

# Implementando Agentes Baseados no Modelo de Processamento de Objetivos Baseado em Crenças\*

**Henrique M. R. Jasinski<sup>1</sup>, Mariela Morveli-Espinoza<sup>1</sup>, Cesar A. Tacla<sup>1</sup>**

<sup>1</sup>CPGEI – Universidade Tecnológica Federal do Paraná (UTFPR)  
Av. Sete de Setembro, 3165 – 80230-901 – Curitiba, Brasil

henriquejasinski@alunos.utfpr.edu.br, morveli.espinoza@gmail.com,  
tacla@utfpr.edu.br

**Abstract.** The objective of this paper is to propose an architecture and discuss the design and implementation of a Belief-Based Goal Processing (BBGP) based agent simulator. An implementation of the model could be employed on the study of the conceptual model BBGP besides simplifying the analysis and explanation of the agent's reasoning process. In this work we show how the model was adapted for the simulator implementation<sup>1</sup>.

**Resumo.** O objetivo deste artigo é propor uma arquitetura e discutir o design e implementação de um simulador de agentes baseados em Processamento de Objetivos Baseado em Crenças (BBGP). Uma implementação do modelo poderia ser empregada em estudos do modelo conceitual BBGP além de tornar mais simples a análise e explicação do processo de raciocínio do agente. Neste trabalho mostramos como o modelo foi adaptado para a implementação do simulador<sup>1</sup>.

## 1. Introdução

Agentes computacionais têm sido amplamente utilizados para diversas finalidades, como negociação automática, controle de robôs autônomos e simulação de fenômenos coletivos. Neste trabalho iremos abordar um modelo específico de agente cognitivo, o modelo de Processamento de Objetivos Baseado em Crenças (*Belief-Based Goal Processing*), proposto por Castelfranchi e Paglieri (2007), modelo ao qual iremos nos referir de forma sucinta como BBGP.

Em [Castelfranchi e Paglieri 2007], os autores afirmam que no modelo *Beliefs-Desires-Intentions* (BDI) [Rao e Gergeff 1995] e em outros modelos a interação entre crenças e objetivos foi abordada com um foco desproporcional na formalização e implementação, com pouco esforço na análise cognitiva e clarificação conceitual. Os autores defendem que não é necessário haver uma primitiva no modelo para representar intenções, uma vez que estas podem ser precisamente definidas em termos de crenças e



\* O trabalho Implementando Agentes Baseados no Modelo de Processamento de Objetivos de Henrique M. R. Jasinski, Mariela Morveli-Espinoza e Cesar A. Tacla está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 - e Centro Nacional de Pesquisa (CNPq).

<sup>1</sup> O código fonte está disponível em: < <https://github.com/henriquemonteiro/BBGP-Agent-Simulator> >

objetivos. No modelo BBGP os objetivos possuem quatro estágios distintos (ativo, perseguiável, escolhido e executivo) enquanto no modelo BDI possuem dois estágios (desejo e intenção). [Castelfranchi, Paglieri 2007] fazem um paralelo entre ambos os modelos, assim o estágio “desejo” corresponde ao estágio “ativo”, enquanto “intenção” corresponde à conjunção dos estágios “escolhido” e “executivo”.

Em [Morveli-Espinoza 2018] é proposta uma formalização computacional para o modelo BBGP, ao qual iremos nos referir como BBGP-argumentativo. O modelo proposto faz uso de argumentação computacional para decidir sobre a transição de estágio dos objetivos do agente. Uma implementação deste modelo poderia ser empregada em estudos do funcionamento e factibilidade do modelo conceitual BBGP, assim como facilitaria a compreensão do processo de raciocínio do agente e consequentemente sua análise e explicação. Este trabalho apresenta os resultados da criação de um simulador do processamento de objetivos de um agente BBGP-argumentativo.

Poucos trabalhos práticos empregando o modelo BBGP foram feitos. Em [Sansores e Pavón 2008] tal modelo foi empregado para estender a linguagem do agente para assim empregar critérios dinâmicos na seleção de papel que o agente irá desempenhar. Entretanto tal abordagem difere do BBGP-argumentativo não apenas no uso de argumentação, mas os processos deliberativos do agente são modificados. Este trabalho serve como um protótipo de um interpretador geral para agentes BBGP-argumentativo, assim se assemelhando com Jason [Bordini, Hübner e Vieira 2005], que apresenta um interpretador para agentes BDI.

Na Seção 2 iremos abordar os conceitos de argumentação computacional necessários. A Seção 3 apresenta uma visão geral do modelo BBGP-argumentativo. A arquitetura do agente é definida na Seção 4. O artefato desenvolvido é abordado na Seção 5. A Seção 6 por fim aborda as conclusões e trabalhos futuros.

## 2. Argumentação Computacional

Argumentação abstrata trata os argumentos como elementos atômicos e aborda as relações de ataques entre si. Um *framework* de argumentação abstrata, como estabelecido em [Dung 1995], é um par  $AF = \langle A, \rightarrow \rangle$ , onde  $A$  é um conjunto de argumentos e  $\rightarrow$  uma relação binária  $\subseteq A \times A$ , onde  $(A, B) \in \rightarrow$  é interpretado como “ $A$  ataca  $B$ ”. Os ataques representam conflitos entre os argumentos, com isso é preciso escolher um subconjunto de argumentos coerentes. Uma semântica de aceitabilidade é uma função responsável por retornar conjuntos de argumentos, denominados ‘extensões’, que representam uma visão coerente, ou seja, internamente livre de conflitos.

Argumentação estruturada trata do conteúdo do argumento. Na implementação do simulador foi utilizado o ASPIC+, que trata de argumentação estruturada baseada em lógica, definido em [Prakken 2010]. Uma linguagem lógica é empregada na definição dos literais, onde juntamente com os conjuntos de regras estritas (que representam uma informação sem exceções, ex.: “pinguins não podem voar”) e derrotáveis (representam informações que podem ter exceções, “aves podem voar”) geram um conjunto de argumentos possíveis. Argumentos são definidos como árvores, podendo ter um conjunto de subargumentos, e sempre um literal como conclusão. Um argumento pode ser derivado de crenças (tendo assim o conjunto de subargumentos vazio) ou derivado de uma regra (onde cada subargumento tem em sua conclusão uma das premissas da regra). A deliberação dos argumentos aceitos é feita através de uma redução do *framework*

estruturado para uma forma abstrata equivalente, a partir da qual é aplicada uma semântica de aceitabilidade e os argumentos estruturados recebem a valoração (aceito, rejeitado, indecidível) de seu equivalente em forma abstrata.

### 3. Modelo BBGP-argumentativo

[Morveli-Espinoza 2018] propõe uma formalização baseada em argumentação para o modelo de agentes BBGP. A argumentação é utilizada para sustentar a passagem dos objetivos de seu estado inicial (ativo) ao final (executivo).

Os objetivos ativos, perseguitáveis, escolhidos e executáveis são representados como átomos *ground* (cujo predicado possui todos os termos como constantes). Antes de um objetivo estar ativo ele é representado como um átomo não-*grounded* (possui variáveis como termos) e é chamado de objetivo latente.

O agente possui um conjunto de regras  $R = \langle R_{pd}, R_{at}, R_{av}, R_{de}, R_{ch} \rangle$  onde cada subconjunto é mutualmente disjuntos e contém as regras padrão, de ativação, de avaliação, de deliberação e de checagem, respectivamente. O subconjunto  $R_{pd}$  é aplicado em todos os estágios, já os demais subconjuntos são aplicados em seus respectivos estágios. Os subconjuntos de regras  $R_{pd}, R_{at}$  e  $R_{av}$  podem conter regras estritas e derrotáveis. As regras padrão não possuem objetivos ou literais sobre objetivos em suas premissas ou conclusão. As regras de ativação e avaliação possuem necessariamente em sua conclusão um objetivo ou negação de um objetivo, respectivamente.

Os três tipos de regras apresentados (padrão, ativação e avaliação) são projetados para serem inseridas pelo programador do agente, de maneira que são dependentes do domínio da aplicação. Já as regras de deliberação e de checagem são axiomas independentes do domínio, sendo definidas a seguir:

$$\begin{aligned} r_{de}^1 &: \neg \text{has\_incompatibility}(g) \rightarrow \text{chosen}(g) \\ r_{de}^2 &: \text{most\_valuable}(g) \rightarrow \text{chosen}(g) \\ r_{ch} &: \text{has\_plans\_for}(g) \wedge \text{satisfied\_context}(g) \rightarrow \text{executive}(g) \end{aligned}$$

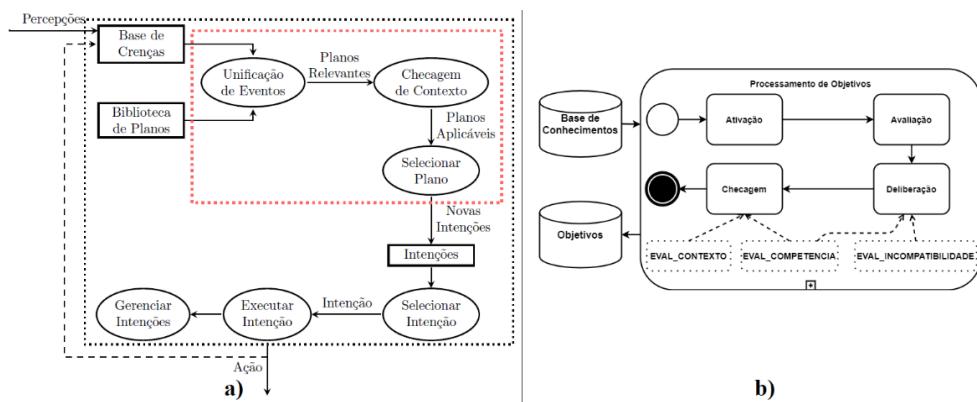
Com isso define-se  $R_{de} = (r_{de}^1, r_{de}^2)$  e  $R_{ch} = (r_{ch})$ . As regras de ativação, avaliação, deliberação e checagem são chamadas de forma genérica de **regras de estágio**.

O agente também possui um conjunto de planos, onde cada plano é definido da seguinte forma:  $g : PC \leftarrow PB$  codificando o corpo do plano ( $PB$ ) para atingir um objetivo  $g$  quando o contexto ( $PC$ ) é satisfeito.

Um agente baseado em BBGP é um tupla  $\langle F, S, D, REC_{res}, O, CP \rangle$ , onde  $F$  é o conjunto de crenças do agente,  $S = R'_{pd} \cup R'_{at} \cup R'_{av} \cup R_{de} \cup R_{ch}$  é o conjunto de regras estritas,  $D = R''_{pd} \cup R''_{at} \cup R''_{av}$  é o conjunto de regras derrotáveis,  $REC_{res}$  é o resumo dos recursos disponíveis ao agente,  $O$  é o conjunto de objetivos e  $CP$  é o conjunto de planos. Note que os conjuntos de regras são mutuamente disjuntos, ou seja, uma regra não pode ser estrita e derrotável ao mesmo tempo.

O processo argumentativo é responsável pelo raciocínio e justificação do processamento de objetivos e pode ser decomposto em quatro etapas: I) construção dos argumentos, II) determinação dos conflitos entre argumentos, III) avaliação da aceitabilidade dos argumentos, e IV) definição das conclusões justificadas.

Os critérios que definem o status de um argumento são estabelecidos pela semântica de aceitabilidade, senso sugerida a “semântica preferida”, cujos critérios são: seja  $E$ , e  $E'$  extensões,  $A$  o conjunto de todos argumentos e  $\rightarrow$  o conjunto de ataques, I)  $\nexists a, b \in E \mid (a, b) \in \rightarrow$ , II)  $\forall a \in E, \forall b \in A, \exists c \in E \mid se (b, a) \in \rightarrow \text{ então } (c, b) \in \rightarrow$  e III)  $\nexists E' \mid E \subset E'$ . Uma vez que pode haver múltiplas extensões segundo a semântica utilizada, é preciso escolher uma única extensão. Para isso existe uma função que escolhe a melhor extensão, os critérios usados podem variar dependendo do domínio do problema.



**Figura 1 a) Ciclo de execução do agente. b) Processamento de Objetivos.**

## **4. Arquitetura do Agente BBGP-argumentativo**

O funcionamento macroscópico do um agente BDI pode ser observado na Figura 1 a). O ciclo de execução para agentes BDI ou BBGP-argumentativo são muito similares, sendo sua principal diferença a maneira que a decisão dos objetivos executivos/intenções ocorre. Observe as funções destacadas em vermelho. Tais funções são realizadas pelo processamento de objetivos. Vale ressaltar que o ciclo é inspirado especificamente em Jason, que é um interpretador de uma versão estendida da linguagem AgentSpeak.

A Figura 1 b) apresenta o esquema do Processamento de Objetivos, onde temos os quatro estágios (retângulos cheios), as funções auxiliares (retângulos pontilhados) e as informações internas do agente que são necessárias (forma de base de dados). Embora os estados dos objetivos apareçam de maneira sequencial é possível que um objetivo retroceda para um estado anterior caso seu suporte sofra alterações. A “Base de Conhecimentos” compreende as crenças, regras (padrão e de estágio), conjunto de recursos e de planos do agente. A Base de “Objetivos” compreende todos os objetivos (não-latentes) que o agente possui, o que inclui os objetivos executivos que compõem as intenções do agente, e é atualizada a cada estágio do processamento.

Os quatro estágios do processamento de objetivos ocorrem sequencialmente, uma vez que o estado dos objetivos tem papel integral no resultado de cada estágio. Cada estágio tem um papel distinto: a **ativação** cria novos objetivos; a **avaliação** verifica se há condições contrárias à execução dos *objetivos ativos*; a **deliberação** encontra os planos conhecidos para atender cada *objetivo persegível* e escolhe dentre os eventuais conflitos quais objetivos devem ser realizados no momento; a **checagem** verifica as precondições para realização do plano de cada *objetivo escolhido*. Os objetivos que com precondições de seu plano satisfeitas tornam-se *objetivos executivos*.

Todo estágio do processamento segue o seguinte subprocesso: **I**) criar os argumentos possíveis a partir da base de conhecimentos; **II**) aplicar a semântica de

argumentação “preferida”, gerando múltiplas extensões; **III**) dentre as extensões geradas, selecionar uma única extensão; **IV**) a extensão selecionada é processada e os objetivos relevantes são criados (ativação) ou modificados (demais estágios).

## 5. Implementação do Modelo

A seguir iremos abordar: a) a adaptação necessária nas regras definidas no modelo BBGP-argumentativo, b) algoritmo responsável por gerar o conjunto de ataques dos objetivos.

### 5.1 Adaptação das Regras

Algumas regras no modelo BBGP-argumentativo utilizam lógica de segunda ordem. Assim foi necessária a conversão das regras de avaliação ( $R_{de}$ ) e checagem ( $R_{ch}$ ) para a primeira ordem, uma vez que os termos empregados foram definidos de forma que  $g$  seja um objetivo, entretanto objetivos também são predicados.

A conversão consistiu em adicionar um novo termo nos predicados de objetivos que age como um identificador, definido no momento da ativação do objetivo, assim os predicados como  $chosen(g)$  e  $executive(g)$  passam a referenciar não o predicado do objetivo mas sim tal identificador. A conversão ocorre nas regras dependentes de domínio (ativação e avaliação), sendo exemplificada a seguir:

$$take\_hospital(X) \mapsto take\_hospital(gHolder, X)$$

O efeito de tal conversão no processo de raciocínio é exemplificado a seguir, considere “g002” como uma constante atribuída ao objetivo convertido:

$$\begin{aligned} \text{Antes: } & most\_valuable(take_{hospital}(X)) \rightarrow chosen(take_{hospital}(X)) \\ \text{Convertido: } & most\_valuable(g002) \rightarrow chosen(g002) \end{aligned}$$

---

#### Algoritmo 1 Compatible Goals

```

1. dungTheory = new abstract argumentation theory
2. for Each goal with state == "Chosen" OR "Executable"
3.   argument = new argument that tracks goal + goal.preferredPlan
4.   dungTheory.addArgument(argument)
5. for Every goal with state == "Pursuable"
6.   for Every plan for goal
7.     argument = new argument that tracks goal + plan
8.     dungTheory.addArgument(argument)
9. for every pair of arguments arg1, arg2
10. if arg1 has conflict with arg2
11.   if one of the goals state is "Chosen" OR "Executable"
12.     preferredArg = argument for the chosen or executable goal
13.     otherArg = argument for the pursuable goal
14.     dungTheory.addAttack(preferredArg, otherArg)
15.   else if both goals state are "Pursuable"
16.     dungTheory.addAttack(preferredArg, otherArg)
17. return = new list of predicates
18. compatibleGoals = selectExtension(dungTheory.CONFLICTFREE())
19. for each argument from compatibleGoals
20.   return.add("!not_has_incompatibility(" + goal.id + ")")

```

---

### 5.2 Algoritmo de Geração de Ataques entre Objetivos

No modelo BBGP-argumentativo são definidas funções auxiliares que o agente emprega para gerar crenças necessárias ao processamento de objetivos. Duas destas funções, responsáveis por avaliar incompatibilidade e avaliar a preferência entre objetivos, foram substituídas por um algoritmo baseado no método apresentado em [Morveli-Espinoza et al. 2019], sendo empregado no estágio de deliberação. Nele é utilizado argumentação

abstrata para modelar os conflitos entre planos e então encontrar os objetivos compatíveis.

O Algoritmo 1 toma como entrada o conjunto de objetivos do agente e os respectivos planos aplicáveis e retorna um conjunto de crenças listando os objetivos no estado perseguiável que são compatíveis com os outros objetivos cujo estado seja escolhido ou executivo. A principal diferença do algoritmo apresentado com o método de [Morveli-Espinoza et al. 2019], é que o algoritmo faz uso das estruturas internas do agente para criar o framework de argumentação abstrata diretamente.

## 6. Conclusões

Mostramos como foi adaptado o modelo BBGP-argumentativo para que ficasse em conformidade com a lógica de primeira ordem. Apresentamos também o algoritmo para o estágio de deliberação, que gera diretamente o framework de argumentação abstrato.

O simulador mostra que a arquitetura proposta é suficiente para executar o processamento de objetivos como definida no modelo de agentes BBGP-argumentativo de Morveli-Espinoza. Com o simulador também é possível avaliar em um ambiente controlado o funcionamento do modelo. Como trabalhos futuros salientamos a análise do comportamento da arquitetura BBGP, em especial quando um objetivo perde seu suporte, e explorar as explicações que podem ser extraídas do agente sobre seu raciocínio.

## Referências

- Bordini, R. H., Hübner, J. F. e Vieira, R. (2005) “Jason and the Golden Fleece of agent-oriented programming”. Em: Bordini, R. H., Dastani, M., Dix, J., e El Fallah Seghrouchni, A., eds., Multi-Agent Programming: Languages, Platforms and Applications. Springer-Verlag. chapter 1, 3-37.
- Castelfranchi, C. e Paglieri, F. (2007) “The role of beliefs in goal dynamics: prolegomena to a constructive theory of intentions”, Em: Synthese, v. 155, n. 2, p. 237–263.
- Dung, P. M. (1995) “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games”. Em: Artificial Intelligence, v. 77, n. 2, p. 321–357.
- Morveli-Espinoza, M. (2018) “Cálculo da Força de Argumentos Retóricos e Sua Utilização em Diálogos de Negociação Persuasiva em Sistemas Multiagentes”. PhD thesis, Universidade Tecnológica Federal do Paraná, Curitiba, Brasil. Disponível em: < <http://repositorio.utfpr.edu.br/jspui/handle/1/3849> >.
- Morveli-Espinoza, M., Nieves, J. C., Possebom, A., Puyol-Gruart, J. e Tacla, C. A (2019) “An argumentation-based approach for identifying and dealing with incompatibilities among procedural goals”. Em: International Journal of Approximate Reasoning, v. 105, p. 1–26. Elsevier Inc.
- Prakken, H. (2010) “An abstract framework for argumentation with structured arguments”. Em: Argument & Computation, v. 1, n. 2, p. 93–124.
- Rao, A. e Georgeff, M. (1995) “BDI Agents: From Theory to Practice”. Em: Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95).
- Sansores, C. e Pavón, J. (2008) “A Motivation-Based Self-organization Approach”. Em: International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008). p.259–268, 2009. Berlin, Heidelberg: Springer Berlin Heidelberg.