

Agregação de Julgamento usando Argumentação Abstrata para JaCaMo: Especificação e Estudo de Caso

Elton M. Sato¹, Cesar A. Tacla¹, Mariela Morveli-Espinoza¹

¹Coordenação do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI) – Universidade Tecnológica Federal do Paraná (UTFPR) Curitiba – PR – Brasil
elton.m.sato@gmail.com, tacla@utfpr.edu.br, morveli.espinoza@gmail.com

Abstract. The problem of judgement aggregation has the objective of unifying the judgement of several individuals in a single set to reach an agreement within the group. Using JaCaMo, a three-layered tool: environment, multi-agent system, and organization, this work describes a specification of the judgment aggregation model of Caminada (2009) for the framework. These layers allows us to model more complex multi-agent systems, with stigmergic agents, or with a hierarchy structure. Also, a simple case study was successfully developed to validate the specification detailed.

Resumo. O problema de Agregação de Julgamento tem como objetivo unir o julgamento de vários indivíduos em um único conjunto para se chegar a um acordo do grupo. Utilizando a plataforma JaCaMo, uma ferramenta de três camadas, ambiente, sistema multiagente, e organizacional, este trabalho descreve a especificação do modelo de agregação de julgamento de Caminada (2009) para a plataforma. Essas camadas permitem a modelagem de sistemas multiagente mais complexos, como agentes estigméricos, ou com uma estrutura hierárquica. Também foi realizado com sucesso um simples estudo de caso para validar a especificação descrita.

1. Introdução

Segundo o Teorema de Impossibilidade de Arrow (1950), os problemas de tomada de decisão em grupo são difíceis, ou podem chegar a ser impossíveis de se obter um resultado com que todos possam concordar sem o uso de uma arbitrariedade. Um problema que demonstra esse teorema, e deu origem a toda a área de Agregação de Julgamento [Grossi 2014] é exemplificado pela Tabela 1.

Tabela 1. Paradoxo Doutrinal

	Está ensolarado?	Está quente?	Vamos para a praia.
Indivíduo 1	Sim	Sim	Sim
Indivíduo 2	Sim	Não	Não
Indivíduo 3	Não	Sim	Não
Decisão da Maioria	Sim	Sim	Não

A Tabela 1 mostra 3 pessoas decidindo se irão à praia por decisão da maioria. Todas as três pessoas concordam com a seguinte fórmula: “ $p \wedge q \rightarrow r$ ”, sendo p =“Está ensolarado”, q =“Está quente”, e r =“Vamos para a praia”. O problema é que apesar de que a maioria concorde com as premissas “ p ” e “ q ”, a maioria discorda da conclusão “ r ”. List e Pettit (2004) acreditam que este paradoxo mostra como não existe uma maneira simples de agregar opiniões individuais, e assim chegar a uma única e coerente decisão do grupo.



O trabalho Agregação de Julgamento usando Argumentação Abstrata para JaCaMo: Especificação e Estudo de Caso de [Elton Masaharu Sato, Cesar Augusto Tacla, Mariela Morveli-Espinoza](#) está licenciado com uma Licença [Creative Commons - Atribuição-NãoComercial-Compartilhável 4.0 Internacional](http://creativecommons.org/licenses/by-nc-sa/4.0/). <http://creativecommons.org/licenses/by-nc-sa/4.0/>

A teoria da argumentação, o estudo das regras de inferência e lógica em debates, diálogos, conversas e persuasão [Frans & Rob 2004], é um dos campos de estudo tentando resolver o problema de tomada de decisão. A Figura 1, mostra como problemas de decisão de grupo podem ocorrer com apenas dois indivíduos e dois argumentos.

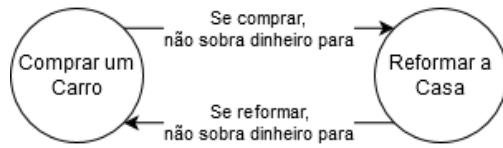


Figura 1. Problema de Decisão de Compra.

Na Figura 1, há duas opções presentes, “Comprar um Carro”, e “Reformar a Casa”, porém as duas opções se atacam, pois como só há dinheiro para realizar uma delas, não sobra dinheiro para realizar a outra. Somente com essas duas opções, não é possível escolher uma vencedora sem utilizar um método arbitrário, ou se utilizar de mais informações para defender as escolhas, como “Não saber dirigir”, tornando a opção “Comprar um Carro” menos interessante.

Um estudo de Caminada (2009) realizou uma análise de problemas de agregação de julgamento. O autor abordou a questão de como combinar as avaliações de cada indivíduo em um coletivo se utilizando de argumentação abstrata.

Para a modelagem e testes de agregação de julgamento, modelos de inteligência artificial de sistemas multiagente cognitivos têm sido utilizados com sucesso (por exemplo em [Rahwan, 2010] [Thimm, 2014]). No entanto, os trabalhos analisados não apresentam a capacidade de representação de ambiente virtual, e segundo Weyns et al. (2005), o ambiente tem um papel fundamental no desenvolvimento de sistemas multiagente. Os trabalhos analisados também não apresentam uma estrutura que organizasse os agentes, portanto não seria possível o desenvolvimento de uma das duas principais abordagens de modelagem multiagente, a modelagem por organização [Lemaitre, 1998].

Para satisfazer essas necessidades de modelagem, foi escolhida a plataforma JaCaMo, um *framework* para o desenvolvimento de sistemas multiagente integrando importantes resultados e tecnologias de três direções de pesquisa, orientado a agentes (Jason), organizados socialmente (Moise), e situados em ambientes onde há artefatos concretos e abstratos (CArtAgO) [Boissier et al, 2011][Bordini e Hubner, 2006] [Hannoun, Sichman, Sayettat, 2000].

Baseando-se nos modelos de agregação de julgamento de Caminada (2009) para argumentação abstrata, e na plataforma JaCaMo, este trabalho pretende responder à pergunta de como seria possível especificar um modelo de agregação de julgamento com argumentos abstratos para um sistema multiagente com ambiente virtual e organização de agentes. A avaliação do mapeamento da especificação será realizada por meio de um estudo de caso.

A motivação de se usar JaCaMo está nas ferramentas adicionais que a plataforma proporciona. Os artefatos CartAgO permitem a descrição de uma camada de ambiente para a plataforma, pois ele permite o desenvolvimento de sistemas multiagente completamente reativos, assim como sistemas multiagente estigméricos, o qual usam o ambiente como meio de coordenação [Duan, 2012]. A posição local de um agente no ambiente também pode influenciar a sua posição em um julgamento, por exemplo estar perto de um objetivo, ou longe de outros agentes podem influenciar nas suas escolhas.

As funções do Moise permitem a modelagem de uma camada de organização através de três principais conceitos: os papéis, os vínculos organizacionais, e os grupos. Possibilitando assim, a modelagem de sistemas de agregação com divisão hierárquica, em que o grupo ou papel a qual um agente pertence influencia o peso, posição, e ou missão de um agente, alterando assim, a saída da agregação de julgamento.

Este artigo foi estruturado da seguinte forma: Na seção 2 são apresentados os conceitos fundamentais para o entendimento da especificação. Seção 3 descreve o Mapeamento realizado para a especificação do modelo de agregação de julgamento para a plataforma JaCaMo. Na seção 4 é apresentado um estudo de caso, avaliando o mapeamento especificado. E na seção 5 será mostrada a conclusão do trabalho, e as sugestões de trabalho futuro.

2. Fundamentação Teórica

Nesta seção são apresentados os estudos de Argumentação e de Agregação de Julgamento. Apresenta-se também, a plataforma JaCaMo.

2.1. Argumentação

A argumentação é um estudo interdisciplinar que analisa como conclusões são atingidas através do raciocínio lógico, sendo utilizados o debate, o diálogo, conversação e a persuasão. Estuda também as regras de inferência, lógica, e procedural em ambos os ambientes reais e virtuais [Frans e Rob 2004].

Um *Framework* de Argumentação Abstrata (FA) é um conjunto de argumentos chamados de abstratos, representando dados ou proposições, e uma relação binária representando o conflito entre eles [Dung 1995]. Neste trabalho, as opções de decisão serão representadas por argumentos abstratos. Esses dois elementos formam o FA, e podem ser representados por um grafo direcionado (Figura 2), da seguinte forma:

- Os argumentos são representados pelos círculos.
- As relações de ataque são representados pelas setas.

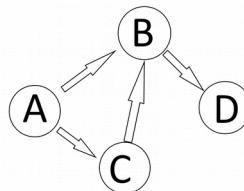


Figura 2. Exemplo de um *Framework* de Argumentação.

A partir desse grafo de argumentos e de ataques, podem ser extraídas características, como:

- **Defesa:** um argumento é defendido quando os argumentos que o atacam são atacados por outros argumentos. Normalmente isso requer que o argumento que defende e o defendido sejam ambos parte do mesmo conjunto, por exemplo, o argumento (A) e (C) defendem o argumento (D).
- **Livre de Conflito:** qualquer conjunto de argumentos que não possua relações de ataque entre si, por exemplo, os conjuntos (A,D), (C,D), (A), (B), (C), (D), e (\emptyset).

- **Admissível:** qualquer conjunto de argumentos livre de conflitos que são defendidos por argumentos do próprio conjunto, por exemplo, os conjuntos (A), (A, D).

Aceito é o termo utilizado para demonstrar que uma opção, ou argumento, foi escolhida. Extensões de semânticas de argumentação são regras para escolher argumentos aceitos, e tem como objetivo separar os conjuntos que possuem características em comum. A partir do grafo de argumentos e ataques, o intuito é determinar qual(is) conclusão(ões) são aceitas por um indivíduo ou grupo. Para tal, algumas das semânticas propostas por Dung (1995) são:

- **Completa:** qualquer conjunto admissível contendo todos os argumentos que ele defende.
- **Preferida:** qualquer conjunto contendo a quantidade maximal de argumentos entre os conjuntos admissíveis.
- **Estável:** qualquer conjunto livre de conflitos que ataca todos os outros argumentos que não fazem parte do conjunto.
- **Grounded:** os menores conjuntos possíveis entre os conjuntos que são completos.

Com os argumentos, ataques, e a semântica de argumentação, é possível analisar quais argumentos foram aceitos e quais foram recusados, além de argumentos que podem não ter sido aceitos e nem recusados. Para evitar que seja necessário reaplicar uma semântica toda vez que for verificar a condição de um argumento, é realizado o processo de “Rotulagem”, aplicando rótulos a cada argumento da FA. Esses rótulos podem ser *IN* (argumento aceito), *OUT* (argumento recusado), ou *UNDEC* (argumento nem aceito nem recusado, sem posição). É a partir da rotulagem que se representa qual é o conjunto de argumentos aceitos via a semântica escolhida.

2.2. Agregação de Julgamento

Segundo Caminada (2009), a agregação de julgamento é um nome dado ao grupo de problemas de preferências na agregação, normalmente devido às diferentes preferências possuírem diferentes resultados. O modelo de Caminada (2009) se utilizou de argumentação abstrata, e para se aplicar a agregação de julgamento, deve-se seguir os seguintes conceitos de condições:

- **Domínio Universal:** esta condição diz que o conjunto Domínio do FA consiste de todos os possíveis rotulagens. Essa condição diz respeito à entrada do sistema.
- **Racionalidade Coletiva:** a semântica da FA deve ser admissível, livre de conflito, e completa. Também chamado de Conjunto de Julgamento [Grossi 2014]. Essa condição diz respeito à saída do sistema.
- **Anonimato:** uma avaliação pode ser qualquer tipo de informação relacionada que um indivíduo possui. Esta condição requer que cada avaliação enviada por um indivíduo seja indistinguível de avaliações enviadas por outros indivíduos. Esta condição normalmente não é quebrada, pois não é comum guardar uma identificação de qual indivíduo enviou qual avaliação.
- **Independência:** caso dois ou mais indivíduos aceitem ou recusem um argumento, então o Julgamento Agregado deles também deve aceitar ou recusar

esse argumento ou fórmula. Ou seja, o Julgamento Agregado de indivíduos que concordam não pode ser alterado por indivíduos que não fazem parte da Agregação de Julgamento.

Um exemplo de problema de preferência de agregação é o da Figura 1. Caso seja utilizada a semântica preferida, há três conjuntos que podem ser aceitos, {Comprar um Carro}, {Reformar a Casa}, e também o conjunto vazio. Caso dois indivíduos votem em diferentes conjuntos, haverá o problema da preferência, caindo assim no Teorema da Impossibilidade.

Segundo Caminada (2009), seria possível utilizar semânticas que gerem somente uma extensão, como a semântica Grounded, mas afirma que isso trivializaria a agregação de julgamento, já que não seria possível o desacordo entre os indivíduos, e portanto, não haveria a necessidade de agregação de julgamento.

O trabalho de Caminada (2009) foca-se na agregação de julgamento, e parte do princípio de que toda informação relevante sobre a decisão dos indivíduos estão contidos nos agentes. Dessa forma, os agentes do sistema trocam argumentos que julgam necessários, e um agente que trata da agregação aplica uma semântica de argumentação, chegando então a uma rotulagem que mostra quais argumentos foram aceitos, recusados, ou indefinidos.

2.3. JaCaMo

JaCaMo é um *framework*, segundo Boissier et al. (2013), consistindo da combinação de três outros projetos, sendo esses ilustrados pela Figura 3:

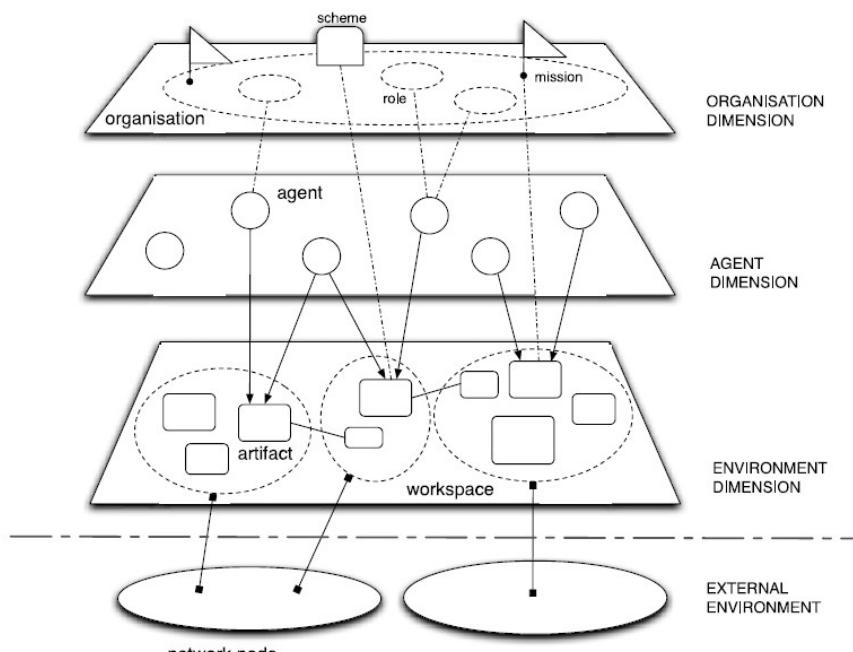


Figura 3. Visão Geral do JaCaMo. (Extraído de Boissier et al, 2013)

Jason

Desenvolvido por Bordini e Hubner (2006), é uma plataforma de desenvolvimento de sistemas multiagente, funcionando como um interpretador para uma versão extendida do

AgentSpeak. Na Figura 3, o Jason é a camada de agentes (*Agent Dimension*), ao qual trabalha nas questões de comunicação dos agentes. Os agentes Jason seguem o modelo BDI(*Belief-Desire-Intention*) para a sua lógica interna.

CArtAgO

Acrônimo de “*Common ARTifact infrastructure for Agents Open environments*”, é um framework de uso genérico para a simulação de ambientes virtuais. Baseado no metamodelo A&A de Omicini et al. (2008), e representado na Figura 3 pela camada de ambiente (*Environment Dimension*), ele utiliza os chamados artefatos para representar objetos ou abstrações do ambiente virtual e pode ser separado em espaços de trabalho (*workspaces*), além de poder serem operados pelos agentes da camada de agentes.

Moise

Moise é um modelo organizacional para sistemas multiagente baseado nas noções de papéis, grupos, e missões. Desenvolvido por Hannoun, Sichman, e Sayettat (2000), e representado na Figura 3 pela camada de organização (*Organization Dimension*), tem como objetivo criar uma estrutura que agrupa e separa os agentes pelas suas características sociais, tendo seus objetivos alinhados com o papel que cada agente atua dentro de seu grupo.

3. Mapeamento

Para realizar a especificação, precisa-se mapear e traduzir todos os termos de agregação de julgamento (Indivíduos, FA, Semântica) para a plataforma JaCaMo, bem como protocolar os procedimentos necessários (Comunicação e Agregação dos Argumentos).

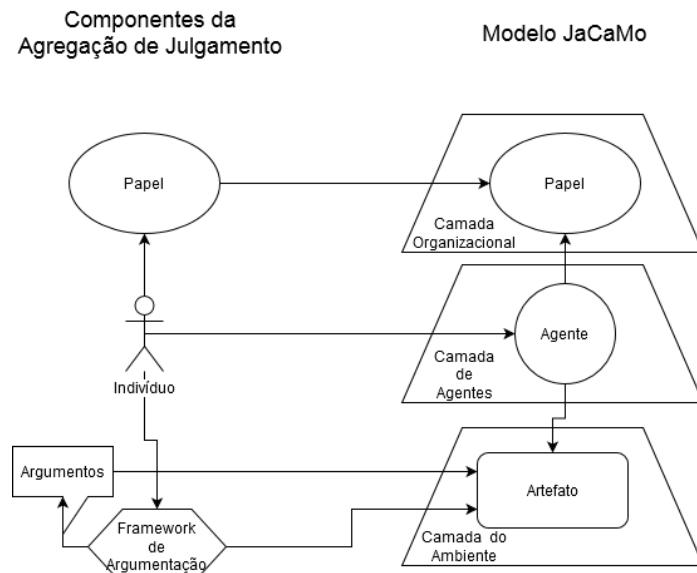


Figura 4. Mapeamento dos Modelos.

3.1. Mapeamento dos Indivíduos

Os indivíduos da agregação de julgamento são as entidades que possuem estados mentais, assim como mostra a Figura 4, são mapeados para os agentes do JaCaMo. Ambos são conceitualmente similares, sendo representações em software de membros do sistema.

- Visão do Mundo: crenças do agente sobre si, o ambiente, os outros agentes e, também, crenças relativas a domínios específicos e crenças gerais.
- Interação e Comunicação: contém as formas que o indivíduo tem como interagir com outros agentes, ou outros objetos do sistema. Os protocolos utilizados para a comunicação são detalhados no final da seção 3. Ambos as interações partindo dos agentes e os protocolos de comunicação são planos de cada agente.
- Objetivo: o que o indivíduo quer atingir, no caso da agregação de julgamento, é o conjunto de argumentos que o indivíduo quer que vença.

3.2. Mapeamento do Framework de Argumentação

Os argumentos são os componentes principais para a criação do FA, conceitualmente, os argumentos representam as proposições e dados que agente sabe, logo, fariam parte das crenças do agente. Porém, como os argumentos são manipulados pelo FA, eles são mapeados para uma classe de artefato do JaCaMo. O FA pode ser representado por grafos direcionados [Besnard & Hunter, 2001] em CArtAgO, utilizando a linguagem Java.

Os argumentos são mapeados como uma classe Java, possuindo as seguintes características:

- “ID”: identificador único para um argumento.
- “Nome”: o nome do argumento pode ser o próprio argumento em sua forma textual, como por exemplo “Está Ensolarado”, utilizado para facilitar a identificação de um argumento para o usuário.
- Vetor de “Ataques”: representa todos os ataques que partem deste argumento. Cada Inteiro é o ID do argumento que ele ataca.
- Vetor de “Atacado”: guarda o valor de ID de todos os argumentos que estão atacando este argumento.
- As demais funções que permitem ler e escrever as variáveis listadas.

O FA também tem a sua própria classe, com as seguintes características:

- Vetor de Argumentos “Lista de Argumentos”: um vetor com todos os argumentos recebidos.
- Matriz de Argumentos “Listas de Argumentos Aceitáveis”: uma matriz com todos os argumentos que estão rotulados como *IN*, ou seja, os argumentos que foram aceitos. Cada vetor da matriz representa um conjunto de rotulagem aceita. No caso de situações em que há somente um conjunto resultado possível, pode se utilizar um vetor em vez de uma matriz.
- Argumento “Adicionado”: quando o agente passa os valores para o FA, ele deve criar um Argumento que possui as características passadas.
- Vetores de Argumentos de Apoio: servem para auxiliar as funções de adição ou remoção de argumentos.
- As funções necessárias são a de criação, adição e remoção de argumento no FA. Bem como as funções que rotulam os argumentos com a semântica desejada.
- Caso haja um agente centralizador fazendo a agregação de julgamento, a informação do FA é pública. Para que os outros indivíduos possam ver o FA,

eles devem usar a função de Focar do CArtAgO no artefato que possui o FA.

3.3. Semântica de Argumentação

O raciocinador semântico é o mecanismo que faz as inferências lógicas a partir de um conjunto de argumentos e relações. Conceitualmente, o raciocinador está associado com as crenças e objetivos do agente, e portanto faria parte da camada de agentes. Porém, para a aplicação do raciocinador semântico, houve uma restrição devido às linguagens empregadas, e a semântica foi mapeada para uma função de um artefato do CartAgO.

Utiliza-se a variável “Listas de Argumentos Aceitáveis” para salvar o conjunto escolhido, o pseudocódigo para determinação da extensão da semântica preferida é apresentado em seguida:

1. Enquanto *REPEAT* for verdade:
2. Para cada argumento do FA
3. Se a quantidade de ataques sofridos é zero
4. Para cada argumento da lista de aceitáveis
5. Se o ID já está na lista, então já foi adicionado à lista de aceitáveis
6. Se ainda não foi adicionado, então coloca-o na lista de aceitáveis
7. Se a quantidade de argumentos aceitáveis não mudou, então *REPEAT* é falso
8. Senão atualiza a quantidade de argumentos aceitáveis
9. Para cada argumento aceitável
10. Para cada ataque que a lista de argumentos aceitáveis realiza
11. Remove os argumentos atacados

3.4. Protocolo de Comunicação

O método mais simples para a agregação de julgamento, é existindo um agente centralizador fazendo a agregação de todos os agentes que entram no sistema.

Agregador (único):

- Quando entra no sistema, envia um *broadcast* anunciando ser o Agregador.
- Se receber uma mensagem Procurando o Agregador, responde somente para quem enviou uma mensagem anunciando ser o Agregador.
- Caso receba uma mensagem de envio de Argumentos, atualiza o seu FA.

Indivíduo (múltiplos):

- Quando entra no sistema, envia um *broadcast* procurando o Agregador.
- Se receber uma mensagem de outro agente anunciando ser o Agregador, guarda a informação de quem é o Agregador, e envia os seus argumentos. A informação de quem é o Agregador é guardada, pois caso haja a necessidade de enviar mais argumentos, o Indivíduo saberá para quem mandar os argumentos.

Como parte-se do princípio de que os agentes já possuem uma posição e argumentos antes de se iniciar os diálogos, não são necessárias trocas de mensagem para os indivíduos reavaliarem suas posições e argumentos.

4. Estudo de Caso

Para testar o mapeamento descrito, foi feito um pequeno estudo de caso adaptando e estendendo o problema da Figura 1, do problema de decisão de compra. Neste caso, são dois advogados tentando convencer um mesmo juiz, cada um defendendo a inocência de seu respectivo cliente.

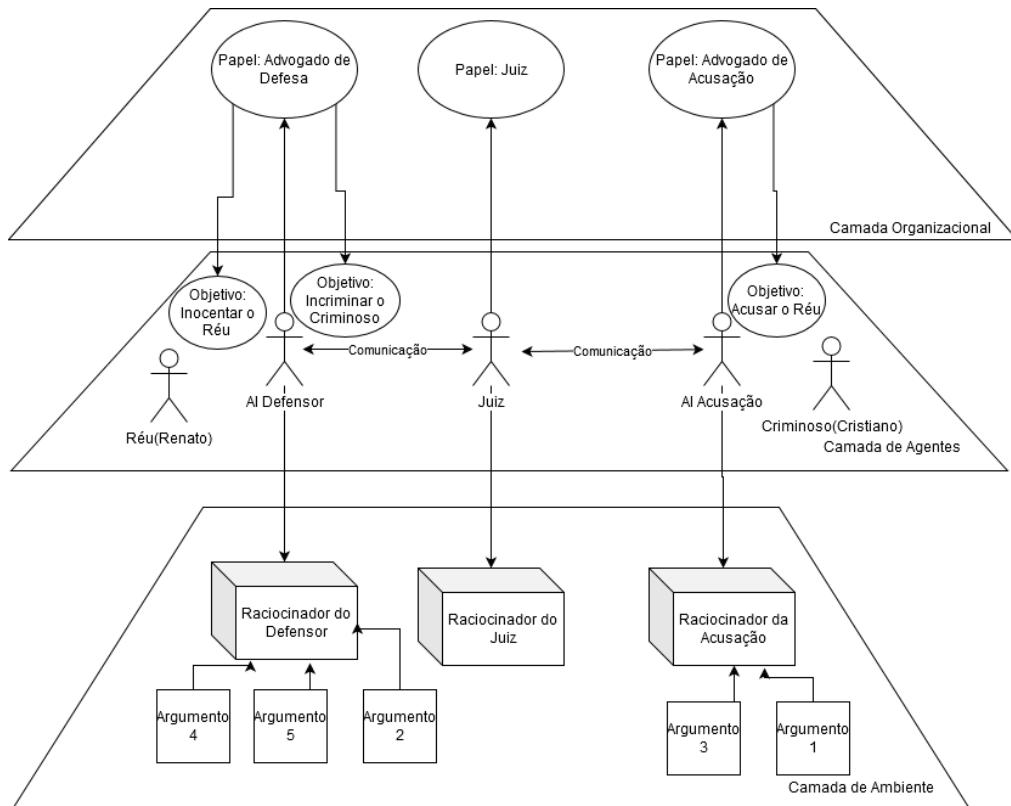


Figura 5. Visão Geral do Estudo de Caso.

A Figura 5 representa a visão geral do nosso estudo de caso, fazendo um paralelo com a Figura 3 que mostrava a visão geral do JaCaMo. Começando por cima, temos 3 papéis, o Advogado de Defesa, o Juiz, e o Advogado de Acusação. Sendo que os agentes (no caso somente um para cada papel) com o papel Advogado de Defesa tem como objetivo Inocentar o Réu e Incriminar o Criminoso, enquanto o Advogado de Acusação tem o objetivo de Acusar o Réu, os objetivos sendo parte do modelo BDI dos agentes.

Na segunda camada, aparecem os agentes que representam o Advogado de Defesa, o Juiz, e o Advogado de Acusação. Além de dois outros agentes que ganharam nomes por questões de abstração, RÉu (RENATO), e o CRIMINOSO (CRISTIANO). Nessa camada, está representado também a capacidade de comunicação entre os agentes.

Na terceira camada, estão os raciocinadores de cada um dos agentes principais, sendo o raciocinador o artefato que contém ambos o FA, e as funções que aplicam a semântica. Nesta camada também contém os seguintes argumentos:

- Argumento 1: Renato é Culpado.
- Argumento 2: Cristiano é Culpado.

- Argumento 3: Faca do Renato na Cena do Crime. Este argumento ataca o Argumento 2, pois inocenta o Cristiano.
- Argumento 4: Pistola do Cristiano na Cena do Crime. Este argumento ataca o Argumento 1, pois inocenta o Renato.
- Argumento 5: Autópsia Indica Morte por Arma de Fogo. Este argumento ataca o Argumento 3, pois a causa da morte foi por Arma de Fogo.

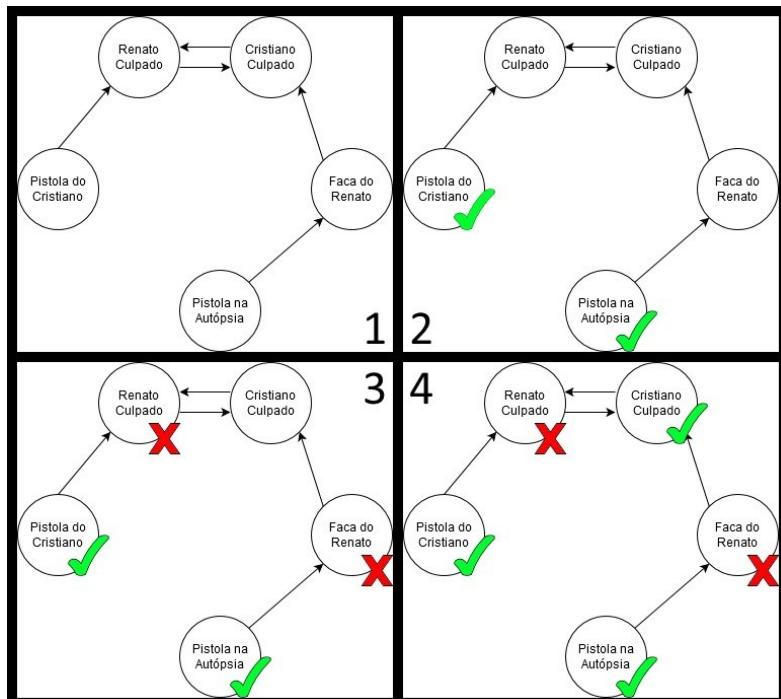


Figura 6. Sequência de Seleção dos Argumentos.

Ao iniciar o programa, ambos os advogados pegam seus argumentos e repassam para o Juiz. A Figura 6 mostra o passo a passo da semântica do Juiz para chegar a um veredito. Ele se utiliza da extensão semântica preferida, começando a selecionar os argumentos que não são atacados, e removendo os argumentos que são atacados por eles. Desconsiderando os ataques realizados por argumentos removidos, repete-se o processo até que não se adicione mais argumentos ao conjunto.

```

MAS Console - court
common [juiz] joinned workspace provas
Cartago [juiz] focusing on artifact raciocinio (at workspace provas) using namespace default
juiz [juiz] 2CristianoCulpado0
acusacao [juiz] 5Pistola na Autopsia3
defensor [juiz] 4Pistola do Cristiano1
[juiz] 1RenatoCulpado0
[juiz] 3FacaDoRenato2
[juiz] Argumento Válido Pistola na Autopsia
[juiz] Julgado
[juiz] Argumento Válido Pistola do Cristiano
[juiz] Argumento Válido CristianoCulpado

```

Figura 7. Resultado do Estudo de Caso.

A Figura 7 nos mostra o resultado prático da teoria. Após receber todos os argumentos, o Juiz decide válidos os argumentos 2, 4, e 5, que correspondem à “Cristiano Culpado”, “Pistola do Cristiano na Cena do Crime”, e “Autópsia Indica Morte por Arma de Fogo”.

5. Conclusão

Os problemas de Agregação de Julgamento não são triviais de se resolver, porém é possível conseguir resultados partindo de algumas restrições ou arbitrariedades. O estudo de caso demonstrou os resultados esperados enquanto satisfazia todas as quatro condições descritas na seção 2.2 o Domínio Universal, pois todas as configurações de rotulagem são possíveis; a Racionalidade Coletiva, pois a semântica preferida engloba as extensões livre de conflito, admissível e completa; o Anonimato, pois não é guardada e nem utilizada a informação de quem enviou cada mensagem; e a Independência, pois todos os agentes estão de acordo com a rotulagem do agregador.

Podemos concluir que neste trabalho realizamos com sucesso a especificação de um modelo de agregação de julgamento para a plataforma JaCaMo, expandindo ainda mais o seu repertório de funções. Para a avaliação do estudo de caso, dois dos mapeamentos foram mais diretos, os indivíduos para os agentes, e os objetivos para os papéis com objetivos e missões. Os mapeamentos mais abstratos se tratam dos argumentos, FAs, e a semântica, que foram representados via artefatos do CArtAgO. A decisão do mapeamento do FA foi devido ao esforço de programação necessário para se representar e manipular um grafo direcionado em *AgentSpeak*, uma linguagem orientada à agentes, em relação à Java, uma linguagem de uso genérico baseado em classes e orientada a objetos. Da mesma forma, foram utilizados recursos da linguagem Java para se implementar os argumentos, que são classes do sistema, e a semântica, que foi implementada via funções do FA.

Para os trabalhos futuros, há três caminhos possíveis que envisionamos. O primeiro seria a descentralização da agregação, pois o agregador é o maior gargalo do sistema, já que ele se comunica com todos os outros agentes e também tem o trabalho de agragar os argumentos. O segundo seria utilizar os papéis para aumentar ou reduzir a credibilidade de um agente, influenciando os pesos de seus argumentos. E o terceiro caminho é um modelo mais completo de comunicação com a possibilidade de retrair argumentos, questionar outros agentes, e desafiar agentes a provar sua posição, assim como mostrado por Panisson et al. (2015).

Referências

- Boissier, O. Bordini, R. Hubner, J. Ricci, A. Santi, A. (2011) “Multi-agent oriented programming with JaCaMo”, *Science of Computer Programming* 78 747-761.
- Bordini, R. Hubner, J. (2006) “BDI Agent Programming in AgentSpeak Using Jason”, *International Workshop on Computational Logic in Multi-Agent Systems*, Lecture Notes in Computer Science, LNCS, volume 3900.
- Omicini, A. Ricci, A. Viroli, M. (2008) “Artifacts in the A&A meta-model for multi-agent systems”, *Autonomous Agents and Multi-Agent Systems* 17 (3)432–456.
- Hannoun, M. Boissier, O. Sichman, J. Sayettat, C. (2000) “MOISE: An Organizational Model for Multi-agent Systems”, *IBERAMIA 2000, SBIA 2000*. Lecture Notes in Computer Science, vol 1952.
- Arrow, K. (1950). ["A Difficulty in the Concept of Social Welfare"](#). *Journal of Political*

- Economy. **58** (4): 328–346. [doi:10.1086/256963](https://doi.org/10.1086/256963). JSTOR 1828886.
- List, C. Pettit, P. (2004) “[Aggregating Sets of Judgments: Two Impossibility Results Compared](#)”, *Synthese* 140 207–235.
- Rahwan, I. Tohmé, F. (2010) “Collective Argument Evaluation as Judgement Aggregation”, Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010).
- Thimm, M. (2014) “Strategic Argumentation in Multi-Agent Systems”. *Künstl Intell* 28, 159–168.
- Cardoso, C. Krausburg, T. Baségio, T. et al. (2018) “SMART-JaCaMo: an organization-based team for the multi-agent programming contest”. *Ann Math Artif Intell* **84**, 75–93 .
- Dung, P. (1995) "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n–person games". *Artificial Intelligence*. **77** (2): 321–357.
- Caminada, M. Pigozzi, G. (2009) “On judgment aggregation in abstract argumentation”. *Auton Agent Multi-Agent Syst* 22, 64–102.
- Grossi D., Pigozzi G. (2014) “Introduction to Judgment Aggregation”. In: Bezhanishvili N., Goranko V. (eds) *Lectures on Logic and Computation*. ESSLI 2011, ESSLI 2010. Lecture Notes in Computer Science, vol 7388. Springer, Berlin, Heidelberg
- Besnard, P. Hunter A. (2001). "A logic-based theory of deductive arguments". *Artificial Intelligence*. 128 (1–2): 203–235.
- Panisson, A. Megeguzzi, F. Vieira, R. Bordini, R. (2015) “Towards Practical Argumentation in Multi-Agent Systems”, *Brazilian Conference on Intelligent Systems*, BRACIS.2015.30.
- Duan, J. Zhu, Y. Huang, S. (2012) "Stigmergy agent and swarm-intelligence-based multi-agent system," *Proceedings of the 10th World Congress on Intelligent Control and Automation*, Beijing, pp. 720-724.
- Lemaitre, C. Excelente, C. (1998) “Multi-Agent Organization Approach”, In *Proceedings of the second Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent systems*, Toledo, Spain.
- Weyns, D. Schumacher, M. Ricci, A. Viroli, M. (2005) “Environments in multiagent systems”. *The Knowledge Engineering Review*, Volume 20, Issue 2, pp.127-141. Cambridge University Press.