

# AGENTES EMOCIONAIS HEDONISTAS PARA COMPORTAMENTO AUTÔNOMO

PATRICIA ROCHA DE TORO\*, RICARDO RIBEIRO GUDWIN\*, MAURO SÉRGIO MISKULIN†

\*DCA/FEEC/UNICAMP, c.p.6101, CEP: 13083-970, Campinas, SP, Brasil

†DSIF/FEEC/UNICAMP, c.p.6101, CEP: 13083-970, Campinas, SP, Brasil

Emails: {patbgi, gudwin}@dca.fee.unicamp.br, mauromiskulin@terra.com.br

**Abstract**— In this article, we present an architecture for an Hedonistic Emotional Agent and a controller that confers autonomous behavior to an artificial creature. Two implementations were performed: the first one is based on an heuristic algorithm, and the second one uses a genetic algorithm. Both implementations consider two layers of control: one level of direct control and one 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. Among the presented results, there is 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 new purpose-oriented way of programming.

**Keywords**— Hedonistic Emotional Agent, Autonomous Behavior, Emotions, Agents.

**Resumo**— Neste artigo, apresentamos uma arquitetura para um Agente Emocional Hedonista e desenvolvemos um controlador que confere comportamento autônomo a uma criatura artificial. Duas implementações foram realizadas: uma baseada em um algoritmo heurístico e outra, em um algoritmo genético. Ambas 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, Agentes.

## 1 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. Esse 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, onde 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 esse problema (Bates, 1994; Blumberg, 1996; Reilly, 1996; Picard, 1997; Velásquez, 1998; Cañamero 1997, 1998, 2000; Ventura, 2000; Almeida et al., 2004; Sarmento et al., 2004; Septseault and Nédélec, 2005; Budakova and Dakovski, 2006; Meyer, 2006).

Uma abordagem sofisticada é a de Ortony et al. (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. Picard (1997) desenvolve toda uma área de estudos, chamada de “computação afetiva” envolvendo o relacionamento entre emoções e computadores.

Neste artigo, 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.

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 proposada e adaptabilidade. O elo que une o uso de emoções para o desenvolvimento de agentes está em se dizer “de acordo com seus propósitos”. Segundo (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 avalia se esses propósitos estão sendo cumpridos. As emoções do agente correspondem portanto, a uma codificação implícita dos propósitos inatos que orientam seu comportamento. Em inteligência artificial, temos algo semelhante nas funções de utilidade (Russell and Norvig, 1995), que podem ser utilizadas para orientar 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 poderia simplificar o desenvolvimento de funções de utilidade, que uma vez implementadas, gerariam o comportamento autônomo em agentes. Observando, estudando as emoções humanas, e analisando seu propósito implícito, poderíamos utilizar as emoções como *design patterns* para o desenvolvimento de funções de utilidade. Partindo desse pressuposto, construímos uma arquitetura para testar a viabilidade da idéia.

A arquitetura na qual este trabalho se baseia, foi proposta originalmente por Gudwin (1996), onde se define uma criatura autônoma dotada de um aparato sensorial e motor capaz de trafegar por um ambiente repleto de objetos com características diferenciadas. Os objetos podem ser de diferentes “cores”, sendo que a cada “cor” se associa uma “dureza”, que restringe a movimentação da criatura por lugares onde os mesmos se encontram (podendo variar de 0 a 1), um “gosto”, que pode ser bom ou ruim (variando de -1 a 1) e uma habilidade de fornecer ou drenar “energia” (que também pode variar entre -1 a 1). A criatura possui uma fonte interna de energia recarregável, e objetos do ambiente nos quais a criatura pode tocar são capazes de carregar ou descarregar esta fonte. A criatura pode navegar neste ambiente, e diversos objetivos podem ser determinados. Um primeiro objetivo poderia ser o deslocamento de pontos iniciais até pontos finais, evitando uma colisão com diferentes objetos do ambiente. Entretanto, em alguns casos (como no caso de objetos que são fontes de energia), essa colisão é desejável. O objetivo é gerar o comportamento do agente se-

gundo diversos propósitos atribuídos a ele.

Foram propostos dois níveis de controle: um nível mais baixo, ou de controle direto e um segundo nível hierarquicamente superior, de decisão de metas (ver Figura 1). No nível de controle direto, assume-se uma meta que a criatura deve atingir, definida na forma de um ponto do ambiente ao qual a criatura deve chegar. A criatura deve deslocar-se do ponto em que se encontra até a meta, sem colidir com obstáculos. No nível de controle superior, o sistema deve decidir as metas, ou seja, os pontos para o qual a criatura deve se deslocar. Juntos, os níveis de controle direto e superior geram um comportamento complexo para a criatura. Em (Gudwin, 1996) estudou-se extensivamente o nível de controle direto, por meio de uma estratégia deliberativa que gera planos que são executados neste nível. Neste trabalho, estendemos o trabalho de (Gudwin, 1996) incorporando o nível superior de controle.

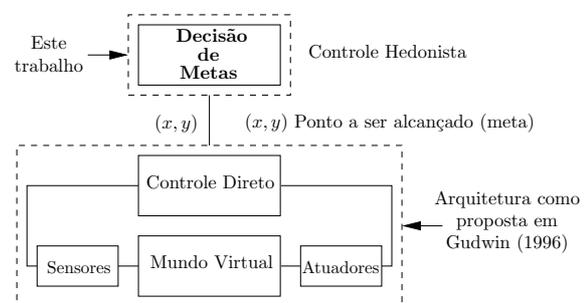


Figura 1: Arquitetura da Criatura Artificial

O nível de controle direto trata somente do deslocamento da criatura. Já o nível de controle de decisão, permite que a criatura considere múltiplos objetivos, dentre explorar o ambiente, cuidar da manutenção de seu nível de energia e regular a sensibilidade do algoritmo anti-colisão. Para gerar esse comportamento com múltiplos objetivos, nos servimos de três metáforas emocionais: *medo*, *fome* e *curiosidade*. O *medo* deve influenciar, de alguma forma, o comportamento anti-colisão. A *fome*, o controle do balanceamento energético da criatura, e a *curiosidade*, o comportamento exploratório da criatura, fazendo-a escolher metas em pontos do ambiente que sejam desconhecidos. O agente aqui desenvolvido, com estas características, foi denominado *agente emocional hedonista*.

## 2 Agentes Emocionais Hedonistas

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 de uma expectativa futura. Todo o comportamento, sob

uma 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. Programar um comportamento autônomo por meio de um *agente emocional hedonista* 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 em 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. 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 o comportamento em grupo, e as emoções de outros agentes também são levados em consideração nas suas decisões comportamentais.

Para o presente experimento, utilizamos a seguinte codificação das emoções:

- *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.

### 3 Experimentos e procedimentos

Um esquema da estrutura sensório-motora da criatura pode ser visto na Figura 2. À esquerda vemos a criatura, localizada em um ponto de seu ambiente, colidindo em um objeto de cor amarela. À direita da criatura, um *grid* de sensores na forma de um quadrado cheio de pontos demonstra o posicionamento do sensor visual da criatura. 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 atual, o estado do sensor visual, sua posição em termos de ângulo e distância e 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. 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 no mesmo. Baseada nas informações sensoriais colhidas, a criatura constrói um mapa do ambiente, de maneira incremental. Esse mapa é utilizado para gerar planos de movimentação sobre o ambiente. Maiores detalhes sobre a estrutura da criatura, bem como o modelo dinâmico utilizado no simulador e a estrutura do controle direto (nível baixo) podem ser encontrados em Gudwin (1996).

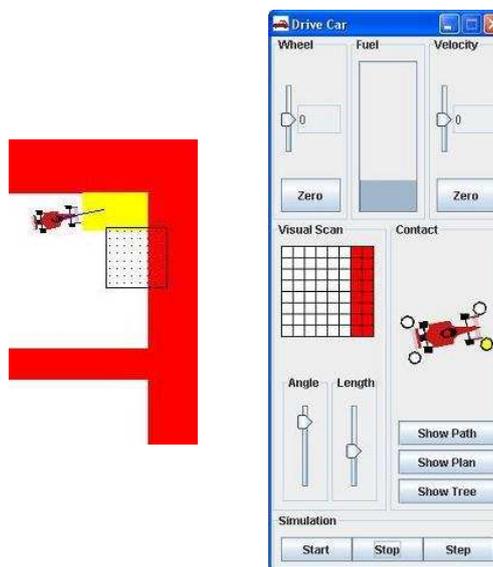


Figura 2: Estrutura Sensório-Motora da Criatura

Neste trabalho, conforme ressaltado anteriormente, visou-se somente a determinação do controle de nível hierárquico superior, ou seja, a decisão sobre os pontos que a criatura deveria atingir a cada instante. Segundo (Balkenius, 1995), uma criatura artificial deve, em princípio seguir quatro tipos de comportamento padrão: (i) apetitivo: foco para um objeto ou uma situação atrativa; (ii) aversivo: afastamento de situações negativas; (iii) exploratório, direcionado a um aumento no conhecimento sobre o ambiente e, por fim, (iv) neutro onde descreve uma classe de comportamentos que se relacionam aos objetos que não são nem apetitivos e nem aversivos. A escolha das emoções “medo”, “fome” e “curiosidade” visou contemplar os comportamentos apetitivos, aversivos e exploratórios. Neste experimento, não contemplamos o comportamento “neutro”.

Algumas questões são importantes para a integração entre estes comportamentos:

- o sensor visual fornece a localização de objetos no ambiente;
- o sensor de contato informa as propriedades do objeto, quando a criatura entra em contato com os objetos;
- à medida que a criatura se desloca pelo ambiente, cai seu nível de energia;
- o contato com objetos pode drenar energia da criatura ou pode re-energizá-la, dependendo das propriedades do objeto em questão.

O nível de controle direto (conforme Gudwin, 1996), naturalmente incorpora o *medo* na geração de um plano para, a partir da posição atual da criatura, traçar uma rota até a meta, considerando-se os objetos que a criatura conhece do ambiente. Entretanto, em Gudwin (1996) esse *medo* era determinado heurísticamente, por meio de uma função de utilidade definida a priori. Aqui, desenvolvemos um modelo incremental, onde esse *medo* pode aumentar ou diminuir, à medida que a criatura entra em contato com objetos que representam obstáculos. Além disso, esse *medo* também é utilizado na determinação da próxima meta pela criatura. A determinação da meta não é uma tarefa simples. Dependendo da situação, diferentes comportamentos podem ser desejáveis. Caso o nível energético da criatura esteja alto, pretende-se que a criatura desenvolva um comportamento exploratório, procurando movimentar-se para áreas do ambiente que desconhece (ou seja, que não possui objetos conhecidos no mapa incremental que gera do ambiente). Entretanto, caso o nível de energia esteja baixo, deseja-se que a criatura busque em seu mapa a fonte de energia mais próxima e se dirija a ela, para reabastecimento. Após o abastecimento, a criatura pode retornar ao comportamento exploratório. Para implementar esse comportamento híbrido, desenvolveu-se duas estratégias distintas, uma baseada em um algoritmo heurístico e sistemático, que explicitamente impõe um conjunto de prioridades nas emoções, e outro que tenta fazer um “blend” automático destas emoções, utilizando um algoritmo genético.

### 3.1 Descrição do algoritmo heurístico

No algoritmo heurístico, as emoções são implementadas de maneira *hard-coded*, 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 se utiliza do controle direto determinado em Gudwin (1996) para desenvolver um plano que movimenta a criatura até a meta, e esse plano é colocado em execução. Esse passo implementa de maneira implícita os comportamentos *medo* e *curiosidade*. O *medo* é utilizado para determinar uma trajetória que contorne posições em que a criatura tenha *medo* de estar (de acordo com a função de *medo*). E a *curiosidade* é contemplada por meio de uma função de *curiosidade*, que avalia o conhecimento que a criatura tem sobre o ponto sorteado. Caso o ponto sorteado tenha um alto índice de *medo* ou um baixo índice de *curiosidade*, novo ponto é sorteado, até que níveis compatíveis de *medo* e *curiosidade* sejam atingidos. Esse é o comportamento padrão da criatura.

Entretanto, caso o nível de energia caia abaixo de 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. Tenta traçar trajetórias para esses objetos, e escolhe o plano que leva ao objeto mais próximo de ser atingido (que nem sempre é aquele que está mais próximo segundo a distância euclidiana), pois obstáculos no ambiente podem torná-lo mais distante do ponto de vista de trajetórias. Após entrar em contato com esse objeto, a criatura aguarda seu nível de energia interna atingir ao máximo, e em seguida traça um novo plano para chegar à meta anterior, que havia abandonado.

Observe-se que este algoritmo heurístico impõe explicitamente uma seqüência de prioridades para as emoções: 1-*Medo*; 2-*Fome*; 3-*Curiosidade*. 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 eleva o “fator de cautela” do algoritmo.
2. *Fome* (manter balanço energético): *Fome* responde pelo nível energético da criatura. Quando está muito baixo, a criatura ignora a meta que deveria atingir e a substitui temporariamente pela busca do ponto fornecedor de energia mais próximo.
3. *Curiosidade* (busca de pontos desconhecidos do ambiente): *Curiosidade* responde pela escolha de metas em lugares do ambiente ainda desconhecidos do agente.

O algoritmo heurístico possui diversas deficiências. 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 quisesse sofisticar o comportamento do agente, o acréscimo de novas emoções demandaria mudanças profundas 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 re-analisado e re-organizado. 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, tentou-se conceber um algoritmo que pudesse ser generalizado de maneira uniforme, onde o acréscimo de novas emoções não demandasse toda uma re-estruturação do algoritmo. O resultado é o algoritmo a seguir.

### 3.2 Descrição do método por algoritmo genético

Para tentar resolver os problemas apontados anteriormente, concebeu-se um algoritmo genético, onde uma população de possíveis metas deve evoluir, utilizando como função de *fitness* uma combinação de todas as funções de utilidade, uma para

cada emoção que se deseje implementar. Essa combinação resulta em uma função de “desejabilidade”, que inclui o *medo*, a *fome* e a *curiosidade*.

Inicialmente essa população de “metas” é gerada aleatoriamente. A cada passo evolutivo, essas “metas” vão sendo confinadas a regiões do espaço onde a “desejabilidade” é maior. Para o algoritmo genético, cada meta é codificada na forma de uma posição  $(x, y)$  no espaço bi-dimensional. O *crossover* entre duas metas gera uma combinação linear entre estes dois pontos, fornecendo um ponto aleatório na reta que liga as duas possíveis metas (Figura 3). Além disso, uma mutação pode deslocar uma possível meta em torno de sua vizinhança. Todas as possíveis metas são avaliadas quanto à sua “desejabilidade”, e um processo de seleção elitista mantém na população somente as metas com maior “desejabilidade”.

O Tamanho da População= 1000 indivíduos; a Prob.*Crossover*= 0.5 e a Prob.Mutação= 0.1.

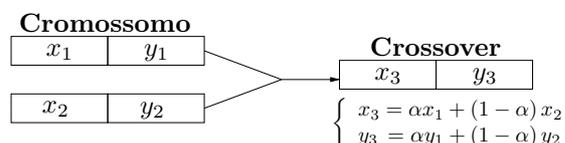


Figura 3: Dados do Algoritmo Genético

As Figuras 4, 5 e 6 demonstram uma simulação da criatura, utilizando o algoritmo genético. A Figura 4 mostra um determinado ambiente, onde a criatura deve navegar. As Figuras 5 e 6 representam o mapa interno construído pela criatura, em diferentes instantes na simulação. A Figura 5 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 6 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.

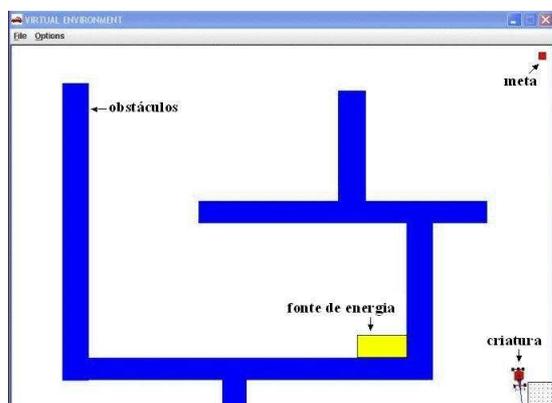


Figura 4: Visão do ambiente



Figura 5: Representação do fator emocional *Curiosidade* (em diferentes tons na cor cinza)

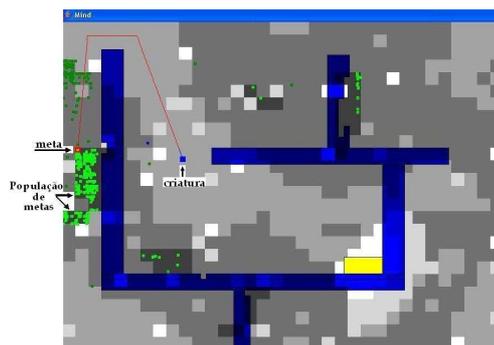


Figura 6: Situação otimizada: criatura com alto grau de conhecimento sobre o ambiente



Figura 7: Fator emocional *Fome* onde a prioridade é o abastecimento de energia

A Figura 7 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.

## 4 Conclusões

Apresentamos neste artigo uma arquitetura que denominamos de *agente emocional hedonista*. A idéia é utilizar uma metáfora computacional para

emoções de forma a gerar comportamentos sofisticados para criaturas artificiais (agentes autônomos inteligentes).

Mostramos que a motivação 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*, não sendo ainda conclusivos. Comparações com outros modelos e técnicas ainda precisam ser efetuados. Mas o presente estudo pelo menos mostra a viabilidade de tal assertiva, e nos motiva para continuar essa linha de pesquisa.

Como trabalho futuro, dando prosseguimento aos resultados aqui alcançados, vislumbramos novos caminhos para pesquisa em conceitos como alcançabilidade e controlabilidade.

### Agradecimentos

Patrícia R. de Toro agradece à CAPES pelo suporte financeiro na forma de bolsa de estudo.

### Referências

- 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.
- Balkenius, C. (1995). *Natural Intelligence in Artificial Creatures*, Lund Univ. Cognitive Studies 37.
- Bates, J. (1994). The role of emotion in believable agents, *Communications of the ACM* **37**(7): 122–125.
- 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.
- Budakova, D. and Dakovski, L. (2006). *Computer Model of Emotional Agents*, Vol. 4133 of *Lect. Notes in Computer Science*, 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.
- 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.
- Gomes, A. R. (2007). Vocabulário de Filosofia, <http://ocanto.esensiveu.net/lexh.htm>.
- Gudwin, R. R. (1996). *Contribuições ao Estudo Matemático de Sistemas Inteligentes*, PhD thesis, DCA-FEEC-UNICAMP.
- Meyer, J. J. C. (2006). Reasoning about emotional agents, *Internat. Journal of Intelligent Systems* **21**(6): 601–619.
- 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., Clore, G. and Collins, A. (1998). *The Cognitive Structure of Emotions*, Cambridge Univ. Press.
- Picard, R. W. (1997). *Affective Computing*, MIT Press, Cambridge.
- Reilly, W. S. N. (1996). *Believable Social and Emotional Agents*, PhD thesis, Carnegie Mellon Univ., Pittsburgh, PA.
- Russell, S. J. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs, NJ.
- Sarmento, L., Moura, D. and Oliveira, E. (2004). Fighting fire with fear, *The Second European Workshop on Multi-Agent Systems*, Barcelona, Spain.
- Septseault, C. and Nédélec, A. (2005). A model of an embodied emotional agent, *Intelligent Virtual Agents, 5th Internat. Working Conference*, Vol. 3661 of *Lect. Notes in Computer Science*, Springer, Kos, Greece, p. 498.
- Sloman, A. (1998). Damasio, descartes, alarms, and meta-management, *In Proc. of the IEEE International Conf. on Systems, Man, and Cybernetics*, pp. 2652–2657.
- Sloman, A. (2001). Beyond shallow models of emotion, *Cognitive Processing*, Vol. 2, pp. 177–198.
- 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.