 4.10 | Situação otimizada: criatura com alto grau de conhecimento sobre o ambiente | 74 |
| 4.11 | Fator emocional fome onde a prioridade é o abastecimento de energia | 74 |
| 4.12 | Distribuição no tempo do fator emocional fome - Heurístico | 75 |
| 4.13 | Distribuição no tempo do fator emocional medo - Heurístico | 76 |
| 4.14 | Distribuição no tempo do fator emocional curiosidade, nível de desconhecimento da criatura sobre o ambiente - Heurístico | 76 |
| 4.15 | Distribuição no tempo do fator emocional fome - AG | 77 |
| 4.16 | Distribuição no tempo do fator emocional medo - AG | 78 |
| 4.17 | Distribuição no tempo do fator emocional curiosidade, nível de desconhecimento da criatura sobre o ambiente - AG | 78 |
| 4.18 | Comparação de emoções: Sarmiento e a dissertação aqui apresentada | 84 |
| 4.19 | Adaptado de Balkenius (1995) - abordagens de comportamento dirigido a meta | 85 |

Lista de Tabelas

| | | |
|-----|---|----|
| 2.1 | Adaptado de Cañamero (2000) - fatores emocionais | 20 |
| 4.1 | Comparação entre os algoritmos | 79 |
| 4.2 | Tempo de execução do ciclo de decisão | 80 |
| 4.3 | Comparação de emoções: Balkenius e a dissertação aqui apresentada | 87 |

Lista de Acrônimos

| | |
|------------|--|
| AFP | - Attention Filter Penetration |
| AG | - Algoritmo Genético |
| BCA | - Camada Básica de Controle / Basic Control Agent |
| BDA | - Camada Deliberativa do Agente / Basic Deliberative Agent |
| BDI | - Beliefs, Desires, Intentions |
| DV | - Desirability Vector |
| OCC | - Modelo Ortony, Clore e Collins |

Trabalhos Publicados Pelo Autor

1. P.R. Toro, R.R. Gudwin, M.S. Miskulin. “Agentes Emocionais Hedonistas para Comportamento Autônomo”. *VIII SBAI - Simpósio Brasileiro de Automação Inteligente (SBAI'2007)*, Florianópolis, Santa Catarina, Brasil, Outubro 2007.

Capítulo 1

Introdução

O conceito de emoção sempre atraiu o interesse da comunidade acadêmica, principalmente nas áreas de psicologia e filosofia da mente. Associada a uma característica eminentemente humana, que dava ao ser humano (e também a outros animais) um caráter de distinção na evolução das espécies, a emoção rompeu seu lado humanístico e passou a servir de inspiração para pesquisas na área de computação. Observa-se um interesse crescente por pesquisas em sistemas que, de alguma forma, envolvam emoções, de forma a fazer com que os sistemas computacionais tornem-se mais próximos da maneira humana de atuar sobre o mundo. Dentre outras atividades, o uso de emoções em computação parece facilitar a execução de atividades voltadas à segurança (navegação autônoma de veículos de vigilância), aplicações médicas (engenharia biomédica), educacionais (sistemas tutoriais), situações de risco ao ser humano, etc.

Nossa motivação inicial, ao começarmos este trabalho, era compreender o fenômeno emocional e sua importância na determinação do comportamento inteligente. No desenrolar desta pesquisa, enquanto rastreávamos cruzamentos entre emoções e comportamentos com vistas à implementação computacional, deparamo-nos com uma série de autores de áreas e pensamentos muito distintos. Cada um deles buscando, a seu modo, responder uma pergunta essencial:

O que são emoções?

O que se entende por emoções está longe de ter uma interpretação consensual. Tomada em um sentido amplo, emoção pode ser vista como uma espécie de impulso neural que move o organismo para uma ação, sendo, portanto, um estado psico-fisiológico (BrasilEscola, 2007).

Em um primeiro momento, as noções de comportamento e de emoções já parecem de alguma forma associadas: a palavra **Emoção** provém do latim *emotionem* “movimento, comoção, ato de

mover” (Sensagent, 2003). Portanto, já vislumbra comportamento.

Recentemente inúmeros autores de áreas e de interesses distintos têm enriquecido o debate acerca das emoções e suas conexões com comportamentos. Algumas dessas idéias são aqui apresentadas. As mais relevantes para esta dissertação serão aprofundadas e retomadas mais adiante.

Gardner (1994) é um psicólogo americano mundialmente conhecido e autor da teoria das inteligências múltiplas. Segundo ele, o ser humano possui sete categorias de inteligência: musical, intrapessoal, interpessoal, corporal-cinestésica, espacial, linguística e lógico-matemática. Destas, a inteligência intrapessoal e a inteligência interpessoal estariam intimamente ligadas às emoções.

Um nome bastante conhecido na literatura recente em psicologia é Daniel Goleman cujo trabalho relacionado à Inteligência Emocional tornou-se bastante popular, ao final da década de 90. Goleman (1995) entende que emoção:

(..) se refere a um sentimento e seus pensamentos distintos, estados psicológicos e biológicos, e a uma gama de tendências para agir. Há centenas de emoções, juntamente com suas combinações, variações, mutações e matizes. Na verdade, existem mais sutilezas de emoções do que as palavras que temos para defini-las. (Goleman, 1995, p.303).

Se as emoções não podem ser definidas de forma fechada, isso não impede porém que tentemos associá-las a outros conceitos. Por exemplo, a idéia de associar racionalidade com emoção, segundo nossas leituras, parte de Damásio (1994), neurocientista, que desenvolveu novas teorias sobre inteligência e emoções bem como a necessidade de um corpo no processo. Damásio também procura diferenciar claramente emoção de sentimento. Segundo este autor, emoção é a modificação no corpo em função de estímulos externos, desempenhando um papel central na tomada de decisões; e sentimento surge em função da tomada de consciência das emoções.

A abordagem das emoções, segundo Damásio, sugere conexões diretas entre aspectos físicos e cognitivos e nos possibilita analogias interessantes para a implementação de emoções em sistemas inteligentes, o que se daria por meio de agentes computacionais.

O termo “agente” é amplamente utilizado na literatura, mas também não há um consenso total sobre seu significado. Isto porque, dependendo da aplicação, os atributos associados a esse conceito assumem diferentes níveis de importância. Agentes podem ser programas de computadores, processos computacionais, robôs ou outros tipos de dispositivos físicos, entidades autônomas em ambientes virtuais e até mesmo seres humanos (que interfaceiam com outros agentes em um sistema multi-agentes).

Segundo (Bradshaw (1997); Murch and Johnson (1999); Franklin and Graesser (1996); Wooldridge and Jennings (1995)), um agente autônomo é capaz de agir e sentir de acordo com seus propósitos em um ambiente, possuindo autonomia, pró-atividade, reatividade, aprendizado, ação propositada e adaptabilidade.

O elo que une o uso de emoções para o desenvolvimento de agentes, de acordo com o que entendemos, está exatamente em se dizer “de acordo com seus propósitos”, isto é, as ações de um agente devem estar relacionadas aos seus propósitos, e as emoções são a maneira por meio da qual o agente avalia se eles estão ou não sendo cumpridos. As emoções de um agente presidem uma codificação capaz de atender os propósitos geradores de seu comportamento.

Assim, o significado das emoções é múltiplo e complexo, influencia e é influenciado pela noção de comportamento e é dependente da área e dos interesses de quem formula tal significado. Ao mesmo tempo, a implementação de agentes computacionais à luz das emoções ganha novo impulso.

Colocadas estas idéias iniciais, procuramos, a seguir, desenvolver as motivações e objetivos desta dissertação.

1.1 Motivações

A emergência da chamada inteligência emocional tem revelado novos aspectos da inteligência humana. Emoções mostraram ter um grande impacto em muitas de nossas funções diárias, incluindo a tomada de decisão, planejamento, comunicação e comportamento. Estudar assunto tão rico e humano convida-nos a desbravar novos territórios do lado da Computação, nossa primeira e mais geral motivação.

Por outro lado, sabe-se que o processo emocional não é simples, depende da intensidade de cada emoção e da interação delas. Isso significa que programar emoções em computadores não se limita a empregar técnicas convencionais, pois isso representaria o risco de se valorizar mais os efeitos das emoções do que seus mecanismos causadores. Sem emoções, resolver certos problemas demandaria muito mais esforço e tempo, uma vez que o número de possibilidades de ação ou decisão a ser considerado seria muito mais elevado. Essa idéia de redução de complexidade ou de “encapsulamento”, tão comum na área de Engenharia, constitui nossa segunda motivação.

O estudo das emoções é também um “estado da arte” em sistemas inteligentes e quando estes estão voltados à interação com o ambiente, mecanismos de adaptação e aprendizado são imprescindíveis.

Uma premissa importante é caracterizar o que define um sistema como sendo inteligente. Em níveis mais elevados, inclui a capacidade de reconhecer objetos e eventos, representar conhecimentos e fazer planos para o futuro. O sistema inteligente deve também ser capaz de rever e corrigir seu comportamento, segundo avaliações do cumprimento ou não de suas metas parciais. Em outras palavras, o sistema ao interagir com o mundo, obtém informações por meio de processos perceptivos, construindo assim seu próprio conhecimento.

Muitos autores reconhecem que, associadas às emoções, existem componentes de valores e padrões éticos. Para uma abordagem dessa questão, recorreremos a (Stocker and Hegeman, 2002). Estes con-

sideram que as emoções são imprescindíveis para o raciocínio e o conhecimento; sendo importantes também para o auto aperfeiçoamento e o entendimento do que significa relacionar-se dentro de um processo.

Segundo Goleman (1995), o reconhecimento das emoções em nós mesmos ou nas outras pessoas, ajuda-nos a modular e civilizar o nosso próprio comportamento, daí a importância do estudo de avaliação das emoções.

Para Nussbaum (2001), as emoções estão relacionadas a um julgamento de valor, uma apreciação implícita realizada pelo organismo, com referência a um conjunto de propósitos inatos de um indivíduo. Em outras palavras, as ações de um agente devem estar relacionadas aos propósitos deste agente, e as emoções são a maneira por meio da qual o agente gera avaliações se esses propósitos estão sendo cumpridos.

Assim, se as emoções desempenham importante papel em nosso funcionamento inteligente cotidiano, é de grande interesse examinar o quanto as emoções podem ser aproveitadas em modelos artificiais com fins semelhantes.

Em inteligência artificial, vemos algo muito semelhante ao conceito de emoções nas assim chamadas funções de utilidade (Russell and Norvig, 1995), que podem ser utilizadas para orientar o comportamento de um agente. Poderíamos, portanto, comparar os efeitos das emoções sobre o comportamento de um indivíduo ao uso que se faz de funções de utilidade em agentes artificiais.

Observando e estudando as emoções humanas, e analisando-se seu propósito implícito, poderíamos utilizar as emoções como *design patterns* para o desenvolvimento de funções de utilidade: nossa principal motivação para este trabalho.

Para testar a viabilidade desta idéia, partindo dos pressupostos até aqui colocados, construímos um *framework* de simulação onde desenvolvemos um estudo de caso por meio do qual procuramos observar de maneira direta os fenômenos que motivaram o nosso interesse.

1.2 Objetivos

Apresentamos nesta dissertação uma arquitetura voltada ao controle de comportamento autônomo em uma criatura artificial, utilizando o conceito de emoções.

Partimos de diversas hipóteses, algumas delas já comentadas. Outras, serão discutidas em detalhes mais adiante neste trabalho:

- Emoções são facilitadoras tanto de ações quanto de decisões, mas sua implementação computacional representa um grande desafio;
- Agentes computacionais são capazes de aprender com a experiência, reagir ao ambiente, ter autonomia, adaptabilidade, pró-atividade e ação propositada (Wooldridge and Jennings, 1995);

- Metáforas emocionais em agentes podem representar um caminho para a implementação de comportamento inteligente;
- Tais metáforas podem servir como *design patterns* para o desenvolvimento de algoritmos de controle para criaturas artificiais, condicionando comportamentos razoavelmente sofisticados;
- Utilizar o conceito de emoções para criar um novo paradigma de projeto de sistemas inteligentes.

A arquitetura que procuramos desenvolver volta-se para o controle do comportamento autônomo de uma criatura artificial, tanto no aspecto da tomada de decisões quanto nas ações que ela deve empreender. Um repertório básico de metáforas emocionais (**Medo, Fome e Curiosidade**) é proposto a fim de sustentar esse comportamento. As emoções são aqui entendidas como estados internos que avaliam propósitos do agente e cujas funções contribuem para a tomada de decisão e ação.

No experimento aqui apresentado, a criatura deve ser capaz de manter a sua integridade física, cuidar da manutenção dos seus níveis internos de energia e explorar territórios desconhecidos.

Ao codificarmos cada possível propósito da criatura em uma emoção distinta, geramos funções de utilidade que medem o quanto as ações (ou planos) do agente contemplam os propósitos em questão. O agente aqui desenvolvido, com estas características, foi denominado **Agente Emocional Hedonista**.

Programar um comportamento autônomo por meio de um agente emocional hedonista implica decidir quais emoções o agente deve instanciar, e definir de que forma essas emoções (na forma de funções de utilidade) podem ser calculadas.

Duas implementações foram realizadas: a primeira tendo como base um algoritmo heurístico e a outra utilizando um algoritmo genético. Ambas as implementações foram analisadas e discutidas, compondo o cerne desta dissertação.

1.3 Organização da Dissertação

A dissertação está dividida em seis capítulos:

No Capítulo 2: “Emoções em Sistemas Artificiais”, apresentamos um apanhado da literatura sobre as várias propostas para o uso de emoções em sistemas computacionais: visão da ciência cognitiva; neurociência; emoções em máquinas e criaturas artificiais. Além disso, apresentamos a conceitualização básica do agente emocional hedonista, na qual abordamos o significado de hedonismo e a sua justificativa na implementação do agente.

No Capítulo 3: “Simulação da Criatura Artificial”, apresentamos os detalhes da aplicação desenvolvida para a navegação de uma criatura artificial. Fazemos um histórico e uma descrição do problema, analisando os principais detalhes sobre o simulador, suas etapas, interface gráfica e, por fim, apresentamos a estrutura geral do sistema de controle a ser desenvolvido.

No Capítulo 4: “Sistema de Controle Hedonista de Decisão de Metas”, apresentamos a base completa desse modelo, fazendo uma descrição elaborada do problema e desenvolvendo uma proposta de solução, baseada em dois algoritmos. Apresentamos, então, os resultados dos experimentos de simulação realizados e, por fim, a elaboração de uma discussão conectando o trabalho desenvolvido com outros trabalhos.

No Capítulo 5: “Conclusão e Trabalhos Futuros”, analisam-se as principais contribuições e a indicação de trabalhos futuros.

No Capítulo 6: “Referências Bibliográficas”, contém as referências utilizadas.

Capítulo 2

Emoções em Sistemas Artificiais

2.1 Prólogo

Neste Capítulo, apresentamos uma visão geral de utilização das emoções em sistemas artificiais e a nossa proposta sobre o Agente Emocional Hedonista.

2.2 Introdução

A área de pesquisa em **Agentes Autônomos Inteligentes** abriga uma abstração poderosa para diversos tipos de aplicações práticas, desde robótica móvel até jogos de computador. Um dos problemas recorrentes neste contexto é o problema da navegação autônoma em ambientes complexos. Este problema surge tanto no caso de robôs móveis que precisam decidir uma trajetória desde um ponto inicial até uma meta sem colidir com obstáculos (eventualmente minimizando a distância percorrida e/ou o tempo de percurso), como no desenvolvimento de oponentes inteligentes em jogos de computador. Neste caso, sistemas de controle inteligente devem decidir as ações de um agente para proporcionar um bom entretenimento ao usuário do sistema, simulando com realismo o comportamento de um oponente humano.

Diversas abordagens são possíveis para esse problema, dependendo de que aspecto se deseja enfatizar. Uma possível forma de sistematizar o problema é considerar uma criatura artificial que existe em um certo ambiente, movimentando-se sobre esse ambiente e realizando ações sobre ele (Balkenius, 1995).

Soluções clássicas para esse problema são encontradas em profusão dentro da Inteligência Artificial (Russell and Norvig, 1995) e envolvem normalmente máquinas de estados, algoritmos de busca e eventualmente programação em lógica. Algoritmos mais avançados podem utilizar redes neurais, lógica *fuzzy* ou computação evolutiva.

Existem na literatura diversas tentativas de se apropriar do conceito de **emoção**, trazido da psicologia cognitiva e da filosofia, de forma a orientar a criação de soluções para problemas de navegação autônoma, computação gráfica, sistemas inteligentes entre outros. Exemplos incluem (Blumberg, 1996; Reilly, 1996; Cañamero, 1997, 1998, 2000, 2003; Ventura, 2000; Almeida, Silva and Bazzan (2004); Sarmiento, Moura and Oliveira (2004); Septseault and Nédélec, 2005; Budakova and Dakovski, 2006; Meyer, 2006; Levine, 2007), além de outros que serão detalhados mais à frente.

Uma abordagem canônica é a de Ortony, Clore and Collins (1998) que desenvolveu um modelo de estruturação de emoções sobre o ponto de vista da Ciência Cognitiva diretamente aplicável à inteligência artificial. Da mesma forma, Sloman (1998, 2001) destaca algumas propriedades funcionais das emoções e Picard (1997) desenvolve toda uma área de estudos, chamada de “computação afetiva” envolvendo o relacionamento entre emoções e computadores. Esse relacionamento entre emoções e computadores pode ser explorado em diversos tipos diferentes de aplicações. Uma das mais utilizadas, entretanto, envolve o uso de agentes computacionais incorporando, de alguma forma, algum modelo de emoção.

2.3 Agentes Computacionais

O termo “agente” é amplamente utilizado na literatura, mas não há um consenso total sobre seu significado. Isto porque dependendo da aplicação, os atributos associados a esse conceito assumem diferentes níveis de importância. Agentes podem ser programas de computadores, processos computacionais, robôs ou outros tipos de dispositivos físicos, entidades autônomas em ambientes virtuais e até mesmo seres humanos (que interfaceiam com outros agentes em um sistema multi-agentes).

Apesar de ser um termo de escopo amplo, o conceito de agente é bastante útil, e vem sendo utilizado de maneira crescente na literatura. Um agente, segundo diversos autores (Wooldridge and Jennings, 1995; Franklin and Graesser, 1996; Bradshaw, 1997 e Murch and Johnson, 1999), deve possuir, pelo menos, as seguintes características (Figura 2.1):

- **Autonomia:** o agente não depende de intervenção humana e possui certo controle sobre suas ações e estados internos;
- **Pró-atividade:** o agente não apenas reage em resposta ao ambiente, mas é capaz de exibir um comportamento baseado em objetivos, tomando iniciativas;
- **Reatividade:** o agente percebe o ambiente no qual está inserido e responde prontamente a mudanças que nele ocorre;
- **Aprendizado:** o agente acumula experiência no sentido de tomar decisões baseando-se em eventos passados;

- **Ação propositada:** a ação do agente é orientada a um propósito;
- **Adaptabilidade:** o agente se adapta às modificações no ambiente.

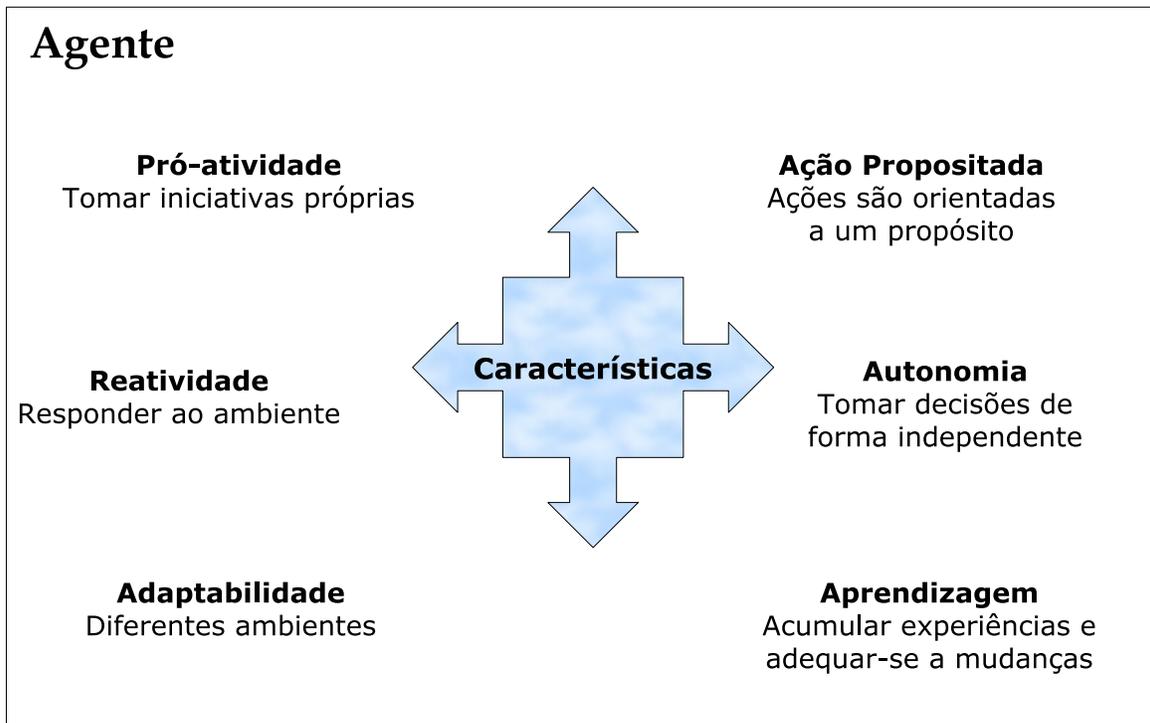


Figura 2.1: Características típicas de agentes

Agentes podem ser classificados segundo diversas tipologias (El-Nasr, Yen and Ioerger (2000); Septseault and Nédélec, 2005; Meyer, 2006; Budakova and Dakovski, 2006; Adam, Gaudou, Herzig and Longin (2006)), de acordo com os mais diferentes critérios.

Uma distinção importante que aparece na literatura é aquela entre agentes reativos e deliberativos.

2.3.1 Agentes Reativos

Agentes reativos (algumas vezes também chamados de reflexivos) são baseados no modelo ação-reação: o agente se comporta simplesmente reagindo aos estados do ambiente detectados pelos seus sensores. Nesta categoria, os agentes não elaboram planos, nem possuem explicitamente metas a serem alcançadas. Os agentes simplesmente reagem às entradas sensoriais, aplicando algum tipo de função às entradas sensoriais para gerar as saídas de controle. A função que determina esse comportamento pode ser qualquer. Alguns exemplos de funções incluem sistemas baseados em regras (clássicas ou *fuzzy*), redes neurais ou algum outro tipo de técnica de inteligência computacional. Apesar de sua simplicidade, comportamentos sofisticados podem ser gerados a partir de agentes reativos.

Normalmente esse comportamento é um comportamento emergente, gerado a partir da função de comportamento. Um agente reativo não possui uma representação explícita do conhecimento, nem a representação do ambiente. Como suas reações dependem unicamente de sua percepção por meio dos sensores, não é capaz de acumular experiências, nem ter memória de suas ações passadas ou planejar ações futuras (Junior and Costa, 2007).

2.3.2 Agentes Deliberativos

Agentes deliberativos, ao contrário dos agentes reativos, que simplesmente reagem aos dados sensoriais colhidos do ambiente, são agentes que planejam antes de executar ações. Para que esse planejamento seja possível, normalmente se utiliza algum modelo interno ou mapa do ambiente. Os dados obtidos por meio de sensoriamento são utilizados para estimar as condições atuais do ambiente, considera-se um estado futuro como meta e algum algoritmo de planejamento é utilizado para determinar quais as ações necessárias para levar do estado atual até o estado final desejado. Diferentes algoritmos planejadores podem ser utilizados, sendo que existe toda uma área de pesquisa somente em algoritmos de planejamento.

Uma classe bastante estudada de agentes deliberativos é a dos assim chamados agentes BDI (Koch, Meyer and Dignum (2006)). Em agentes BDI, três tipos de estados mentais são considerados: *Beliefs*, *Desires* e *Intentions* (Crenças, Desejos e Intenções), onde crenças são asserções construídas a partir de informações colhidas do ambiente, desejos são os objetivos e intenções são os planos.

2.3.3 Agentes Emocionais

Uma classe de agentes que vem sendo cuidadosamente estudada na literatura e que nos é particularmente interessante é a dos agentes emocionais, ou agentes dotados de emoção. Agentes emocionais podem tanto ser reativos como deliberativos. O conceito de emoção tem sua origem na psicologia cognitiva e na filosofia, sendo normalmente aplicado a seres humanos e eventualmente a animais. O conceito de emoção em agentes artificiais é apenas uma abstração deste conceito de emoção, estendido para ser aplicado a entidades artificiais que, de alguma forma, exibem propriedades similares ou próximas daquelas encontradas em seres vivos que possuem emoções. Em nenhum momento estamos querendo dizer que as emoções que encontramos em agentes artificiais são as mesmas que ocorrem em seres vivos, mas tão somente uma abstração útil deste conceito. Agentes emocionais possuem internamente um conjunto de estados chamados de estados emocionais, que são utilizados para diversas finalidades distintas. Por exemplo, podem ser utilizados para expressar emoções (ou simular emoções), em agentes corporificados (*embodied agents*), alterando fatores físicos do corpo normalmente associados a emoções em seres humanos. Esse tipo de reação em agentes pode ser bastante

interessante quando os agentes devem interagir de alguma forma com seres humanos, de tal forma que os agentes passem a ilusão de serem seres vivos e dessa forma proporcionar maior conforto na interação entre seres humanos e agentes. Agentes emocionais podem ser capazes, de alguma forma, de detectar reações emocionais de seres humanos e se utilizar dessa informação para conduzir algum tipo especial de interação com estes seres humanos. Mas, além desses exemplos, e o que mais nos interessa em particular neste trabalho, agentes emocionais podem se utilizar do conceito de emoções para agir de maneira inteligente, em sua interação com seu ambiente. Essa relação entre emoção e inteligência, será explorada em seções mais a frente.

2.3.4 Funções de Utilidade

Um conceito que nos é particularmente interessante quando pensamos em estudar agentes emocionais é o conceito de função de utilidade (Russell and Norvig, 1995). Segundo Russell e Norvig, “uma *função de utilidade* mapeia um estado (ou uma sequência de estados) em um número que descreve o grau de felicidade associado ... como *felicidade* não soa muito científico, a terminologia habitual é dizer que, se um estado do mundo for preferido em detrimento de outro, ele terá maior utilidade para o agente”. Observamos que, a despeito do preconceito de Russell e Norvig com o termo “felicidade”, os conceitos de emoção e de função de utilidade, até certo ponto se assemelham bastante. O conceito de função de utilidade é utilizado por muitos autores para criar algoritmos que determinam o comportamento de um agente. Poderíamos comparar os efeitos das emoções sobre o comportamento de um indivíduo, ao uso que se faz de funções de utilidade em agentes artificiais. Gerar funções de utilidade não é uma tarefa fácil quando se leva em consideração múltiplos objetivos. Neste sentido, utilizar metáforas emocionais como fonte de inspiração poderia simplificar o desenvolvimento de funções de utilidade, que uma vez implementadas, poderiam ser utilizadas para gerar o comportamento autônomo em agentes.

Observando e estudando as emoções humanas, e analisando seu propósito implícito, poderíamos utilizar as emoções como *design patterns* para o desenvolvimento de funções de utilidade. Nossa proposta neste trabalho parte portanto dessa premissa. A idéia é codificar cada possível propósito que se almeja para um agente em uma emoção distinta, gerando funções de utilidade que governem o desempenho do agente.

2.4 Agentes Emocionais na Literatura

Apresentamos a seguir, uma revisão de alguns dos trabalhos mais significativos encontrados na literatura, envolvendo diferentes abordagens do conceito de emoção em máquinas.

2.4.1 Modelo OCC - Ortony, Clore, Collins

Esse modelo foi desenvolvido em 1988 por Andrew Ortony, professor de psicologia, educação e ciência da computação na Universidade de Northwestern (EUA), Gerald L. Clore, professor de psicologia da Universidade de Virginia (EUA) e Allan M. Collins, professor de educação e política social da Universidade Northwestern. É chamado de OCC devido às letras iniciais dos sobrenomes dos autores.

Esse modelo, considerado clássico por alguns, tem influenciado uma vasta gama de trabalhos na área computacional. Inspira-se na Ciência Cognitiva e propõe um dos primeiros modelos de estruturação de emoções, contribuindo para a compreensão de como as pessoas experimentam emoções e sob quais condições isso ocorre.

Para os autores, emoções são “reações valoradas a eventos, agentes ou objetos”, que podem ter diferentes intensidades. Esta definição incorpora duas características que os autores consideram fundamentais na compreensão do fenômeno da emoção: a idéia de que emoções são “despertadas” por situações experienciadas e a idéia de que promovem uma avaliação, ou apreciação sobre a causa desse “despertar”. Essa apreciação envolve diretamente um propósito, o qual motiva o sujeito que experimenta a emoção.

O modelo OCC pode ser estruturado na forma de regras e considera 22 emoções divididas em aspectos positivos e negativos, cujos potenciais ativadores são os seguintes fatores: Eventos, Agentes e Objetos (Figura 2.2).

- **Eventos** e suas conseqüências: fator capaz de estimular (ou não), objetivos e preferências (carregam portanto já uma noção de “desejabilidade”);
- **Agentes** e suas ações: avaliadas em conformidade a normas sociais e comportamentais;
- **Objetos** e suas propriedades: as emoções surgem das percepções sobre objetos inanimados, que são avaliados como atrativos ou não. (Ortony et al., 1998; Ortony, 2003).

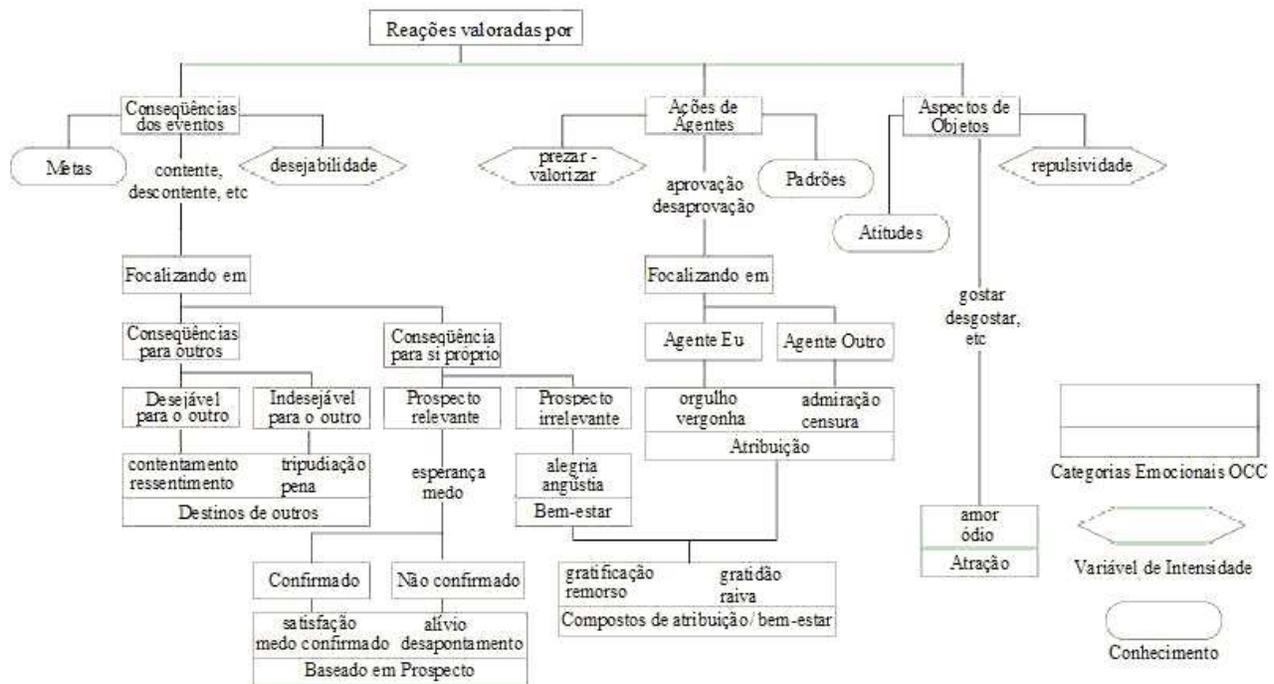


Figura 2.2: Adaptada de Ortony et al. (1988) - modelo OCC

Segundo os autores, este modelo visa o reconhecimento de emoções associadas a cognições humanas, mais do que propriamente a implementação de emoções em máquinas. Apesar disso, diversos autores utilizam o modelo OCC como ponto de partida para implementações de emoções em máquinas, como veremos mais a frente. Por exemplo, Picard (1997), aproveita o modelo OCC para formular um mecanismo de síntese de emoções.

2.4.2 Computação Afetiva - Rosalind Picard

A pesquisadora Rosalind W. Picard do Instituto de Tecnologia de Massachusetts (MIT) / EUA desenvolve projetos em uma área que denominou de “computação afetiva”, onde busca relacionar emoções e computadores.

A computação afetiva busca por um lado, reconhecer emoções humanas ou expressar emoções por máquinas na interação homem-computador e, por outro, sintetizar emoções em máquinas a fim de investigar mais sobre as emoções humanas e também construir robôs que pareçam mais reais (Jaques and Vicari, 2005).

Para Picard (1997; 2006), emoções são capazes não só de influenciar o aprendizado e a tomada de decisão, mas de atuar sobre a recuperação de memória e ainda em processos cognitivos que conectam conceitos, idéias, planos e todas as experiências armazenadas (interações corpo-mente). Portanto,

os sistemas construídos por Picard são capazes de reagir tanto “racionalmente” quanto “emocionalmente”, dependendo do estado emocional do usuário.

O estudo da computação afetiva pode ser dividido entre três grandes áreas:

- i. reconhecimento de emoções;
- ii. expressão das emoções;
- iii. influência das emoções sobre as ações.

No reconhecimento de emoções, busca-se a detecção de emoções humanas por meio do sensoriamento dos fatores físicos que o ser humano emite quando está emocionado. Esse reconhecimento propicia ao sistema agir de modos diferentes dependendo do estado emocional do ser humano com o qual interage.

Na expressão de emoções, estudam-se os mecanismos necessários para que sistemas artificiais pareçam ter emoções aos olhos de um ser humano. Busca-se reproduzir em avatares digitais ou robôs os fatores físicos que usualmente estão relacionados à expressão de emoções no ser humano.

Na influência das emoções sobre as ações, estuda-se como emoções e inteligência estão relacionados entre si, e como aparentemente a emoção é utilizada para a tomada de decisões e como estas afetam as ações executadas.

Picard sugere que é possível avaliar o desempenho emocional de um sistema. Para tanto, sugere a aferição de uma série de parâmetros:

- Comportamento emocional: *Parece ter emoções?*
- Emoções primárias rápidas: *Responde rapidamente a estímulos específicos?*
- Emoções geradas através da cognição: *Interpreta a situação e a avalia?*
- Experiências emocionais: *Diferentes sentimentos se associam a diferentes emoções?*
- Interações corpo-mente: *Emoções influenciam o comportamento e vice-versa?*

2.4.3 Contribuição da Neurociência Cognitiva - Antonio Damásio

Uma das importantes contribuições ao estudo das emoções e sua possível síntese em máquinas vem da área da neurobiologia do cérebro humano, principalmente a partir das contribuições de Antonio Damásio.

Antonio Damásio é um neurologista e neurocientista português, atuando no Departamento de Neurologia da Universidade de Iowa (EUA) onde estuda as bases cerebrais associadas a emoções e sentimentos.

Damásio utiliza conceitos da Biologia, Psicologia e Filosofia para desenvolver todo um conjunto de novas teorias sobre inteligência e comportamento humano.

Para Damásio, o corpo é a gênese do pensamento. Ao examinar como os processos fisiológicos contribuem para o funcionamento da mente, o autor desenvolve a idéia de que o ato de pensar é inerente a um corpo.

Encarar emoções como parte de processos fisiológicos na perspectiva de Damásio tem influenciado enormemente a comunidade de pesquisadores, entre eles Sloman, Picard, Cañamero, Ventura, Velásquez e muitos outros.

Para Damásio, emoções compõem um conjunto complexo de reações químicas e neurais, formando um padrão. São desencadeadas após um processo mental de avaliação que produz um efeito de filtro na proporção e intensidade dos estímulos recebidos.

Para o autor, emoções desempenham papel fundamental na cognição pois são capazes de:

- i. influir fortemente na tomada de decisões e na própria inteligência, e,
- ii. estarem voltadas a um propósito.

Em suas próprias palavras:

Emoção é a combinação de um processo avaliatório mental, simples ou complexo, com respostas dispositivas a esse processo, em sua maioria dirigidas ao corpo propriamente dito, resultando num estado emocional do corpo, mas também dirigidas ao próprio cérebro, resultando em alterações mentais adicionais. (Damásio, 1994, p.169).

Damásio propõe a divisão das emoções em duas categorias: emoções **primárias** e emoções **secundárias**.

As **emoções primárias** constituem os processos básicos. Com elas, estamos programados para reagir de modo pré-organizado quando certas características dos estímulos no mundo ou nos nossos corpos são detectadas. Entre elas se incluem o medo, a alegria, a tristeza, a raiva. Lesões do sistema límbico cerebral limitam o processamento das emoções primárias.

As **emoções secundárias** são mais complexas. Demandam maior processo de pensamento, influem no estilo e eficiência dos processos cognitivos, mas dependem das emoções primárias para que funcionem. Entre elas se incluem o ciúme, a culpa, o orgulho. Lesões nos córtices pré-frontais do cérebro limitam o processamento das emoções secundárias.

É importante ressaltar que Damásio diferencia **emoções** de **sentimentos**.

Emoção implica modificações no corpo em função de estímulos externos, enquanto sentimento é um processo mental, com origem na tomada de consciência das emoções. Enquanto todas as emoções originam sentimentos, nem todos os sentimentos provêm de emoções.

A abordagem biológica das emoções, como sinalizado pelo trabalho de Damásio (1994; 1999; 2003), sugere conexões diretas com aspectos físicos e cognitivos e permite analogias interessantes para a implementação de emoções em sistemas artificiais.

2.4.4 Arquitetura baseada em Cognição e Emoção - Aaron Sloman

Aaron Sloman é um filósofo e pesquisador de Inteligência Artificial e Ciência Cognitiva, trabalhando atualmente na Universidade de Birmingham, Inglaterra, onde lidera o *Cognition and Affect Group (CogAff)*.

Sloman (1998; 2001) propõe uma arquitetura hierárquica: *Attention Filter Penetration (AFP)*, em três camadas, com ênfase em uma descrição cognitiva baseada em propriedades funcionais das emoções.

A arquitetura AFP objetiva suportar a implementação de agentes com diferentes tipos e níveis de complexidade, em termos de controle e de processamento de informação. Para esse autor, estados emocionais envolvem a capacidade de mudar prioridades do agente sem implicar necessariamente na interrupção dos objetivos atuais.

A configuração proposta por Sloman aproveita a idéia tradicional das “três torres” (ver Figura 2.3), isto é, prevê um subsistema de **Percepção**, responsável por extrair dados do ambiente, um subsistema de **Ação** que possibilita que o agente interaja com o ambiente e, finalmente, um subsistema **Central** que faz a mediação entre a percepção e a ação e preside o controle de ambas.

Uma divisão horizontal das “três torres” em três camadas cognitivas compõe um cruzamento que define a arquitetura proposta pelo autor.

Tais camadas cognitivas são associadas a diferentes tipos de processamento, operando de forma simultânea e influenciando-se mutuamente (Figura 2.3).

Para Sloman, emoções funcionam como alarmes, que podem exercer influência nas três camadas cognitivas que integram o Processamento Central. Ele divide as emoções em três grupos: primária, secundária (como sugerido por Damásio, 1994) e acrescenta um terceiro grupo, o das emoções terciárias.

As emoções primárias são alarmes que influenciam as camadas Reativa e Deliberativa, já as emoções secundárias exercem influência somente na camada Deliberativa. Para Sloman, as emoções terciárias exercem influência na camada de Meta-Gerenciamento e aparentemente são exclusividade dos seres humanos.

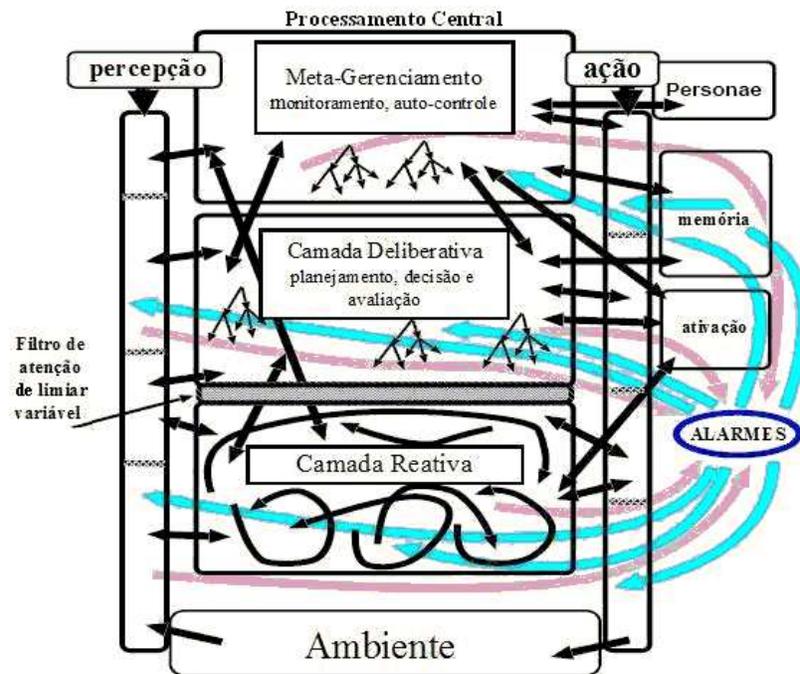


Figura 2.3: Adaptada de Sloman (2001) onde as setas representam fluxos de informações e interações entre os diversos elementos da arquitetura

a) Camada Reativa

Nesta camada ocorrem as respostas automáticas do sistema cognitivo. Não há avaliação cognitiva prévia, dependendo-se apenas de que as condições apropriadas sejam satisfeitas. Portanto, nessa camada não ocorre aprendizado e nela o sistema é incapaz de entender ou avaliar planos.

b) Camada Deliberativa

Nesta camada há uma correspondência com as “emoções secundárias” de Damásio (1994). As emoções são geradas através de processos cognitivos, os quais requerem algum tipo de raciocínio a respeito de metas, situações, objetos e eventos. Nesta camada o sistema é capaz de planejar, avaliar opções, tomar decisões e alocar recursos. Uma característica importante é a antecipação e explicação causal de acontecimentos, isto é, raciocínio sobre o passado, presente e futuro.

c) Camada de Meta-Gerenciamento

A terceira camada possui capacidades de meta-gestão como a monitoração, avaliação e controle dos processos internos. Esta camada impede que certas metas interfiram umas nas outras; procura formas mais eficientes de operar sobre a camada deliberativa, bem como refinar escolhas de estratégias e alocação de recursos. Considera também que as emoções possuem intensidade e tem certo caráter social levando em conta como as pessoas avaliam a si mesmas e às demais, por exemplo: noções de

sofrimento, vergonha, humilhação e perturbação (Sarmiento, 2004).

Parte importante da arquitetura refere-se à alocação de recursos segundo a atividade cognitiva em curso. Dado que os recursos são limitados, vem a idéia da necessidade de arbitrar interrupções por meio de filtros. A variação do limiar do filtro possibilita que a proteção da atividade cognitiva seja ajustada de acordo com o contexto.

Para apoiar essa idéia, Sloman implementa mecanismos de Alarmes os quais recebem informação de cada componente do sistema e, com base nisso, ativa mudanças de estratégias internas no processamento central do agente ver Figura 2.3 (setas claras).

Esta arquitetura AFP possui uma analogia com a evolução das espécies biológicas: a camada reativa estaria associada ao comportamento de criaturas primitivas; a camada deliberativa associada ao comportamento animal de forma geral, e, a camada de meta-gerenciamento associada a comportamentos complexos mais próximos do ser humano.

A idéia de emoções na perspectiva de Sloman se desenvolve segundo necessidades da arquitetura (interrupção e troca de prioridades), portanto mais voltado ao papel cognitivo das emoções e menos focado nas propriedades funcionais das emoções como outros autores. Sloman não propõe um modelo explícito para mecanismos emocionais, como faz Cañamero.

2.4.5 Motivações e Emoções Fisiológicas - Lola Cañamero

Professora de Ciência da Computação na Universidade de Hertfordshire (Inglaterra), Lola Cañamero busca uma fisiologia sintética inspirada em fenômenos emocionais, bioquímicos e homeostáticos que caracterizam os organismos biológicos.

Cañamero (1997, 1998, 2000) construiu um simulador para agente autônomo baseado na idéia de que emoções surgem como resultado de mudanças fisiológicas. Tal experiência representa um exemplo notável de ambiente para o estudo de seleção de ações e mecanismos emocionais.

A arquitetura *Abbott/Gridland* contempla criaturas artificiais implementadas por meio de agentes que habitam um mundo virtual.

Existem no ambiente *Gridland* três elementos: **criaturas** (*Abbotts* e *Enemies*, estes devoradores dos *Abbotts* e ainda competem pelos mesmos recursos que os primeiros), **recursos** (comida e água, quando uma fonte se esgota reaparece aleatoriamente) e **obstáculos** (variam de tamanho e forma).

São características do ambiente: bi-dimensionalidade, dinamismo (objetos mudam de posição), comportar certo nível de incerteza, prever interações sociais (colaboração e competição) e oferecer riscos internos e externos.

Ameaças aos *Abbotts* podem corresponder tanto a fatores internos ligados a sua sobrevivência (condições fisiológicas) quanto a fatores externos (*Enemies* em ação e ou *Abbotts* irritados).

Enquanto as criaturas *Enemies* possuem comportamento simples com mecanismo pré-definido de decisão e não carregam emoções, as criaturas *Abbotts* correspondem à implementação do modelo de agentes dotados de emoções.

Cada *Abbott* possui três sensores: **sensor somático** que provê informação sobre sua fisiologia interna; **sensor táctil** para detecção de propriedades dos objetos ao redor da criatura e **sensor visual** capaz de prover informação sobre brilho e distância dos objetos vizinhos. Quanto aos atuadores, cada *Abbott* possui: mão, pé e boca que possibilitam comportamentos como comer, caminhar, desviar de obstáculo, brincar, atacar, procurar, entre outros. A dor aparece quando há conflitos entre comportamento pretendido e restrições do ambiente e também quando a criatura é atacada pelo inimigo.

Uma série de variáveis fisiológicas são utilizadas para definir o estado corporal de um agente (adrenalina, dopamina, frequência cardíaca, temperatura, entre outras) que ativam motivações, que por sua vez são associadas a *drives*.

A autora desenvolve conceitos de **motivações**, **drives** e **emoções** e os utiliza como base para selecionar e coordenar o comportamento de um agente:

- **Motivações:** são processos homeostáticos¹ relacionados a variáveis fisiológicas como fome, sede, curiosidade, frio, fadiga, auto-preservação, etc. Em cada momento a motivação mais intensa tentará ativar comportamentos que satisfaçam necessidades mais urgentes. Tal execução, pode afetar além do ambiente, a própria fisiologia do agente;
- **Drives:** impulsionam a ação que garante a homeostase das variáveis fisiológicas associadas às motivações. Exemplo: a motivação Frio associa-se ao *drive* Aumentar temperatura; a motivação Fome associa-se ao *drive* Aumentar açúcar no sangue; a motivação Curiosidade associa-se ao *drive* Aumentar Endorfina;
- **Emoções:** trabalham paralelamente ao sistema motivacional fornecendo suporte à interação agente/ambiente e ajudando assim a selecionar comportamentos mais adequados (Tabela 2.1). Possuem as seguintes características:
 - i. são amplificadores de motivações, isto é, tornam um processo homeostático mais “intenso” em situações especiais, fazendo com que ele se destaque dos demais;
 - ii. mecanismo que proporciona uma resolução do comportamento final do agente, diante de diversas motivações concorrentes; e
 - iii. são “disparadas” a partir de um evento específico que ocorre no ambiente (externo ou interno).

¹Necessidade de operar em equilíbrio funcional, pois quando ocorre o desequilíbrio o sistema pode entrar em colapso.

| Fator Emocional | Objetivo | Condição para o Disparo |
|------------------------|---|---|
| Raiva | Mecanismo para bloquear influências do ambiente | Objetivo é fracassado |
| Tédio | Mecanismo para interromper comportamento repetitivo e ineficiente | Comportamento prolongadamente ineficiente |
| Medo | Funciona como um mecanismo de defesa | Presença de inimigos |
| Felicidade | Buscar um re-equilíbrio interno | Objetivo alcançado |
| Interesse | Estimular novas interações | Presença de um novo objeto |
| Tristeza | Efeito de “espera” por mudanças no ambiente ou estado interno | Incapacidade de atingir um objetivo |

Tabela 2.1: Adaptado de Cañamero (2000) - fatores emocionais

A Tabela acima evidencia que cada emoção é afetada pelas variáveis fisiológicas de controle (entre elas os hormônios sintéticos) ao mesmo tempo em que exerce influência indireta na seleção de comportamentos.

Em Cañamero (2001) a autora se pergunta como definir agentes autônomos utilizando emoções. Para isso, relata as principais abordagens até então e justifica com o argumento de que agentes autônomos fornecem um bom *framework* para estudar a relação entre emoções e adaptação.

Interessante declaração é dada pela mesma autora em outro artigo:

... o que a modelagem de emoções em robôs autônomos podem fazer no sentido de compreensão das emoções humanas “como atividade cerebral” e como podem ser empregadas em nossas interações com o ambiente. Tais contribuições são associadas, por um lado, ao uso potencial de modelos robóticos como ferramentas e “laboratórios virtuais” para testar e explorar sistematicamente teorias e modelos de emoções humanas e, ainda, por outro lado, para modelar abordagens que ajudem a esclarecer e operacionalizar aspectos relevantes de noções teóricas e modelos. Como ilustrado em um *overview* de avanços recentes, esta área está ainda em sua infância. Entretanto, o trabalho desenvolvido já mostra que nós compartilhamos muitos problemas conceituais e interesses com outras disciplinas em ciências afetivas e que o progresso na área necessita de esforços multidisciplinares. (Cañamero, 2005, p.445) (tradução da autora).

2.4.6 Mecanismos Neurais das Emoções - Juan Velásquez

Juan Velásquez pesquisador do Laboratório de Inteligência Artificial do MIT (EUA) desenvolveu um modelo para suportar emoções com arquitetura modular e incremental denominada *Cathexis*

(Velásquez, 1998).

A arquitetura *Cathexis* utiliza um modelo com inspiração fisiológica para definir mecanismos emocionais e comportamentais resultantes da interação entre sistemas que configuram estruturas neurológicas.

Os elementos desta arquitetura estão baseados em redes neurais/comportamentais com unidades computacionais não lineares equivalentes a neurônios (chamados *releasers*).

Os *releasers* realizam uma filtragem sensorial dos dados ao identificarem condições específicas, produzindo reações amplificadoras ou inibidoras para os sistemas a que estão ligados. Possuem ainda memória de curto prazo que permite adaptação a estímulos inatos ou aprendidos (Figura 2.4).

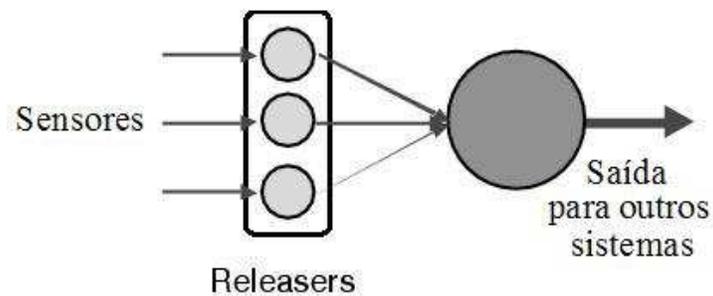


Figura 2.4: Adaptada de Velásquez (1998) - *releasers: Basic Computational Units (BCU)*

Conjuntos de *releasers* são interligados em redes e compõem 5 diferentes subsistemas: sistema sensorial, sistema motor, sistema comportamental, sistema *drive* e sistema gerador de emoção (Figura 2.5).

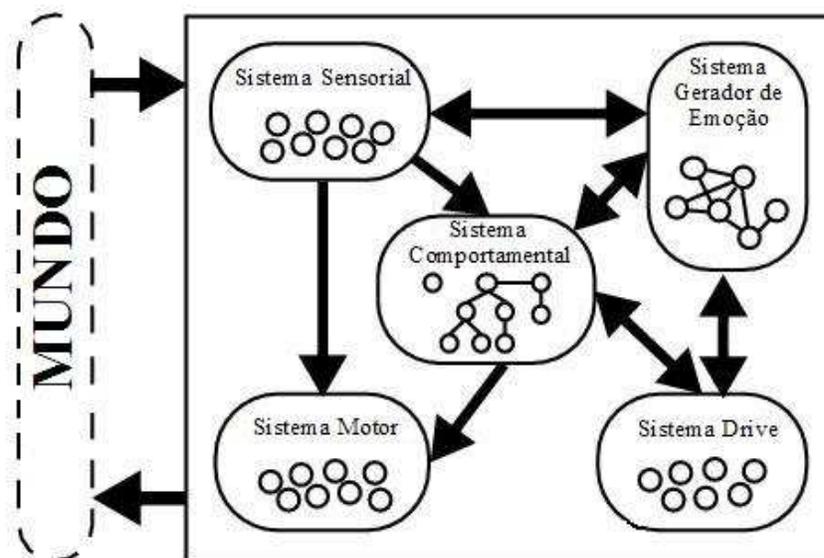


Figura 2.5: Adaptada de Velásquez (1998) - *framework* de controle baseado em emoção

Cada sistema possui um conjunto de *releasers* que dita suas funcionalidades e provê saídas para outros sistemas:

1. **Sensorial:** obtém dados do ambiente sob a forma de características dos estímulos e objetos detectados, e os distribui para os sistemas comportamental e emocional;
2. **Motor:** executa as ações indicadas pelo sistema comportamental e sensorial;
3. **Comportamental:** troca dados com os demais sistemas;
4. **Drive:** Seus *releasers* monitoram variáveis motivacionais verificando se estão dentro de certas faixas de valores, com isso enviam sinais de erro para outros sistemas (sistema gerador de emoções e comportamental). Contribui para compor as motivações do agente;
5. **Gerador de Emoção:** neste sistema os *releasers* podem pertencer a quatro categorias:
 - Neural: sujeito a condições neuro-fisiológicas, ação de neurotransmissores, condições do ambiente, etc;
 - Sensorio-motor: ligado à expressão facial, postura corporal;
 - Motivacional: abriga também *releasers* do sistema *drive* e todos os demais;
 - Cognitivo: disparados por processos cognitivos tais como crenças, lembranças, etc.

Baseado em Damásio, há uma distinção clara entre emoções primárias e emoções secundárias. As emoções primárias são mais primitivas (como medo, aversão) e conferem ao agente a base para que ele desenvolva suas emoções secundárias (ciúme, orgulho e outras). Estas resultam de um processamento mais complexo envolvendo memória e aprendizado, portanto ligados à capacidade de aprendizagem e adaptação ao ambiente.

Velásquez (1998) faz parte de uma linha de pesquisadores que associam fatores cognitivos e não cognitivos às emoções. Entende emoções como indissociáveis de componentes como percepção, motivação, comportamento e controle motor. O modelo por ele proposto é algo modular e incremental no sentido de que à medida que a compreensão sobre emoções se aprofunda, novas possibilidades e propriedades podem ser facilmente incorporadas à arquitetura de base neural.

Um desafio para se construir este tipo de agente, tal qual Cañamero (1997), está na necessidade de especificação detalhada dos sistemas emocionais considerados e suas relações com estímulos ambientais.

2.4.7 Agentes Emocionais: Reproduzindo Damásio - Rodrigo Ventura

Rodrigo Martins de Matos Ventura é professor no Instituto Superior Técnico (IST) de Lisboa, na área de Inteligência Artificial, onde tem desenvolvido trabalhos de pesquisa no Instituto de Sistemas e Robótica (ISR).

Ventura (2000) descreve uma arquitetura em dois níveis para o controle de agentes simples, que incorpora o conceito de emoção, interpretado como processo mental. Esta arquitetura tenta reproduzir o processamento cognitivo que ocorre nas regiões cerebrais do neocórtex e da amígdala cerebral. É fortemente inspirado no trabalho de Damásio.

Opera sob dois níveis de processamento de estímulos oriundos do ambiente: um complexo (Cognitivo) e um básico (Perceptivo).

O nível perceptivo cria uma imagem perceptiva, ou seja, estabelece um conjunto reduzido de características dos estímulos, definidas como sendo as mais significativas para o agente. Essa camada é capaz de trabalhar de forma independente e representa o mesmo tipo de processamento que ocorre na amígdala.

No nível perceptivo, há ainda um vetor de apreciação emocional associado à desejabilidade (DV - *desirability vector*), nele se dá o caráter emocional associado às imagens perceptivas ou cognitivas que o agente forma. Cada componente do DV contém um avaliador de estímulos, podendo estar ativo ou neutro - uma vez ativado, o estímulo será avaliado em termos de ser favorável ou desfavorável.

Já o nível cognitivo cria uma imagem cognitiva, uma representação detalhada que permite uma posterior reconstrução dos estímulos. Esta camada cognitiva depende da camada perceptiva e representa o tipo de processamento que ocorre no neocórtex (Figura 2.6).

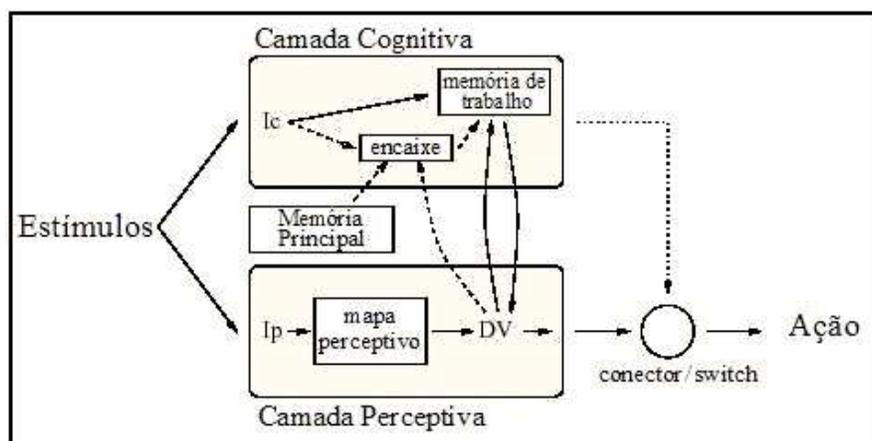


Figura 2.6: Adaptada de Ventura (2000) - a arquitetura proposta

O modelo opera também segundo vários níveis de memória, os quais permitem conectar as imagens cognitivas e perceptivas ao vetor de desejabilidade, de acordo com as circunstâncias de mo-

mento.

2.4.8 Agentes Emocionais em Ambiente de Tempo Real: Sarmento

Luís Morais Sarmento é pesquisador da Faculdade de Engenharia Informática da FEUP da Universidade do Porto, Portugal, na qual pesquisa agentes emocionais que atuam em ambientes em tempo real. Assim como outros autores, Sarmento (2004) argumenta que o uso de agentes facilita a implementação de sistemas que envolvem tomada de decisão, aprendizado e gerenciamento de memória, e que o uso de agentes emocionais, especialmente, permite gerenciar melhor recursos quando estes são limitados tanto do lado do agente quanto do ambiente.

Como a implementação da solução de Sarmento (2004) e Sarmento et al. (2004) tem especial interesse para esta dissertação, iremos descrevê-la com mais detalhes do que os trabalhos mencionados anteriormente.

Descrição Inicial do Simulador

O ambiente proposto por Sarmento (2004) denomina-se *Pyrosim* e corresponde a uma plataforma onde focos de incêndio em uma grande área devem ser combatidos por um conjunto de agentes emocionais colaborativos representados na figura de bombeiros (Figura 2.7).

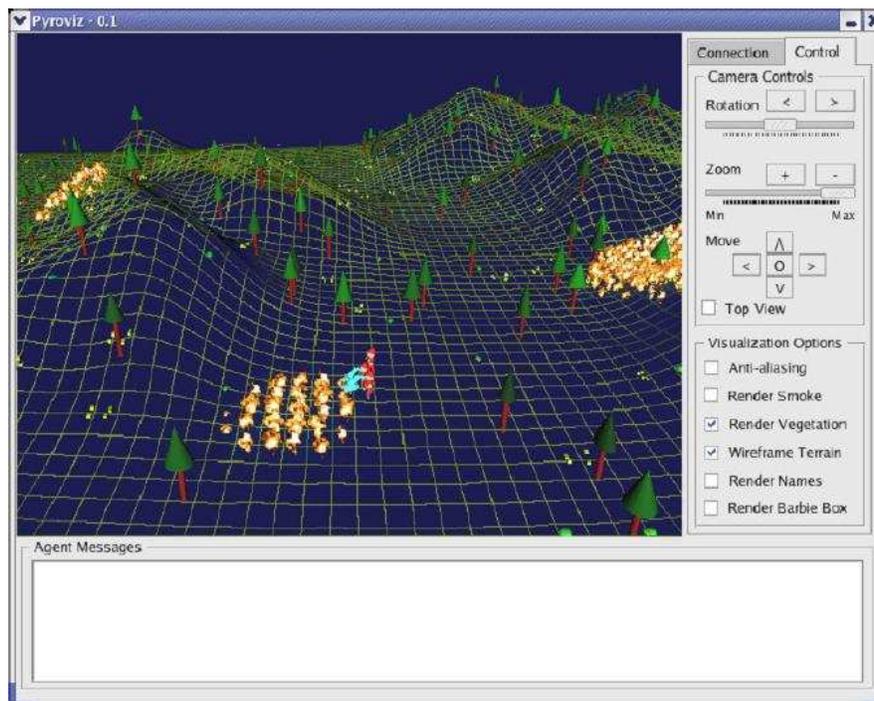


Figura 2.7: Adaptada de Sarmento (2004) - ambiente *Pyrosim* e alguns agentes em ação

Além do objetivo de combater o fogo e minimizar os prejuízos com o incêndio, cada bombeiro deve assegurar sua integridade física, cooperar com os colegas (por exemplo, jato de água é um equipamento que não pode ser utilizado isoladamente), saber lidar com limitações físicas próprias, transpor irregularidades do terreno e outras restrições.

Em resumo, há dois objetivos no simulador: (i) combater o fogo eficientemente e, (ii) manter a integridade física dos combatentes.

Existe comunicação direta entre agentes (por exemplo, para comunicar um risco iminente) apenas na modalidade de um agente para outro agente. Um agente também recebe informação de parâmetros visíveis de outros agentes como localização, nível de energia e ação.

Arquitetura *Pyrosim*

A execução do simulador pode ser vista como uma luta contra o tempo (incêndio tende a se alastrar se o combate a ele não for eficiente), por esse motivo é muito importante para esta estrutura a forma como é feito o gerenciamento de recursos e de tempo.

O processamento ocorre de baixo para cima, isto é, das camadas inferiores (voltadas à sobrevivência) para as camadas superiores (voltadas ao planejamento e decisões). Os agentes têm que lidar com interesses múltiplos e simultâneos, que, de certa forma, correspondem a programas (módulos) que rodam em paralelo e que compartilham os mesmos recursos computacionais de acordo com as metas colocadas.

Nesse contexto é que entra o papel das emoções, mecanismos capazes de configurar três dos elementos principais da arquitetura: Módulos Funcionais, Camadas e Metas, como veremos mais adiante.

Os principais elementos na arquitetura *Pyrosim* são:

1. Módulos Funcionais;
2. Camadas;
3. Metas;
4. Mecanismos Emocionais.

1. **Módulos Funcionais** são programas que compõem os blocos constitutivos da arquitetura e dividem-se em cinco categorias: perceptores, raciocinadores, gerenciadores, controladores e mecanismos emocionais.

Módulos Funcionais utilizam a Memória de Trabalho para armazenar e ler Crenças. A memória permite a troca de informações entre módulos de diferentes camadas. Perceptores recebem e tratam dados sensoriais locais providos pelo *Skeleton* do Agente (exemplo: temperatura local), armazenam na Memória de Trabalho as novas Crenças produzidas que se tornam, por sua vez, disponíveis como entrada para os módulos Raciocinadores. Os Raciocinadores poderão, então, produzir Crenças de alto nível (exemplo: um plano, uma análise de fogo global). Verificar na Figura 2.8.

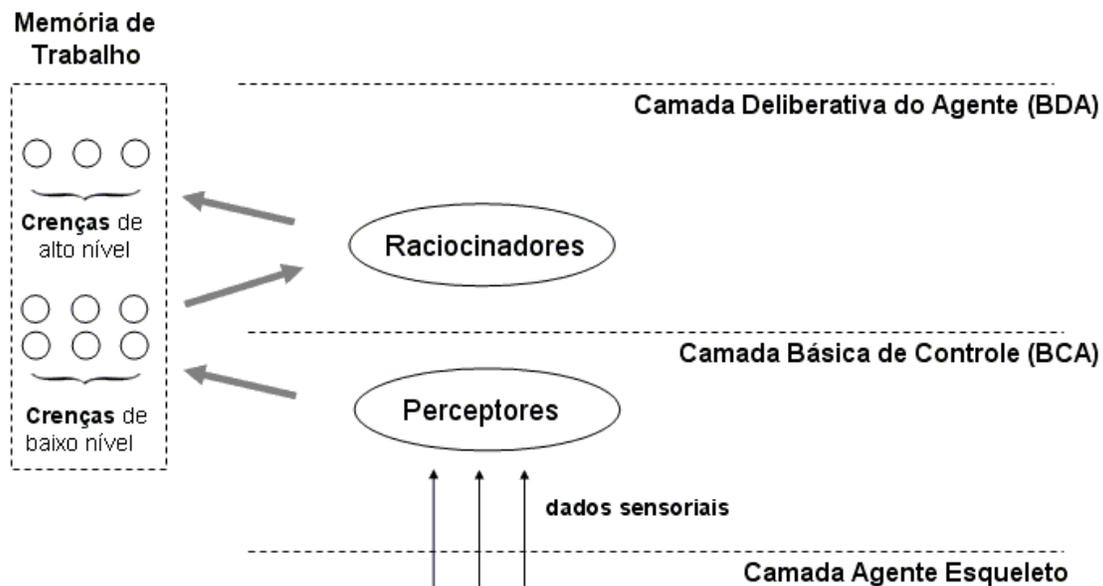


Figura 2.8: Elementos da arquitetura

2. **Camadas** correspondem a grupos de módulos processadores de informação cuja execução no tempo é voltada para o cumprimento de metas específicas.

A arquitetura do simulador *Pyrosim* compõe-se de três camadas: Camada Básica de Controle (BCA) e Camada Deliberativa do Agente (BDA), que operam sobre uma camada básica inferior de comunicação denominada de *Skeleton* do Agente.

A **Camada Básica de Controle (BCA)** está associada a grupos de módulos encarregados de processar funções mais básicas e essenciais visando garantir a seguinte meta: a sobrevivência do agente.

Opera com módulos Perceptores: um perceptor de dor, um perceptor de temperatura, um perceptor de fogo, entre outros. As tarefas de processamento associadas a estes perceptores são executadas a cada ciclo de execução do BCA produzindo Crenças que são disponibilizadas globalmente pela Memória de Trabalho.

Nesta camada atua o módulo Gerenciador de Metas que é responsável por escalonar a execução das metas do agente.

Já a **Camada Deliberativa do Agente** (BDA) inclui módulos associados a metas menos urgentes, mais sofisticadas, ou que dependem de informações produzidas nas camadas inferiores.

Os Módulos Funcionais presentes nesta camada são do tipo Raciocinadores e conferem ao agente as capacidades mentais necessárias para metas associadas ao combate ao fogo. Por exemplo, metas como “vá para um local específico” ou “apague o fogo no local vizinho”.

Outro grupo de metas também presentes, possui caráter mais geral, associado ao monitoramento da situação como, por exemplo, “verifique se há necessidade de reposicionar-se”. Inclui a Memória de Execução que permite raciocinar sobre ações anteriores. A Memória de Execução localizada na Camada BDA está relacionada ao estado do agente e auxilia o mesmo a tomar decisões no presente e futuro baseando-se em ações do passado.

3. **As Metas** utilizam informações sensoriais, Crenças (da Memória de Trabalho) e aproveitam valores dos Acumuladores Emocionais produzindo como saídas comandos de alto nível ou novas metas.

O gerenciamento do processamento de recursos/tempo ocorre de baixo para cima, da camada inferior para a camada superior. Módulos associados às camadas mais altas só serão executados depois dos Módulos da camada inferior e rodarão menos do que estes, pois os módulos da camada inferior são sempre mais solicitados.

Como já mencionamos, a idéia desta arquitetura é de que todos os módulos rodem em paralelo e que compartilhem os recursos computacionais de acordo com a importância das metas. Nesse âmbito é que entram as Emoções.

Em outras palavras, a arquitetura composta de vários módulos distribuídos em camadas, inclui um conjunto de Mecanismos Emocionais cuja atividade está diretamente relacionada à performance do agente em cumprir metas. Mecanismos Emocionais podem ser usados para elevar a adaptação do agente ao ambiente por influenciar vários parâmetros internos, entre eles o de otimizar a alocação do processamento de recursos e aperfeiçoar as decisões de estratégias.

4. **Mecanismos Emocionais**, a operação dos módulos é altamente configurável e sofre influência dos Mecanismos Emocionais.

Metas possuem parâmetros configuráveis. Por exemplo, a meta “combata o fogo em uma área próxima” envolve limiares de temperatura influenciadores da decisão se o agente deve avançar, ficar onde está ou recuar para uma posição menos perigosa. Alterar esses limiares de temperatura afeta a eficiência global do sistema (menor efeito do jato de água).

Pode-se também alterar o comportamento interno de cada camada. Como já vimos, Camadas são responsáveis por decidir quais módulos serão escalonados a cada ciclo de execução. Na prática,

decidem quanto tempo gastar em análises de percepção, de controle das ações, em tomar decisões ou explorar possíveis cenários.

Assim, Emoções representam uma forma de regular os parâmetros das metas, camadas e módulos de forma a tornar o método mais eficiente como um todo, ou visto de outro ângulo, capacitar eficientemente o processamento do agente em resposta às condições do ambiente: saber investir recursos onde é mais urgente investir.

Cada Mecanismo Emocional possui um Acumulador Emocional a ele associado que afeta dinamicamente o comportamento do Agente.

A arquitetura *Pyrosim* contempla três Mecanismos Emocionais: Medo, Ansiedade e Auto-Confiança (Sarmiento, 2004 e Sarmiento et al., 2004).

Nesta dissertação existe um paralelo entre as emoções definidas por Sarmiento e a nossa (Medo e Curiosidade).

Mais detalhes desta comparação serão retomados no Capítulo 4 desta dissertação.

2.4.9 Agentes Emocionais Reativos: Balkenius

Christian Balkenius é professor do departamento de Ciência Cognitiva da Universidade de Lund, Suécia, onde pesquisa mecanismos cognitivos cerebrais e redes neurais artificiais visando aplicações em criaturas artificiais.

Balkenius (1995) menciona que desde a década de 30 do século XIX, estudiosos já buscavam modelos matemáticos para a área da psicologia. O problema embutido era a impossibilidade de se definir uma métrica para o espaço psicológico. Isso muda com o advento de criaturas artificiais. Nele, o comportamento pode ser definido a cada momento como uma função do espaço-vida do indivíduo.

$$B = f(P, E), \text{ onde } P = \text{personalidade e } E = \text{ambiente}$$

Segundo Balkenius, uma criatura artificial deve, em princípio seguir quatro tipos de comportamento padrão:

1. **Apetitivo:** foco para um objeto ou uma situação atrativa;
2. **Aversivo:** afastamento de situações negativas;
3. **Exploratório,** direcionado a um aumento no conhecimento sobre o ambiente;
4. **Neutro:** descreve uma classe de comportamentos que se relacionam aos objetos que não são nem apetitivos e nem aversivos.

Balkenius volta suas atenções mais propriamente para Agentes Reativos, embora o aprendizado embutido nas redes neurais possa conferir às suas criaturas comportamentos bastante sofisticados.

2.5 Emoções: Buscando um Consenso

Sloman (2001) nos adverte que: “Existem atualmente muitos “*surveys*” bastante úteis apontando questões relacionadas a emoções, mas ainda é difícil a alguém novo no ramo obter uma visão geral balanceada, e não somente porque existe uma grande variedade de definições de emoções oferecidas por diferentes pesquisadores com diferentes pontos de vista ... o termo “Emoção”, da mesma forma que “Inteligência”, é um conceito do tipo *cluster*, que se refere a um grupo variável de capacidades e admite uma grande variedade de tipos e instâncias, sem uma fronteira bem definida”.

Esta variedade de modelos para emoções pode ser vista a partir dos modelos apresentados nas seções anteriores. Por exemplo, em (Ortony et al., 1998) emoções são definidas como “reações valoradas a eventos, agentes ou objetos, com sua natureza particular sendo determinada e por meio dela a situação elicitada é construída”.

Cañamero (2000) define emoções como “processos dinâmicos complexos que integram vários sub-sistemas relacionados de maneira causal” (Cañamero, 2000, p.03). Ou ainda (Cañamero, 2000, p.04): “emoções são funções biológicas do sistema nervoso envolvidas tanto na sobrevivência do indivíduo como da espécie, em ambientes sociais complexos, dinâmicos, incertos e com recursos limitados, sobre os quais os agentes têm pouco controle”.

Balkenius (1995), seguindo (Rolls 1986; Gray 1982), diz que emoções são “estados elicitados por estímulos de reforço”.

Damásio (1994) e também Ventura (2000) pregam que “Emoção é a combinação de um processo avaliatório mental, simples ou complexo, com respostas dispositivas a esse processo, em sua maioria dirigidas ao corpo propriamente dito, resultando num estado emocional do corpo, mas também dirigidas ao próprio cérebro, resultando em alterações mentais adicionais” (Damásio, 1994, p.169).

Velásquez (1998) nos diz que “emoções são funções biológicas do sistema nervoso que foram conservadas por meio da evolução, e que são necessárias para a satisfação das condições básicas que são cruciais à sobrevivência (escapar de um perigo, obter comida, abrigo e acasalamento)” (Velásquez, 1998, p.01).

Desta forma, vemos que, para uns, emoções são estados, para outros, são processos, para outros são funções. Para alguns autores, o mais importante é o aspecto valorativo/apraisivo da emoção (o chamado aspecto *appraisal*, sua capacidade de avaliar se objetivos ou propósitos implícitos do indivíduo estão sendo cumpridos). Outros enfocam mais o aspecto dinâmico das emoções (o chamado aspecto *arousal*), como reações específicas a situações, eventos ou objetos específicos. Outros au-

tores ressaltam o aspecto homeostático das emoções, fazendo um paralelo com o processo de regulação em sistemas de controle. Alguns autores ressaltam a importância das emoções como “alarmes” que disparam comportamentos de exceção no agente. Em alguns casos, falam de emoções como “amplificadores” de valores que afetam o julgamento (e, portanto, o comportamento).

Em diversas situações, ao invés de simplesmente se referirem às emoções, desenvolve-se todo um “sistema emocional”, onde outros termos como “motivações”, “afetos”, “necessidades”, “*drives*”, “sentimentos”, “desejos”, “impulsos”, “vontades” voluntárias/involuntárias, e outros são utilizados, às vezes em conjunto, às vezes não, com o próprio termo emoção, para designar diferentes componentes ou aspectos do sistema emocional. Entretanto, o uso destes termos não é realizado sempre de maneira consistente, entre os autores.

Apesar dessa aparente confusão, como aponta Sarmiento et al. (2004): “a noção essencial sobre mecanismos emocionais é a de que estão relacionados ao sucesso com o qual um indivíduo atinge seus propósitos”.

Desta forma, resolvemos, nesta dissertação, explorar um caminho minimalista, adotando a idéia que nos pareceu mais consensual dentre todas as propostas envolvendo emoções, que é a de que emoções codificam de maneira implícita os propósitos de um indivíduo. Desta forma, as emoções funcionariam como uma espécie de conhecimento *default* que temos *a priori*, obtido por herança genética (ou seja, por construção) de nossa espécie, de como reagir e executar comportamentos prototípicos que têm maiores chances de contemplar nossos objetivos implícitos como membros de uma espécie que busca a sobrevivência em um ambiente pleno de recursos, mas também cheio de armadilhas. O resultado foi a arquitetura que denominamos de Agente Emocional Hedonista, que utiliza uma metáfora computacional relativamente simples para emoções de forma a gerar comportamentos sofisticados para criaturas artificiais (agentes autônomos inteligentes).

2.6 Agente Emocional Hedonista

O Hedonismo é uma teoria ou doutrina filosófico-moral que coloca como maior valor, o prazer e a satisfação individual em primeiro lugar (Gomes, 2007).

Assim, de um ponto de vista hedonista, as ações e tomadas de decisão são determinadas em relação ao prazer e à dor que proporcionam, tanto em um sentido imediato como em termos futuros. Todo o comportamento, sob a perspectiva hedonista, visa diminuir a dor atual e minimizar a expectativa de dor futura e, da mesma forma, incrementar o prazer atual e maximizar a expectativa de prazer futura.

Na verdade, prazer e dor aqui podem ser ampliados a um grande espectro de emoções, onde cada emoção codifica um sub-objetivo implícito do agente, aqui chamado **Agente Emocional Hedonista**.

Programar um comportamento autônomo por meio deste **Agente** implica decidir quais emoções o agente deve instanciar, e definir de que forma essa emoção (função de utilidade) pode ser calculada. É importante ressaltar que pode existir uma hierarquia inicial entre as emoções, que implicarão uma ordem de prioridade, quando diferentes emoções levarem a resultados conflitantes. A influência de um dos fatores emocionais pode preponderar sobre os demais acarretando um comportamento complexo para a criatura artificial (Figura 2.9).

Agente Hedonista

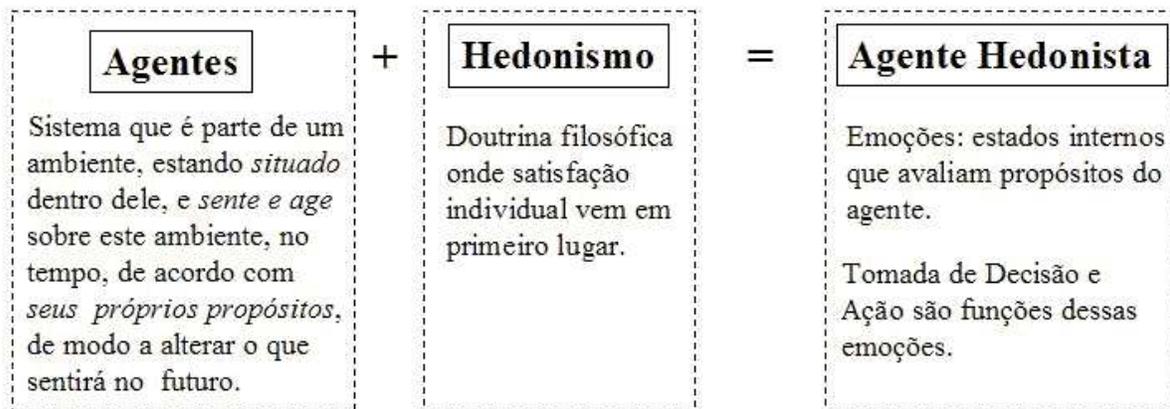


Figura 2.9: Agente emocional hedonista

Entretanto, o **Agente Emocional Hedonista** leva em consideração somente suas próprias emoções, desconsiderando outros tipos de comportamento mais sofisticados que ocorrem em agentes humanos, onde, por exemplo, o comportamento em grupo e as emoções de outros agentes também são também relevantes em suas decisões comportamentais.

Neste trabalho, utilizamos a seguinte codificação das metáforas emocionais (mais detalhes nos Capítulos 3 e 4):

- **Medo:** sinal sensorial interno que avalia o quanto uma situação pode representar riscos à integridade física da criatura.
- **Fome:** avalia as condições energéticas internas da criatura quanto à manutenção de energia em um nível suficiente.
- **Curiosidade:** o quanto uma determinada ação é capaz de aumentar o conhecimento da criatura sobre o ambiente.

A escolha das emoções **Medo**, **Fome** e **Curiosidade** tem paralelo com os comportamentos Aversivo, Apetitivo e Exploratório propostos por Balkenius, embora nesta dissertação, não contemplemos o comportamento Neutro (maiores detalhes no Capítulo 4).

Nesta dissertação, inspirados por essas idéias colhidas da literatura, apresentamos uma arquitetura de um controlador para a navegação autônoma de uma criatura artificial, utilizando metáforas emocionais para a implementação de agentes.

2.7 Criaturas Artificiais

Nos diversos trabalhos encontrados na literatura, observamos o uso dos termos “agentes” e “criaturas” empregados de maneira inter-cambiável, para designar um mesmo tipo de idéia. Alguns autores utilizam “agente”, sendo que outros preferem o termo “criatura”. Entretanto, apesar disso parecer somente uma questão de gosto pessoal, entendemos que existe uma questão terminológica mais séria neste assunto, que merece ser tratada.

O termo “agente” tem sido empregado em uma acepção mais ampla, a diversas idéias que, mesmo tendo uma abstração comum, podem representar coisas bem diferentes. Por exemplo, utiliza-se o termo “agentes de internet” (ou *spiders*, ou ainda *web-crawlers*) para designar certos programas de computador que varrem a internet por meio do acesso sistemático a *links* encontrados em páginas *web*, visando indexar e catalogar a informação na internet. Utiliza-se ainda o termo “agente móvel”, para designar certos componentes de *software* que possuem a habilidade de se deslocar dentre diferentes máquinas, de maneira pró-ativa. Agentes como os agentes de internet ou os agentes móveis possuem uma diferença bastante marcante quando comparados aos agentes revisados na Seção 2.4. A grande diferença que existe é que os agentes referenciados na Seção 2.4 possuem um “corpo”, ou avatar, por meio do qual são corporificados no ambiente em que atuam. Em outras palavras, ele não somente sentem e agem sobre um ambiente, mas “existem” nesse mesmo ambiente, por meio de seu corpo/avatar². Esse detalhe poderia, a princípio, parecer insignificante. Entretanto, existe todo um movimento dentro das ciências cognitivas, que ficou conhecido como *Embodied Cognitive Science* (Varela, Thompson and Rosh (2003); Clark 1999), onde certas questões relacionadas à existência de um corpo, um ambiente local e a complexa interação entre estes parece ser de crucial importância para a compreensão do processo de cognição. Desta forma, o uso do termo “criatura”, para representar um agente corporificado (ou um agente com corpo/avatar), seja este corpo real, como no caso de um robô, ou virtual, como no caso de uma simulação, nos parece mais adequado do ponto de vista terminológico.

Desta forma, estaremos adotando deste ponto em diante, o termo “criatura” para designar nosso **Agente Emocional Hedonista**.

²avatar é a corporalização de um modelo ou categoria.

2.8 Resumo do Capítulo

Neste Capítulo, mostramos uma visão geral de utilização das emoções em sistemas artificiais e a nossa proposta sobre o **Agente Emocional Hedonista**, principal elemento desta dissertação. No próximo Capítulo, apresentaremos o problema de simulação que vamos estudar, ressaltando os detalhes do simulador e a estrutura geral do sistema de controle a ser desenvolvido.

Capítulo 3

Simulação da Criatura Artificial

3.1 Prólogo

Neste capítulo, apresentamos os principais detalhes sobre o simulador construído neste trabalho: seus objetivos, detalhes internos e sua interface gráfica com o usuário.

3.2 A Simulação de Criaturas e seus Objetivos

O estudo de criaturas artificiais envolve basicamente dois tipos de esforços tecnológicos. O primeiro diz respeito ao corpo (avatar) da criatura em si, e eventualmente a criação/preparação de seu ambiente. Por exemplo, em casos de criaturas robóticas, é necessário se desenvolver o hardware do robô, seu sistema de sensoriamento e atuação, e posteriormente temos que preparar o ambiente onde o robô desenvolverá seu comportamento. Este ambiente pode ter obstáculos especialmente preparados, envolver marcações no solo ou disponibilizar outras informações que possam ser utilizados para orientação e navegação do robô. Em alguns casos, este ambiente será um ambiente do mundo real, como laboratórios, corredores e salas. No caso de criaturas virtuais, é necessário desenvolver um simulador para a criatura. Este simulador deve não somente simular o corpo da criatura, mas também todo o ambiente onde a criatura desenvolverá seu comportamento. Este ambiente deve ser tão mais complexo quanto se deseja que o comportamento da criatura seja complexo. Em alguns casos, esses ambientes virtuais podem tentar se aproximar de ambientes reais. Nesses casos, teríamos ambientes tridimensionais, com objetos de múltiplas formas e atributos. Em outros casos, onde deseja-se estudar fenômenos específicos, esses simuladores normalmente optam pela simplicidade e o minimalismo do ambiente, introduzindo-se complexidade somente em função da necessidade de se explorar/estudar algum comportamento especializado. Normalmente, ambientes desse tipo são ambientes bi-dimensionais, com objetos bastante simples como retângulos, ou outras formas

geométricas simplificadas.

O segundo esforço tecnológico ao qual aludimos anteriormente é o esforço no desenvolvimento de um sistema de controle para a criatura artificial. Este sistema de controle deve colher dados sensoriais do ambiente e gerar dados de atuação, que coordenarão o comportamento da criatura artificial em seu ambiente, em tempo real. Esse comportamento normalmente visa cumprir um conjunto de objetivos, que podem ser especificados tanto em alto nível como em baixo nível. Esses objetivos podem ser mais específicos, tais como desenvolver uma trajetória, deslocando-se no ambiente a partir de um ponto inicial até um ponto final, sem colidir com obstáculos. Ou então, podemos ter objetivos mais genéricos, tais como aumentar o conhecimento sobre o seu ambiente, minimizar o risco de colisão da criatura ou manter o balanço energético da criatura. Objetivos como esses são de mais longo prazo, que dizem respeito não somente a um comportamento específico, mas sim ao próprio propósito da existência da criatura.

Para cumprir seus objetivos, a criatura artificial demanda um sistema de controle que considere esses objetivos e determine, a cada instante, as saídas de atuação que devem ser enviadas aos atuadores mecânicos (no caso de robôs reais) ou aos atuadores simulados, em um simulador de ambiente virtual. Diferentes estratégias podem ser implementadas na construção destes sistemas de controle, desde estratégias puramente reativas, que para cada entrada sensorial x determina uma atuação $y = f(x)$, até estratégias deliberativas, que envolvem planos de mais longo prazo e incorporam memória e algum tipo de antecipação no comportamento da criatura.

Nesta dissertação nosso propósito foi o de trazer contribuições principalmente a este segundo esforço tecnológico, o desenvolvimento de sistemas de controle para criaturas artificiais. Desta forma, nosso maior esforço está no desenvolvimento de estratégias de controle para a criatura artificial, e menos para o desenvolvimento de um corpo/ambiente para a criatura (embora este seja também necessário). Assim, adotamos uma estratégia minimalista no desenvolvimento de um simulador para um ambiente virtual, inserindo no mesmo somente a complexidade necessária para estudarmos o comportamento específico que é objeto de nosso interesse. Como já dissemos anteriormente, nosso maior objetivo é utilizar a metáfora emocional para direcionar o desenvolvimento de um sistema de controle para a criatura artificial, em que os diferentes propósitos para o comportamento da criatura sejam traduzidos em emoções, e estas possam ser utilizadas para derivar as saídas de controle para a criatura.

Levando-se em conta as considerações anteriores, desenvolvemos um simulador bi-dimensional, utilizando a linguagem Java, tendo como referência o simulador desenvolvido em Gudwin (1996), originalmente desenvolvido em linguagem C (utilizando a biblioteca XFORMS) e rodando somente em ambiente Unix. O uso da linguagem Java aumenta a portabilidade do simulador, permitindo que o mesmo possa ser executado tanto em ambiente *Windows* como em ambientes Unix (Linux, Solaris,

etc).

O simulador estabelece a interação de uma criatura autônoma com um ambiente repleto de objetos com características diferenciadas. Os objetos podem ser de diferentes “cores”, sendo que cada “cor” está associada a um conjunto de propriedades de “dureza”, “gosto” e “energia”. A “dureza” de um objeto representa certo grau de restrição ao movimento da criatura ao tentar atravessá-lo; “gosto” pode variar de ruim até bom, portanto está associado a certo grau de satisfação que o objeto proporciona; e, finalmente, “energia”, é a propriedade de um objeto fornecer ou drenar energia da criatura quando em contato com a mesma.

O comportamento da criatura artificial neste ambiente deve obedecer conjuntamente a um certo número de objetivos, onde a cada instante, um destes objetivos pode ser mais importante do que os outros. Em nossos experimentos, consideramos os seguintes objetivos:

- Alcance das metas previamente estabelecidas (alcançar alvos);
- Exploração do ambiente (elevar conhecimento sobre o ambiente);
- Manutenção da integridade física da criatura (acionar sistema anti-colisão);
- Manutenção da carga das baterias em níveis operacionais (mudar prioridades).

Tipicamente, o comportamento da criatura deve seguir o seguinte padrão: o objetivo de mais longo prazo é a exploração do ambiente. Dessa forma, a criatura deve explorar todo o ambiente, conhecendo e integrando a seu modelo de mundo os objetos nele existentes. Para consolidar esse objetivo, a criatura determina “metas”, ou seja, pontos do ambiente que ela deseja alcançar, a partir de sua posição corrente. Dada então uma “meta”, a criatura deve encontrar uma trajetória que a leve de sua posição atual até a meta, sem colidir com os obstáculos no caminho. A escolha dessas “metas” pode ser implementada de diferentes maneiras. De um modo geral, a idéia é escolher metas que façam a criatura visitar partes do ambiente que ainda não conhece. Entretanto, caso a criatura identifique que está com suas reservas energéticas baixas (baixa carga na bateria), ela deve, ao invés de privilegiar a exploração do ambiente, buscar a manutenção de suas reservas. Para tanto, deve escolher como “meta” um ponto do ambiente onde esteja localizado algum objeto fornecer de energia, para que a criatura possa se reabastecer. Após o reabastecimento, a criatura deve retornar ao seu objetivo de explorar o mundo.

Para a execução dos objetivos acima descritos, foram propostos dois níveis de controle para a criatura: um primeiro nível mais baixo, também chamado de **nível de controle direto** e um segundo nível hierarquicamente superior, chamado de **nível de decisão de metas** (Figura ??). O nível de controle direto possui como entrada uma meta que se deseja alcançar (um ponto do ambiente, em termos de suas coordenadas (x, y)), além dos sinais sensoriais obtidos do ambiente e eventualmente alguma memória interna do sistema de controle. Como saída, o nível de controle direto determina

os sinais de controle a serem enviados aos atuadores da criatura. O nível de decisão de metas é responsável pela determinação das metas (em termos de coordenadas (x, y)), que alimentarão o nível de controle direto. Esse nível deve, a partir da especificação dos objetivos de longo prazo da criatura, determinar para onde a criatura deve se deslocar.

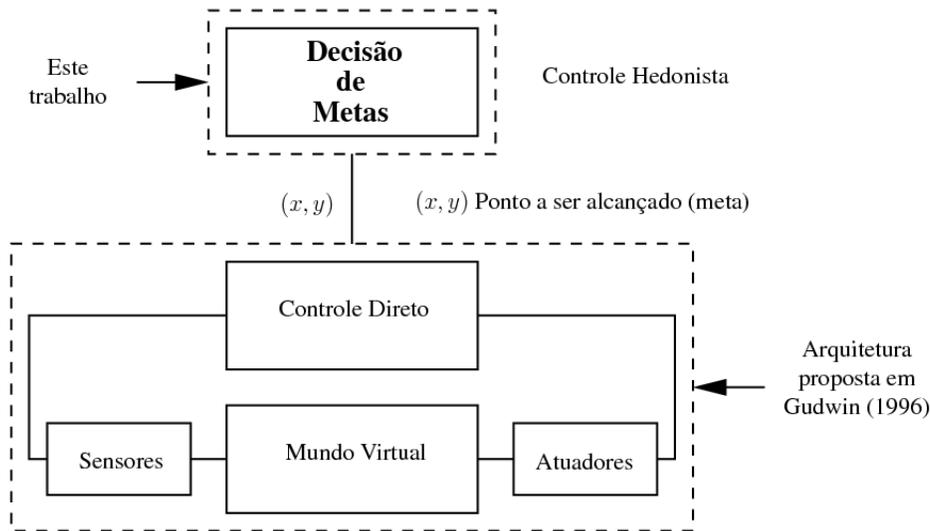


Figura 3.1: Arquitetura da criatura artificial

O nível de controle direto utilizado neste trabalho é basicamente o mesmo desenvolvido em (Gudwin, 1996), sendo que neste caso foi feita uma nova implementação do mesmo algoritmo na linguagem Java. Esse algoritmo segue uma estratégia deliberativa, que tenta determinar um conjunto variável de sub-metas, de tal forma que uma trajetória linear entre a posição atual da criatura e cada uma das sub-metas não cruza nem colide com nenhum obstáculo conhecido no modelo de mundo da criatura. Esse modelo de mundo é incremental, sendo construído a partir da interação da criatura com o ambiente, e à medida que alguma modificação no modelo do mundo inviabiliza um plano prévio em execução, o sistema refaz um novo plano, a partir da posição corrente da criatura e passa a executar esse novo plano. Para maiores detalhes sobre o nível de controle direto, veja a subseção 3.6.1 mais à frente neste Capítulo.

Nesta dissertação, estendemos o trabalho de (Gudwin, 1996) desenvolvendo uma estratégia para o nível de Decisão de Metas. Essa estratégia, que chamamos de controle hedonista, utiliza o conceito de emoções para traduzir os diferentes objetivos globais do sistema e determinar a cada instante uma meta condizente com esses objetivos globais, que é enviada ao nível de controle direto. Duas possíveis variações dessa estratégia são desenvolvidas mais à frente no Capítulo 4.

Juntos, os níveis de controle direto e de decisão de metas geram um comportamento complexo para a criatura, para que a mesma apresente um comportamento global que atenda aos objetivos

propostos.

3.3 Descrição da Criatura

A seguir, faremos uma síntese a respeito do processo de navegação autônoma da criatura e seus elementos constituintes (sensores e atuadores).

Existem três tipos diferentes de sensores: o sensor visual, os sensores de contato e o sensor de energia. Já os atuadores determinam a posição do sensor visual, o ângulo entre as rodas dianteiras e o eixo longitudinal da criatura bem como a sua velocidade nominal.

3.3.1 Sensor Visual

O sensor visual consiste de uma simplificação de um mecanismo de visão. Foi implementado por meio de uma matriz ($8 \times X8$) capaz de cobrir uma área de varredura em movimento contínuo à frente da criatura (Figura 3.2).

Esse sensor consegue detectar a cor de um objeto e prover informação da sua localização no ambiente. Entretanto, o sensor visual não é capaz de determinar as propriedades individuais de um dado objeto (sua “dureza”, “gosto” e “energia”), somente sua cor e localização.

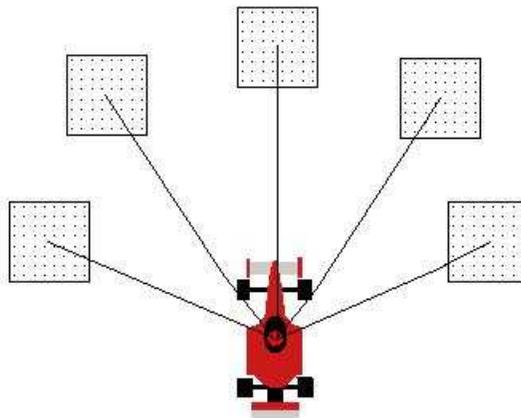


Figura 3.2: Sensor visual

3.3.2 Sensores de Contato

Os sensores de contato consistem de quatro elementos localizados nas extremidades da criatura (Figura 3.3). Diferentemente do sensor visual que “varre” áreas à frente da criatura, os sensores de contato são acoplados à criatura e se movimentam apenas segundo o movimento desta. Esses sensores

têm a função de perceber as propriedades dos objetos (“dureza”, “gosto” e “energia”) no ambiente, de forma que um dado objeto possa variar de pouco atrativo a muito atrativo para a criatura em diferentes graus. Essa consideração deve interferir no cumprimento do objetivo do sistema em um dado instante.

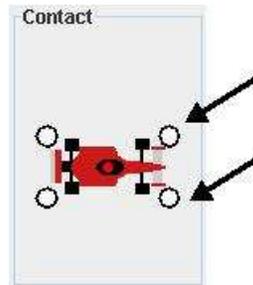


Figura 3.3: Sensores de contato

3.3.3 Sensor de Energia

Esse sensor mede o nível de energia interna da criatura relativa a uma carga inicial máxima (Figura 3.4). Assim, se o nível de energia estiver zerado, o sensor indicará área sombreada vazia. Se estiver plenamente carregado, o sensor indicará área sombreada total. Esse sensor tem uma descarga gradual ao longo do tempo, porém ao entrar em contato com alguns objetos, pode haver tanto uma descarga acelerada quanto uma recarga dependendo da propriedade “energia” do objeto em contato. A situação de carga e descarga pode ser visualizada a qualquer momento durante a execução da simulação, observando-se a indicação do sensor de energia.

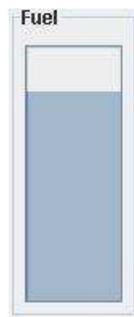


Figura 3.4: Sensor de energia em situação parcialmente carregado

3.3.4 Atuadores de Posição do Sensor Visual

A posição do sensor visual, é determinada por meio de dois atuadores:

- i. ângulo em função da direção frontal da criatura (θ_0);
- ii. raio de ação (l_0) (Figura 3.5).

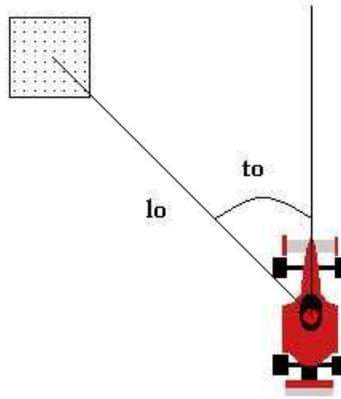


Figura 3.5: Atuadores de posição do sensor visual

3.3.5 Atuadores de Movimentação da Criatura

Existem dois atuadores de movimentação da criatura. O primeiro é o atuador de tração, que confere uma velocidade nominal *speed* à criatura. Se *speed* for positivo, a criatura estará se movimentando para frente, se negativo, para trás, e se *speed* for igual a zero, a criatura estará parada. O valor de *speed* não corresponde exatamente à velocidade da criatura, mas sim à velocidade que a criatura estará se o atrito for nulo. Para valores de atrito superiores a zero, a velocidade real da criatura será menor que a velocidade nominal. O atrito é determinado a partir das propriedades dos objetos por onde a criatura estiver navegando (ver seção 3.4).

Outro atuador de movimentação determina o ângulo *wheel* da posição das rodas em relação ao eixo longitudinal da criatura. Esse atuador determinará a direção que a criatura deverá seguir, quando a velocidade nominal for não-nula. O ângulo das rodas poderá variar de -45° a $+45^\circ$, sendo que um ângulo 0° corresponde a seguir em frente.

3.4 Descrição do Ambiente

O ambiente é limitado por um retângulo ocupando a tela do computador com diversos objetos em seu interior, sendo possível definir diferentes dimensões para este ambiente. Como já vimos, os objetos são caracterizados por suas propriedades físicas: sua “cor”, sua “dureza”, seu “gosto” e sua “energia”.

Cada objeto pode possuir uma cor diferente, que é detectada pelo sensor visual. A cada cor, estarão associados graus configuráveis (em tempo de simulação) de “dureza”, “gosto” e “energia”. A “dureza” do objeto mede a capacidade da criatura em trafegar sobre ele. Diferentes graus de dureza corresponderão a diferentes graus de mobilidade da criatura, facilitando ou dificultando seu trajeto. O valor da “dureza” varia de “0” até “1”. Objetos com “dureza” “1” são considerados intransponíveis, correspondendo a obstáculos sólidos, capazes de obstruir o movimento da criatura.

O “gosto” de um objeto corresponde a um grau de desejo ou repulsa (agradável ou desagradável) proporcionado pelo contato com este objeto. Este gosto é detectado pelos sensores de contato, sendo que seu valor varia de “-1” até “1”, onde “-1” representa total repulsa, “1” representa total contentamento e “0” representa indiferença.

A “energia” de um objeto corresponde à capacidade de fornecer ou absorver energia da criatura durante o contato. Valores positivos de “energia” irão promover um aumento da carga da criatura e valores negativos correspondem à dissipação acelerada de energia da criatura. Valores neutros (zero) correspondem a uma não modificação da carga, exceto a queda gradual da “energia” da criatura em função do tempo da mesma no ambiente. O valor da energia pode variar de “-1” até “1”.

Como vimos, então, as propriedades de “dureza”, “gosto” e “energia” têm um vínculo direto com a “cor”, reconhecendo-se a “cor” de um objeto (mediante sensor visual e de contato), se obtém os valores para essas três propriedades, sendo que objetos definidos pelo usuário com uma mesma cor, reunirão os mesmos valores.

3.4.1 Orientação da Criatura Autônoma

O modelo da criatura é descrito a seguir. As variáveis de interesse correspondem a suas coordenadas x , y , $pitch$, de acordo com a Figura 3.6, além do nível de energia das baterias, dado por *fuel*.

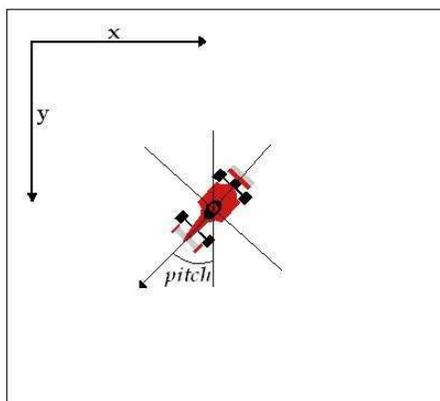


Figura 3.6: Sistema de coordenadas da criatura

Outros parâmetros do sistema são a constante de atrito *friction*, a velocidade nominal *speed*, a distância entre eixos **D** e o ângulo das rodas *wheel* em relação ao eixo longitudinal (Figura 3.7).

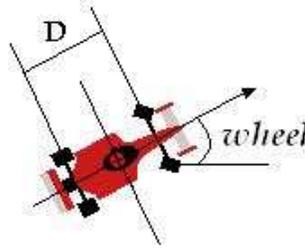


Figura 3.7: Distância entre eixos e o ângulo das rodas

Para modelar a criatura, os seguintes sistemas de equações são utilizados. O primeiro, é utilizado quando o valor da variável *wheel* for igual a zero. O segundo é utilizado quando o valor da variável *wheel* for diferente de zero.

Para *wheel* igual a zero, tem-se as seguintes equações:

$$\begin{aligned}x(k+1) &= x(k) + (1 - \text{friction}) \cdot \text{speed} \cdot \cos(\text{pitch} + \pi/2) \\y(k+1) &= y(k) + (1 - \text{friction}) \cdot \text{speed} \cdot \sin(\text{pitch} + \pi/2) \\ \text{pitch}(k+1) &= \text{pitch}(k) \\ \text{fuel}(k+1) &= \text{fuel}(k) - 10^{-5} + 10^{-2} \cdot \Delta\text{Fuel}\end{aligned}$$

Para *wheel* diferente de zero, temos:

$$\begin{aligned}x(k+1) &= x(k) + D \cdot \sin(\Delta\text{pitch})/\sin(\text{wheel}) \cdot \cos(\text{pitch} + \pi/2) \\ &\quad - D \cdot (1 - \cos(\Delta\text{pitch}))/\sin(\text{wheel}) \cdot \sin(\text{pitch} + \pi/2) \\ y(k+1) &= y(k) + D \cdot \sin(\Delta\text{pitch})/\sin(\text{wheel}) \cdot \sin(\text{pitch} + \pi/2) \\ &\quad + D \cdot (1 - \cos(\Delta\text{pitch}))/\sin(\text{wheel}) \cdot \cos(\text{pitch} + \pi/2) \\ \text{pitch}(k+1) &= \text{pitch}(k) + \Delta\text{pitch} \\ \text{fuel}(k+1) &= \text{fuel}(k) - 10^{-5} + 10^{-2} \cdot \Delta\text{Fuel}\end{aligned}$$

onde

$$\Delta\text{pitch} = (1 - \text{friction}) \cdot \text{speed} \cdot \sin(\text{wheel})/D$$

Analisando-se as equações, observa-se que além das variáveis já comentadas, existem parâmetros e entradas do sistema: o parâmetro **D** que é constante e o parâmetro *friction* que não é constante.

Para *friction*, definimos a seguinte regra: se a criatura estiver se movimentando para frente (ve-

locidade nominal positiva), obtém-se a dureza dos objetos por meio dos sensores de contato esquerdo e direito frontal da criatura, tomando-se o maior valor obtido entre eles. Isso garante que se a dureza de um deles for igual a 1, o que indica uma iminente colisão frontal com um obstáculo intransponível, a criatura terá sua movimentação bloqueada. Neste caso, o controlador da criatura deve colocar um valor negativo na variável *speed*, para que a criatura volte a se movimentar.

Com a criatura se movimentando para trás, o procedimento é semelhante, só que considerando-se a dureza dos objetos que se encontram sobre os sensores de contato esquerdo e direito traseiros da criatura, o que detecta também colisões em marcha a ré, e permite ainda, na possibilidade de uma colisão traseira, que a criatura possa ser movimentada para frente.

Outra variável não-controlável do sistema é ΔFuel , que corresponde à quantidade de energia fornecida à criatura. Essa variável é determinada de modo semelhante ao parâmetro *friction*. Nesse caso, são obtidos os valores de energia dos objetos que se encontram nos quatro sensores de contato e é determinada uma média desses valores.

A variável ΔFuel é calculada conforme o trecho de pseudo-código a seguir:

```
double  $\Delta\text{Fuel}$  = 0;
for (int i = 0; i < 4; ++i)
    if (contactSensors[i].getMaterial() != null)
         $\Delta\text{Fuel}$  += contactSensors[i].getMaterial().getEnergy();
 $\Delta\text{Fuel}$  /= 4;
```

Para viabilizar a simulação como um todo (comportamento autônomo da criatura em um ambiente), construímos uma interface gráfica associada a um controlador que atende a diferentes necessidades de configuração e monitoramento.

3.5 Descrição do Simulador

O simulador da criatura artificial pode ser dividido em duas partes, desacopladas entre si. A primeira parte corresponde à simulação do ambiente em si, além do corpo (avatar) da criatura, seus sensores e atuadores. A segunda parte corresponde ao sistema de controle da criatura. A conexão do sistema de controle com o ambiente de simulação se dá por meio de *sockets*, em ambiente de rede TCP/IP. Desta forma, a simulação do ambiente pode ser executada em um computador e o sistema de controle pode ser executado em outro computador. Esse desacoplamento entre sistema de controle e simulação do ambiente permite que diferentes controladores possam ser desenvolvidos e testados, além de permitir uma melhor visualização da simulação, onde em um computador vemos

o que acontece no ambiente, e no outro podemos monitorar as variáveis internas do controlador, seu modelo de mundo sendo construído e seus planos de atuação.

O simulador de ambiente possui um grande número de parâmetros que podem ser configuráveis pelo usuário para preparar uma simulação:

- em relação ao ambiente: configuração do *socket* (endereço IP do controlador), tamanho do ambiente;
- em relação à criatura: tamanho da criatura, posição inicial, *pitch*, *wheel*, velocidade inicial;
- em relação aos objetos do ambiente: definição de objetos e suas propriedades físicas (“cor”, “dureza”, “gosto”, “energia”).

Configurados os parâmetros, pode-se iniciar o processo de simulação. Neste momento a criatura nada conhece sobre o ambiente, ou seja, seu modelo de mundo é inicialmente nulo.

Por meio dos sensores de contato e sensor visual, a criatura começa a descobrir o ambiente. Tal exploração ocorre mediante as propriedades dos objetos que ali estão. A partir dessa exploração, a criatura começa a construir um modelo do mundo. Esse modelo é continuamente atualizado, para incorporar as novas informações oriundas da interação da criatura com seu ambiente.

A Figura 3.8 representa a tela de configuração do *socket*, primeira caixa de diálogo do simulador, onde o usuário é instado a informar o endereço IP onde se encontra o sistema de controle da criatura. O controlador tanto pode rodar na mesma máquina que o simulador do ambiente, ou em outra máquina. A utilização de duas máquinas, entretanto, favorece a visualização dos resultados da simulação.



Figura 3.8: Caixa de diálogo do socket

Após a conexão com o sistema de controle, é aberta a janela principal do simulador de ambientes. Nesta janela, há dois menus que permitem a configuração de uma simulação: o menu *File* (que contém as opções *New*, *Open*, *Save* e *Exit*) e o menu *Options* (contendo *Add Object*, *Edit Car* e *Drive Car*) conforme a Figura 3.9 a seguir:

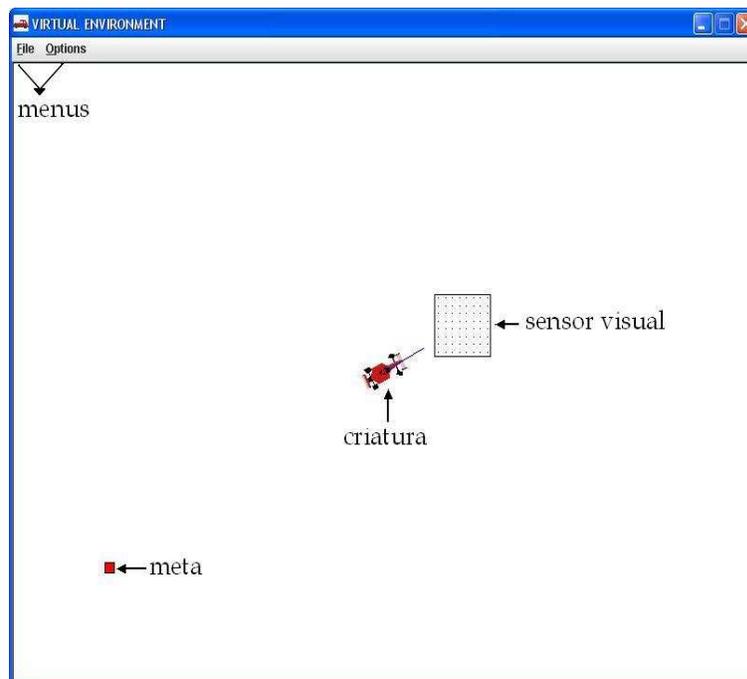


Figura 3.9: Tela inicial do simulador

3.5.1 Configuração do Ambiente

Utilizando-se a opção *New* (Figura 3.10) do menu *File*, o usuário pode escolher o tamanho do ambiente em *pixels*.



Figura 3.10: Caixa de diálogo para o tamanho do ambiente

Nos itens *Open* e *Save* (Figura 3.11) do menu *File*, o usuário poderá abrir ou salvar diferentes configurações de ambiente a qualquer momento. Desta forma, é possível configurar um ambiente com um tamanho determinado, preenchê-lo com diversos objetos, configurar as características de cada cor de objeto e aproveitar essa configuração em diferentes simulações.

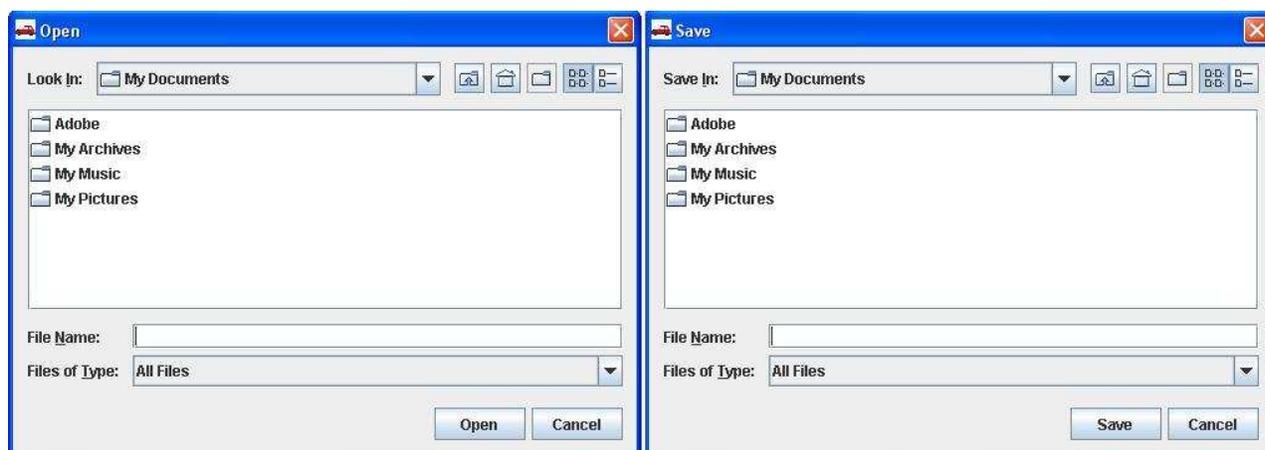


Figura 3.11: Caixas de diálogo para a opção abrir e salvar a simulação

Por meio da opção *Exit* do menu *File*, é possível encerrar o simulador de ambientes.

Por meio da opção *Add Object*, no menu *Options*, o usuário pode adicionar um objeto ao ambiente. Diferentes materiais, representados por cores, podem ser selecionados. Os objetos são adicionados na forma de retângulos traçados com o mouse na área de visualização do ambiente (Figura 3.12).



Figura 3.12: Add Object: escolha de materiais

Para editar o retângulo, basta o usuário clicar com o mouse sobre ele e aparecerão quatro pontos guias. Ao arrastar o ponteiro do mouse sobre um desses pontos, o usuário poderá alterar o tamanho e a posição do retângulo. Outra forma é clicar com o botão direito, quando aparecerá outra janela para definir com maior precisão o tamanho do objeto (Figura 3.13). Para eliminar o retângulo, basta clicar

duas vezes com o botão esquerdo do mouse sobre ele.

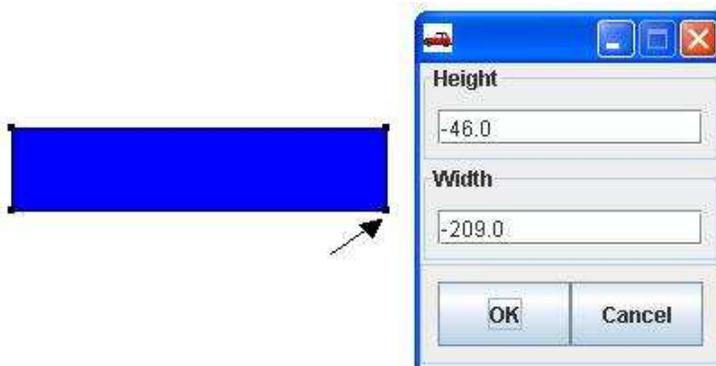


Figura 3.13: Caixa de diálogo: tamanho do objeto

Os valores de “dureza”, “gosto” e “energia” para cada material (cada cor) podem ser definidos pelo usuário. Para tal, deve-se clicar sobre o botão *Define* (Figura 3.12). Aparecerá então, uma nova caixa de diálogo (Figura 3.14), que permitirá ao usuário definir os valores de dureza - *hardness*, gosto - *taste* e fluxo de energia - *energy* associados a esse material (cor). Esses valores afetarão todos os objetos de mesma cor existentes no ambiente.

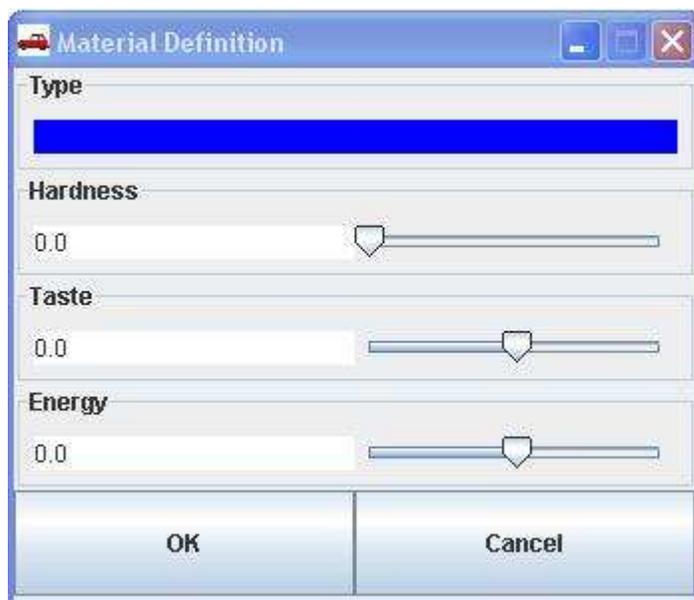


Figura 3.14: Definição das propriedades do objeto

3.5.2 Configurações do Veículo

Por meio da opção *Edit Car* no menu *Options* (Figura 3.15), o simulador entra em modo de edição da criatura.

Neste modo, o usuário pode ajustar a posição inicial da criatura, bem como seu ângulo de *pitch* e o próprio tamanho da criatura. Quanto à posição inicial da criatura e seu ângulo de *pitch*, estes também podem ser definidos mediante um clique sobre a criatura e posicionando-a onde se desejar no ambiente.

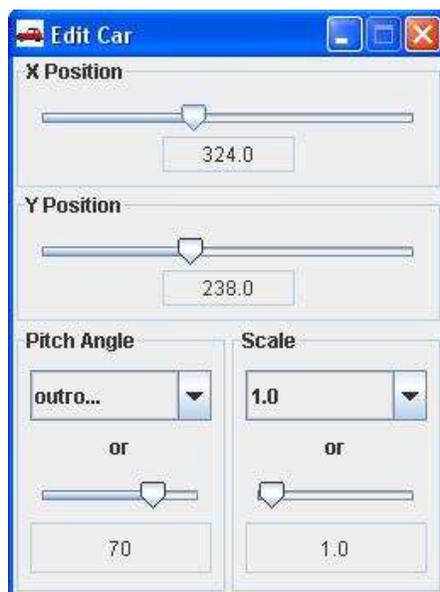


Figura 3.15: Edit Car: edição da criatura

Por meio da opção *Drive Car*, no menu *Options*, é possível iniciarmos uma sessão de simulação. Nesta situação, a janela mostrada na Figura 3.16 aparece, mostrando os valores dos sensores e atuadores correntes para a criatura, e exibindo também botões para o controle da simulação. Os valores dos sensores e atuadores é continuamente alterado, de acordo com a evolução da simulação.

O controle *Wheel* mostra a direção do ângulo das rodas da criatura, o botão *Zero/Wheel* permite que o usuário posicione a criatura no ângulo 0 que corresponde a criatura seguir em frente. O controle *Fuel* indica o nível da energia da criatura. O controle *Velocity* ao mesmo tempo mostra a velocidade da criatura e permite que o usuário a altere durante a simulação. O botão *Zero/Velocity* permite que o usuário interrompa a movimentação da criatura, sua velocidade cai a zero. O controle *Visual Scan* representa a imagem visualizada pelo sistema de controle, bem como os valores de comprimento (*Length*) e o ângulo do foco de atenção (*Angle*).

O controle *Contact* mostra o estado dos sensores de contato, mostrando a cor do objeto que se encontra em contato com a criatura, em cada uma de suas extremidades.

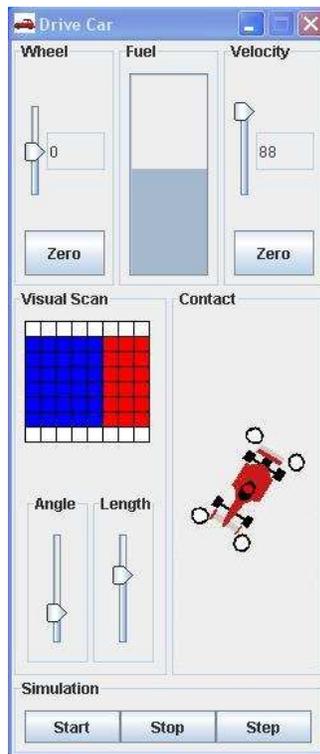


Figura 3.16: Drive Car

O painel auxiliar de controle (*Simulation*) permite que o usuário conduza a simulação por meio de três botões, permitindo-lhe começar (*Start*), terminar (*Stop*) ou executar passo a passo a simulação (*Step*).

3.5.3 Acompanhamento dos Dados Sensorio-Motores

O simulador permite um acompanhamento visual dos dados sensoriais da criatura, bem como dos dados de atuação, em tempo de simulação. Para isso, a mesma interface utilizada para configurar o veículo (Figura 3.16) é utilizada para monitorar os valores dos sensores e atuadores durante o andamento da simulação. Um exemplo da estrutura sensorio-motora da criatura em execução pode ser visto na Figura 3.17.

À esquerda, em um fragmento extraído da tela principal do simulador, vemos a criatura localizada em um ponto de seu ambiente colidindo com um objeto de cor amarela. Neste fragmento ainda, à direita da criatura, indica-se a posição do sensor visual da criatura, na forma de um *grid* de sensores dentro de um quadrado, onde cada sensor é indicado por um ponto. À direita, vê-se o painel de controle da criatura, onde diversos parâmetros podem ser observados: o ângulo da roda dianteira da criatura, o nível energético, a velocidade, o estado do sensor visual, sua posição em termos de ângulo

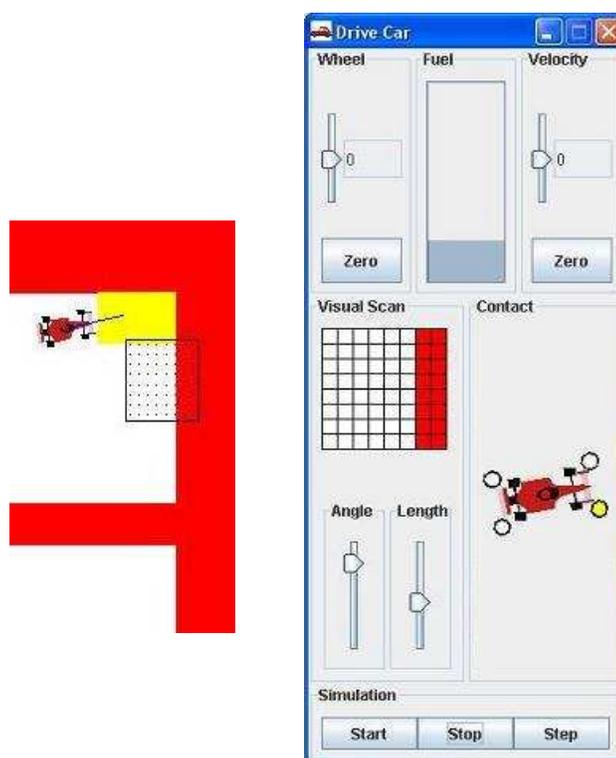


Figura 3.17: Estrutura sensório-motora da criatura

e distância, bem como os sensores de contato. Observe no sensor visual a detecção da parede em vermelho, e no sensor de contato a colisão com um objeto de cor amarela. Todos estes dados estão disponíveis como dados sensoriais ao agente implementado no controlador. Os dados de saída do controlador são o ângulo da roda e a velocidade da criatura. A criatura não possui um mapa *a priori* do ambiente, sendo que o mesmo é construído a partir da interação sensorial entre a criatura e o ambiente, durante sua navegação.

Baseada nas informações sensoriais colhidas, a criatura constrói um mapa do ambiente, de maneira incremental. Tal mapa é utilizado para gerar planos de movimentação sobre o ambiente.

3.5.4 Um Exemplo de Simulação

Para acompanhar o desenvolvimento de uma simulação, o usuário pode visualizar a movimentação da criatura na tela principal do simulador (Figura 3.18). Em nosso caso, existe ainda a possibilidade de se acompanhar os dados conforme os mesmos são processados pelo sistema de controle (Figura 3.19), onde outros dados, tais como o modelo atual do mundo levado em consideração, e outras variáveis internas do controlador podem ser visualizados.

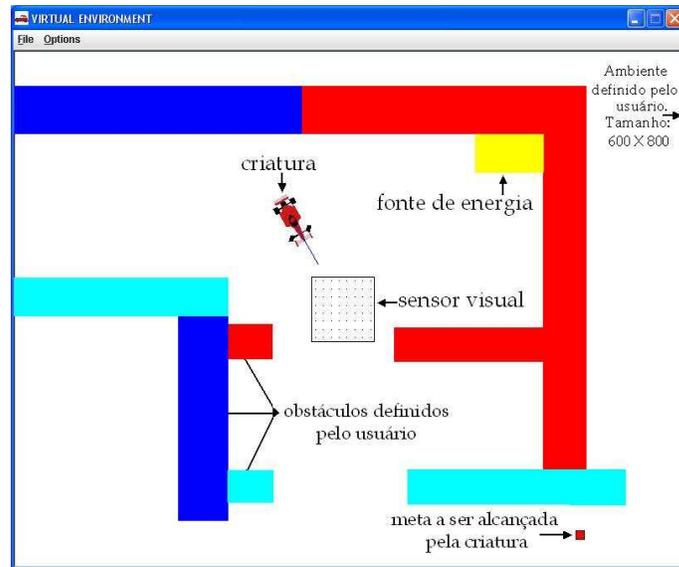


Figura 3.18: Um cenário possível visualizado na janela principal do simulador

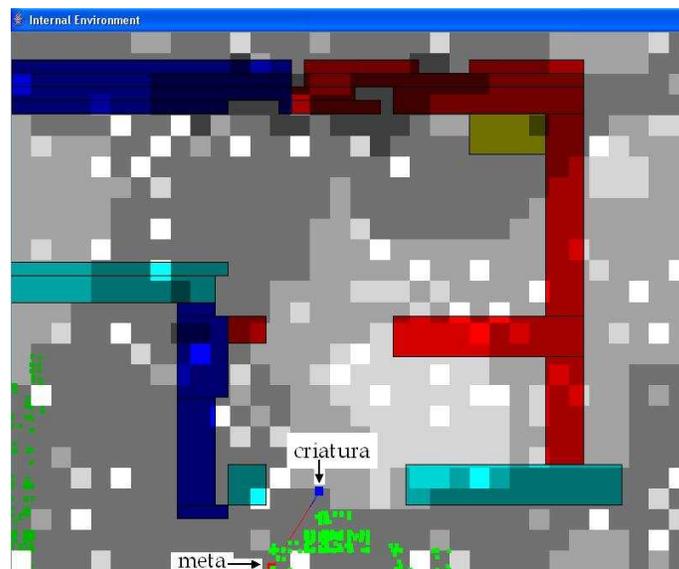


Figura 3.19: O mesmo cenário, visualizado a partir do sistema de controle

O detalhamento específico deste processo trataremos no Capítulo 4.

A Figura 3.20 demonstra uma outra característica interessante de nosso simulador. O ambiente possui uma estrutura toroidal. Com isso, caso a criatura se mova para fora dos limites do ambiente de simulação, ela aparecerá em posição diametralmente oposta no próprio ambiente. Por exemplo, na Figura 3.20, a criatura se move para fora do canto superior direito, e em seguida aparece em posição correlata no canto inferior direito.

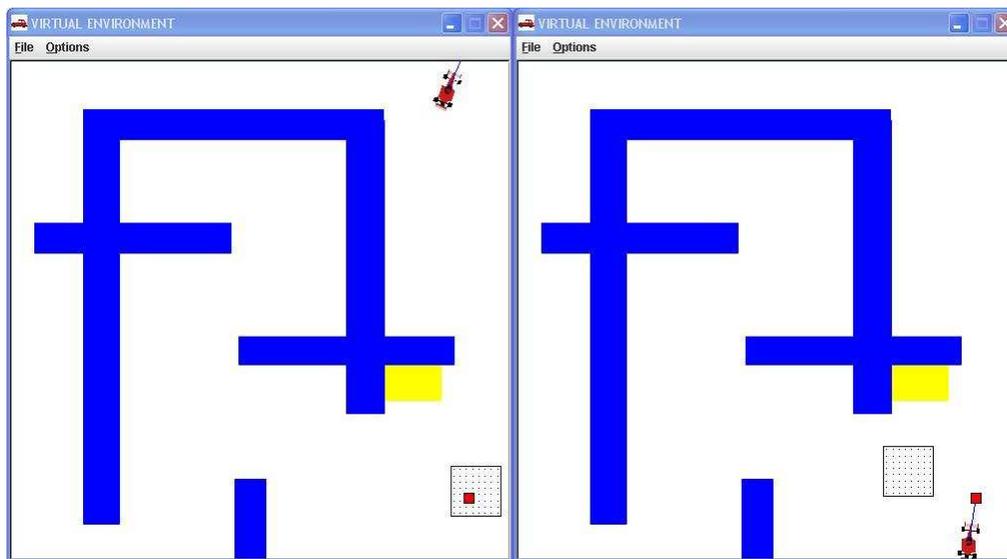


Figura 3.20: Movimentação ilimitada da criatura

3.6 Sistema de Controle Autônomo da Criatura

O sistema de controle autônomo da criatura será descrito em detalhes no Capítulo 4. Nesta seção, de forma a preparar o leitor, apresentamos algumas informações preliminares que podem ser úteis à compreensão do próximo Capítulo.

Como já dissemos anteriormente, o sistema de controle opera de acordo com dois níveis de controle, organizados hierarquicamente: o nível de Controle Direto e o nível superior de Decisão de Metas.

3.6.1 Nível de Controle Direto

O nível de Controle Direto é responsável por determinar, a cada instante de tempo, os valores dos dados a serem enviados aos atuadores da criatura. Este controlador utiliza uma arquitetura deliberativa

bastante sofisticada, desenvolvida originalmente em (Gudwin, 1996). Nesta seção, para facilitar a compreensão de seu comportamento, apresentamos os princípios fundamentais de seu funcionamento. Leitores interessados em uma descrição mais completa devem procurá-la em (Gudwin, 1996).

O sistema de controle direto é composto de diversos sub-sistemas operando em paralelo:

- sub-sistema de controle motor
- sub-sistema de controle visual
- sub-sistema de manutenção do modelo do mundo

O objetivo principal do controle direto é fazer o deslocamento da criatura, desde sua posição atual até uma posição final (chamada de “meta”), que deve ser fornecida como entrada do sistema de controle direto. Essa movimentação deve ser efetuada, em princípio de forma a se evitar colisões entre a criatura e os obstáculos existentes no ambiente.

Para deslocar a criatura no ambiente, existem as variáveis de atuação que determinam a velocidade da criatura (chamada de *speed*), e o ângulo da roda da criatura (chamada de *wheel*). Estas são as principais saídas do sub-sistema de controle motor. Entretanto, a determinação da velocidade e do ângulo da roda se utiliza de uma estratégia bastante sofisticada, baseada em planejamento. Este sub-sistema se serve de um “modelo de mundo” que considera objetos retangulares com atributos de posição e cor. Este modelo é originalmente vazio, sendo construído e adaptado pelo sub-sistema de manutenção de modelo do mundo, descrito um pouco mais à frente.

Na implementação realizada, a velocidade da criatura possui um módulo constante, determinado a partir da configuração da simulação. Dessa forma, a única variável determinada pelo sub-sistema de controle motor é o ângulo da roda. Para determinar o valor do ângulo da roda, utiliza-se a noção de um “plano”¹, construído na forma de um conjunto sequencial de pontos considerados “seguros” (ou seja, não há nenhum obstáculo conhecido que se interponha da posição de um ponto em linha reta até o próximo ponto), que levam a criatura de sua posição atual até a meta. A idéia de um plano é a idéia de se decompor o problema em sub-problemas mais simples. Assim, como não há obstáculos conhecidos entre dois pontos consecutivos de um plano, o problema de realizar o deslocamento entre esses dois pontos torna-se trivial. Basta alinhar as rodas em direção à meta, na metade do ângulo atual entre o alinhamento da criatura e a meta. A construção do plano segue o seguinte algoritmo²:

¹“Plano” aqui é utilizado no sentido de algo que é planejado, não no sentido de uma superfície plana.

²O algoritmo apresentado aqui é apenas uma simplificação do algoritmo real implementado, detalhado completamente em (Gudwin, 1996). A descrição detalhada deste algoritmo neste trabalho, poderia talvez trazer um nível de complexidade desnecessária aos nossos atuais propósitos, e eventualmente confundir o leitor. O algoritmo simplificado apresentado aqui permite uma compreensão geral das linhas básicas do algoritmo real, e é suficiente para que o leitor tenha uma idéia do funcionamento do sistema de controle direto sem entrar em detalhes que não são a preocupação imediata deste trabalho. O algoritmo real utiliza um mecanismo sofisticado de geração de pontos e arcos e de avaliação destes pontos e arcos que é bastante complexo, e sua explicação fugiria ao escopo do que aqui se pretende.

- Inicialmente tenta-se traçar uma linha reta a partir de sua posição atual até a meta. Caso isso seja possível, então o plano está concluído.
- Caso o passo anterior não seja possível, sorteia-se aleatoriamente um conjunto de direções ao redor do ponto atual, e para cada uma das direções projeta-se uma reta, que deve ou colidir com algum obstáculo do ambiente, ou com alguma das fronteiras do ambiente, ou com alguma outra aresta do plano. Na metade da distância entre o ponto original e a colisão, localiza-se um novo ponto. A partir de cada um desses novos pontos, tenta-se de maneira recursiva este mesmo algoritmo.

Este algoritmo gera uma árvore de pontos (Figura 3.21) que deve ter, como um de seus nós terminais, a meta (caso exista um caminho livre que leve da posição atual da criatura até a meta). Caso não seja encontrado um plano, a criatura se move aleatoriamente, procurando uma nova posição, e tenta de novo aplicar o algoritmo em um ciclo posterior.

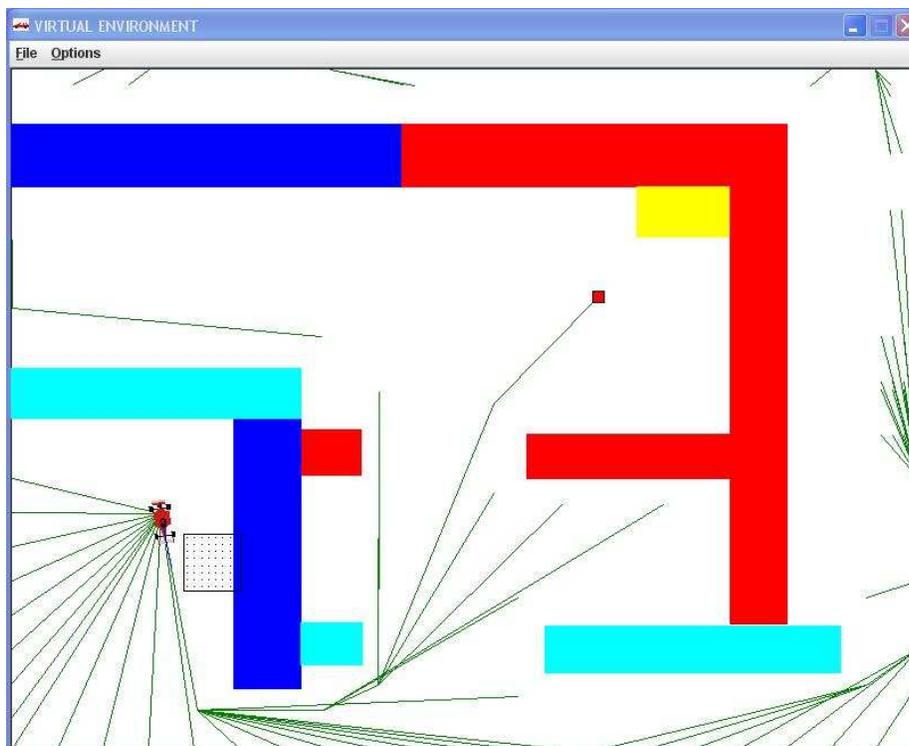


Figura 3.21: Exemplo de uma árvore de planos

Além do sub-sistema de controle motor, que determina a velocidade da criatura e o ângulo da roda, é atribuição do módulo de controle direto determinar o posicionamento do sensor visual, buscando capturar informações sobre os obstáculos do ambiente. O sub-sistema responsável por este controle é o sub-sistema de controle visual. Neste caso, as saídas de controle são as variáveis **lo** e **to**, que

determinam a posição do sensor visual, em relação ao eixo principal de orientação da criatura. A seguinte estratégia de controle é utilizada. Caso o sensor visual esteja posicionado em algum trecho do ambiente onde não há nenhum objeto, o sub-sistema de controle visual começa uma sequência de movimentos aleatórios, em um raio de -45° a 45° ao redor do eixo principal de orientação da criatura. Caso o sensor visual esteja posicionado sobre algum objeto do ambiente, o sub-sistema de controle visual tenta centralizá-lo, para que a máxima informação sobre o ambiente seja capturada. Em seguida, retorna à movimentação aleatória.

Por fim, o sub-sistema de manutenção de modelo do mundo é responsável pela construção e atualização do modelo do mundo. Inicialmente, o modelo do mundo é nulo. À medida que o sensor visual detecta objetos no ambiente, o sistema de percepção insere um novo objeto no modelo, que melhor se adapta à informação visual obtida. Esses objetos são sempre retângulos com cores definidas. Entretanto, na maioria das vezes, o sensor visual é capaz de capturar somente um fragmento dos objetos do ambiente, e não os objetos por completo. Desta forma, o sub-sistema de manutenção do modelo do mundo possui um algoritmo que fica sempre tentando integrar os objetos conhecidos, e por justaposição e pertinência, integra diversos objetos em um, refinando constantemente os objetos que compõem o modelo.

De uma maneira bastante sucinta, este é um resumo do nível de controle direto. Maiores informações, remetemos o leitor a (Gudwin, 1996), onde diversos outros detalhes sobre o nível de controle direto podem ser encontrados.

3.6.2 Nível Superior de Controle - Decisão de Metas

No nível de controle direto, assume-se que uma meta, na forma de uma coordenada (x, y) , deve ser fornecida, de forma que o sistema de controle direto gere os sinais de atuação que levarão a criatura de sua posição atual até a meta, sem colidir com os obstáculos do ambiente. Esse nível de controle visa solucionar um objetivo de curto prazo específico que é o deslocamento da criatura de maneira segura no ambiente. Entretanto, temos que recordar que a criatura deve também ter seus objetivos de longo prazo. Como vimos anteriormente, os objetivos de longo prazo sendo considerados neste trabalho são:

- Manutenção da integridade física da criatura
- Manutenção do balanço energético da criatura
- Aumento do conhecimento da criatura sobre seu ambiente

Destes objetivos, o único que é contemplado adequadamente pelo nível de controle direto é a manutenção da integridade física da criatura, uma vez que as rotas geradas por seu intermédio evitam a colisão com obstáculos do ambiente³.

Entretanto, para que se atinja o balanço energético da criatura e o aumento de seu conhecimento sobre o ambiente, é necessário que exista um nível superior de controle, que escolha de maneira adequada as metas que a criatura deve considerar, a cada instante do tempo. Este é portanto, o trabalho do nível superior de decisão de metas.

Esta decisão envolve, tipicamente, múltiplos objetivos. Em nosso caso, explorar o ambiente, regular a sensibilidade do algoritmo anti-colisão e cuidar da manutenção do nível de energia da criatura.

Nossa proposta para desenvolver o nível superior de controle, que é também a principal contribuição deste trabalho, é o uso de metáforas emocionais que são utilizadas para derivar o algoritmo de controle. Desta forma, cada um dos objetivos de longo prazo da criatura está associado a uma emoção, e o sistema de controle que guia o comportamento da criatura pode ser descrito como um sistema baseado em emoções. Esta é a razão do título deste trabalho: **Agente Emocional Hedonista**. O hedonismo se caracteriza aqui, pelo fato de que todo o comportamento do agente é guiado, tanto pelas emoções atuais da criatura, como sua perspectiva de emoções no futuro, quando suas metas forem alcançadas.

Servimo-nos portanto de três metáforas emocionais que são determinantes no comportamento da criatura: **Medo, Fome e Curiosidade**.

- **Medo**: relacionado ao comportamento anti-colisão;
- **Fome**: relacionado ao controle do balanceamento energético da criatura;
- **Curiosidade**: relacionado ao comportamento exploratório da criatura, fazendo-a escolher metas em pontos do ambiente que sejam desconhecidos.

A determinação da meta não é trivial porque dependendo da situação, diferentes comportamentos podem ser desejáveis, e a interação entre as emoções e uma eventual prioridade entre elas pode ser necessária. Por exemplo, caso o nível energético da criatura esteja alto, a Curiosidade deve ser prioritária à Fome, de tal forma que a criatura desenvolva um comportamento exploratório, procurando movimentar-se para áreas do ambiente que desconhece (ou seja, onde não há objetos conhecidos no mapa incremental gerado a partir do ambiente). Entretanto, caso o nível de energia esteja baixo, a Fome deve ser prioritária à Curiosidade, fazendo com que a criatura busque em seu mapa a fonte de

³Apesar disso, existem parâmetros no sistema de geração de planos que não foram detalhados aqui (veja mais em (Gudwin, 1996)) que podem regular a sensibilidade do algoritmo de geração de planos, fazendo com que o sistema passe a considerar rotas com maior ou menor risco à criatura. Isso pode fazer a diferença, por exemplo, entre o algoritmo encontrar ou não um plano quando a única rota entre a posição atual e a meta deve passar por uma passagem estreita.

energia mais próxima e se dirija até ela, com o intuito de reabastecimento. Após o abastecimento, a criatura pode retomar o comportamento exploratório anterior.

A questão do Medo diz respeito à regulação da sensibilidade do mecanismo anti-colisão. Caso a criatura nunca tenha colidido com qualquer objeto do ambiente (e tenha tido uma sensação desagradável como decorrência dessa colisão), o sistema anti-colisão pode não bloquear rotas que atravessem esses objetos. Uma vez que a criatura tenha já colidido com um obstáculo, e tenha experienciado uma sensação desagradável com isso, obstáculos dessa natureza (cor) já não serão considerados em futuras rotas traçadas pelo gerador de planos. Quando definiu o algoritmo de controle direto utilizado neste trabalho, Gudwin (1996) empregou um esquema de avaliação de pontos possíveis para uma rota baseada em um fator de utilidade que poderia ser comparável ao que entendemos por Medo neste trabalho. Em seu trabalho, entretanto, Medo era determinado heurísticamente, por meio de uma função de utilidade definida *a priori*. A idéia aqui é que esse fator possa ser regulado em função da interação da criatura com os objetos, considerando certos tipos de objetos como tangível (como por exemplo, objetos que fornecem energia) e outros intangíveis.

Utilizando a mesma metáfora emocional, desenvolvemos duas estratégias distintas de consideração das emoções, neste trabalho: uma baseada em um algoritmo heurístico e outra em um algoritmo genético. A primeira impõe explicitamente um conjunto de prioridades nas emoções, e a segunda tenta fazer um *blend*⁴ automático destas emoções. No Capítulo 4, a seguir, apresentaremos de maneira detalhada essas duas estratégias.

3.7 Resumo do Capítulo

Neste Capítulo, apresentamos o problema de simulação e propusemos estratégias de solução para ele. A estrutura geral do sistema de controle que desenvolvemos para o comportamento da criatura justificou a construção de um simulador, onde diferentes estratégias foram associadas ao comportamento da criatura por meio de emoções. No próximo Capítulo, apresentaremos uma descrição mais detalhada do nível superior de controle - Decisão de Metas, principal contribuição deste trabalho.

⁴Utilizamos aqui o termo em inglês “*blend*”, ao invés de simplesmente “combinação” em português, pois em nossa opinião o mesmo exprime de maneira mais completa a idéia de uma “combinação ponderada”, onde os pesos de cada parcela devem ser ajustadas de maneira fina para se obter os resultados de maior qualidade.

Capítulo 4

Sistema de Controle Hedonista de Decisão de Metas

4.1 Prólogo

Neste capítulo, apresentamos o detalhamento do sistema de controle superior, de decisão de metas, utilizando uma estratégia de controle que chamamos de controle hedonista. São apresentadas duas variações desta estratégia. A primeira, chamada aqui de algoritmo heurístico, considera as emoções com uma prioridade fixa entre si. A segunda, baseada em um algoritmo genético, faz um *blend* entre as diversas emoções e permite que novas emoções possam ser acrescentadas ao sistema sem o ônus de determinar todo um conjunto de prioridades entre as emoções, em diferentes situações, como no primeiro caso. Ao final do capítulo, são apresentados os resultados das simulações realizadas.

4.2 Controle Hedonista

Como apresentamos no capítulo anterior, o sistema de controle da criatura artificial foi dividido em dois níveis hierárquicos. O nível mais baixo considera basicamente um problema de navegação de curto prazo, ou seja, determinar a movimentação da criatura a partir de sua posição atual até uma meta (coordenada (x, y) localizada no ambiente). Este nível mais baixo, chamado de nível de controle direto foi implementado de maneira análoga à de (Gudwin, 1996). A entrada do nível de controle direto é uma coordenada (x, y) que deve ser gerada a partir do nível superior de controle. A escolha de diferentes metas (x, y) ao longo do tempo, deve levar em consideração os objetivos de longo prazo da criatura. Neste capítulo, apresentamos de maneira detalhada dois algoritmos utilizados para implementar o nível de decisão de metas. Ambos utilizam-se de uma técnica que chamamos de “controle hedonista”. Antes de iniciar a descrição destes algoritmos, justificaremos o nome que demos

à técnica e como ela funciona. Como dissemos no Capítulo 2, o hedonismo é uma teoria ou doutrina filosófico-moral em que as ações e tomadas de decisão são determinadas em relação ao prazer e à dor que proporcionam. Prazer e dor são duas abstrações para uma série de diferentes emoções que um ser humano (e também animais) podem exibir. O mecanismo emocional humano é bastante sofisticado, e como vimos no Capítulo 2, diversas são as tentativas de se apropriar do conceito de emoção de forma a aproveitá-lo no desenvolvimento de criaturas artificiais. Uma visão do conceito de emoção que nos parece especialmente interessante é a colocada por Marta Nussbaum em seu livro *Upheavals of Thought* (Nussbaum, 2001). De acordo com Nussbaum, o conceito de emoção está relacionado a um julgamento de valor, ou como ela coloca, com uma “percepção de valor”. Essa percepção de valor não é racional, ou seja, não é fruto de uma análise lógica da situação e sua subsequente avaliação. É uma capacidade de valoração que é instantânea, implícita do mecanismo perceptivo. Essa capacidade muito provavelmente desenvolveu-se geneticamente a partir da evolução animal, e nos permite tomar decisões sem que uma avaliação lógica da situação seja necessária. Devido a sua origem evolutiva, essa decisão nem sempre é a mais adequada, mas é útil na tomada de decisões em situações gerais, que permitiram que um certo conjunto de seres de uma mesma espécie pudesse sobreviver, ao passo que outros pereceram. Em nossa aplicação a cada emoção deve existir um objetivo ou propósito específico que é avaliado por meio da mesma emoção. Por exemplo, para a emoção **Fome**, o objetivo é a manutenção do balanceamento energético da criatura. Para a emoção **Medo**, o objetivo é a manutenção da integridade física da criatura, e para a emoção **Curiosidade**, o objetivo é o aumento do conhecimento da criatura sobre seu ambiente. Outras emoções devem ter objetivos específicos também associados. Desta forma, o comportamento da criatura pode ser completamente governado por um conjunto de emoções, ou seja, sinais sensoriais que exprimem uma percepção de valor, que orienta a tomada de decisões por parte da criatura. Neste caso, não existe uma avaliação lógica da situação por parte da criatura, mas ela tão somente reage aos sinais avaliativos gerados por seu sistema emocional. Podemos entender um sistema de controle com essas características como um controle baseado em emoções. E se pudermos agora, abstrair esse conjunto de emoções, de forma a ter apenas prazer e dor, podemos dizer que se trata de um controle hedonista.

Utilizamos esse conceito de um controle hedonista para gerar dois algoritmos diferentes, que se baseiam no mesmo princípio, mas que apresentam diferentes características. Essas estratégias baseiam-se na escolha de um conjunto de emoções que são especificadas para a criatura artificial, de forma que cada emoção esteja associada a um objetivo global para o funcionamento da criatura. A partir da definição destas emoções, construímos funções de avaliação dos dados sensoriais e de planejamentos antecipativos para esses dados. Tais funções devem ser capazes de avaliar diferentes metas (coordenadas (x, y)) para a criatura. A partir dessa avaliação, o sistema decide qual a coordenada (x, y) deve ser enviada ao nível de controle direto, a cada instante. Observe-se que difer-

entes emoções podem levar a condições de concorrência na determinação da decisão da criatura. Por exemplo, em uma situação de baixas reservas energéticas (muita fome) e um mundo totalmente desconhecido (muita curiosidade), a criatura deve dar prioridade ao balanço energético ao invés do aumento sobre o conhecimento do mundo. A diferença entre as duas técnicas apresentadas aqui ocorre principalmente devido à maneira como essa concorrência é administrada. No primeiro algoritmo, é administrada com a criação de uma prioridade específica e arbitrária definida para cada uma das emoções. Apesar de sua factibilidade e simplicidade, esse mecanismo cria um problema quando se pensa na escalabilidade do controle emocional. A inserção de novas emoções no sistema (ou seja, novos objetivos concorrentes) acaba por demandar uma análise de seu impacto e priorização diante de cada uma das outras emoções já instaladas na criatura. Para tentar resolver esse problema da escalabilidade de emoções, criamos um algoritmo genético, que tenta realizar essa priorização de maneira implícita, como veremos mais a frente.

4.3 Algoritmo Heurístico

Nossa primeira implementação de um controle emocional hedonista para a criatura artificial em questão foi baseada em uma solução heurística, ou seja, uma solução arbitrária onde um conjunto de emoções segue uma ordem pré-definida de prioridades na determinação da decisão de controle. Sabe-se que os métodos heurísticos não necessariamente fornecem soluções ótimas em termos de eficiência mas permitem chegar a soluções aceitáveis dentro de certos limites (Kim and Bae, 2004).

No algoritmo heurístico aqui implementado, as metáforas emocionais são implementadas diretamente sobre o código do algoritmo de controle da criatura.

Neste algoritmo, sorteia-se aleatoriamente um ponto que deverá ser a próxima meta. O algoritmo utiliza-se do Controle Direto definido em Gudwin (1996) para desenvolver um plano que movimenta a criatura até a meta. Esse passo implementa de maneira implícita os fatores Medo e Curiosidade que influenciarão o comportamento da criatura. O Medo é utilizado para determinar uma trajetória que contorne obstáculos, isto é, posições onde a criatura tenha medo de estar. Já a Curiosidade avalia o conhecimento que a criatura tem sobre o ponto sorteado. Caso o ponto sorteado inspire um alto índice de Medo ou um baixo índice de Curiosidade, um novo ponto é sorteado. O ciclo se repete até que níveis compatíveis de Medo e Curiosidade sejam atingidos (Figuras 4.1 e 4.2), quando isso ocorrer, uma nova meta é definida. Esse processo determina o comportamento padrão da criatura.

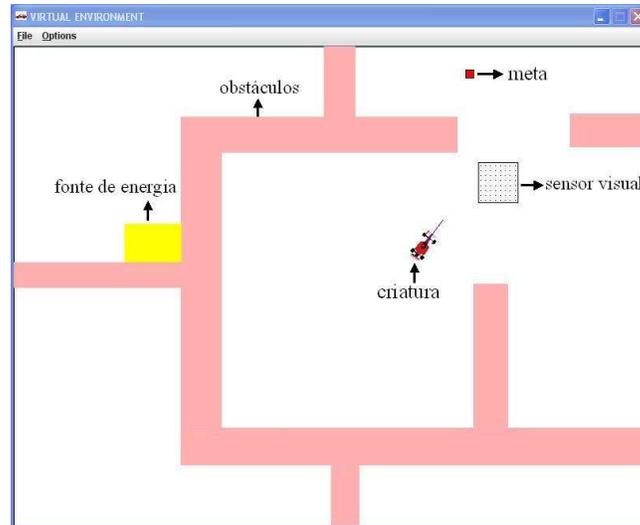


Figura 4.1: Visão do ambiente

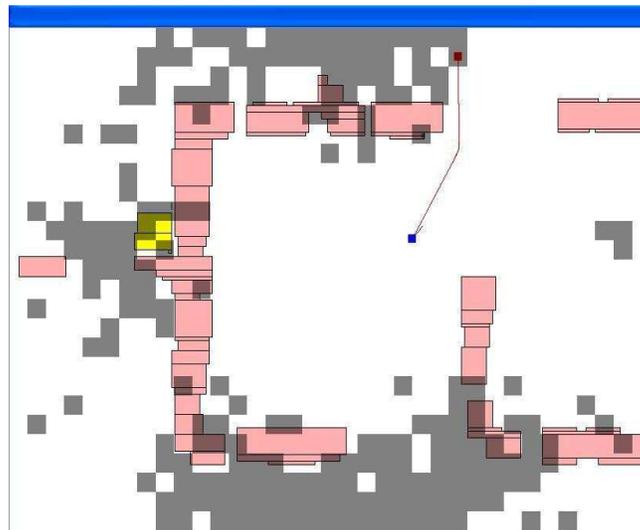


Figura 4.2: Representação do fator emocional curiosidade (em tom na cor cinza). Ocorre alto índice de curiosidade em lugares desconhecidos e baixo índice de medo.

Um terceiro fator relacionado ao comportamento da criatura diz respeito a seu balanceamento energético: o fator Fome. Caso o nível de energia torne-se inferior a 40%, um novo regime de comportamento passa a vigorar. A criatura abandona deliberadamente sua meta, e procura em seu mapa do ambiente, objetos que possam fornecer energia (Figuras 4.3 e 4.4). Passa a considerar novas trajetórias para esses objetos e escolhe o plano que leve ao objeto mais conveniente, não necessariamente o mais próximo uma vez que pode ser necessário contornar obstáculos na trajetória.

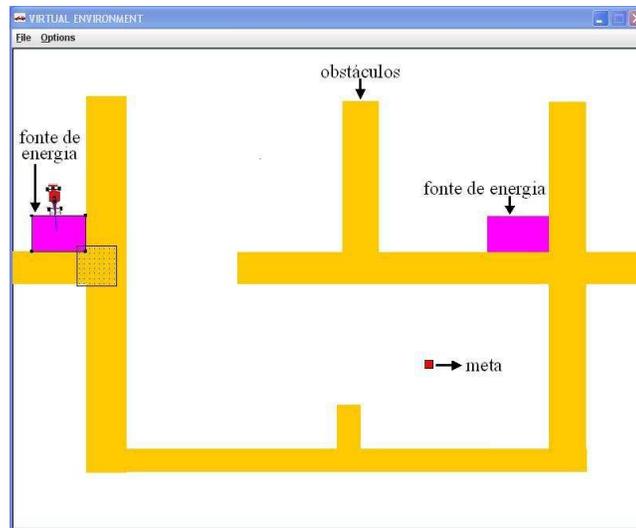


Figura 4.3: Visão do ambiente

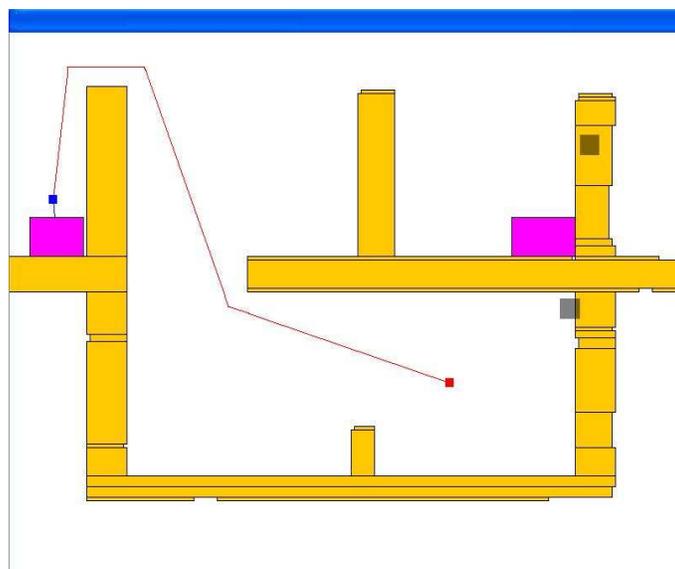


Figura 4.4: Fator emocional fome onde a prioridade é o abastecimento de energia

Após entrar em contato com esse objeto, a criatura reabastece seu nível de energia interna até atingir o máximo, e em seguida traça um novo plano para chegar à meta anterior (ponto vermelho), que havia abandonado quando da queda do nível energético abaixo de 40% (Figura 4.4).

A seguir, mostramos alguns trechos de código associados ao fator Fome:

Este método retorna verdadeiro se a quantidade de energia estiver abaixo do “seguro – 40%”, a criatura deverá procurar um objeto que possa fornecer energia.

```
public boolean needEnergy ( )
{
    return (fuel < 60);
}
```

A sequência seguinte controla a “parada” da criatura para abastecer. Se ela estiver precisando de energia e estiver sobre o objeto “fornecedor de energia”, faz com que a criatura fique parada no local para reabastecer.

```
if (needEnergy( )) {
    //Calcula a quantidade de energia adquirida/perdida na posição
    atual da criatura

    double totalEnergy = 0;
    for (int i = 0; i < 4; i++)
    {
        Material m = contactSensors[i].getMaterial( );
        if (m != null) {
            totalEnergy +=m.getEnergy( );
        }
    }

    //Se a posição atual fornece energia então a criatura para e
    espera o fim do abastecimento.

    abastecendo = totalEnergy > 0;
}
```

```
//Verifica se o abastecimento terminou
if (abastecendo && fuel >= 100)
{
    abastecendo = false;
}
```

Observamos que este algoritmo heurístico impõe explicitamente uma seqüência de prioridades para as emoções: 1-**Medo**; 2-**Curiosidade**; 3-**Fome**. Estas atuam conforme as seguintes regras:

1. **Medo** (de colidir com obstáculos): por meio do Medo, o algoritmo normalmente busca evitar colisões, caso haja a colisão, o agente hedonista eleva o “fator de cautela” do algoritmo;
2. **Curiosidade** (buscar pontos desconhecidos do ambiente): Curiosidade responde pela prioridade em escolher metas em lugares do ambiente ainda desconhecidos da criatura;
3. **Fome** (manter balanço energético): quando a energia está muito baixa, o fator Fome faz com que a criatura ignore a meta que deveria atingir e a substitua temporariamente pela busca do ponto fornecedor de energia mais conveniente.

O algoritmo heurístico possui claras limitações. Talvez a mais óbvia seja que a metáfora emocional, apesar de inspirar o desenvolvimento do algoritmo, não permite uma generalização a ser aplicada em situações diferentes. Por exemplo, caso se deseje sofisticar o comportamento da criatura implementando novas emoções, isso demandaria profundas mudanças na estrutura do algoritmo. A necessidade de determinar comportamentos distintos para situações de conflito como a que ocorre quando as reservas energéticas estão baixas, exige que, a cada caso, o algoritmo seja reanalisado e reorganizado. Além disso, a utilização de funções de utilidade para as diferentes emoções nem sempre é realizada de maneira eficaz. O tratamento de emoções conflitantes obriga a geração de exceções, que demandam quase um algoritmo próprio, a cada caso.

Para contornar essas restrições, partiu-se para a concepção de um algoritmo, baseado em um algoritmo genético, que pudesse ser generalizado de maneira uniforme, onde o acréscimo de novas emoções não demandasse toda uma reestruturação do algoritmo.

Uma das vantagens no uso de algoritmos genéticos é a simplificação que eles permitem na formulação e solução de problemas de otimização. O resultado é o algoritmo a seguir.

4.4 Algoritmo Genético

Para tentar contornar as limitações relatadas no item anterior, inerentes ao algoritmo heurístico, concebeu-se uma solução que, ao contrário da priorização arbitrária e *hard-coded* do algoritmo heurístico, tenta promover um *blend* de avaliações emocionais na determinação da decisão de controle. Esse *blend* é implementado por meio de um algoritmo genético.

Algoritmos genéticos (AG's) são muito utilizados para resolução de problemas complexos, com elevado número de variáveis e soluções possíveis. São aplicados também em problemas de otimização e aprendizado de máquinas.

A analogia dessa técnica com sistemas naturais deve-se a Holland que tomou emprestado da Biologia uma série de termos e processos adaptativos como população, indivíduo, cromossomo, *crossover*, mutação e geração, com o intuito de modelar mecanismos inspirados na evolução natural em sistemas artificiais (Goldberg, 1989).

Comparando AG's com outras técnicas, temos que:

- AG's priorizam regras probabilísticas em vez de determinísticas. Assim, nem sempre há garantia de que se encontrará a solução exata do problema, mas os AG's são capazes de encontrar boas aproximações para a solução;
- trabalha com uma população de pontos por meio da evolução paralela de possíveis soluções, enquanto os métodos de otimização tradicionais tratam um único ponto de cada vez;
- implementa um conjunto de parâmetros, em lugar de utilizá-los diretamente (Deb, 1999).

A definição de um AG demanda os seguintes passos:

- uma codificação (ou representação) de uma possível solução ao problema em questão, na forma de um cromossomo (*string* de parâmetros);
- uma população de cromossomos (definição de seu tamanho);
- uma função de avaliação (que avalia cada cromossomo quanto a sua qualidade em resolver o problema em questão);
- operadores genéticos (*crossover* e mutação, isto é, capacidade de alterar a composição dos indivíduos em uma população) (Mitchell, 1996).

O algoritmo genético que utilizamos neste trabalho segue o fluxograma apresentado na Figura 4.5.

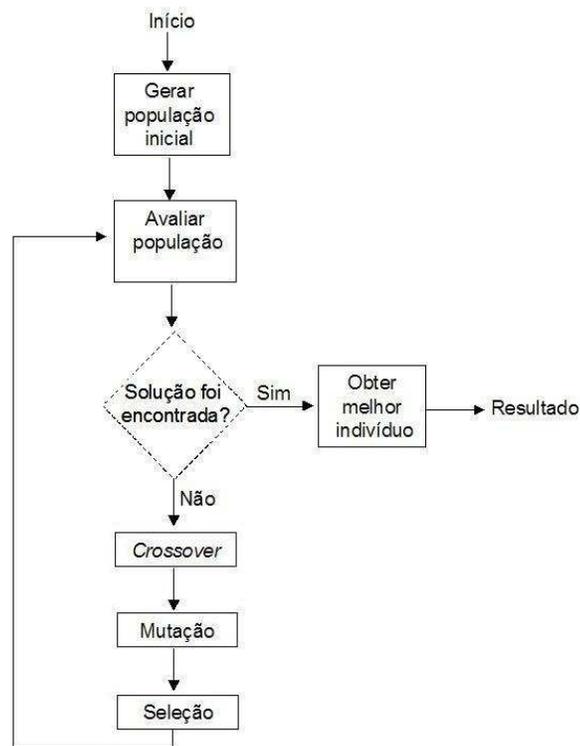


Figura 4.5: Algoritmo genético utilizado neste trabalho

4.4.1 Representação dos Indivíduos e Geração da População Inicial

Em nosso caso, adotaremos a abordagem de Pittsburgh (Bacardit, 2005), onde cada indivíduo da população é uma solução candidata para o problema tratado. Uma população inicial é formada por um conjunto aleatório de “n” indivíduos, onde cada indivíduo representa uma “meta” hipotética, um ponto para onde a criatura teria como objetivo se deslocar.

Considerando que cada indivíduo deve ser representado por um cromossomo único, adotamos como genes, o par x e y . Assim, cada cromossomo (x, y) corresponde a um par de números inteiros associados à posição do ponto no espaço bi-dimensional do ambiente.

4.4.2 Avaliação da População

Depois que uma população de metas hipotéticas é gerada, cada uma delas deve ser avaliada por meio de uma função de avaliação (as vezes também chamada de função objetivo, ou função de *fitness*). Desta forma, cada indivíduo da população se comporta de forma análoga a de um ser vivo em um ecossistema, competindo pela sobrevivência diante de seus adversários. O grau de adaptação e a sobrevivência de um indivíduo no processo evolutivo indicam a qualidade do mesmo em relação à solução que representa ao problema em questão. Em um algoritmo genético, a função de avaliação

da população é um ponto crítico do algoritmo.

Em nosso caso, utilizaremos nossa metáfora emocional para determinar a função de avaliação do algoritmo genético. A função de avaliação corresponderá a uma medida da “desejabilidade” de cada meta representada por um membro da população. Essa “desejabilidade” pode ser vista como uma abstração unificada de todo um conjunto de emoções combinadas na forma de um *blend* ponderado em função da situação presente da criatura. Em nosso experimento, de maneira análoga ao do algoritmo heurístico, utilizamos as emoções **Medo**, **Fome** e **Curiosidade** de forma a gerar a “desejabilidade”. Essa combinação de emoções pode ser melhor visualizada na Figura 4.6. Esse conjunto de emoções poderia entretanto ser estendido facilmente com a inclusão de novas emoções, ao contrário do que acontece no algoritmo heurístico.

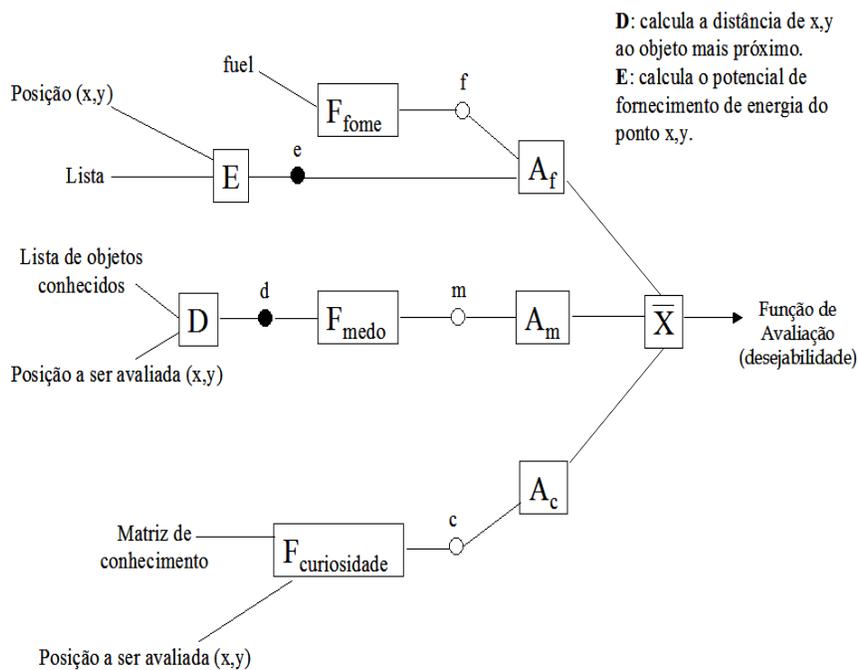


Figura 4.6: Função de avaliação

De acordo com a Função avaliação acima se pode relatar o seguinte esquema de funcionamento das funções emocionais, para se chegar a esta desejabilidade:

A partir da posição desejada (x, y) , da lista de posições dos objetos conhecidos, com suas respectivas energias, e da Função fome, chega-se a função atratividade por fome (A_f).

Para compor a desejabilidade também é utilizada a Função medo, que a partir da lista dos objetos conhecidos e da posição (x, y) a ser avaliada, calcula a função atratividade por medo (A_m).

A função atratividade por curiosidade (A_c) é calculada pela matriz de conhecimento (descrita a seguir) e a posição a ser avaliada (x, y) .

E finalmente a função de avaliação (desejabilidade) é composta pela média ponderada dos valores de saída das funções de Atratividade por Fome, Atratividade por Medo e Atratividade por Curiosidade.

Além disso utilizamos três outras funções auxiliares para calcular as atratividades:

- **Função de Medo:** Dada a distância entre a meta e o objeto conhecido mais próximo calcula o medo “m” para a posição. Metas muito próximas a obstáculos geram um medo maior. Metas mais distantes de obstáculos geram um medo menor;
- **Função de Fome:** Dado o estado energético da criatura, obtém-se a fome “f” da criatura. Se a criatura está em um estado de baixo nível de energia, a fome é maior, caso contrário, a fome é menor;
- **Função de Curiosidade:** Dado o mapa de posições conhecidas, a função calcula uma maior curiosidade “c” para posições desconhecidas que estejam mais próximas da posição atual.

A **Função fome** trabalha com a quantidade de combustível atual, e com a energia da meta, sendo que uma vez a criatura em contato com o objeto energético, a mesma se reabastece.

$$F_{fome} = (1 - fuel) \quad (4.1)$$

$$A_f(x, y) = F_{fome} * E(x, y) \quad (4.2)$$

A **Função medo:** Dada uma possível meta, posição da meta (X_n, Y_n) a criatura descobrirá as coordenadas (X_m, Y_m) do objeto mais próximo da meta. Se (X_n, Y_n) estiver dentro do objeto a Função medo será:

$$F_{medo} = 1.0, \quad (4.3)$$

que representa o maior valor que a F_{medo} pode assumir.

Senão, F_{medo} será descrita pela expressão abaixo:

$$F_{medo}^{(d)} = H * e^{-0,4*d}, \quad (4.4)$$

sendo:

- H = constante de dureza do objeto, $0 \leq H \leq 1$
- 0,4 = constante de atenuação da distância
- d = distância entre a meta e o objeto

$$0 \leq F_{medo} \leq 1 \quad (4.5)$$

Um ponto que causa medo, terá um valor próximo de 1, levando à diminuição da desejabilidade.

$$A_m(m) = -m \quad (4.6)$$

A **Função curiosidade** é calculada utilizando-se uma “matriz de conhecimento”, que codifica o ambiente em zonas, registrando o número de vezes em que a criatura já esteve em cada zona. Inicialmente, como a criatura não conhece nada sobre o ambiente, a matriz de conhecimento encontra-se totalmente zerada. Cada zona ou “célula” cobre uma área do ambiente de 24 x 24 pixels.

A medida que a criatura se movimenta no ambiente, obtém-se a posição média do Sensor Visual, e calcula-se em qual zona este ponto se encontra. Caso uma nova zona seja sensoreada, atualiza-se o valor correspondente na matriz de conhecimento, incrementando-se seu valor em 1. O valor da função curiosidade é calculado então seguindo-se a fórmula a seguir:

$$F_{curiosidade[i,j]} = 1 - \frac{M_{[i,j]}}{K} \quad (4.7)$$

onde M é a Matriz de conhecimento, $M_{[i,j]}$ indica o número de vezes que a criatura sensoriou a zona correspondente a (i,j) e $K = \max_{\forall i, \forall j} M_{[i,j]}$.

$$A_c(c) = c \quad (4.8)$$

4.4.3 Parada do Algoritmo

Feita a avaliação da população, é necessário verificar se o critério de parada do algoritmo genético foi atendido. A cada iteração do algoritmo, diversas metas hipotéticas competem entre si. No início do algoritmo, estas metas estão distribuídas aleatoriamente por todo o ambiente. A cada passo evolutivo, essas metas vão sendo confinadas, cada vez mais, a regiões do espaço onde a “desejabilidade” é maior. Depois de um certo número de passos evolutivos (considerado como um parâmetro do algoritmo), o indivíduo com maior “desejabilidade” é escolhido como a meta definitiva da criatura.

Caso não se tenha chegado a esse número limite, o algoritmo deve continuar, promovendo alterações na estrutura da população.

4.4.4 Operador genético: *Crossover*

A operação de *crossover* promove a recombinação dentre dois indivíduos selecionados da população, gerando um novo indivíduo, que será considerado posteriormente pelo algoritmo. Em um algoritmo genético tradicional, essa recombinação se dá pela permuta entre genes escolhidos aleatoriamente entre os dois indivíduos. A lógica do *crossover* como operação de modificação de elementos da população é tentar se utilizar de “pais” bem qualificados, aproveitando diferentes características de cada um deles de forma a tentar formar um indivíduo melhor. Entretanto, em nosso caso, como nossos cromossomos possuem somente 2 genes (2 parâmetros, x e y da coordenada da meta), a utilização de técnicas como o *crossover* simples ou mesmo o *crossover* uniforme¹ não trariam muitos benefícios. Se dois cromossomos constituem 2 bons candidatos a metas, talvez a melhor recombinação que poderia haver entre eles seria tentar encontrar um novo ponto entre o segmento de reta que conecta os dois pontos “pais”. Desta forma, o *crossover* utilizado neste trabalho efetua uma combinação linear entre os dois pontos pais, conforme mostra a Figura 4.7.

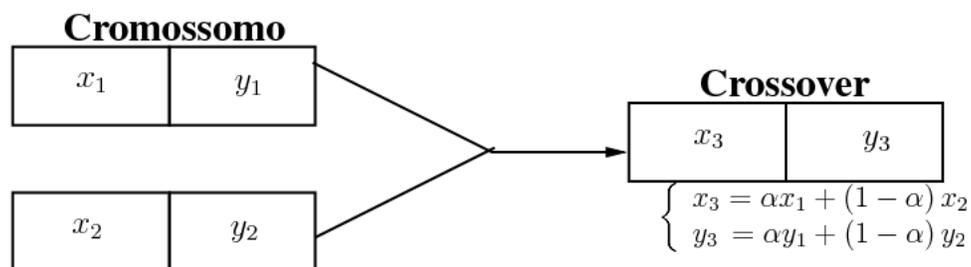


Figura 4.7: Operador de crossover

4.4.5 Operador genético: Mutação

A lógica do operador de mutação é se aproveitar de um indivíduo de boa qualidade, promover pequenas mudanças aleatórias em sua constituição e tentar gerar um indivíduo de qualidade ainda melhor. Pelos mesmos motivos apresentados no caso do *crossover*, adotamos uma operação de mutação diferenciada. Desta forma, a mutação utilizada visa deslocar uma possível meta em torno de sua vizinhança. O operador de mutação sorteia um indivíduo da população, segundo seu *fitness* e aplica sobre ele um pequeno deslocamento aleatório em sua coordenada (x, y) . O novo indivíduo é então incorporado à população.

¹Crossover simples: é um operador onde indivíduos trocam partes de seus genes e as novas gerações herdam características dos seus pais. Crossover uniforme: é o operador capaz de recombinar quaisquer posições entre dois genes.

4.4.6 Seleção

Uma vez que a população de indivíduos tenha sido aumentada a partir das operações de *crossover* e mutação, é necessário agora aplicar-se algum tipo de seleção, que preserve alguns indivíduos dessa população aumentada e descarte outros, de forma que a população possa permanecer constante.

Adotamos neste trabalho um procedimento de seleção chamado de elitista, isto é, aquele que seleciona os melhores indivíduos da população e os envia para a geração seguinte sem realizar sorteio. O método elitista elimina o risco de perda da melhor solução.

Vários parâmetros do algoritmo genético contribuem para melhorar o seu desempenho, entre eles o tamanho da população, o número de gerações, a probabilidade associada aos operadores genéticos *crossover* e mutação, bem como o procedimento de seleção utilizado.

Para encontrar valores adequados para esses parâmetros, é necessário que se promova uma bateria de testes empíricos.

4.5 Cenários de Aplicação dos AG's

As Figuras 4.8, 4.9 e 4.10 mostram uma simulação da criatura, utilizando o algoritmo genético. A Figura 4.8 ilustra um determinado ambiente, onde a criatura deve navegar. As Figuras 4.9 e 4.10 representam o mapa interno construído pela criatura, em diferentes instantes na simulação. A Figura 4.9 mostra uma situação no início da simulação, onde as áreas escuras apresentam grande “desejabilidade”, pois são áreas desconhecidas, e existe grande Curiosidade nessas áreas. Observe como a população de metas se distribui sobre essas áreas (pontos verde-claro). A Figura 4.10 mostra uma situação onde a simulação já avançou, e a criatura já teve a oportunidade de conhecer outras áreas. Nesse caso, o número de áreas escuras é bem menor, e as metas ficam concentradas em focos menores.

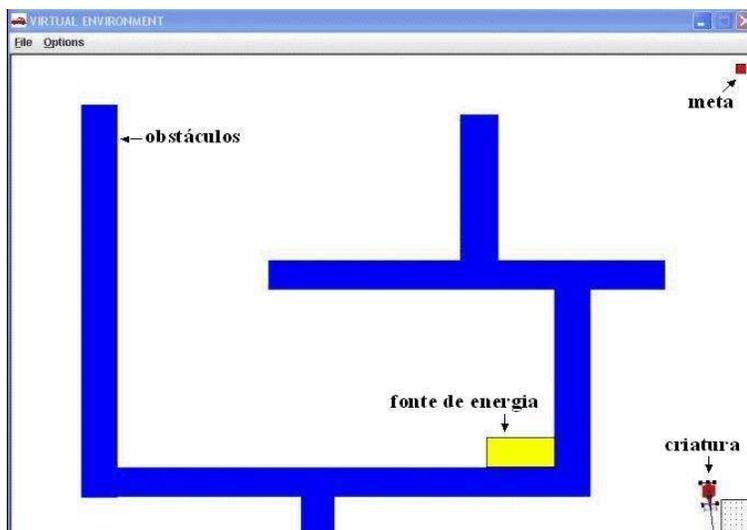


Figura 4.8: Visão do ambiente



Figura 4.9: Representação do fator emocional curiosidade (em diferentes tons na cor cinza)

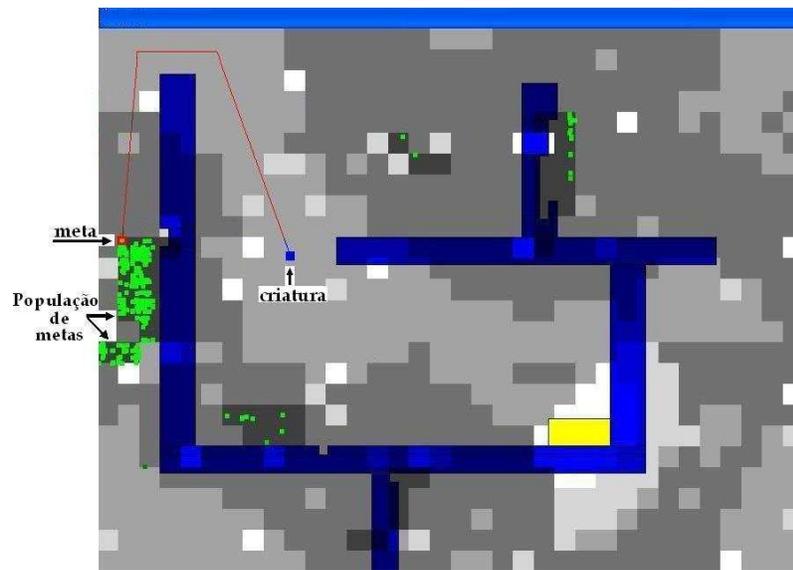


Figura 4.10: Situação otimizada: criatura com alto grau de conhecimento sobre o ambiente

A Figura 4.11 mostra uma outra simulação (em outro ambiente), em uma situação quando as reservas energéticas da criatura estão baixas, o que faz com que as áreas onde existe o fornecimento de energia atinjam uma maior “desejabilidade”. Observe a concentração de metas sob os objetos fornecedores de energia.



Figura 4.11: Fator emocional fome onde a prioridade é o abastecimento de energia

4.6 Resultados

A partir do algoritmo heurístico e do algoritmo genético, implementamos diversos testes, visando caracterizar o comportamento de ambos os algoritmos. Para estudar o desempenho dos dois métodos e poder compará-los de forma quantitativa, utilizamos um mesmo cenário para todos os exemplos abaixo.

4.6.1 Algoritmo Heurístico

Na Figura 4.12 a seguir, podemos ver a distribuição no tempo do fator Fome.

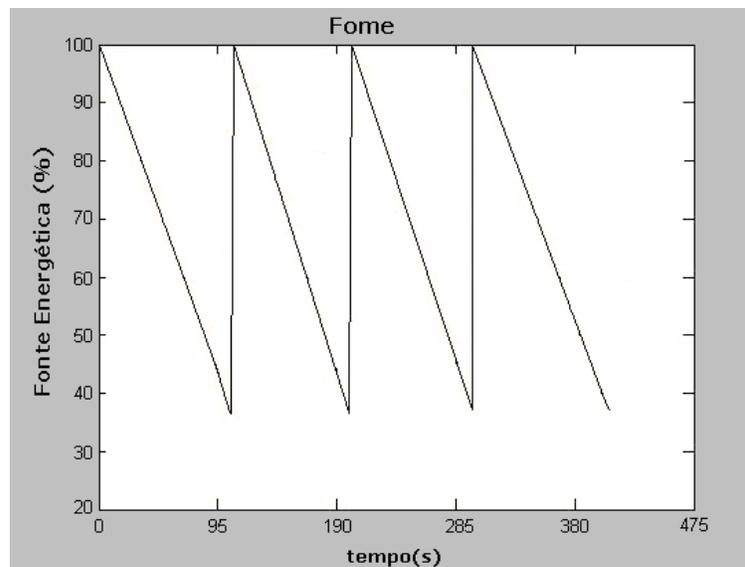


Figura 4.12: Distribuição no tempo do fator emocional fome - Heurístico

A simetria do gráfico do fator Fome, revela um comportamento cíclico e bastante previsível. Quando o nível energético da criatura atinge 60% da sua capacidade, esta ignora sua meta, e a substitui temporariamente pelo ponto fornecedor de energia mais próximo. Pode então haver mais alguns momentos de queda de energia, até que quando chega por volta de 40%, criatura encontra um fornecedor de energia.

Na Figura 4.13 a seguir, podemos ver a distribuição do fator Medo em função do tempo.

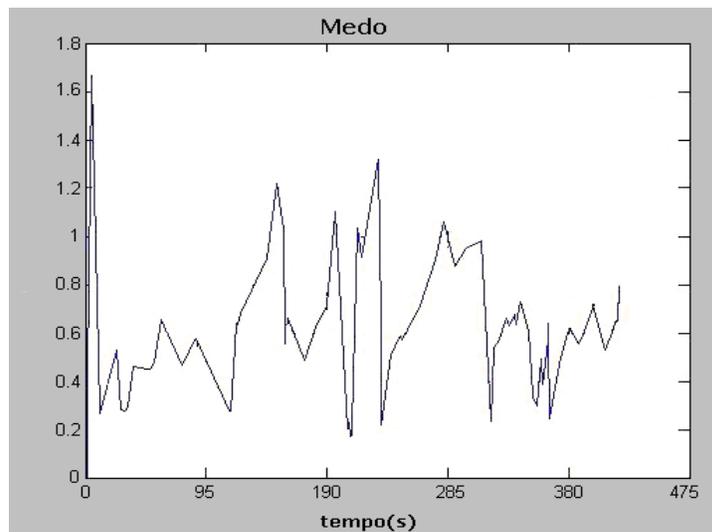


Figura 4.13: Distribuição no tempo do fator emocional medo - Heurístico

Observando a Figura 4.13, notamos grande variabilidade do fator Medo ao longo do tempo. Tal comportamento não nos permite aferir um padrão.

Na Figura 4.14 podemos ver a distribuição do fator Curiosidade ao longo do tempo.

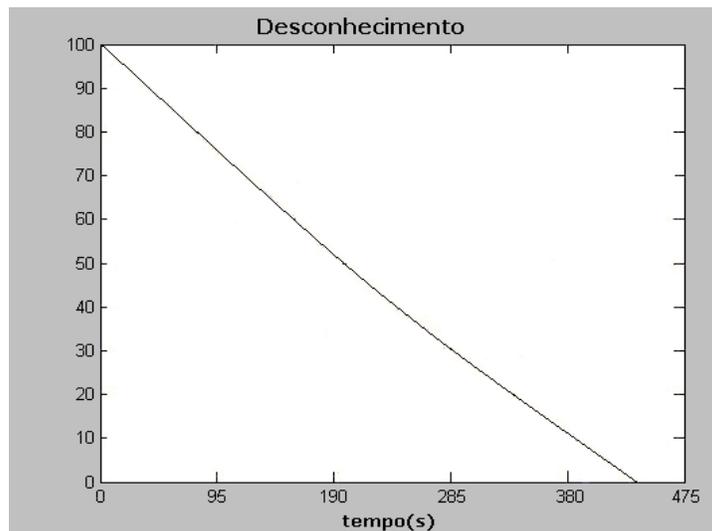


Figura 4.14: Distribuição no tempo do fator emocional curiosidade, nível de desconhecimento da criatura sobre o ambiente - Heurístico

Notamos um decaimento quase linear devido a sua implementação baseada em regras fixas.

4.6.2 Algoritmo Genético

Testamos o algoritmo genético para populações com 1000, 3000 e 5000 indivíduos em 15 gerações, com probabilidade de *crossover* 0,5 e probabilidade de mutação 0,1.

No início da simulação, a criatura possui energia máxima (100%). Ao longo do tempo, a energia cai e se eleva somente quando a criatura encontra um objeto dotado de energia. A mudança de prioridade no comportamento da criatura ocorre quando o nível de energia cai a 40% do total (Figura 4.15).

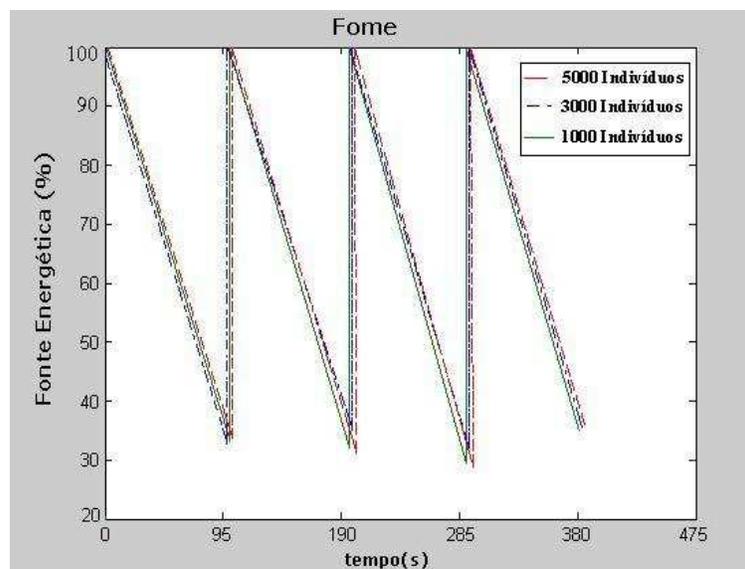


Figura 4.15: Distribuição no tempo do fator emocional fome - AG

Podemos observar que os gráficos obtidos revelam um efeito “gangorra” de decaimento e elevação do nível de energia e isso ocorre de forma linear.

Para plotar os gráficos do fator Medo, selecionamos o obstáculo com menor distância em relação à criatura (Figura 4.16).

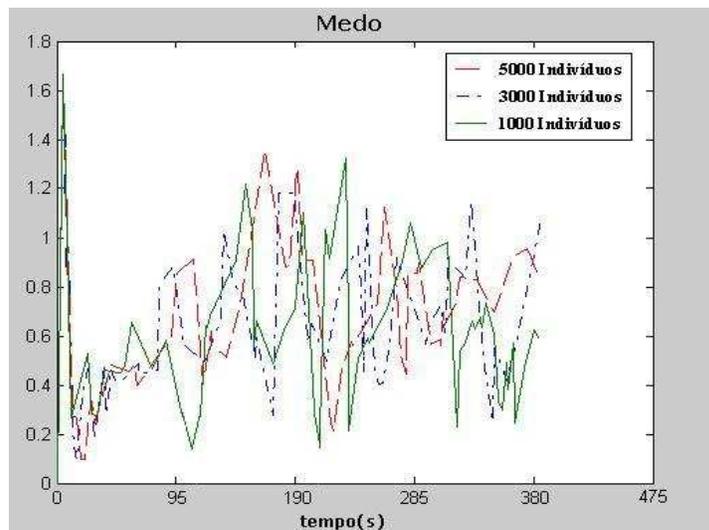


Figura 4.16: Distribuição no tempo do fator emocional medo - AG

Podemos observar que os gráficos obtidos não definem um padrão visível pois cada um deles representa uma multiplicidade de situações de elevação e decrescimento do fator Medo em função do tempo, conforme diferentes obstáculos são considerados ao longo do comportamento da criatura.

Para plotar os gráficos do fator Curiosidade, utilizamos relações entre áreas conhecidas e desconhecidas do ambiente ao longo de um intervalo de tempo. No início a criatura nada conhece do ambiente (Figura 4.17).

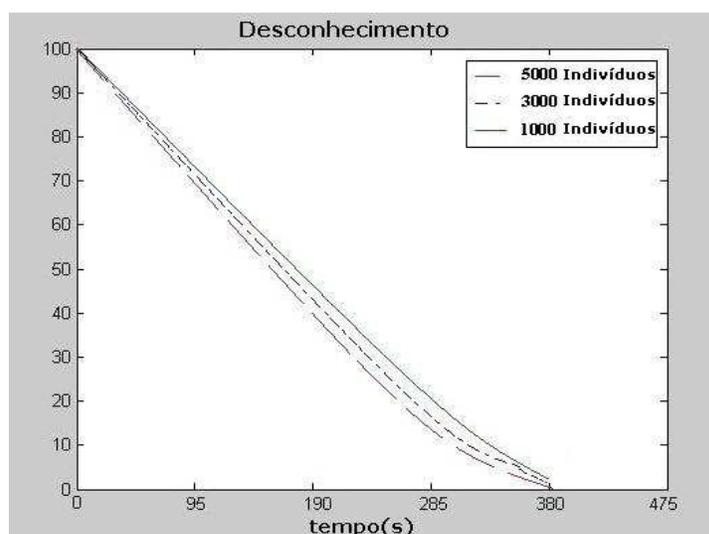


Figura 4.17: Distribuição no tempo do fator emocional curiosidade, nível de desconhecimento da criatura sobre o ambiente - AG

Podemos observar que os gráficos obtidos possuem decaimento da Curiosidade não linear.

4.6.3 Comparação entre os Algoritmos

Apresentamos uma comparação entre o desenvolvimento das duas estratégias utilizadas nesta dissertação: uma baseada em um algoritmo heurístico e a outra em um algoritmo genético (Tabela 4.1).

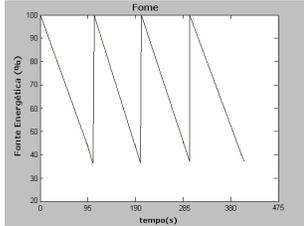
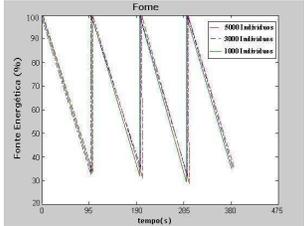
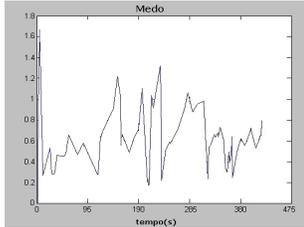
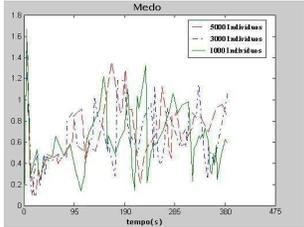
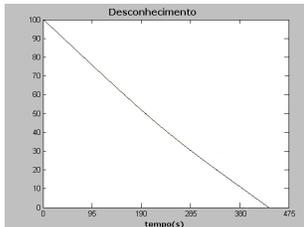
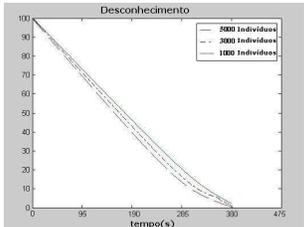
| Fator Emocional | Algoritmo Heurístico | Algoritmo Genético | Comentário |
|--------------------|---|--|---|
| Fome |  |  | Comportamento semelhantes porém quando o nível de energia cai a 40%, a reação do Heurístico é previsível |
| Medo |  |  | Grande variabilidade detectada em ambos os métodos |
| Curiosidade |  |  | Enquanto o fator Curiosidade no Algoritmo Heurístico é quase linear, o mesmo não acontece com o Algoritmo Genético |

Tabela 4.1: Comparação entre os algoritmos

Podemos observar que em ambos os casos, a criatura aumenta seu conhecimento sobre o ambiente até 100% após algum tempo. Neste caso, nós detectamos uma diferença pequena na eficiência entre muitos algoritmos que simulamos. Os algoritmos heurísticos são menos eficientes, ganhando o conhecimento total do ambiente somente em torno de 430s. Já o algoritmo genético com tamanho da população de 5000 indivíduos alcança o mesmo 100% em torno do 380s. O algoritmo genético com

tamanhos da população de 1000 e 3000 indivíduos executou também com o melhor desempenho do que o algoritmo heurístico.

Como uma avaliação geral, nós podemos concluir que ambos, os algoritmos heurístico e genético possuem um comportamento similar, conseguindo seus objetivos principais de exploração do ambiente, de manutenção do seu nível de energia e de evitar colisões com os obstáculos, com uma pequena diferença no algoritmo genético. Mas, no outro lado, o algoritmo genético tem uma maior vantagem em termos de escalabilidade. Para o algoritmo heurístico, cada nova emoção que desejaríamos incluir, todo o algoritmo necessitaria de um replanejamento. No caso do algoritmo genético, somente uma mudança pequena na função do desejabilidade é requerida, caso pretendêssemos incluir uma nova emoção. Como conclusão, nós podemos definir que o algoritmo genético é mais vantajoso do que heurístico.

A Tabela 4.2 a seguir apresenta estatísticas para o tempo necessário para o cálculo da meta, a partir de diferentes situações. As simulações foram realizadas em plataforma *Windows*, em máquinas Athlon XP 2400, com 1GB de memória disponível.

| Gerar Meta | Algoritmo Heurístico | Algoritmo Genético 1000 Indivíduos | Algoritmo Genético 3000 Indivíduos | Algoritmo Genético 5000 Indivíduos |
|---------------------|-----------------------------|---|---|---|
| Tempo Mínimo | 693 ms | 1151 ms | 6255 ms | 33228 ms |
| Tempo Máximo | 1064 ms | 1956 ms | 8508 ms | 42142 ms |
| Média | 878,5 ms | 1553,5 ms | 7381,5 ms | 37685 ms |

Tabela 4.2: Tempo de execução do ciclo de decisão

4.7 Discussão

Conforme mencionamos no Capítulo 2, uma vasta gama de autores (Ortony et al. (1998); Ortony (2003); Picard (1997, 2006); Sloman (1998, 2001); Cañamero (1997, 1998, 2000, 2001, 2005); Velásquez (1998); Sarmiento (2004); Sarmiento et al. (2004) e Balkenius (1995)) têm desenvolvido trabalhos que envolvem de uma maneira ou de outra o conceito de emoções. Tais trabalhos são motivados não apenas pelo caráter otimizador que as emoções exercem sobre o desempenho de um agente em um ambiente (equacionamento de restrições internas e externas), mas também como forma de estudar diferentes teorias de emoções em ciência cognitiva.

Embora já tenhamos rapidamente abordado os autores aqui mencionados no Capítulo 2, retomamos aqui algumas idéias relacionadas às emoções com o intuito de enquadrar melhor o trabalho

desenvolvido nesta dissertação e rastrear indícios de noções que nós mesmos desenvolvemos na arquitetura do Agente Emocional Hedonista.

O modelo OCC de Ortony et al., (1998) é um dos mais referenciados na literatura. Observamos neste modelo, que os eventos e suas conseqüências têm um potencial de impedir ou promover objetivos trazendo, de certa forma, uma idéia de “desejabilidade”. Em nosso trabalho, a “desejabilidade” resulta de um conjunto de funções de utilidade, cada uma relacionada a um fator emocional que desejamos implementar: Medo, Fome e Curiosidade. A esse feixe de funções, chamamos “Função de Desejabilidade”.

No desenvolvimento da arquitetura do Agente Emocional Hedonista, trabalhamos com a idéia da ação do agente sob influência das emoções. A forma como caracterizamos as emoções em nossa arquitetura possui paralelo com a definição de síntese das emoções definida por Picard (1997).

Outro trabalho que nos chama a atenção é o de Sloman (1998, 2001). Nele, os estados emocionais operam como filtros capazes de alterar prioridades e arbitrar interrupções no comportamento do agente a fim de alocar recursos onde estes são mais necessários e ao mesmo tempo, proteger a atividade cognitiva em curso. Os vários componentes do sistema trocam informações e determinam mudanças de estratégia no processamento central do agente. A arquitetura proposta por Sloman inclui as Camadas Reativa (respostas automáticas), Deliberativa (capacidade de planejamento) e Meta-Gerenciamento (auto-avaliação e interação social). Em nosso trabalho, utilizamos uma arquitetura que entendemos como híbrida pois inclui um nível de Controle Direto responsável pelo deslocamento da criatura e um nível de Controle de Decisão de Metas, onde entram múltiplos objetivos, dentre eles explorar o ambiente, manter energia em nível favorável e regular o sistema anti-colisão. A idéia de basear a arquitetura em três subsistemas (três torres): subsistema de Percepção, subsistema de Ação e subsistema Central está presente também em nossa arquitetura, embora de forma não tão explícita como em Sloman.

Também muito influente é o trabalho de Cañamero (1997, 1998, 2000, 2001, 2005) que desenvolve mecanismos emocionais em seu projeto *Abbott/Gridland* onde os agentes utilizam três tipos de sensores: sensor somático (provê informação sobre sua fisiologia interna), sensor tátil (propriedades dos objetos ao redor) e sensor visual (informação sobre brilho e distância de objetos vizinhos). No Agente Emocional Hedonista estão presentes sensores de contato e sensor visual. Além das propriedades imediatas do objeto (dureza, gosto e energia), o sensor de contato fornece um grau de “desejabilidade” como atrativo ou não atrativo para a criatura. Já o sensor visual, fornece dados sobre

a cor e a localização do objeto no ambiente.

Em Cañamero, os fatores Fome e Curiosidade, entre outros, são elementos do tipo Motivação enquanto Medo é um dos elementos do tipo Emoção. No Agente Emocional Hedonista, investimos nos Fatores Emocionais Fome, Medo e Curiosidade, embora não sejam e não devam ser exclusivos pois a arquitetura não só prevê, mas procura favorecer que outras emoções possam ser agregadas em futuras implementações. É possível questionar a adequação de Fome e Curiosidade como Fatores Emocionais. Esta porém foi uma opção deliberada que fizemos e por outro lado, tentamos satisfazer os comportamentos propostos por Balkenius (1995) conforme será apresentado mais adiante.

Para que as arquiteturas sejam mais eficientes, sempre haverá a necessidade de especificações detalhadas dos sistemas emocionais e de suas relações com estímulos ambientais. Concordamos com Damásio, Sloman, Sarmiento e Picard que as metáforas emocionais embora úteis porque permitem vislumbrar melhor os modelos, não alteram o fato de que ainda estamos longe de compreender totalmente a complexidade das emoções humanas.

O trabalho de Velásquez (1998), sugere um modelo modular e incremental no sentido de que à medida que a compreensão sobre emoções se aprofunda, novas possibilidades e propriedades podem ser facilmente incorporadas à arquitetura de base neural. Esse sentido escalável de comportamentos para a criatura é altamente inspirador para nosso trabalho, sobretudo na implementação por meio de Algoritmo Genético. Ao codificarmos cada possível propósito do agente em uma emoção distinta, geramos também funções de utilidade que medem o quanto as ações (ou planos) da criatura contemplam os propósitos atuais. Dessa forma, comportamentos de complexidade escalável podem ser gerados, na medida em que novas emoções podem ser agregadas ao comportamento da criatura.

O trabalho de Sarmiento, (2004) e Sarmiento et al., (2004) visto com mais detalhes no Capítulo 2, exerce particular interesse dada a engenhosidade com que trata os mecanismos emocionais. A idéia desta arquitetura é de que todos os Módulos Funcionais sejam paralelos e que compartilhem recursos computacionais de forma escalonada. Tal escalabilidade depende de Mecanismos Emocionais cuja atividade está diretamente relacionada à performance do agente em cumprir metas, respondendo adequadamente às condições do ambiente. Os mecanismos emocionais constituintes desta arquitetura são: Medo; Ansiedade e Auto-Confiança.

Mecanismo Emocional Medo

Ocorre quando a meta sobrevivência do agente está em risco. Medo reflete a inabilidade do agente em lidar com a situação face aos recursos disponíveis.

Quando o Medo é alto, o nível de atividade computacional relativo ao planejamento é reduzido para obter um plano mais rapidamente. Agir de forma pessimista (com Medo) fará com que o agente corra menos riscos.

Mecanismo Emocional Ansiedade

Relacionado a antecipações de reações, Ansiedade resulta da avaliação do ambiente e das próprias capacidades do agente em prever situações futuras. De certa forma, Ansiedade trabalha como um mecanismo de prevenção. Alta Ansiedade produz um estado de processamento intensivo de forma que o agente detectará melhor as oportunidades. O agente investirá uma grande parte de seus recursos de processamento em analisar a percepção dos dados e produzir Crenças de alto nível para usar na decisão de novas metas. Isso é obtido aumentando a frequência de execução do BDA (Camada Deliberativa) onde os Raciocinadores estão escalados para executar. O agente também melhorará a qualidade do plano final. Ao mesmo tempo, planos são revisados frequentemente para integrar o conhecimento das Crenças atualizadas produzidas pelos Raciocinadores. Obviamente o aumento do processamento precisa ser compensado pela redução do tempo de processamento alocado a outros módulos, por exemplo, dos Perceptores e Controladores e por isso leva a um relaxamento do BCA (Camada Reativa).

Mecanismo Emocional Auto-Confiança

Este mecanismo sinaliza que o ambiente não oferece dificuldades para o agente. Possibilita relaxar as estratégias de processamento de informação e aproveitar melhor os recursos em outras tarefas ou metas que podem ser vantajosas mais tarde.

Em síntese, os Mecanismos Emocionais proposto por Sarmiento, (2004) e Sarmiento et al., (2004) (Medo, Ansiedade e Auto-Confiança) são úteis para: adaptar o esforço computacional a cada situação; alterar prioridades e atribuições anteriores dos objetivos dos agentes; adaptar parâmetros internos de acordo com sucesso próprio do agente; e melhorar a coordenação do trabalho da equipe influenciando o líder.

Consideramos que na implementação do Agente Emocional Hedonista, o papel do Fator Emocional Medo tem paralelo com o Mecanismo Emocional Medo descrito em Sarmiento. No Método Heurístico, o Medo é utilizado para determinar uma trajetória que contorne obstáculos, isto é, posições onde a criatura tenha Medo de estar. Desse modo, Medo favorece um comportamento capaz de evitar colisões, o Agente Emocional Hedonista trabalha com um “fator de cautela” presente no algoritmo. Já para a implementação por meio de Algoritmo Genético, a estratégia é a seguinte: dada a posição da meta, calcula-se a proximidade aos obstáculos conhecidos e emite-se uma desejabilidade para aquela

posição.

Consideramos que o papel do Mecanismo Emocional Ansiedade de Sarmiento tem algum paralelo com os nossos Fatores Emocionais Medo e Curiosidade combinados (Figura 4.18). No Método Heurístico, a Curiosidade está relacionada a determinação de prioridades na escolha de metas em lugares desconhecidos do ambiente. Já no Algoritmo Genético, temos a Função de Curiosidade: dado o mapa de posições conhecidas, a função calcula uma maior desejabilidade para posições desconhecidas que estejam mais próximas da posição atual.

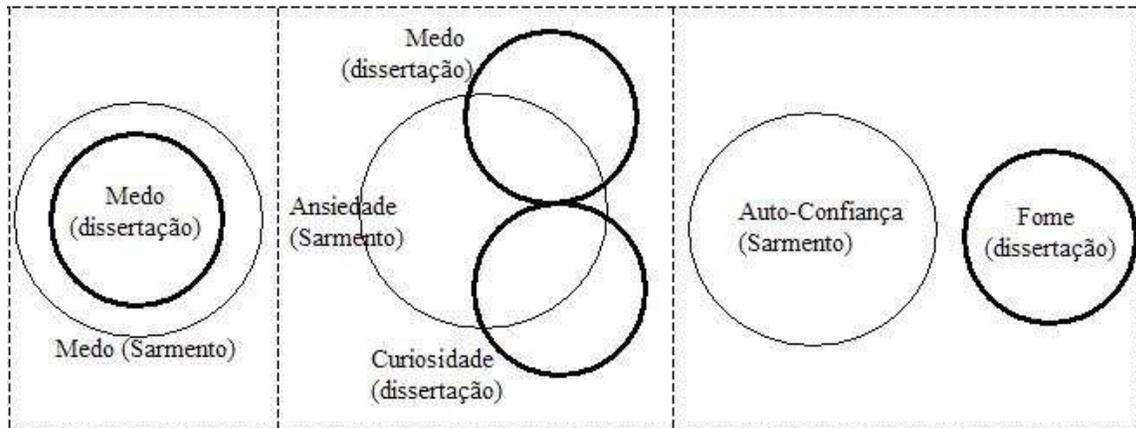


Figura 4.18: Comparação de emoções: Sarmiento e a dissertação aqui apresentada

O trabalho de Balkenius (1995) descreve diversas formas de comportamento de uma criatura em relação a um objeto considerado. De forma simplificada, tais comportamentos incluem-se em quatro categorias principais:

1. Apetitivo;
2. Aversivo;
3. Neutro;
4. Exploratório.

Comportamento Apetitivo

Possui forte analogia com a necessidade de uma criatura precisar se alimentar. Assim, movida por esse comportamento, a criatura precisa mover-se da posição atual até a posição do alimento. Uma vez alcançado o alimento, o comportamento buscar-alimento muda para comer. Este é um exemplo de dois componentes principais de comportamento em relação ao objeto Apetitivo.

Existem quatro tipos de abordagens de comportamentos dirigido a meta (Figura 4.19):

1. Acelerando (Atratividade): a velocidade da criatura torna-se mais intensa à medida que se aproxima da meta (gradiente de atratividade);
2. Desacelerando (Repulsividade): oposto ao Acelerando. A criatura desacelera ao se aproximar da meta. Por exemplo: comportamentos de muitos animais reais combinam esses dois tipos, havendo gradiente de atratividade e gradiente de repulsividade em relação ao objeto;
3. Combinado: velocidade aumenta até uma certa distância de separação da meta, a partir desse ponto, a velocidade tende a diminuir até encontrar um equilíbrio;
4. Orientado: move a criatura para a meta em trajetórias variáveis.

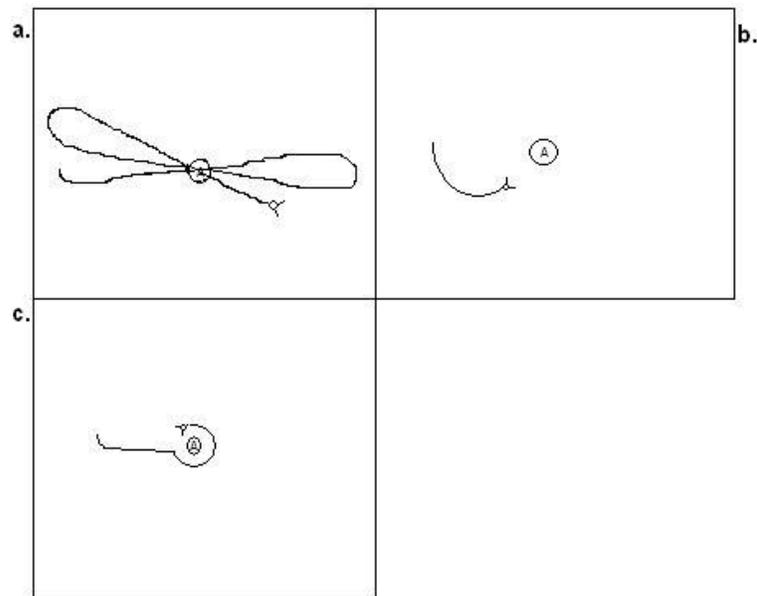


Figura 4.19: Adaptado de Balkenius (1995) - abordagens de comportamento dirigido a meta

Na Figura 4.19 (a): a criatura acelera na direção da meta, não pára ao atingi-la e quando a ultrapassa, à medida que se afasta, desacelera. O movimento é, portanto, pendular; (b): a criatura desacelera em direção à meta, mas pára antes de atingi-la e, por fim, (c): a criatura tem atração e repulsa pela meta. Acelera e depois desacelera em direção a ela dependendo dos gradientes de aproximação e repulsa. A certa altura a criatura inicia um movimento circular de equilíbrio instável em torno da meta.

Comportamento Aversivo

Este comportamento é aquele onde uma situação deve ser evitada em nome da sobrevivência da criatura. Desdobra-se em Repulsa Passiva e Repulsa Ativa. A Repulsa Passiva ocorre em uma situação onde a criatura não executa uma ação mas sim evita esta ação. Por exemplo: não pular de uma árvore muito alta, não ingerir alimento envenenado. Por outro lado, a Repulsa Ativa exige uma ação. Por exemplo: correr de um predador. Embutido neste comportamento pode estar o comportamento atrativo onde a criatura se afasta lentamente no início e depois acelera, talvez para não ser notada pelo predador.

Tanto os comportamentos Apetitivo e Aversivo demandam alguma habilidade de aprendizagem. Sem aprendizado, não há acúmulo de experiências e o comportamento orientado sempre será o mesmo (maiores detalhes em Balkenius (1995)).

Comportamento Neutro

Está relacionado aos elementos do ambiente que não são úteis e nem tampouco perigosos para a criatura (mas podem vir a se tornar mais tarde). Uma parede em um ambiente é um objeto Neutro mas a relação entre a criatura e o objeto pode ser do tipo Repulsa Passiva no sentido de que a criatura deve evitar a colisão e contornar o objeto para alcançar a meta.

Comportamento em relação a um Objeto Desconhecido Exploratório

Este comportamento é conflituoso. Desde que possa ser perigosa uma aproximação, o comportamento não deve ser exclusivamente do tipo atrativo ou repulsivo *a priori*. Uma possível solução é de que a criatura desempenhe um comportamento atrativo porém com precaução. Esse comportamento varia conforme se conheça o objeto. Por exemplo: vamos imaginar uma criatura que tem comportamento de atratividade somente até uma certa distância de um objeto mas tende a evitar esse comportamento à medida que chega mais perto. Isso permite aprender sobre o objeto enquanto ainda for possível evitá-lo, caso seja necessário.

O Agente Emocional Hedonista desempenha uma série de comportamentos que foram descritos no Capítulo 3 e início do presente Capítulo desta dissertação. Tais comportamentos possuem forte analogia com comportamentos descritos por Balkenius. Procuramos sumarizar essa associação na Tabela 4.3.

| Classificação Criatura-Objeto segundo Balkenius | Agente Emocional Hedonista - Criatura Artificial |
|--|---|
| Comportamento Apetitivo | Relacionado ao Fator Emocional Fome |
| Comportamento Aversivo | Relacionado ao Fator Emocional Medo |
| Comportamento Neutro | Este comportamento não é previsto |
| Comportamento Exploratório | Relacionado ao Fator Emocional Curiosidade |

Tabela 4.3: Comparação de emoções: Balkenius e a dissertação aqui apresentada

4.8 Resumo do Capítulo

Neste capítulo, apresentamos uma descrição mais detalhada do nível superior de controle - Decisão de Metas, principal contribuição deste trabalho, na qual implementamos duas estratégias distintas, uma baseada em um método heurístico e a outra em algoritmo genético. Também mostramos os resultados obtidos destes experimentos e, por fim, elaboramos uma discussão conectando o trabalho desenvolvido com outros trabalhos na literatura. No próximo Capítulo, apresentamos a conclusão da dissertação e os trabalhos futuros.

Capítulo 5

Conclusão e Trabalhos Futuros

Nosso intuito com este trabalho foi estudar os diferentes modelos de emoções utilizados em agentes na literatura, e desenvolver um estudo de caso utilizando um modelo de emoções aplicado no controle de uma criatura artificial.

Apesar do farto material disponível na literatura a esse respeito, pudemos constatar que a comunidade, como um todo, está ainda longe de ter uma compreensão consensual do que seja emoção e, desta forma, avaliar de maneira generalizada o impacto de seu uso em agentes.

Mostramos também que a inspiração emocional pode servir como *design patterns* para o desenvolvimento de algoritmos de controle para criaturas artificiais, capazes de gerar comportamentos razoavelmente sofisticados. Mais do que isso, essa proposta sugere um novo paradigma de programação para agentes, orientada a propósitos. A idéia é codificar cada possível propósito do agente em uma emoção distinta, gerando funções de utilidade que medem o quanto as ações (ou planos) do agente contemplam os propósitos em questão.

Dessa forma, comportamentos de complexidade escalável poderiam ser gerados, simplesmente agregando novas emoções ao comportamento do agente. É certo que os atuais experimentos apenas sugerem esse *insight*, mas o presente estudo pelo menos mostra a viabilidade de tal assertiva, e nos motiva para continuar essa linha de pesquisa.

Como trabalhos futuros, dando prosseguimento aos resultados aqui alcançados, vislumbramos a necessidade de um estudo comparativo detalhado entre as diversas abordagens envolvendo emoções, e eventualmente a ampliação da proposta do agente hedonista incorporando outras propriedades emocionais, ou mesmo o desenvolvimento de um sistema emocional unificado, que busque avaliar as vantagens e características das diferentes propostas, de forma a tentar conciliar as diferenças e apontar as semelhanças que possam existir entre as propostas, na tentativa de promover um maior consenso sobre essa questão.

Outro ponto que nos parece demandar um maior esforço de pesquisa é a busca por métodos e

procedimentos mais apropriados para avaliar os progressos das pesquisas na área. Infelizmente, hoje não temos parâmetros normatizados ou consensuais em como avaliar de maneira adequada os resultados de pesquisa em criaturas artificiais. Como dizer que uma pesquisa foi bem ou mal sucedida? Normalmente, diferentes autores apresentam suas arquiteturas, mostram o comportamento, ou particularidades do comportamento obtido em seus experimentos, mas, devido à heterogeneidade de experimentos e de plataformas utilizadas, fica muito difícil uma comparação justa entre resultados de pesquisa. Além de uma avaliação meramente qualitativa, em termos de comportamentos observados que parecem apontar ou sugerir um progresso, é bastante difícil criar-se métricas quantitativas que possam ser utilizadas para promover a comparação entre propostas. A criação de tais métricas nos parece ser uma demanda importante para a continuidade de pesquisas na área, para que a mesma possa atingir maiores níveis de maturidade.

Já do ponto de vista de aplicações tecnológicas de criaturas artificiais (como por exemplo jogos de computadores), nos parece ser uma extensão natural dos experimentos desenvolvidos aqui, a utilização dos conceitos explorados em ambientes tri-dimensionais, com criaturas com maior número de ações possíveis, corpos com maior número de movimentos independentes e um maior número de emoções instanciadas. Tais experimentos poderiam corroborar (ou não) a proposta que apresentamos aqui de uma metodologia de programação orientada a propósitos e sua conveniência em casos mais complexos.

Na verdade, a pesquisa aqui desenvolvida corresponde somente aos primeiros passos de uma jornada ainda longa, em uma maior compreensão da dimensão cognitiva da emoção, e como essa pode afetar nosso comportamento inteligente.

Referências Bibliográficas

- Adam, C., Gaudou, B., Herzig, A. and Longin, D. (2006). Occ's emotions: A formalization in a bdi logic, *AIMSA*, pp. 24–32.
- Almeida, L. B., Silva, B. C. and Bazzan, A. L. C. (2004). Towards a physiological model of emotions: first steps, *Workshop on Architectures for Modeling Emotion: Cross-Disciplinary Foundations*, Vol. 1 of *Proc. of the 2004 AAAI Spring Symp. Series*, pp. 1–4.
- Bacardit, J. (2005). Analysis of the initialization stage of a pittsburgh approach learning classifier system, *Genetic and Evolutionary Computation Conference 2005 (GECCO'05)*, Washington, USA, pp. 1843–1850.
- Balkenius, C. (1995). *Natural Intelligence in Artificial Creatures*, Lund Univ. Cognitive Studies 37.
- Blumberg, B. M. (1996). *Old Tricks, New Dogs: Ethology and Interactive Creatures*, PhD thesis, MIT Media Lab, Cambridge, MA.
- Bradshaw, J. M. (1997). *An Introduction to Software Agents*. In *Software Agents*, AAAI Press, Menlo Park, CA.
- BrasilEscola (2007). Emoção, <http://www.brasilecola.com/psicologia/emocao.htm>. <acesso em 29/10/2007>.
- Budakova, D. and Dakovski, L. (2006). *Computer Model of Emotional Agents*, Vol. 4133, Springer Berlin / Heidelberg.
- Cañamero, L. (1997). A hormonal model of emotions for behavior control, *Proc. of the 4th ECAL (ECAL'97)*.
- Cañamero, L. (1998). Issues in the design of emotional agents, in A. Press (ed.), *In: Proc. of Emotional and Intelligent: The Tangled Knot of Cognition*. AAAI Fall Symposium, Menlo Park, CA, pp. 49–54.

- Cañamero, L. (2000). Designing emotions for activity selection, *Technical Report DAIMI PB 546*, Univ. of Aarhus, Denmark.
- Cañamero, L. (2001). Emotions and adaptation in autonomous agents: a design perspective, *Cybernetics and Systems: An International Journal* **32**: 507–529.
- Cañamero, L. (2003). *Designing Emotions for Activity Aelection in Autonomous Agents*, MIT Press.
- Cañamero, L. (2005). Emotion understanding from the perspective of autonomous robots research, *Neural Networks* **18**(4): 445–455.
- Clark, A. (1999). An embodied cognitive science?, *Trends in Cognitive Sciences* **3**(9).
- Damásio, A. R. (1994). *O Erro de Descartes: Emoção, Razão e Cérebro Humano*, Europa-América.
- Damásio, A. R. (1999). *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*, New York: Harcourt.
- Damásio, A. R. (2003). *Looking for Spinoza: Joy, Sorrow and the Feeling Brain*, London: William Heinemann.
- Deb, K. (1999). An introduction to genetic algorithms, *Sadhana* **24**: 293–315.
- El-Nasr, M. S., Yen, J. and Ioerger, T. R. (2000). Flame - fuzzy logic adaptive model of emotions, *Autonomous Agents and Multi-agent Systems* **3**: 219–257.
- Franklin, S. and Graesser, A. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents, in Springer-Verlag (ed.), *Proc. of the Third ATAL*, London, UK.
- Gardner, H. (1994). *Estrutura da Mente: A Teoria das Inteligências Múltiplas*, Porto Alegre: Artes Médicas Sul. Tradução por Sandra Costa.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.
- Goleman, D. (1995). *Inteligência Emocional*, Rio de Janeiro: Objetiva.
- Gomes, A. R. (2007). Vocabulário de Filosofia, <http://ocanto.esensiveu.net/lexh.htm>. <acesso em 29/10/2007>.
- Gray, J. A. (1982). *The neuropsychology of anxiety*, Oxford University Press, New York.

- Gudwin, R. R. (1996). *Contribuições ao Estudo Matemático de Sistemas Inteligentes*, PhD thesis, DCA-FEEC-UNICAMP.
- Jaques, P. A. and Vicari, R. M. (2005). Estado da arte em ambientes inteligentes de aprendizagem que consideram a afetividade do aluno, *Informática na educação - UFRGS: Porto Alegre* **8**(1): 15–38.
- Junior, O. S. and Costa, A. L. (2007). Meca team 2006: Um sistema multiagente reativo para o futebol de robôs simulado, *Universidade Federal da Bahia - SBC*.
- Kim, K. H. and Bae, J. W. (2004). A look-ahead dispatching method for automated guided vehicles in automated port container terminals., *Transportation Science* **38**(2): 224–234.
- Koch, F., Meyer, J. J. C. and Dignum, F. Rahwan, I. (2006). *Programming Deliberative Agents for Mobile Services: The 3APL-M Platform*, Vol. 3862, Springer Berlin / Heidelberg. Lecture Notes in Computer Science.
- Levine, D. S. (2007). Neural network modeling of emotion, *Physics of Life Reviews* **4**: 37–63.
- Meyer, J. J. C. (2006). Reasoning about emotional agents, *Int. J. Intell. Syst* **21**(6): 601–619.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*, MIT Press.
- Murch, R. and Johnson, T. (1999). *Intelligent Software Agents*, Prentice Hall, NJ.
- Nussbaum, M. (2001). *Upheavals of Thought- The Intelligence of Emotions*, Cambridge Univ. Press.
- Ortony, A. (2003). *On making believable emotional agents believable*, Cambridge, Massachusetts London, England: MIT Press.
- Ortony, A., Clore, G. and Collins, A. (1998). *The Cognitive Structure of Emotions*, Cambridge Univ. Press.
- Picard, R. W. (1997). *Affective Computing*, MIT Press, Cambridge.
- Picard, R. W. (2006). Building an affective learning companion, *Intelligent Tutoring Systems*, pp. 811–811.
- Reilly, W. S. N. (1996). *Believable Social and Emotional Agents*, PhD thesis, Carnegie Mellon University, Pittsburgh, PA.
- Rolls, E. T. (1986). *Neural systems involved in emotion in Primates. in Emotion: Theory, Research, and Experience*, Vol. 3.

- Russell, S. J. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs, NJ.
- Sarmiento, L. M. (2004). *An emotion-based agent architecture*, Master's thesis, FC University of Porto.
- Sarmiento, L., Moura, D. and Oliveira, E. (2004). Fighting fire with fear, *2nd European Workshop on Multi-Agent Systems*, Barcelona, pp. 627–633.
- Sensagent (2003). Dictionary on your website - emotion, <http://dictionary.sensagent.com/EMOTION/fr-fr/>. <acesso em 29/10/2007>.
- Septseault, C. and Nédélec, A. (2005). A model of an embodied emotional agent, *IVA*, p. 498.
- Slovan, A. (1998). Damasio, descartes, alarms, and meta-management, *In Proc. of the IEEE International Conf. on Systems, Man, and Cybernetics*, pp. 2652–2657.
- Slovan, A. (2001). Beyond shallow models of emotion, *Cognitive Processing*, Vol. 2, pp. 177–198.
- Stocker, M. and Hegeman, E. (2002). *O Valor das Emoções*, Palas Athena - São Paulo.
- Varela, F. J., Thompson, E. and Rosch, E. (2003). *A mente incorporada: ciencias cognitivas e experiencia humana*.
- Velásquez, J. (1998). A computational framework for emotion-based control, *In Proc. of the Grounding Emotions in Adaptive Systems - workshop SAB'98*, Zurich, Switzerland.
- Ventura, R. (2000). *Emotion-based agents*, Master's thesis, Inst. Sup. Tec. - Lisboa, Portugal.
- Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents theories, architectures, and languages: a survey, *Lecture notes in AI* **890**: 1–32. Springer-Verlag.