

Criação de Agentes BDI a partir de Modelos do UPPAAL*

João Vicente Markovicz¹, Gleifer Vaz Alves¹, André Pinz Borges¹

¹Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná (UTFPR)
Ponta Grossa – PR – Brasil

joao.080603@alunos.utfpr.edu.br gleifer@utfpr.edu.br

apborges@utfpr.edu.br

Abstract. *This paper presents the extension of ethical models as automata built using the UPPAAL tool. These models are translated into prototypes of agents using the BDI architecture. This translation aims to establish a direct relationship between formal modelling and agent implementation.*

Resumo. *Este artigo apresenta a extensão de modelos éticos em forma de autômatos construídos na ferramenta UPPAAL. Tais modelos são traduzidos em protótipos de agentes BDI. Esta tradução busca estabelecer uma relação direta entre a modelagem formal e a implementação dos agentes.*

1. Introdução

Sistemas inteligentes para automação de tarefas simples ou complexas (como o controle de um veículo autônomo) estão cada vez mais presentes no nosso dia-a-dia e demandam tomadas de decisão conforme princípios éticos [Alves et al. 2021]. Esta conformidade garante que tais sistemas operem de maneira responsável, seguindo regras éticas. Por exemplo, um VA não deve tomar decisões imprudentes perante as normas de trânsito, de modo a proteger as vidas dos seus ocupantes e demais envolvidos. A automação dos sistemas pode ser desenvolvida com uso de agentes inteligentes [Alves et al. 2021]. Os aspectos éticos destas soluções podem ser baseados em trabalhos como Bench-Capon [Bench-Capon 2020], que disserta sobre três abordagens éticas aplicadas em sistemas autônomos (Consequencialismo, Deontologia e Ética das Virtudes) e como elas poderiam ser usadas para tomar decisões. Os comportamentos dos agentes pode ser modelado como autômatos, o que leva ao trabalho de [Markovicz and Alves 2023], onde é apresentada uma modelagem formal desses três modelos éticos usando autômatos temporais com a ferramenta UPPAAL [Bengtsson et al. 1996]. O objetivo deste artigo é apresentar uma extensão dos modelos construídos em [Markovicz and Alves 2023], e a tradução dos modelos formais para geração de protótipos visando a programação de agentes seguindo o paradigma de agentes BDI.

2. Modelos Éticos e Agentes BDI

Bench-Capon descreve as abordagens éticas Consequencialista, Deontológica e Ética das Virtudes, e faz apontamentos sobre aplicar cada abordagem em agentes éticos. Ele utiliza

*Copyright© 2024 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

um cenário específico, a história “A Cigarra e a Formiga“, onde a Formiga trabalha no verão e a Cigarra joga. No inverno, a Formiga tem comida suficiente para si e uma sobra que pode ser usada para uma festa no fim da estação. Já a Cigarra dependerá da ajuda da formiga para sobreviver. Este dilema ético que pode tomar diferentes caminhos dependendo da abordagem escolhida.

Em [Markovicz and Alves 2023] foram construídos autômatos para cada uma das abordagens utilizando o *UPPAAL* [Bengtsson et al. 1996], uma ferramenta que permite a modelagem e verificação formal de autômatos temporais de forma fácil e dinâmica, possibilitando até mesmo a simulação do autômato construído. Nesse trabalho, será usada uma nova versão dos modelos construídos, descritos na Seção 3.

Conforme declarado, esse trabalho busca estabelecer uma conexão entre os modelos formais previamente construídos e a sua implementação usando agentes BDI [Bratman 1987], que é um tipo de arquitetura de agentes baseada na estrutura: *Beliefs* (Crenças), *Desires* (Objetivos ou Desejos) e *Intentions* (Intenções ou Planos). As crenças representam o conhecimento que o agente possui sobre si ou o ambiente que está inserido. Os objetivos estabelecem aquilo que o agente pretende alcançar. Os planos organizam as ações que o agente deve executar para chegar aos seus objetivos, esses planos tem o seguinte formato: *TriggerEvent* : {*Context*} \leftarrow *Body*, onde *Trigger Event* é definido pela adição de numa nova crença ou objetivo, o *Context* é determinado por um conjunto de crenças e o *Body* é uma sequência de ações a serem executadas pelo agente [Bordini et al. 2007].

3. Modelagem formal das abordagens éticas

Os modelos descritos em [Markovicz and Alves 2023] são distintos na etapa da decisão de cada um. O modelo Deontológico usa na decisão apenas a variável *played_times*, que conta a quantidade de vezes que o agente joga. Já o modelo Consequencialista usa a variável *balance*, que incrementa quando um agente trabalha e decrementa quando ele joga. Por fim, o modelo da Ética das Virtudes, faz o uso de ambas as variáveis para a decisão. Todos esses modelos possuem um agente único que pode ser instanciado várias vezes e se comportar como formiga ou cigarra.

Neste artigo será utilizado apenas o modelo da Ética das Virtudes, que pode ser visto como o mais robusto dentre os três. Aqui em uma nova versão onde os papéis (formiga e cigarra) foram separados gerando dois agentes distintos, um representando a Formiga e outro a Cigarra. É importante notar que foram usadas técnicas de abstração para desenvolver modelos que representem o que é necessário para o entendimento do comportamento de cada agente do sistema, tendo em vista o processo de tradução que será descrito na seção 4.

3.1. Autômato Formiga

No autômato da Formiga (Fig. 1), tem-se um agente que, saindo do estado inicial, pode ir para o estado **working** (a variável *food* aumenta 2), após isso, pode ir para **eating** (*food* diminui 1). Então, este agente pode ir para **feasting**, que seria uma festa no fim da estação (*food* diminui 1) ou pode receber um pedido por comida de um agente Cigarra, indo para **negotiating**. Neste estado, o agente ativa um outro autômato que acessa as informações da Cigarra e faz o processo de decisão, ao voltar para a

Formiga, esta envia uma mensagem à Cigarra aceitando (**give_food**) ou recusando o pedido (**refuse_food**).

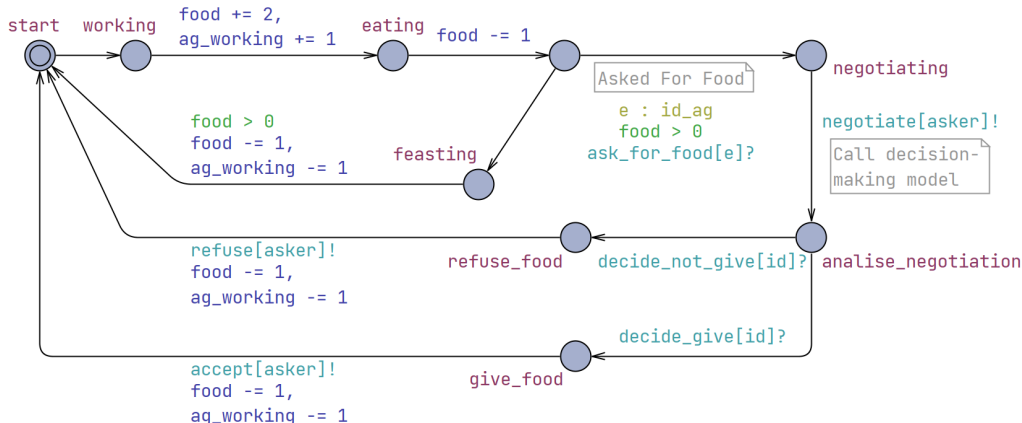


Figura 1. Autômato Formiga (UPPAAL)

3.2. Autômato Cigarra

Já o autômato da Cigarra (Fig. 2), saindo do estado inicial, possui uma escolha entre trabalhar ou jogar (respectivamente os estados **working** e **playing**, com um peso de 50% para cada). Além disso, são atualizadas as variáveis `played_times` (incrementa 1 caso o agente escolha jogar) e `balance` (decrementa 1 caso o agente jogue, incrementa 1 caso o agente trabalhe).

Caso o agente decidir trabalhar, segue para os estados **eating** e **feasting** similarmente ao que ocorre com a Formiga. Caso escolha jogar, vai para o estado **ask_for_food** e deverá recorrer a ajuda de algum agente Formiga utilizando troca de mensagens. A Cigarra espera até receber um sinal `refuse` (o pedido foi recusado e a Cigarra morre) ou o sinal `accept` (a Cigarra recebe a comida chegando a **received_food**, segue para **eat_given_food**, então usando a comida volta ao estado inicial).

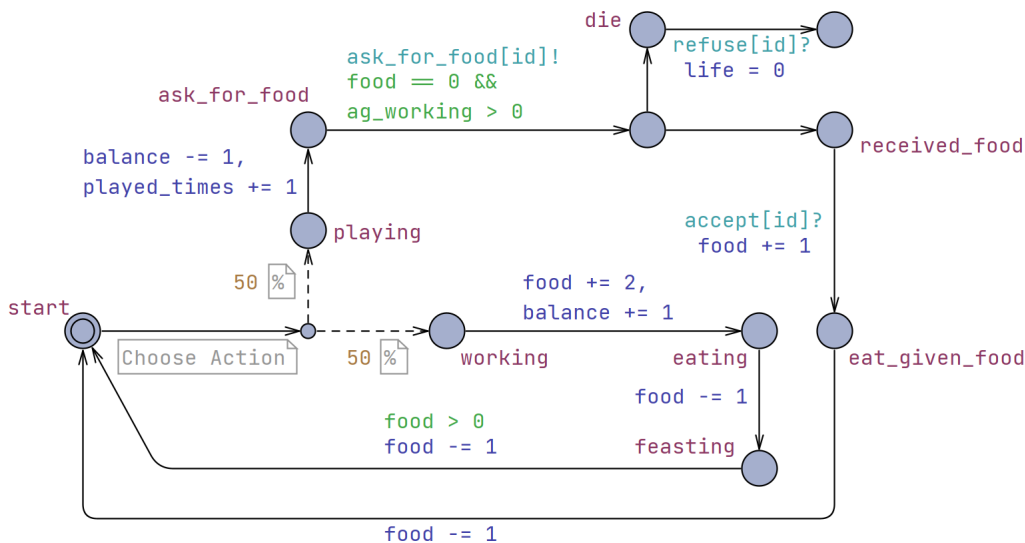


Figura 2. Autômato Cigarra (UPPAAL)

4. Tradução dos Modelos para Agentes BDI

Com os autômatos previamente apresentados, busca-se estabelecer uma conexão entre o modelo ético construído e a sua devida implementação como agentes BDI. Neste momento, é descrita uma tradução para um formato comumente utilizado por linguagens de agentes BDI, mas sem determinar uma linguagem alvo específica.

Nos protótipos 1 e 2 tem-se a definição das crenças e objetivos dos agentes Formiga (Ant) e Cigarra (GH) respectivamente. Nesse caso, as crenças foram mapeadas com base nas variáveis declaradas no UPPAAL e os objetivos são os estados do autômato. Cabe ressaltar que foram colocados apenas os objetivos iniciais, visto que outros deverão ser adicionados à medida que o agente executa seus planos.

A Formiga possui as crenças iniciais *life(1)* que controla a vida da formiga, e *food(0)* que controla a quantidade de comida que ela possui. Já como objetivo inicial tem-se apenas *start*.

```
1 Beliefs: life(1), food(0);
2 Goals: start;
```

Protótipo 1. Crenças e Objetivos - Agente Formiga

A Cigarra, possui as mesmas crenças *life(1)* e *food(0)*, além de *balance(0)* e *played_times(0)*, que eram as variáveis de controle presentes no autômato construído. Para objetivo inicial tem-se apenas *start*.

```
1 Beliefs: life(1), food(0), balance(0), played_times(0);
2 Goals: start;
```

Protótipo 2. Crenças e Objetivos - Agente Cigarra

Nos protótipos 3 e 4 há a definição dos planos para ambos os agentes. Para efetuar a tradução foi utilizado como base o mecanismo ilustrado na Figura 3, onde os estados e transição do autômato são usados para gerar o respectivo plano do agente. Especificamente, obtém-se o seguinte plano: $State1 : \{Conditions\} \leftarrow Commands, send(Agent, Communication), State2$; onde State 1 determina o *Trigger Event* do plano, o *Context* do plano é obtido a partir das *conditions*, enquanto o *Body* é determinado pelos *commands*, *communication* e ainda o próprio State 2. Note que nem toda transição do autômato necessariamente tem os três elementos: *conditions*, *commands* ou *communication*.

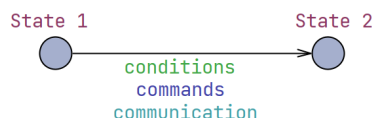


Figura 3. Visão geral do Autômato

No protótipo 3, há a definição dos planos do agente Formiga:

- **start** - O agente inicia, no contexto não possui pré-requisito, no corpo adiciona o objetivo *working*;

- **working** - O agente trabalha, no contexto seleciona a crença *food*, no corpo aumenta *food* em 2 e adiciona o objetivo *eating*;
- **eating** - O agente utiliza sua comida, no contexto seleciona a crença *food*, no corpo diminui 1 em *food*;
- **negotiate(GH)** - Recebe o pedido de negociação do agente GH, utiliza uma função do ambiente *analise_negotiation()*, que irá analisar o pedido com base nas informações de GH, assim como é realizado no autômato;
- **give_food(GH)** - Plano que dá comida ao agente GH, no contexto é necessário possuir *food(1)*, no corpo envia uma mensagem ao GH informando que o pedido foi aceito, então é adicionado o objetivo *start*;
- **refuse_food(GH)** - Plano que recusa comida ao agente GH, no corpo envia uma mensagem a ele recusando o pedido, e então adiciona o objetivo *feasting*;
- **feasting** - Realiza a festa, no contexto seleciona a crença *food* que precisa ser maior que 0, no corpo diminui 1 em *food* e adiciona o objetivo *start*.

```

1 start : {true} <- working;
2 working : {food(X)} <- +food(X+2), eating;
3 eating : {food(X)} <- food(X-1);
4 negotiate(GH) : {true} <- analise_negotiation(GH);
5 give_food(GH) : {food(1)} <- send(GH, accept), start;
6 refuse_food(GH) : {true} <- send(GH, refuse), feasting;
7 feasting : {food(X) and X>0} <- food(X-1), start;

```

Protótipo 3. Planos - Agente Formiga

No protótipo 4, há a definição dos planos do agente Cigarra:

- **start** - O agente inicia, no contexto não possui pré-requisito, no corpo adiciona o objetivo *choose_action*;
- **choose_action** - No contexto não possui pré-requisito, no corpo utiliza a função do ambiente *choose_action()* para decidir randomicamente se irá trabalhar ou jogar;
- **working** - O agente trabalha, no contexto seleciona as crenças *food* e *balance*, no corpo aumenta *food* em 2, *balance* em 1 e adiciona o objetivo *eating*;
- **playing** - O agente joga, no contexto seleciona as crenças *balance* e *played_times*, no corpo diminui 1 em *balance*, aumenta 1 em *played_times* e adiciona o objetivo *ask_for_food*;
- **ask_for_food** - O agente pede por comida, no contexto precisa ter a crença *food(0)* e a função do ambiente *ag_working()* precisa retornar *true*, no corpo envia uma mensagem ao Ant pedindo por uma negociação;
- **received_food** - O agente recebe comida, no contexto precisa da crença *accept* e seleciona a crença *food*, no corpo *food* aumenta em 1 e adiciona o objetivo *eat_given_food*;
- **eat_given_food** - O agente usa a comida que recebeu, no contexto seleciona a crença *food*, no corpo diminui 1 em *food* e adiciona o objetivo *start*;
- **eating** - O agente utiliza sua comida, no contexto seleciona a crença *food*, no corpo diminui 1 em *food* e adiciona o objetivo *feasting*;
- **feasting** - Realiza a festa, no contexto seleciona a crença *food* que precisa ser maior que 0, no corpo diminui 1 em *food* e adiciona o objetivo *start*;

- **die** - O agente morre, no contexto precisa da crença *refuse* indicando que o pedido por comida foi recusado, no corpo atualiza *life* para 0;

```

1 start : {true} <- choose_action;
2 choose_action : {true} <- choose_action();
3 working : {food(X), balance(Y)} <- food(X+2), balance(Y+1), eating ;
4 playing : {balance(X), played_times(Y)} <-
5     balance(X-1), played_times(Y+1), ask_for_food;
6 ask_for_food : {food(0) and ag_working()} <- send(Ant, negotiate(GH));
7 received_food : {accept and food(X)} <- food(X+1), eat_given_food;
8 eat_given_food : {food(X)} <- food(X-1), start;
9 eating : {food(X)} <- food(X-1), feasting;
10 feasting : {food(X)} <- food(X-1), start;
11 die : {refuse} <- life(0);

```

Protótipo 4. Planos - Agente Cigarra

5. Considerações Finais

Nesse artigo os modelos éticos construídos em [Markovicz and Alves 2023] foram estendidos, de forma a favorecer a transição para um protótipo usando o paradigma de agentes BDI. Como trabalho futuro, pretende-se implementar o protótipo apresentado além de suas ações internas, em uma linguagem para programação de agentes, como JASON [Bordini et al. 2007] ou GWENDOLEN [Dennis and Müller 2008]. Além de realizar uma extensão para agentes que representem veículos autônomos em cenários específicos que necessitem tomada de decisão ética.

Referências

- Alves, G., Dennis, L., and Fisher, M. (2021). An Agent-based architecture with support to Ethical Decisions on a Road Traffic Scenario. Publisher: Zenodo.
- Bench-Capon, T. (2020). Ethical approaches and autonomous systems. *Artificial Intelligence*, 281:103239.
- Bengtsson, J., Larsen, K., Larsson, F., Pettersson, P., and Yi, W. (1996). UPPAAL — a tool suite for automatic verification of real-time systems. In Alur, R., Henzinger, T. A., and Sontag, E. D., editors, *Hybrid Systems III*, number 1066 in Lecture Notes in Computer Science, pages 232–243. Springer Berlin Heidelberg.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, Inc., Hoboken, NJ, USA.
- Bratman, M. (1987). *Intention, Plans, and Practical Reason*. Cambridge: Cambridge, MA: Harvard University Press.
- Dennis, L. and Müller, B. (2008). Gwendolen: A bdi language for verifiable agents.
- Markovicz, J. V. and Alves, G. V. (2023). Modelagem formal de abordagens éticas para comportamento de agentes. In *Anais do Workshop-Escola de Informática Teórica (WEIT)*, pages 134–138.