

# Composição de Agentes EBDI: Integração WASABI-Jason

Diego Couto, Jerusa Marchi, Thiago Â. Gelaim

<sup>1</sup>Departamento de Informática e Estatística - INE  
Programa de Pós-Graduação em Ciência da Computação  
Universidade Federal de Santa Catarina - UFSC  
Florianópolis – SC – Brasil

diego.gcouto@gmail.com, jerusa.marchi@ufsc.br, t.gelaim@gmail.com

**Abstract.** *This work presents the integration of WASABI emotions simulator and JASON BDI framework in order to provide emotional agents. The results suggest that the performance of the agents in the environment is susceptible not only to their behavior, but to their emotional character and environmental conditions.*

**Resumo.** *Este trabalho apresenta a integração do simulador de emoções WASABI com o arcabouço para agentes BDI JASON, para prover agentes emocionais. Os resultados obtidos sugerem que o desempenho dos agentes no ambiente, assim como em sociedades humanas, é suscetível não só ao seu comportamento programado, mas à sua personalidade emocional e às condições do ambiente.*

## 1. Introdução

Decisões tomadas por seres humanos estão intimamente relacionadas a um estado emocional [Lerner et al. 2015]. Contudo, o impacto efetivo da tomada de decisão baseada em razão permeada por emoções não é identificável uma vez que não se pode estabelecer um comparativo com um mesmo cenário em que não exista a influência de emoções.

Em um ambiente computacional, no entanto, a simulação de sociedades de agentes permite que tal impacto seja mensurado, pois é possível estabelecer cenários, verificar e diferenciar o comportamento de agentes que deliberam somente de forma racional daqueles que fazem uso de emoções simuladas. Conhecer tal comportamento é útil não só para modelar comportamentos mais refinados mas também para determinar se o uso de emoções produzirá os resultados esperados.

Este trabalho apresenta a integração do simulador de emoções WASABI [Becker-Asano 2008] com o framework Jason [Bordini et al. 2007], onde estados emocionais passam a compor crenças do agente e são utilizados para a tomada de decisão. Um cenário simples é utilizado para verificação da proposta. O artigo está estruturado da seguinte forma: a seção 2 apresenta alguns conceitos básicos sobre emoções em agentes, abordando em mais detalhes o modelo WASABI. A seção 3 apresenta detalhes do desenvolvimento do *middleware* que operacionaliza a inserção e manutenção dos estados emocionais de agentes concebidos em Jason, bem como a forma encontrada para modelar a "personalidade" dos agentes. Na seção 4 é descrito o cenário utilizado na verificação e as simulações realizadas, visando verificar a diferença de comportamento entre agentes que fazem uso de emoções em seus processos de raciocínio dos demais. Por fim, a seção 5 apresenta algumas considerações e apontamentos para trabalhos futuros.

## 2. Emoções em Agentes Computacionais

A introdução de emoções em sistemas computacionais objetiva melhorar a interação humano-máquina; testar, refinar e desenvolver hipóteses emocionais; e colaborar no refinamento de técnicas de Inteligência Artificial (IA) [Marsella et al. 2010]. As duas principais vertentes de modelos computacionais de emoções são: teorias apreciativas (do inglês *appraisal theories*) [Scherer 1999] e teorias dimensionais. Nas teorias apreciativas, elicitação e diferenciação de emoções são feitas por meio da avaliação, ou apreciação subjetiva do indivíduo considerando uma situação, objeto ou evento [Scherer 1999]. O modelo OCC, proposto por Ortony, Clore e Collins [Ortony et al. 1990], é um modelo de teorias apreciativas frequentemente utilizado. Em teorias dimensionais, emoções e outros fenômenos afetivos, como humor, são visualizados como pontos em um espaço dimensional contínuo [Marsella et al. 2010]. Embora existam outras, estas duas vertentes são as mais comumente utilizadas para alicerçar modelos computacionais de emoções, como por exemplo, WASABI [Becker-Asano 2008], FLAME [El-Nasr et al. 2000], ALMA [Gebhard 2005] e EMA [Marsella and Gratch 2009].

Outro campo de estudo sobre emoções e IA, está em como integrar emoções em mecanismos de tomada de decisão de agentes cognitivos. Em diversas propostas, emoções são implementadas em dois níveis: o primeiro é usualmente associado a um comportamento instintivo [Puică and Florea 2013] e por isso são geralmente associados diretamente às percepções do agente, enquanto as de segundo nível são utilizadas como um componente que influenciará o mecanismo cognitivo e deliberativo do agente, atualizando seu estado emocional [Hernández et al. 2004].

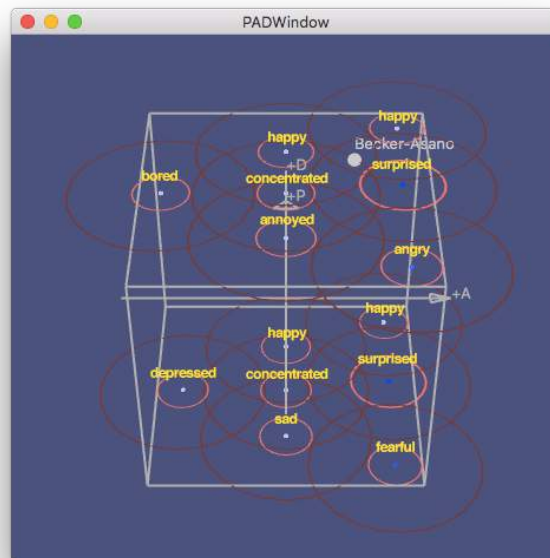
Quanto a sua duração, percebe-se intuitivamente que é necessário um determinado período para que emoções sejam ativadas e desativadas. No trabalho de [Van Dyke Parunak et al. 2006] este fenômeno é modelado com auxílio do conceito de feromônios, fazendo com que emoções sejam dissipadas com o passar do tempo. Além disso, existem feromônios externos, que representam uma maneira de externalizar emoções de modo que outros agentes possam percebê-las. Baseando-se na estrutura do modelo emocional OCC para calcular a intensidade de uma emoção, [da Silva Bitencourt 2014] faz uso de uma *função limiar*, considerando também a potencialidade da emoção e o tempo corrente. A função limiar determina qual é o valor mínimo necessário para que a emoção passe a influenciar o comportamento do agente, enquanto a função de potencialidade dita as chances de uma determinada emoção ocorrer. Quando essas duas funções são associadas ao tempo corrente, pode-se gerar um efeito similar ao uso de feromônios, em que o nível de uma emoção varia gradativamente ao longo do tempo.

Por fim, a arquitetura proposta por [Jiang et al. 2007] fornece resultados que são bastante importantes: em um sistema multiagente, agentes com comportamento egoísta podem apresentar melhorias em suas habilidades de cooperação e agentes BDI com emoção podem atingir seus objetivos com um menor número de ciclos se comparado a implementações sem o uso de emoções.

### 2.1. WASABI

A proposta do modelo WASABI, [Becker-Asano 2008] tem origem no aprimoramento da interação humano-computador. Por conta de limitações existentes em um modelo de

emoções estrutural – como o OCC – no que tange o comportamento crível de agentes emocionais, a ideia que suporta o WASABI envolve a utilização de um espaço tridimensional denominado *Pleasure-Arousal-Dominance Space* ou Espaço PAD [Becker-Asano 2014]. Cada eixo é independente e bipolar representando prazer (*pleasure*), ativação (*arousal*) e dominância (*dominance*), respectivamente. O eixo  $x$  (prazeroso / não-prazeroso), corresponde aos julgamentos avaliativos, quanto maior a avaliação, mais prazeroso. O eixo  $y$  (ativação / desativação), representa o nível de atividade. O eixo  $z$  (dominância / submissão), é uma medida de controle, ou poder [Mehrabian 1996]. As emoções são definidas no espaço PAD através de polígonos e diz-se que um agente percebe determinada emoção quando seu ponto de referência pertence a um dos pontos delimitados pelo círculo externo de uma dada emoção, como pode ser verificado na figura 1. Impulsos, positivos ou negativos, fazem com que o ponto de referência mova-se no espaço e, como num sistema de massa-mola, este ponto tende a voltar para seu local de origem (ou estado homeostático).



**Figura 1. Representação gráfica do espaço PAD**

Ao alterar a massa de um ponto de referência ou os coeficientes das forças que agem sobre os eixos, pode-se fazer com que pontos de referência diversos possuam comportamentos distintos no espaço PAD para um mesmo estímulo e a partir de um mesmo estado atual [Becker-Asano 2008], o que será explorado neste trabalho como um artifício para a definição de "personalidades" dos agentes.

Por ter sido desenvolvido para ser executado em um sistema distribuído, o WASABI tem suporte nativo a execução das principais ações da aplicação através de duas portas de comunicação UDP, uma para leitura, em que podem ser enviados comandos predefinidos, e outra para escrita, em que serão impressos os estados afetivos correntes dos pontos de referência existentes. O código 1 apresenta a estrutura básica para a realização de qualquer ação sobre o WASABI. Os diferentes componentes da mensagem são sempre separados pelo símbolo  $\&$  e todas as mensagens enviadas precisam, obrigatoriamente, começar com um identificador do agente que esta enviando a mensagem,

representado no código por *senderID*, embora, na implementação atual do WASABI, este valor não seja utilizado na avaliação da mensagem. Como uma consequência de ações como a geração de novos impulsos, o estado afetivo de agentes em simulação pelo WASABI deve ser alterado. O WASABI envia mensagens, como a exemplificada no código 2, em que são listadas todas as emoções elicitadas com suas respectivas valências para cada um dos agentes em execução.

**Código 1. Geração de um impulso com intensidade 100**

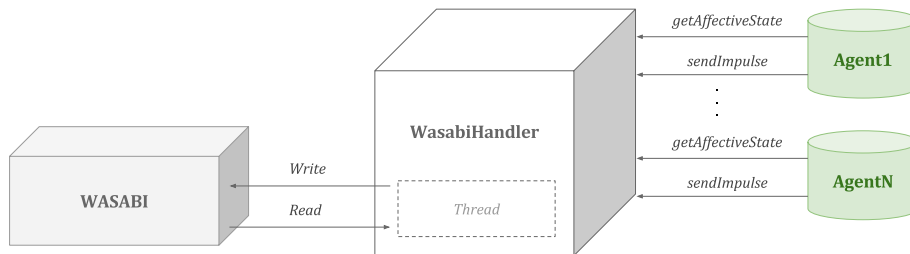
```
senderID&IMPULSE&agentId&100
```

**Código 2. Exemplo de mensagem de retorno.**

```
(10:51:02) Transmit (success) [
  agents: timeStamp=244344992
  total=2
  ID1=ag1 (
    happy=0.5625
    concentrated=0.75
    P=15
    A=-100
    D=0 )
  ID2=ag2 (
    bored=0.72932
    P=15
    A=-100
    D=100 ) ]
```

**3. Integração WASABI-Jason**

O objetivo deste trabalho é o desenvolvimento de uma interface entre o framework Jason e o WASABI, ou seja, um *middleware*, para que o agente, descrito em Jason, possa ter *crenças emocionais*, e possa fazer uso destas crenças no processo de raciocínio BDI. De modo a garantir que a integração pudesse ocorrer sem alterações estruturais tanto no Jason quanto no WASABI, o *middleware* opera basicamente obtendo os estados afetivos, estruturando-os adequadamente para serem utilizados por agentes e abstraindo comandos sobre o WASABI, como o envio de impulsos ou alterações de dominância. A figura 2 ilustra a proposta.



**Figura 2. WasabiHandler**

Para que os estados afetivos calculados pelo WASABI possam ser atribuídos a agentes em execução pelo Jason e que impulsos possam ser disparados diretamente do ambiente para o sistema de simulação afetiva, faz-se necessária a definição de algumas especificações para orientar a solução proposta:

- Impulsos podem ser disparados a partir do ambiente sobre agentes específicos, assim como alterações de dominância, e devem ser propagados tão logo quanto possível;
- Durante a etapa de revisão de crenças, é importante que o agente tenha acesso ao seu estado afetivo mais recente, de modo a permitir a atualização das crenças relacionadas ao comportamento emocional;
- A operação do *middleware* deve ser iniciada e finalizada em conjunto com a execução do sistema multiagente.

Além disso, percebe-se, por exemplo, que na etapa de revisão de crença todos os agentes em simulação esperam ter acesso a uma representação do seu estado afetivo. Para que esta demanda seja suprida, seria preciso monitorar a escrita de uma nova mensagem do WASABI com os valores atualizados e, posteriormente, efetuar o *parsing* desse conteúdo em busca dos valores específicos para o agente requerente. Esta única característica é responsável por influenciar boa parte do projeto do *middleware* e impulsiona a definição de alguns requisitos básicos:

- Considerando a existência de diversos agentes em execução em um mesmo instante de tempo, a obtenção dos estados afetivos não pode ser executada por demanda, a fim de evitar um *overhead* de comunicação e também impactos negativos no tempo de execução da função de revisão de crenças, aumentando assim, a duração do ciclo de tomada de decisão.
- Faz-se necessário o desenvolvimento de um *buffer*, que conterà os valores mais recentes dos estados afetivos de todos os agentes até que uma nova atualização seja indicada pelo WASABI.
- As informações armazenadas no *buffer* precisam estar disponíveis para todos os agentes em simulação.

Para desenvolver a solução, uma *thread* específica para a comunicação com o WASABI foi criada, a fim de obter os valores dos estados afetivos e evitar o *overhead* das requisições individuais geradas pelos agentes. A operação desta *thread* restringe-se apenas à obtenção dos estados afetivos calculados pelo WASABI e à sua estruturação de modo a permitir acesso rápido ao ser requisitada pelos agentes. Como a busca e a inserção em tabelas *hash* possuem complexidade  $O(1)$ , esta é a estrutura utilizada para armazenar as emoções dos agentes, em que o nome do agente serve como chave e o valor representa a emoção mais expressiva da verificação mais recente. Todos os agentes em operação possuem acesso aos valores desta lista pois, antes de operarem sobre o *middleware*, a instância da classe é obtida.

É importante perceber que, como apenas a emoção de maior intensidade é armazenada, a utilização desta informação pelos agentes é simplificada, uma vez que se faz necessário verificar apenas se o agente está feliz, e não se está feliz com uma intensidade determinada. Por outro lado, perde-se a possibilidade de se construir planos mais subjetivos e baseados em o agente estar incomodado com intensidade 0.54 e, ao mesmo tempo, irritado com intensidade 0.21, por exemplo.

### 3.1. Preparação do WASABI

A posição dos pontos de referência no espaço PAD são atualizados com uma frequência de 50Hz, podendo ser customizada para qualquer valor no intervalo [0, 50]. Esta taxa de

atualização, no entanto, não influencia a frequência de envio das informações sobre os estados afetivos através da porta de comunicação UDP, que é definida separadamente.

Ainda que a frequência de execução seja mais pertinente como uma característica do problema em questão que da própria integração, é importante notar que os tempos de execução do WASABI e do Jason não são sincronizados por definição, o que compromete o *debug*. Já quanto a taxa de envio, ou *sendrate*, recomenda-se que a frequência seja alta o suficiente para que as estruturas que representam os estados afetivos dos agentes sejam atualizadas consistentemente. Como referência, neste trabalho as frequências de execução e envio foram definidas em 25Hz.

Um detalhe importante para o sucesso da implementação consiste na correta criação de novos pontos de referência no espaço PAD, que precisam ser definidos com o mesmo identificador utilizado no Jason durante a execução de uma simulação.

### 3.2. Integração com Jason

Considerando as especificações e requisitos necessários à proposta, pode-se restringir o escopo da integração com Jason às seguintes etapas:

1. Controle de execução (inicialização e finalização) do *middleware*;
2. Sincronização entre crenças e os estados afetivos correntes.
3. Suporte a geração de novos impulsos e atualizações de dominância em agentes;

Atrelar a inicialização do *middleware* à execução do sistema multiagente (MAS) foi bastante simples, uma vez que a estrutura que define o ambiente faz uso de métodos *init* e *stop* para estabelecer o que ocorre quando uma simulação é iniciada ou finalizada. O desenvolvimento de um novo cenário geralmente levará a extensão desta classe que define o ambiente (ainda que não seja uma obrigatoriedade), permitindo a sobrescrita dos métodos descritos como demonstrado no código 3 que, ainda que trivial, explicita a criação da thread a partir de uma instância global da classe, respeitando o padrão *Singleton*.

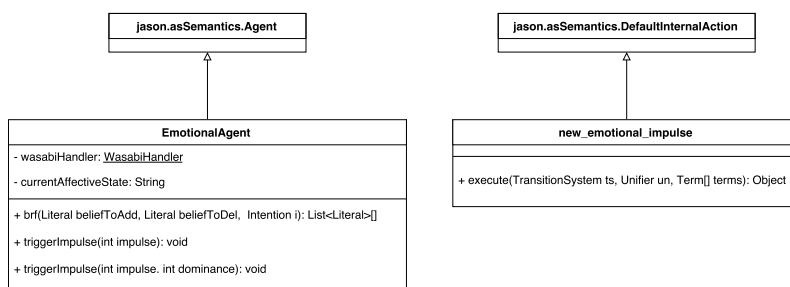
#### Código 3. Inicialização da thread de atualização do middleware

```
@Override
public void init(String [] args) {
    initWorld();

    wasabiHandler = new Thread(WasabiHandler.getInstance());
    wasabiHandler.start();
}
@Override
public void stop() {
    super.stop();
    wasabiHandler.stop();
}
```

A sincronização de crenças com os estados afetivos, por sua vez, ocorre na etapa de revisão de crenças do ciclo de tomada de decisões de agentes BDI. Seguindo a mesma estratégia anterior, que também será reproduzida na próxima etapa, a criação de uma nova classe para a definição de agentes, denominada *EmotionalAgent* (figura 3), que estenda a definição original do Jason é suficiente para que se possa sobrescrever a definição das operações que ocorrem no método *brf*, destinado à revisão de crenças.

Durante esta etapa, o estado afetivo atualizado é requerido através de uma instância do *WasabiHandler*. Se o valor resultante for igual ao valor armazenado pelo agente (e já repassado como uma crença), nada ocorre, no entanto, se houver variação, o valor armazenado é utilizado para indicar a crença que precisa ser removida e o novo estado afetivo determinará a crença a ser adicionada.



**Figura 3. Diagrama das classes EmotionalAgent e newEmotionalImpulse**

O envio de impulsos, por fim, ocorre através de ações internas que podem ser disparadas utilizando a linguagem Jason durante a própria definição dos planos do agente. Esta facilidade torna-se possível ao definir uma nova classe que represente a ação desejada e estenda a classe *DefaultInternalAction*. De funcionamento bastante simples, a única tarefa desta ação é a de, a partir de uma instância do agente, requisitar o método responsável por enviar um determinado impulso para o WASABI através do *middleware* e, opcionalmente, informar um novo valor de dominância para solicitar sua alteração.

### 3.3. WASABI e Personalidade

Atribuir personalidades aos agentes através do WASABI consiste em manipular a massa e as forças  $F_x$  e  $F_y$  que agem sobre o ponto de referência. Na tese que o originou [Becker-Asano 2008], discute-se um parâmetro  $\alpha$  que, dependendo da magnitude dos valores utilizados para defini-lo, poderia culminar na concepção de agentes letárgicos ou temperamentais. Ainda que a utilização deste parâmetro estivesse completamente alinhada aos interesses deste trabalho, não foi encontrada uma interface para manipulação desse valor, seja ela visual ou por código. Por conta deste contratempo, a massa do ponto de referência foi utilizada para prover a diferenciação emocional entre os agentes, uma vez que, influencia na ação das forças  $F_x$  e  $F_y$  sem que as mesmas precisem ser alteradas.

Para a definição dos pesos a serem aplicados, manteve-se o valor padrão utilizado pelo WASABI como um valor intermediário e outros dois novos valores foram empregados para que fosse possível verificar o comportamento dos agentes nos pontos extremos. Com isso, os pesos utilizados compõem o conjunto:  $\{500, 5000, 9900\}$ . Coerente com os resultados esperados de um sistema massa-mola, com uma massa menor (valor 500), rapidamente o ponto de referência retorna ao seu estado homeostático, enquanto com uma massa maior (valor 9900), este retorno é mais gradual. Na prática, o que deve ocorrer é que, após o recebimento de um impulso positivo, por exemplo, um agente que possui seu comportamento emocional atrelado a um ponto de referência com um peso pequeno para sua massa deve retornar rapidamente para o estado *concentrated*, enquanto o comportamento oposto é esperado de um agente representado por um ponto de referência com peso consideravelmente superior.

#### 4. Cenário de teste e Experimentos

Para verificar a atuação do WASABI como simulador de emoções em um sistema de agentes foi utilizado o cenário, proposto em [Bordini et al. 2006]. Este cenário é composto por um grupo de agentes encarregados de coletarem pepitas de ouro disponíveis em um ambiente (figura 4). Embora tenha sido desenvolvida uma versão em que existam equipes distintas que competem entre si, a versão pertinente para este estudo supõem a existência de um ambiente orientado à colaboração que, em decorrência da sua extensão para contemplar o uso de emoções, apresentará alguns componentes que sugerem um funcionamento competitivo entre os agentes.

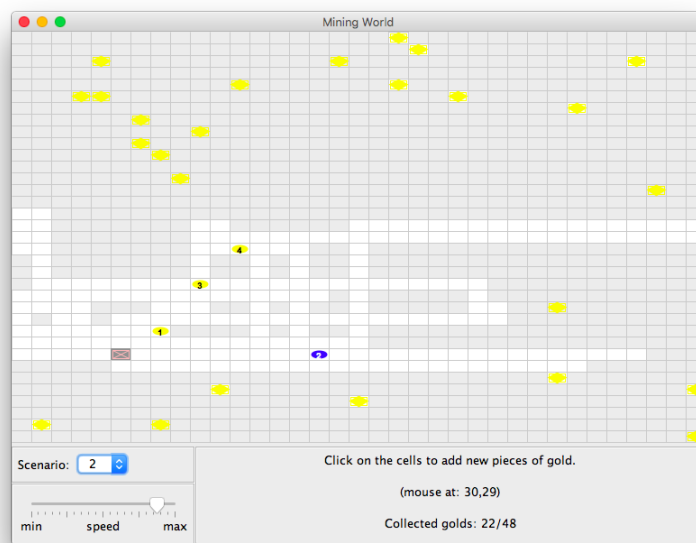


Figura 4. Representação gráfica do problema dos mineiros

Naturalmente, em ambientes em que se assume a existência de colaboração, é esperado que algum tipo de comunicação ocorra entre os agentes. Neste caso em particular, também para aprimorar os resultados obtidos, existe não apenas a comunicação entre agentes mas pequenas negociações para determinar qual agente será responsável por coletar o recurso. Esta negociação ocorre quando um determinado agente já está comprometido a coletar uma pepita de ouro e, com isso, move-se em direção a este objeto para coletá-lo, por exemplo, mas durante seu trajeto, depara-se com uma outra pepita. Neste momento, como a nova pepita está mais próxima que a anterior, o agente decide por coletar a mais próxima, abandonando o interesse pela anterior, e comunicando aos demais a localização da pepita que desejava obter inicialmente. A figura de um líder é definida como um artifício para simplificar as negociações entre os agentes. Assim que um agente informa a localização de uma pepita de ouro disponível aos demais, cada agente interessado em coletá-la irá comunicar ao líder sua posição corrente no ambiente. O líder, com isso, atribuirá como responsável pela coleta aquele que estiver mais próximo do objetivo. Todas as características deste cenário discutidas até então, foram definidas como tais pelos seus próprios autores. Não foram realizadas modificações no comportamento padrão dos agentes, apenas extensões para contemplar o uso de emoções, como será discutido na seção a seguir.



## 4.1. Integração do Componente Emocional

O cenário dos mineiros<sup>1</sup> foi estendido de modo que, ao coletar uma pepita de ouro, um estímulo positivo seja gerado para o agente coletor. Ao receber uma determinada quantidade de estímulos positivos, e neste caso a quantidade necessária varia de acordo com a personalidade do agente, um agente eventualmente deve encontrar-se em um estado afetivo que denote a felicidade. Quando está feliz, entretanto, um agente passa a tornar-se ganancioso e, no lugar de informar para os demais agentes a localização de uma pepita que ele mesmo não possa carregar, ele mente, informando uma localização aleatória. Deste modo, a pepita de ouro avistada tende a ficar reservada apenas para si, como demonstrado esquematicamente em 4.

### Código 4. Influência de crenças emocionais no comportamento do agente

```
+cell(X,Y,gold) : not committed(gold(X,Y)) & happy
  <- +gold(X,Y);
     .broadcast(tell , gold(RandomX,RandomY)).

+cell(X,Y,gold) : not committed(gold(X,Y))
  <- +gold(X,Y);
     .broadcast(tell , gold(X,Y)).
```

No entanto, como uma penalização indireta, ao perceber que na localização informada não existe nenhuma pepita de ouro disponível, o agente que recebeu a informação passa a tratar o emissor como mentiroso e deixa de informar para este agente a localização de pepitas disponíveis enquanto estiver ocupado.

## 4.2. Experimentos

As simulações foram realizadas tomando como medida de qualidade a quantidade de pepitas de ouro obtidas pelos agentes. Desta forma, foram realizadas 10 simulações e o valor considerado é a média de objetos coletados. O desvio padrão também foi calculado para determinar a consistência dos resultados.

Todos os experimentos foram executados utilizando a mesma frequência de execução, até que um limite de 100 objetos coletados tivesse sido atingido e partindo de uma nova instância do WASABI, garantindo que não houvesse influência de simulações anteriores. Em ambos os casos, o desempenho de quatro agentes distintos é verificado, um BDI e três agentes EBDI com valores de massa iguais a 500, 5000 e 9900, conforme discutido na seção 3.3.

Os resultados obtidos a partir da simulação e apresentados na Tabela 4.2, foram bastante consistentes e sugerem que a utilização de agentes emocionais pode efetivamente promover benefícios.

A maior consistência entre as simulações é observada no agente BDI, visto que apresenta o menor desvio entre os agentes avaliados. No entanto, este também apresenta o pior desempenho. Para que se possa compreender as razões que fundamentam os resultados obtidos é preciso considerar que:

---

<sup>1</sup>Foram desenvolvidos dois cenários distintos como uma prova de conceito mas, por falta de espaço, cita-se apenas o problema dos mineiros.

- O tamanho do mapa utilizado (36x36) favorece a acumulação de ouros próximos uns dos outros, o que faz com que um agente emocional gere impulsos positivos com bastante frequência, o que deve deixá-lo feliz e conseqüentemente ganancioso;
- Como os agentes emocionais tendem a estar felizes com frequência, mesmo que os impulsos negativos ao descobrirem que foram enganados tenha valência máxima, isso não deve deixá-los irritados diretamente, uma vez que, a influência que os impulsos oferecem varia de acordo com o estado corrente do agente e um agente feliz é mais sensível a impulsos positivos e menos sensível a impulsos negativos.
- Por não considerar emoções, o agente BDI não torna-se ganancioso, o que faz com que continue sempre divulgando, ao estar ocupado, a localização de pepitas de ouro encontradas para todos os demais agentes. Não há a distinção entre agentes que mentiram ou não, uma vez que a identificação de uma mentira está atrelada ao agente estar irritado.

**Tabela 1. Resultados obtidos a partir do cenário dos mineiros.**

Total de Ouros Obtidos				
Experimento	BDI	EBDI-500	EBDI-5000	EBDI-9900
<b>1</b>	20	23	26	31
<b>2</b>	24	22	26	28
<b>3</b>	22	26	26	26
<b>4</b>	20	29	23	28
<b>5</b>	21	23	25	31
<b>6</b>	22	27	26	25
<b>7</b>	23	27	22	28
<b>8</b>	22	24	27	27
<b>9</b>	22	24	27	27
<b>10</b>	24	26	22	28
<b>Média</b>	<b>22</b>	<b>25</b>	<b>26</b>	<b>28</b>
Variância	2,000	4,989	3,778	3,656
<b>Desvio Padrão</b>	<b>1,414</b>	<b>2,234</b>	<b>1,944</b>	<b>1,912</b>
Variância Populacional	7,6			

Quão maior a massa do ponto de referência no WASABI, maior será o tempo em que um agente permanecerá em um mesmo estado afetivo. Sabendo que, por conta das características do cenário os agentes tendem a manter-se felizes, o que leva diretamente à ganância, percebe-se que o melhor desempenho decorreu justamente dos agentes que mais mentiram e guardaram a localização dos ouros encontrados para si.

## 5. Considerações Finais

Este trabalho apresentou um *middleware* que permite que estados emocionais sejam adicionados como crenças e utilizados no ciclo de raciocínio do agente, influenciando na tomada de decisão. Diferentemente dos trabalhos de [Van Dyke Parunak et al. 2006] e

[da Silva Bitencourt 2014] que enfatizam como emoções são modeladas, ou da proposta de [Jiang et al. 2007] que define um modelo de agente emocionais, o trabalho ora apresentado integra o modelo de emoções WASABI com a ferramenta para desenvolvimento de agentes Jason.

Ainda que uma avaliação mais profunda dependa de um volume muito maior de experimentos, a verificação feita neste trabalho fornece um panorama bastante interessante sobre a influência da personalidade ao se modelar agentes emocionais. É importante observar que a determinação de uma "personalidade" ao agente, pode resultar em um comportamento que impeça o agente de atingir seus objetivos (modelando um agente que tem medo, por exemplo). Tais variações mostram que, ao utilizar-se de agentes dotados de emoções, não basta definir o comportamento do agente e as limitações do meio. Todas as nuances que constituem a personalidade emocional dos agentes estão diretamente associadas ao seu desempenho no ambiente. Para determinados casos de uso, agentes com uma menor variação entre estados afetivos, como no cenário apresentado, podem atingir seus objetivos com maior facilidade, no entanto, quando expostos a um ambiente distinto, este agente pode ter um desempenho bastante similar ou mesmo inferior ao de um agente BDI.

Durante a implementação, diversos desafios foram encontrados e nem todos puderam ser endereçados adequadamente. Um dos mais importantes está na própria definição da arquitetura WASABI que, por ser constituída de um sistema massa-mola, promove um retorno gradual ao estado homeostático de cada agente, mas não possibilita que a evolução para um determinado estado, após receber um impulso qualquer, ocorra também de maneira gradual, o que impede de modelar agentes que mudam gradativamente seu estado emocional. Apesar desta limitação, integrar o *framework* Jason com a arquitetura WASABI mostrou-se um conjunto bastante versátil para a construção de agentes EBDI.

## 5.1. Trabalhos Futuros

As aplicações WASABI e Jason permanecem operando de maneira independente e com frequências de execução possivelmente distintas, característica esta que pode vir a ser um problema quando da utilização do *mind inspector* do Jason. Uma possível melhoria seria permitir a definição de um passo customizado, através da ampliação da API do Jason.

Ainda, além de se considerar a utilização de um outro sistema que não um de massa-mola para determinar as forças que agem sobre um ponto de referência, seria possível fazer com que fossem utilizados quatro sistemas massa-mola, no lugar de apenas dois para as forças  $F_x$  e  $F_y$ . Os dois primeiros seriam mantidos como originalmente definidos, enquanto os dois últimos passariam a existir assim que tenha sido calculada uma nova posição para o ponto no espaço PAD, com sua origem neste novo ponto, permitindo assim, a variação gradual também ao receber impulsos.

O próprio *middleware* responsável pela integração WASABI-Jason poderia ser otimizado para, no lugar de enviar um impulso para a arquitetura tão logo o tenha recebido, passasse a adicioná-los em um *buffer* e a enviá-los todos ao mesmo tempo como parte do *loop* de verificação de novas mensagens.

Visto que a interação entre o WASABI e o Jason contempla apenas alterações na base de crenças dos agentes, o próprio componente de raciocínio do Jason pode ser melhor explorado, de modo a permitir a influência das emoções de maneira mais direta.

## Referências

- Becker-Asano, C. (2008). *WASABI: Affect simulation for agents with believable interactivity*, volume 319. IOS Press.
- Becker-Asano, C. (2014). Wasabi for affect simulation in human-computer interaction. In *Proc. Intern. Workshop on Emotion Representations and Modelling for HCI Systems*.
- Bordini, R. H., Hübner, J. F., and Tralamazza, D. M. (2006). Using jason to implement a team of gold miners. In *Comput. Logic in Multi-Agent Systems*, pages 304–313. Springer.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons.
- da Silva Bitencourt, G. K. (2014). Um modelo de confiança baseado em emoções para agentes. Master's thesis, Universidade Federal de Santa Catarina.
- 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(3):219–257.
- Gebhard, P. (2005). Alma: a layered model of affect. In *Proc. of the 4<sup>th</sup> Intern. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 29–36. ACM.
- Hernández, D. J., Déniz, O., Lorenzo, J., and Hernández, M. (2004). Bdie: a bdi like architecture with emotional capabilities. In *American Association for Artificial Intelligence Spring Symposium*.
- Jiang, H., Vidal, J. M., and Huhns, M. N. (2007). EBDI: an architecture for emotional agents. In *Proc. of the 6<sup>th</sup> Intern. Joint Conf. on Autonomous Agents and Multiagent Systems*, page 11. ACM.
- Lerner, J. S., Li, Y., Valdesolo, P., and Kassam, K. S. (2015). Emotion and decision making. *Psychology*, 66.
- Marsella, S., Gratch, J., and Petta, P. (2010). Computational models of emotion. *A Blueprint for Affective Computing-A sourcebook and manual*, pages 21–46.
- Marsella, S. C. and Gratch, J. (2009). Ema: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1):70–90.
- Mehrabian, A. (1996). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292.
- Ortony, A., Clore, G., and Collins, A. (1990). *The Cognitive Structure of Emotions*. Cambridge University Press.
- Puică, M.-A. and Florea, A.-M. (2013). Emotional belief-desire-intention agent model: Previous work and proposed architecture. *Intern. Journal of Advanced Research in Artificial Intelligence*, 2(2):1–8.
- Scherer, K. R. (1999). Appraisal theory. *Handbook of cognition and emotion*, pages 637–663.
- Van Dyke Parunak, H., Bisson, R., Brueckner, S., Matthews, R., and Sauter, J. (2006). A model of emotions for situated agents. In *Proc. of the 5<sup>th</sup> Intern. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 993–995. ACM.