

UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

**ARQUITETURA BASEADA NO MODELO TOULMIN
PARA O RACIOCÍNIO BASEADO EM
ARGUMENTAÇÃO EM AGENTES BDI**

Vágner de Oliveira Gabriel

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Prof. Dr. Cleo Zanella Billa

Co-orientadora: Prof. Dra. Diana Francisca Adamatti

Rio Grande, 2019

Ficha catalográfica

G118a Gabriel, Vágner de Oliveira.
Arquitetura baseada no modelo Toulmin para o raciocínio baseado em argumentação em agentes BDI / Vágner de Oliveira Gabriel. – 2018.
76 p.

Dissertação (Mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2018.
Orientador: Dr. Cleo Zanella Billa.
Coorientadora: Dra. Diana Francisca Adamatti.

1. Raciocínio baseado em argumentação 2. Modelo Toulmin 3. Sistemas multiagente 4. Arquitetura BDI I. Billa, Cleo Zanella II. Adamatti, Diana Francisca III. Título.

CDU 004



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

DISSERTAÇÃO DE MESTRADO

Arquitetura Baseada no Modelo Toulmin para o Raciocínio Baseado em Argumentação em Agentes BDI

Vágner de Oliveira Gabriel

Banca examinadora:

Prof. Dr. Rafael Heitor Bordini

Prof. Dr. Antonio Carlos da Rocha Costa

Prof. Dr. Cleo Zanella Billa
Orientador

Profa. Dra. Diana Francisca Adamatti
Coorientadora

*Dedico os esforços deste trabalho aos meus pais, os quais sempre foram os
incentivadores da minha busca por conhecimento.*

AGRADECIMENTOS

Primeiramente, agradeço a Deus por ter iluminado a minha jornada e me dar forças para concluir mais uma etapa na minha vida.

Agradeço aos meus pais, Ernestina Terezinha Boeira de Oliveira e Edgar Moreira Gabriel, o qual não se encontra mais entre nós, porém sua lembrança estará sempre em nossos corações. O apoio deles foi essencial para alcançar os meus objetivos, uma vez que nunca mediram esforços para me proporcionar uma base sólida na minha caminhada estudantil.

Agradeço ao meu orientador Cleo Zanella Billa, por todo conhecimento construído, ensinamentos compartilhados, orientação em todos os momentos que foram necessários, obrigado por estar sempre presente. Sua dedicação foi essencial para a realização deste trabalho e para concluir esta etapa da minha vida.

Agradeço imensamente a minha coorientadora Diana Francisca Adamatti, por ter me incentivado durante esta jornada, por sempre estar disposta a me ajudar e ajudar ao grupo, nos estimulando e incentivando para atingirmos o nosso melhor. Além de ser uma excelente coorientadora é um exemplo de pessoa e profissional, a qual admiro e me espelho. Obrigado por ser essa pessoa simples, justa e amiga dos alunos.

Agradeço a Aline Portantiolo Lettnin, que esteve sempre ao meu lado, me incentivando, me apoiando e sempre disposta a me ajudar quando necessário. Obrigado por toda dedicação e carinho. Seu apoio e incentivo foram essenciais em minha evolução acadêmica.

Agradeço aos colegas do grupo de pesquisa LAMSA que estiveram diariamente convivendo comigo nesses últimos dois anos, onde tivemos momentos de muito trabalho e também momentos de diversão com boas risadas.

A inteligência é o farol que nos guia, mas é a vontade que nos faz caminhar.
ÉRICO VERÍSSIMO

RESUMO

GABRIEL, Vágner de Oliveira. **ARQUITETURA BASEADA NO MODELO TOULMIN PARA O RACIOCÍNIO BASEADO EM ARGUMENTAÇÃO EM AGENTES BDI**. 2019. 76 f. Dissertação de Mestrado (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

Sistemas multiagente são um conjunto de agentes autônomos, que atuam de forma organizada em um ambiente compartilhado, colaborando entre si e buscando de uma forma conjunta atingir um objetivo satisfatório para o grupo. Em sistemas multiagente, existem distintas arquiteturas que possuem a finalidade de especificar a estrutura interna de um agente, neste trabalho utilizamos a arquitetura racional chamada BDI. A arquitetura BDI é uma das arquiteturas mais conhecidas para modelar o comportamento de agentes racionais e possibilita construir um agente com algumas atitudes mentais, sendo estas: crenças, desejos e intenções. Prover técnicas e métodos para proporcionar o raciocínio dos agentes, evitando conflito de crenças e possibilitando tomadas de decisões bem fundamentadas é muito importante. Neste contexto, a argumentação surge como uma ferramenta que possibilita os agentes utilizarem um raciocínio baseado em argumentação para tomarem decisões e reduzirem conflitos de informações. A teoria da argumentação abrange diversas áreas do conhecimento e tem ganhado um espaço significativo na comunidade de sistemas multiagente, uma vez que, pode possibilitar ao agente raciocinar sobre suas decisões, baseado em percepções adquiridas sobre o ambiente ou através da comunicação com outros agentes. Toulmin, no campo da filosofia, desenvolveu um modelo, o qual indica seis componentes para montar a estrutura de um argumento, sendo eles: dados, garantias, alegação, apoio, qualificador e refutação. Este trabalho descreve o desenvolvimento de uma arquitetura para inferência baseada em argumentos em agentes BDI, a qual foi desenvolvida sobre o modelo de Toulmin. A arquitetura desenvolvida possibilita analisar e justificar uma alegação, proporcionando a representação de incertezas através da geração de quantificadores de força, os quais acompanham a alegação, como forma de justificar a decisão tomada. Para demonstrar o funcionamento da arquitetura desenvolvida, foram descritos quatro estudos de caso distintos, os quais apresentam situações reais dentro do cenário de aplicação da arquitetura.

Palavras-chave: Raciocínio Baseado em Argumentação, Modelo Toulmin, Sistemas Multiagente, Arquitetura BDI.

ABSTRACT

GABRIEL, Vágner de Oliveira. **Architecture Based in Toulmin Model to Argumentation-Based Reasoning in BDI Agents**. 2019. 76 f. Trabalho Individual (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

Multi-agent systems are a set of autonomous agents, which act in an organized way in a shared environment, collaborating among themselves and seeking a joint form a satisfactory goal for the group. In multi-agent systems, there are different architectures that have the purpose of specifying the internal structure of an agent, in this work we use the rational architecture called BDI. The BDI architecture is one of the most well-known architectures for modeling the behavior of rational agents, and it makes possible to construct an agent with some mental attitudes: beliefs, desires and intentions. With techniques and methods that provide agents reasoning, it is possible avoid conflict of beliefs and enable informed decision making. In this context, argumentation emerges as a tool that enables agents to use argumentation-based reasoning to make decisions and reduce information conflicts. The theory of argumentation embraces several areas of knowledge and it has gained significant space in the community of multi-agent systems, since it can enable agents to reason about their decisions, based on acquired perceptions about the environment or through communication with other agents. Toulmin, in the philosophy field, developed a model, which indicates six components to assemble the structure of an argument: data, warrant, claim, backing, qualifier and rebuttal. This work describes the development of an argument-based inference architecture in BDI agents, which was developed on the Toulmin model. In the developed architecture, it possible to analyze and justify a claim, providing the representation of uncertainties through the generation of force quantifiers, which accompany the claim, as a way of justifying the decision making of the agents. In order to demonstrate the operation of the developed architecture, four different case studies were described, which present real situations using the proposed architecture.

Keywords: Argumentation-Based Reasoning, Toulmin Model, Multi-Agent Systems, BDI Architecture.

LISTA DE FIGURAS

Figura 1	Arquitetura BDI (HÜBNER; BORDINI; VIEIRA, 2004)	19
Figura 2	Gramática de especificação de um agente na linguagem <i>AgentSpeak(L)</i> (BORDINI; VIEIRA, 2003)	21
Figura 3	Interpretador <i>AgentSpeak(L)</i> (MACHADO; BORDINI, 2001)	22
Figura 4	Modelo de Toulmin com os componentes Dados (D), Garantias (W) e Alegação (C). (TOULMIN, 2003)	27
Figura 5	Exemplo do modelo de Toulmin com os componentes Dados (D), Garantias (W) e alegação (C)	28
Figura 6	Modelo de Toulmin com a adição dos componentes Refutação (R) e Qualificador (Q) (TOULMIN, 2003). (TOULMIN, 2003)	28
Figura 7	Exemplo do Modelo de Toulmin com a adição dos componentes Refutação (R) e Qualificador (Q)	29
Figura 8	Modelo de Toulmin com a adição do componente de Apoio (B) (TOULMIN, 2003)	29
Figura 9	Exemplo do Modelo de Toulmin com a adição do componente de Apoio (B)	30
Figura 10	Sistema de Raciocínio Processual (PSR) (BORDINI; HÜBNER; WOOLDRIDGE, 2007)	41
Figura 11	Modelo Toulmin em Agentes BDI (Arquitetura Toulmin)	42
Figura 12	Modelo Toulmin em Agentes BDI (Funcionamento)	43
Figura 13	Crenças em <i>AgentSpeak(L)</i> na plataforma Jason	44
Figura 14	Planos em <i>AgentSpeak(L)</i> na plataforma Jason	45
Figura 15	Exemplo Arquitetura Toulmin	48
Figura 16	Modelo implementado na linguagem <i>AgentSpeak(L)</i>	49
Figura 17	Estudo de Caso 1 - Nacionalidade em <i>AgentSpeak(L)</i>	53
Figura 18	Estudo de Caso 2 - Ir ao parque em <i>AgentSpeak(L)</i>	56
Figura 19	Estudo de Caso 3 - Comprar passagem para conferência em Paris em <i>AgentSpeak(L)</i>	58
Figura 20	Estudo de Caso 4 - Diagnóstico da Dengue em <i>AgentSpeak(L)</i>	66
Figura 21	Parte do código do Plano $+!DengueGrupo_A$ em <i>AgentSpeak(L)</i>	67

LISTA DE TABELAS

Tabela 1	Trabalhos Relacionados	34
Tabela 2	Trabalhos Relacionados - Mecanismos de raciocínio baseados em Argumentação	35
Tabela 3	Trabalhos Relacionados - Modelo Toulmin	38
Tabela 4	Componentes da Arquitetura	40
Tabela 5	Componentes Estudo de Caso 1 - Nacionalidade	52
Tabela 6	Componentes Estudo de Caso 2 - Ir ao parque	55
Tabela 7	Componentes Estudo de Caso 3 - Comprar passagem para conferência em Paris	57
Tabela 8	Classificação de risco de acordo com os sinais e sintomas	59
Tabela 9	Garantias e Refutações - Grupo A	62
Tabela 10	Garantias e Refutações - Grupo B	63
Tabela 11	Garantias e Refutações - Grupo C	64
Tabela 12	Garantias e Refutações - Grupo D	65

LISTA DE ABREVIATURAS E SIGLAS

BDI	Belief, Desire, Intentions.
SMA	Sistemas Multiagente.
SPG	Soma dos Pesos das Garantias
SPR	Soma dos Pesos das Refutações
NGV	Número de Garantias Válidas
NRV	Número de Refutações Válidas
PGV	Peso de Garantias Válidas.
PRV	Peso de Refutações Válidas.

SUMÁRIO

1	INTRODUÇÃO	14
2	FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA	17
2.1	Agentes	17
2.2	Sistemas Multiagente	18
2.3	Arquitetura BDI	18
2.4	AgentSpeak(L)	20
2.4.1	Sintaxe Abstrata	20
2.4.2	Semântica informal	22
2.5	Plataforma Jason	23
2.6	Argumentação	24
2.6.1	Argumentação em Sistemas Multiagente	25
2.7	Modelo Toulmin	26
2.8	Considerações Finais do Capítulo	30
3	TRABALHOS RELACIONADOS	31
3.1	Mecanismos de raciocínio baseados em argumentação	31
3.2	Modelo Toulmin na Computação	35
3.3	Considerações Finais do Capítulo	37
4	ARQUITETURA PROPOSTA	39
4.1	Arquitetura Baseada no Modelo de Toulmin para o Raciocínio Baseado em Argumentos entre Agentes BDI	39
4.1.1	Função Qualificador	41
4.2		42
4.3	Implementação da Arquitetura	42
4.3.1	Crenças	42
4.3.2	Planos	44
4.3.3	Função Qualifier	45
4.3.4	Exemplo do processo de raciocínio	47
4.3.5	Implementação da Estrutura da Arquitetura em AgentSpeak(L)	49
4.4	Considerações Finais do Capítulo	49
5	ESTUDOS DE CASO	51
5.1	Estudo de Caso 1 - Nacionalidade	52
5.2	Estudo de Caso 2 - Ir ao parque	54
5.3	Estudo de Caso 3 - Comprar passagem para conferência em Paris	56
5.4	Estudo de Caso 4 - Classificação de Risco de Paciente com Dengue	58

5.4.1	Modelagem do Estudo de Caso	61
5.4.2	Possíveis Resultados Gerados	61
5.5	Considerações Finais do Capítulo	69
6	CONCLUSÃO	70
	REFERÊNCIAS	72

1 INTRODUÇÃO

Em uma sociedade, pessoas devem interagir e se comunicar para conseguir resolver seus problemas a fim de alcançar seus objetivos. Dentro de uma sociedade de agentes não é diferente, os agentes necessitam argumentar para resolver problemas, reduzir conflitos e informar fatos uns aos outros (OLIVA; MCBURNEY; OMICINI, 2007). Dessa forma, torna-se necessário a existência de modelos de argumentação que possibilitem uma tomada de decisão bem fundamentada por parte dos agentes.

Agentes podem ser definidos como entidades virtuais que atuam em um ambiente, no qual interagem e colaboram com outros agentes movidos por um objetivo (FERBER; GASSER, 1991). Agentes podem agir de uma forma isolada ou em grupos, formando sistemas multiagente (REZENDE, 2003). Ao longo dos anos, surgiram diversas arquiteturas para sistemas multiagente, RAO; GEORGEFF et al. (1995) apresentaram a arquitetura racional denominada BDI, a qual foi fundamentada sobre modelo de BRATMAN (1987) e permite aos agentes englobar algumas atitudes mentais: *Belief* (B) - crenças, *Desire* (D) - desejos, *Intentions* (I) - intenções.

A teoria da argumentação é estudada em distintos campos do conhecimento, como na retórica, no jurídico, na filosofia e na inteligência artificial. Na área da inteligência artificial, a argumentação tem evoluído constantemente, sendo utilizada para prover a conciliação de informações e tomadas de decisão. Alguns modelos de argumentação considerados influentes em IA são apresentados por (DUNG, 1995), (PRAKKEN; SARTOR, 1997) e (RAHWAN et al., 2003).

Nos últimos anos, a argumentação ganhou um espaço significativo na comunidade multiagente devido à possibilidade de prover a um agente conciliar informações conflitantes dentro de si, conciliar novas informações adquiridas através de percepções do ambiente e para conciliar conflitos de informações entre múltiplos agentes através da comunicação. Na literatura, é possível encontrar diversos trabalhos na comunidade multiagente envolvendo argumentação (KRAUS; SYCARA; EVENCHIK, 1998) (SIERRA et al., 1998) (AMGOUD; PARSONS; MAUDET, 2000) (AMGOUD; MAUDET; PARSONS, 2000) (AMGOUD; CAYROL, 2002) (RAHWAN et al., 2003).

Em sistemas multiagente, a argumentação pode ser dividida em duas linhas principais

de pesquisa: (i) Raciocínio baseado em argumentação, a qual é usada sobre informações incompletas, conflitantes ou incertas, onde os argumentos a favor e contra são construídos e comparados para atingir conclusões (crenças, objetivos, etc.); (ii) Diálogos baseados em argumentação, onde existe uma interação entre agentes, trocando argumentos para justificar uma posição e fornecer justificativas que apoiem suas reivindicações.

Este trabalho apresenta como proposta o desenvolvimento de uma arquitetura para inferência baseada em argumentos para agentes BDI, a qual tem como objetivo prover componentes para a fundamentação de uma alegação lançada no ambiente por um agente. A arquitetura idealizada no trabalho tem sua estrutura composta de acordo com os fundamentos abordados por TOULMIN (2003). Toulmin apresentou um modelo de argumento alternativo ao modelo racional tradicional, o qual descreve como ponto chave a análise estrutural de um argumento.

O modelo de Toulmin define seis componentes como importantes na construção de um argumento: dados, que são fatos e informações recorridos para sustentar uma alegação; garantias, as quais servem como apoiadores dos dados da alegação; qualificador, que serve como um quantificador de força que as garantias possuem na alegação; refutações, que possuem a função de refutar ou invalidar uma alegação; e apoios, que servem como apoiadores (estatutos, documentos ou leis) das garantias.

A arquitetura apresentada neste trabalho, além de possibilitar agentes utilizarem um raciocínio baseado em argumentação para tomarem decisões, propicia examinar porque uma determinada alegação é falsa ou verdadeira. Esta possibilidade de argumentar sobre uma decisão tomada ocorre pelo fato da arquitetura desenvolvida ser dividida em componentes, os quais podem ser utilizados para justificar uma tomada de decisão ou criação de novas crenças.

Neste trabalho desenvolveu-se um modelo teórico, inserindo os componentes apresentados por TOULMIN (2003) e os adaptando para o modelo BDI. Dessa forma, construindo uma arquitetura que empregue uma estrutura robusta para quantificar a força de uma alegação. Ao final do processo de construção do modelo teórico da arquitetura, o mesmo foi implementado na linguagem *AgentSpeak(L)*, a fim de avaliar sua adaptação em distintas situações, sendo quatro estudos de caso apresentados ao final do trabalho.

Este trabalho está organizado em 6 capítulos. O primeiro capítulo apresenta uma introdução sobre o trabalho. O segundo capítulo aborda a fundamentação teórica e tecnológica do trabalho, onde são conceituados agente e sistemas multiagente, teoria da argumentação, *AgentSpeak(L)* e a Plataforma Jason. Já o terceiro capítulo descreve os trabalhos correlacionados a este trabalho. O quarto capítulo descreve a arquitetura proposta, abordando o desenvolvimento da arquitetura e posteriormente sua implementação na linguagem *AgentSpeak(L)*. No quinto capítulo são apresentados distintos estudos de caso, com o objetivo de comprovar que a arquitetura implementada suporta diferentes situações. Finalmente, no sexto capítulo são descritas as conclusões e possíveis trabalhos

futuros.

2 FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA

Neste capítulo, serão apresentados os componentes teóricos e as tecnologias que servirão como base para o desenvolvimento deste trabalho. Este capítulo aborda uma breve introdução sobre agentes, sistemas multiagente, arquitetura BDI, *AgentSpeak(L)* e plataforma Jason. Em seguida, apresenta uma introdução sobre Argumentação e finalmente, descreve o modelo apresentado por Toulmin para estruturar um argumento.

2.1 Agentes

Na literatura é possível encontrar distintas definições para o termo agente, algumas utilizadas para compor a definição apresentada neste trabalho podem ser encontradas em (FERBER; GASSER, 1991) (FRANKLIN; GRAESSER, 1997) (REZENDE, 2003) (RUSSELL; NORVIG, 1995)(WOOLDRIDGE; JENNINGS, 1994).

Agentes são entidades reais ou virtuais situadas em um ambiente, no qual são capazes de perceber componentes, interagir com outros agentes e trabalhar de forma colaborativa buscando alcançar seus objetivos (FERBER; GASSER, 1991). Os agentes podem realizar suas tarefas de forma isolada ou trabalhar em comunidades formando assim, sistemas multiagente (REZENDE, 2003). Segundo WOOLDRIDGE; JENNINGS (1994) agentes inteligentes apresentam as seguintes propriedades:

- **Autonomia:** os agentes operam sem conexão direta com seres humanos ou outros agentes, possuindo o total controle sobre suas ações e estado interno.
- **Habilidades sociais:** os agentes se comunicam com outros agentes e também com humanos através da utilização de um agente de comunicação de linguagem.
- **Reatividade:** agentes podem perceber alterações em seu ambiente e reagir em um tempo satisfatório para mudanças ocorridas no mesmo.
- **Pró atividade:** o agente, além de reagir as alterações em seu ambiente, toma iniciativas para alcançar seus objetivos.

2.2 Sistemas Multiagente

Sistemas multiagente (SMA) são um conjunto de agentes autônomos, que atuam de forma organizada em um ambiente compartilhado, colaborando entre si, objetivando resolver um determinado problema (AMANDI, 1997) (WOOLRIDGE, 2001). WEISS (1999) aponta duas propriedades consideradas fundamentais em sistemas multiagente: autonomia dos agentes e sua organização. Autonomia refere-se ao agente ter sua existência independentemente da existência de outros agentes, mesmo que para alcançar seus objetivos necessite da ajuda de outros agentes. Já a organização define restrições aos comportamentos dos agentes, objetivando desenvolver um comportamento em grupo entre os agentes, como cooperação, coordenação, negociação e argumentação.

No universo multiagente existem dois grandes modelos que especificam o comportamento dos agentes: os modelos reativo e cognitivo. Agentes reativos são caracterizados por possuírem um conhecimento implícito, o qual se manifesta apenas através de seu comportamento. O comportamento dos agentes reativos baseia-se no que é percebido a cada instante no ambiente, porém sem existir uma representação explícita deste. Este modelo de agentes possui uma organização etológica, ou seja, suas atitudes são similares a de animais e o número de membros de suas organizações são bem elevados, chegando a dezenas, centenas ou até mesmo milhares. Agentes reativos não podem armazenar um histórico de suas ações, dessa forma uma ação futura não depende de uma experiência adquirada em uma ação realizada no passado (ALVARES; SICHMAN, 1997).

Já os agentes cognitivos mantêm uma representação explícita do ambiente em que estão situados e dos agentes que também estão nesse ambiente. Estes agentes podem armazenar um histórico de ações passadas que podem vir a influenciar uma ação futura, e a comunicação realizada entre agentes cognitivos é feita através da troca de mensagens. Agentes cognitivos possuem um mecanismo deliberativo, de forma que possuem um raciocínio lógico e definem quais ações devem tomar. O modelo organizacional dos agentes cognitivos é sociológico, ou seja, sua organização é similar com as organizações humanas e suas sociedades são formadas geralmente por pouco agentes (FERBER; GASSER, 1991).

Em sistemas multiagente existem distintas arquiteturas que especificam a estrutura de um agente, que dependem da aplicação a ser realizada. Neste trabalho, será conceituada a seguir a arquitetura para agentes cognitivos chamada BDI, a qual foi apresentada por (RAO; GEORGEFF et al., 1995) e permite os agentes englobar as seguintes atitudes mentais: crenças (B), desejos (D), intenções (I).

2.3 Arquitetura BDI

Os agentes além de possuírem suas propriedades básicas tais como autonomia, proatividade e habilidade social, os mesmos podem possuir a capacidade de raciocínio e

aprendizagem (NUNES, 2007). Estudos envolvendo ciência cognitiva e técnicas da inteligência artificial são usados para proporcionar a modelagem de agentes virtuais com características similares à de humanos. Em 1987, BRATMAN (1987) propôs o modelo BDI como uma teoria filosófica sobre o raciocínio prático, onde o comportamento humano é modelado com as seguintes atitudes:

- *Belief* (crenças), as quais representam o conhecimento que um indivíduo possui sobre o ambiente e sobre outros indivíduos situados naquele ambiente incluindo o próprio indivíduo.
- *Desire* (desejos), os quais representam os objetivos que o indivíduo busca atingir no ambiente;
- *Intentions* (intenções), as quais representam sequencias de ações que o indivíduo se compromete a realizar para alcançar seus objetivos.

O modelo BDI é baseado na ideia de que as ações são derivadas do processo de raciocínio prático, constituído de dois passos. O primeiro passo é denominado deliberação, neste passo realiza-se uma triagem de um conjunto de desejos a serem alcançados, e no segundo passo é definido como os desejos selecionados podem ser obtidos.

RAO; GEORGEFF et al. (1995) desenvolveram um modelo de raciocínio para agentes virtuais, utilizando o modelo BDI como arquitetura para agentes de software, onde foi apresentada uma teoria formal para o modelo BDI. A Figura 1 ilustra a estrutura da arquitetura BDI.

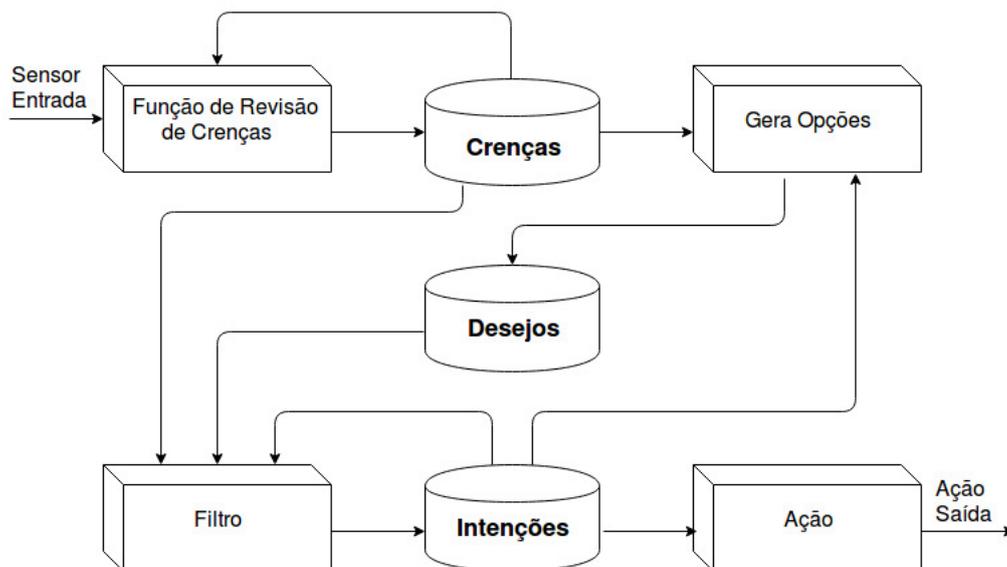


Figura 1: Arquitetura BDI (HÜBNER; BORDINI; VIEIRA, 2004)

Na Figura 1, as crenças representam o conhecimento que o agente possui sobre o ambiente e agentes que integram o ambiente incluindo si próprio. Os desejos representam

os objetivos que o agente almeja alcançar. As intenções representam os planos que o agente deve seguir para atingir seus objetivos. Revisão de crenças percebe informações do ambiente e consulta crenças anteriores do agente.

A função gera opções objetiva verificar as novas alternativas de estados a serem atingidos, uma vez que, busca encontrar as que apresentam maior relevância para os interesses particulares daquele agente. Este procedimento de gerar opções deve ser conduzido com base no estado atual do mundo, tendo base nas crenças e intenções que o agente possui. Logo, é possível afirmar que a atualização dos objetivos pode ocorrer quando as observações do ambiente possivelmente determinam novos objetivos para agente, ou a execução de intenções de um nível maior acarreta na necessidade de criar objetivos mais específicos a serem atingidos.

Com atualização dos objetivos e conhecimentos que o agente possui é preciso decidir as ações específicas a serem utilizadas buscando atingir os novos objetivos do agente, contudo, é preciso levar em conta as demais ações com que o agente se comprometeu para que não haja nenhuma eliminação de intenções ou que alguma intenção seja impossível de ser alcançada. A função de filtro tem a tarefa de atualizar o conjunto de intenções do agente, com base nas crenças e desejos atualizados e nas intenções já existentes. Por fim, a escolha de qual ação específica será a próxima a ser executada pelo agente é feita pela função ação.

2.4 AgentSpeak(L)

AgentSpeak(L) é uma linguagem de programação introduzida por (RAO; GEORGEFF et al., 1995) para o desenvolvimento de agentes racionais, a qual apresenta um modelo abstrato sendo desenvolvida sobre a arquitetura BDI possibilitando modelar os agentes com crenças, desejos e intenções.

Um agente modelado na linguagem *AgentSpeak(L)* equivale a especificação de uma base de crenças e um conjunto de planos. A base de crenças de um agente é a coleção de átomos de crenças, onde um átomo de crença representa um predicado de primeira ordem e literais de crença representam átomos de crenças ou suas negações (HÜBNER; BORDINI; VIEIRA, 2004).

2.4.1 Sintaxe Abstrata

Para especificar um agente *ag* em *AgentSpeak(L)* é preciso seguir a gramática apresentada por MOREIRA; BORDINI (2002) e ilustrada na Figura 2, de forma que um agente é modelado com crenças *bs*, a qual representa a base crenças inicial que o agente possui e um conjunto de planos *ps* que constitui a biblioteca de planos do agente. Segundo HÜBNER; BORDINI; VIEIRA (2004), a gramática de especificação de um agente na linguagem *AgentSpeak(L)* possui as seguintes propriedades:

$$\begin{array}{ll}
ag & ::= \quad bs \ ps \\
bs & ::= \quad b_1 \dots b_n & (n \geq 0) \\
at & ::= \quad P(t_1, \dots, t_n) & (n \geq 0) \\
ps & ::= \quad p_1 \dots p_n & (n \geq 1) \\
p & ::= \quad te : ct \leftarrow h \\
te & ::= \quad +at \mid -at \mid +g \mid -g \\
ct & ::= \quad at \mid \neg at \mid ct \wedge ct \mid \top \\
h & ::= \quad a \mid g \mid u \mid h; h \\
a & ::= \quad A(t_1, \dots, t_n) & (n \geq 0) \\
g & ::= \quad !at \mid ?at \\
u & ::= \quad +at \mid -at
\end{array}$$

Figura 2: Gramática de especificação de um agente na linguagem *AgentSpeak(L)* (BORDINI; VIEIRA, 2003)

- Fórmulas atômicas(*at*) são predicados, de forma que P é um símbolo predicativo e $t_1 \dots t_n$ são os termos padrão da lógica de primeira ordem. Uma fórmula atômica (*at*) sem variável é denominada crença e b é uma meta-variável para crenças, já um conjunto inicial de crenças é simbolizado como uma sequência de crenças(*bs*).
- Em *AgentSpeak(L)* um plano(*p*) é constituído de um evento ativador(*te*), de um contexto do plano(*ct*) e de uma sequência de ações, objetivos ou atualizações de crenças(*h*). É possível conceituar $te : ct$ como cabeça do plano e h como o corpo. Já um conjunto de planos pertencentes a um agente é representado por *ps*.
- Um evento ativador(*te*) é utilizado para a adição e remoção de crenças ou também adição ou remoção de objetivos:

+at Adiciona Uma Crença

-at Remove Uma Crença

+g Adiciona Um Objetivo

-g Remove Um Objetivo

Ações são expostas através de predicados usuais com a exceção de que um símbolo de ação A , a qual é utilizado no lugar do símbolo predicativo. Os objetivos podem ser objetivos de realização(!*at*) ou de teste(?*at*). As atualizações (*update*) da base de crença(*u*) são realizadas através da inserção de $+at$ e $-at$ no corpo de um plano, de forma que $-at$ apaga um crença existente e $+at$ adiciona uma nova crença na base.

- A fórmula !*g* no corpo de um plano gera um evento o qual o evento ativador é +!*g*. Dessa forma, os planos que possuem um evento ativador que permite ser unificado com +!*g* correspondem a alternativas de planos que devem ser considerados no

tratamento de tal evento. Planos que possuem eventos ativadores do tipo $+at$ e $-at$ são utilizados em eventos gerados quando crenças são adicionadas ou removidas, já planos ativadores do tipo $-!g$ são utilizados para o tratamento de falhas de planos. Por fim, eventos do tipo $+?g$ e $-?g$ são usados para a implementação da linguagem.

2.4.2 Semântica informal

Em *AgentSpeak(L)*, um interpretador abstrato necessita ter acesso a base de crenças, a biblioteca de planos e gerencia um conjunto de eventos e um conjunto de intenções. Para o funcionamento do interpretador em *AgentSpeak(L)* é necessário a existência de três funções de seleção: função de seleção de evento (S_E), que seleciona um único evento do conjunto de eventos; função de seleção de opção (um plano aplicável) (S_{Ap}) entre o conjunto de planos aplicáveis para um evento selecionado. E a função de selecionar intenção (S_I), a qual seleciona uma intenção em um conjunto de intenções. Intenções são rotas de ações que o agente se compromete para tratar eventos específicos e eventos que causam o início da execução de planos que possuem eventos ativadores relevantes.

O funcionamento do interpretador abstrato para *AgentSpeak(L)* é ilustrado na Figura 3, onde o conjunto de crenças, eventos, planos, e intenções são representados por retângulos. A seleção de um elemento de um conjunto é representado por um losango e os processos envolvidos na interpretação de programas são representados por círculos.

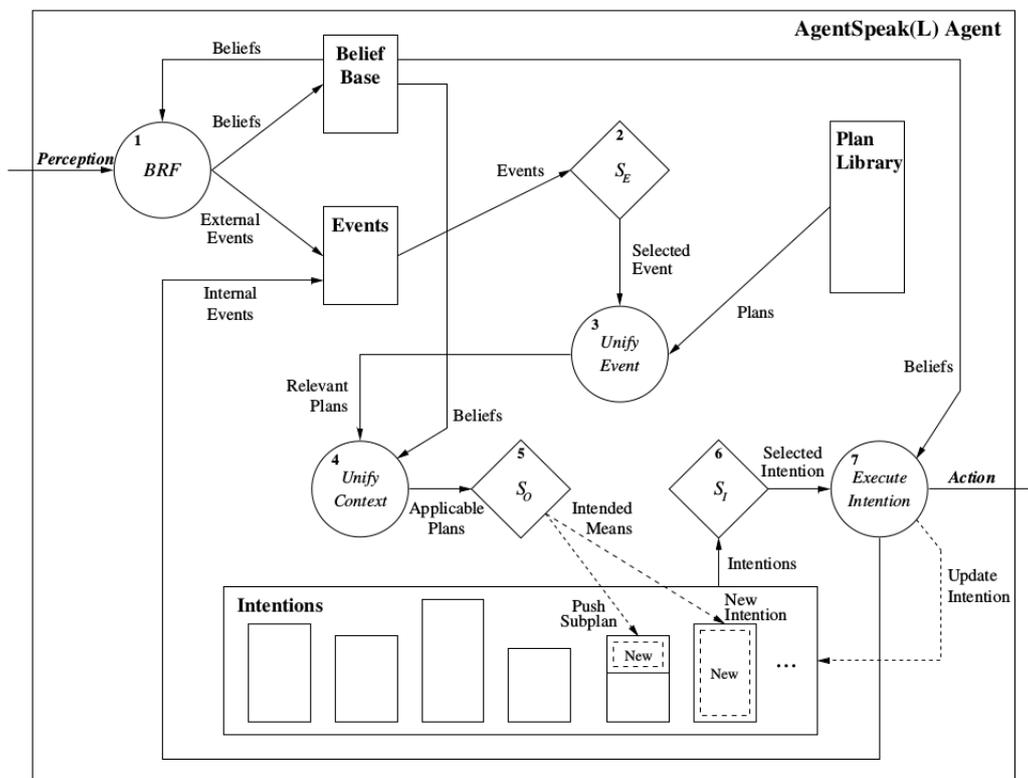


Figura 3: Interpretador *AgentSpeak(L)* (MACHADO; BORDINI, 2001)

A lista de eventos é atualizada a cada ciclo de interpretação do programa *AgentSpeak(L)* com o resultado do processo de revisão de crença. A função de revisão de crença não integra o interpretador *AgentSpeak(L)*, porém é um componente que deve integrar a arquitetura geral de um agente. Geralmente as implementações de interpretadores *AgentSpeak(L)* fornecem uma função simples como *default*.

Após S_E selecionar um evento, o interpretador *AgentSpeak(L)* tem a função de unificar o evento escolhido por S_E com eventos ativadores nas cabeças dos planos. Estes eventos ativadores estão presentes na biblioteca de planos, e o objetivo dessa unificação é gerar um conjunto constituído de todos planos importantes para o evento selecionado.

O *AgentSpeak(L)* determina um conjunto de planos aplicáveis, os quais são planos que podem ser executados para tratar o evento selecionado naquele ciclo. Após, S_A seleciona entre os planos aplicáveis um único plano aplicável, o qual vira o meio pretendido para o tratamento daquele evento e insere o plano no topo de uma intenção existente, caso o evento seja interno. Caso o evento seja externo, gera uma nova intenção no conjunto de intenções.

Ao final, resta apenas selecionar uma única intenção no ciclo. A função S_I seleciona uma intenção do agente, e no topo da intenção selecionada existe uma instância de um plano da biblioteca de planos, de forma que a fórmula do início do plano é executada. Esse processo gera uma ação básica a ser realizada pelo agente no ambiente, sendo estas ações um objetivo de realização ou um objetivo de testes. Maiores informações sobre *AgentSpeak(L)* podem ser encontradas em RAO (1996), MACHADO; BORDINI (2001) e BORDINI; VIEIRA (2003).

2.5 Plataforma Jason

Existem diversas plataformas para o desenvolvimento de agentes, algumas que apresentam linguagens específicas da ferramenta e outras que utilizam linguagens já existentes como Java e XML. Algumas das ferramentas para o desenvolvimento de agentes mais conhecidas são NetLogo¹, Jade², Jadex³, Mason⁴ e Jason⁵. Neste trabalho foi utilizada a plataforma Jason, a qual possibilita a modelagem de agentes racionais de acordo com os fundamentos da arquitetura BDI.

O Jason é uma plataforma de código aberto para uma versão estendida da linguagem *AgentSpeak(L)*. A plataforma Jason foi desenvolvida na linguagem Java, atribuindo as características multiplataforma e Open source, além de ser facilmente personalizável e apropriada para o desenvolvimento de arquiteturas BDI (BORDINI; HÜBNER; WO-

¹<https://ccl.northwestern.edu/netlogo/>

²<http://jade.tilab.com/>

³<https://www.activecomponents.org/>

⁴<http://cs.gmu.edu/eclab/projects/mason/>

⁵<http://jason.sourceforge.net/wp/>

OLDRIDGE, 2007). Outra característica importante é que um SMA desenvolvido na plataforma Jason pode ser facilmente distribuído em uma rede de computadores.

Segundo HÜBNER; BORDINI; VIEIRA (2004) além de interpretar a linguagem *AgentSpeak(L)* original a plataforma Jason apresenta os seguintes recursos:

- Negação forte (*strong negation*), dessa forma são possíveis tanto sistemas que consideram mundos fechados quanto sistemas que consideram mundos abertos.
- São possíveis tratamentos de falhas em planos;
- São possíveis comunicações baseadas em atos de fala;
- Existe a possibilidade de anotações em identificadores de planos, as quais podem ser utilizadas na elaboração de funções customizadas para a seleção de planos.
- O Jason possui suporte para o desenvolvimento de ambientes, o que normalmente é realizado na linguagem Java;
- O Jason possibilita executar um SMA distribuído em uma rede através do uso do SACI;
- No Jason existe a possibilidade de especializar na linguagem Java funções de seleção de planos, funções de confiança e toda a arquitetura do agente (percepção, revisão de crenças, comunicação e atuação);
- O Jason possui uma biblioteca básica de ações internas;
- O Jason possibilita estender a biblioteca de ações internas;

2.6 Argumentação

Para atingir um acordo, os indivíduos, em uma sociedade, precisam negociar, a fim de que todos os interessados consigam alcançar seus determinados objetivos. A argumentação, neste contexto, surge como uma forma de auxiliar na negociação, possibilitando fundamentar, através de argumentos ou alegações, um determinado ponto de vista. A argumentação pode ser definida como um processo verbal e social que busca qualificar a aceitabilidade de um ponto de vista de um indivíduo, apresentando fatos que justifiquem ou refutem uma alegação diante de um juiz racional (VAN EEMEREN et al., 2013).

A teoria da argumentação desperta atualmente o interesse em múltiplos campos do conhecimento, como nos campos da retórica, do jurídico, do filosófico e da inteligência artificial. Alguns autores são destaques em seus campos, como (TOULMIN, 2003) na filosofia, em que o mesmo aponta que um modelo ideal de argumento é constituído por seis componentes (dados, garantia, refutação, apoio, qualificador e alegação).

No campo da retórica destaca-se (PERELMAN, 1971) que desenvolveu a teoria chamada Nova Retórica, a qual define que o efeito da argumentação é medido com o peso de sua repercussão na audiência alvo. Esta audiência alvo pode ser um grupo em particular ou ninguém, dessa forma, o orador está a arguir consigo próprio.

Na área da inteligência artificial, os modelos formais de argumentação têm ganhado grande destaque ao longo dos anos, uma vez que possuem uma ampla aplicação na especificação de semânticas para a programação lógica. Estes modelos formais de argumentação podem ser utilizados para prover a conciliação de informações, facilitar diálogos entre agentes e auxiliar na tomada de decisão (WALTON; REED; MACAGNO, 2008).

Os primeiros trabalhos envolvendo argumentação no campo da IA foram apresentados por (FLOWERS; MCGUIRE; BIRNBAUM, 1982)(BIRNBAUM; FLOWERS; MCGUIRE, 1980) e tiveram um foco no estudo do processamento de linguagem natural. Demais trabalhos envolvendo argumentação e IA surgiram no decorrer dos anos seguintes, alguns modelos considerados influentes na IA foram apresentados por (DUNG, 1995) (PRAKKEN; SARTOR, 1997) (BONDARENKO et al., 1997) (VREESWIJK, 1997) (KRAUSE et al., 1995) e (RAHWAN et al., 2003).

2.6.1 Argumentação em Sistemas Multiagente

Nos últimos anos, a pesquisa, envolvendo a teoria da argumentação em sistemas multiagente, tem recebido um crescente interesse (MAUDET; PARSONS; RAHWAN, 2006) o qual surge como uma forma de facilitar o raciocínio do agente para a tomada de decisões e na troca de informações através de diálogos. A argumentação fortalece a interação para obtenção de acordos em negociação em que os agentes não partilham de um acordo, de forma a possibilitar que agentes lidem com informações incompletas e entrem em acordo com os demais agentes da sociedade (MOULIN et al., 2002). Segundo MAUDET; PARSONS; RAHWAN (2006), a argumentação pode auxiliar na resolução de dois tipos principais de problemas no campo de pesquisa de sistemas multiagente.

- **Formação e revisão de crenças e decisões**

A argumentação fornece recursos sistemáticos para a resolução de conflitos entre diferentes argumentos. Dessa forma, a argumentação possibilita para o agente tomar decisões e gerar crenças a partir de informações incertas, conflitantes e incompletas.

- **Interação racional**

Conforme já foi mencionado, uma característica essencial em sistemas multiagente é a comunicação para atingir suas metas individuais e coletivas. A argumentação inserida na comunicação fornece recursos para estruturar diálogos entre agentes com pontos de vistas conflitantes, objetivando que a interação entre os agentes respeite certos princípios.

A argumentação na comunidade multiagente pode ser dividida em duas linhas principais de pesquisa: a linha de raciocínio baseado em argumentação e a linha de diálogos baseados em argumentação. A linha de pesquisa sobre raciocínio baseado em argumentação tem seu foco voltado para o raciocínio (raciocínio não-monotônico) sobre informações conflitantes e incompletas em que argumentos são construídos contra e a favor de certas conclusões, objetivando uma tomada de decisão.

Já a linha de pesquisa de diálogos baseados em argumentação tem seu foco na comunicação e interação entre agentes em que permite que agentes dialoguem com intercâmbio de argumentos para justificarem e defenderem suas reivindicações sobre alguma alegação, crença ou objetivo. A argumentação inserida em diálogos fornece uma forma de estruturar interações racionais entre agentes, possibilitando que questionem a validade de reivindicações e permitindo que possam persuadir outros participantes do diálogo a aceitarem suas alegações.

2.7 Modelo Toulmin

Stephen Toulmin, em 1958, apresentou um modelo de argumento alternativo ao silogismo tradicional (TOULMIN, 2003), o qual especifica a estrutura de um argumento e detalha diferentes componentes relevantes na fundamentação de um argumento.

O modelo de Toulmin não se restringe apenas a troca de argumentos envolvendo dois ou mais indivíduos, este modelo também pode ser usado para um raciocínio individual, onde o indivíduo extrai suas conclusões a partir de informações de seu conhecimento (HITCHCOCK, 2005).

Segundo TOULMIN (2003), normalmente existem fatos que oferecem apoio a uma alegação caso a mesma seja desafiada, uma vez que, esta alegação não seja feita de forma irresponsável. No caso de uma alegação ser desafiada, cabe ao indivíduo apresentador da alegação recorrer a fatos e apresentá-los como fundamentos no qual se baseou para fazer tal alegação.

Um exemplo que ilustra o desafio a uma alegação pode ser um indivíduo fazer a seguinte alegação: “Paulo é brasileiro”. Este indivíduo pode ser questionado sobre essa afirmação feita. “O que você tem para seguir com essa alegação”? Logo o indivíduo precisa informar no que ele se baseou para chegar a tal alegação. “Tenho o conhecimento que Paulo nasceu no Rio Grande do Sul”. Este é o dado que o indivíduo possui, o qual serve como base para a sua alegação.

A partir deste ponto, é possível nomear os dados como (D - *Data*), este que são os fatos e dados recorridos para sustentar uma alegação e nomear a alegação como (C - *Claim*), que são as alegações que o indivíduo estabelece como verdadeiras. Seguindo o exemplo apresentado, pode surgir um novo questionamento solicitando uma relação entre os dados apresentados e a alegação.

O próximo passo é focado em apresentar preposições que apoiem o dado apresentado como suporte para a alegação, desde que não sejam novos dados. Este componente de apoio aos dados (D) é chamado de garantias (W - *Warrants*), o qual é utilizado como suposição implícita que funciona como uma ponte entre dados e alegação. Neste caso o exemplo ficaria da seguinte forma: o conhecimento que o indivíduo possui sobre Paulo ter nascido no Rio Grande do Sul dá o direito de descartar a sugestão que Paulo não seja brasileiro, uma vez que, por conta da garantia que “se alguém nasce no Rio Grande do Sul é brasileiro”.

A partir do conhecimento mostrado até o momento já se torna possível montar o primeiro esqueleto para analisar um argumento, conforme Figura 4.

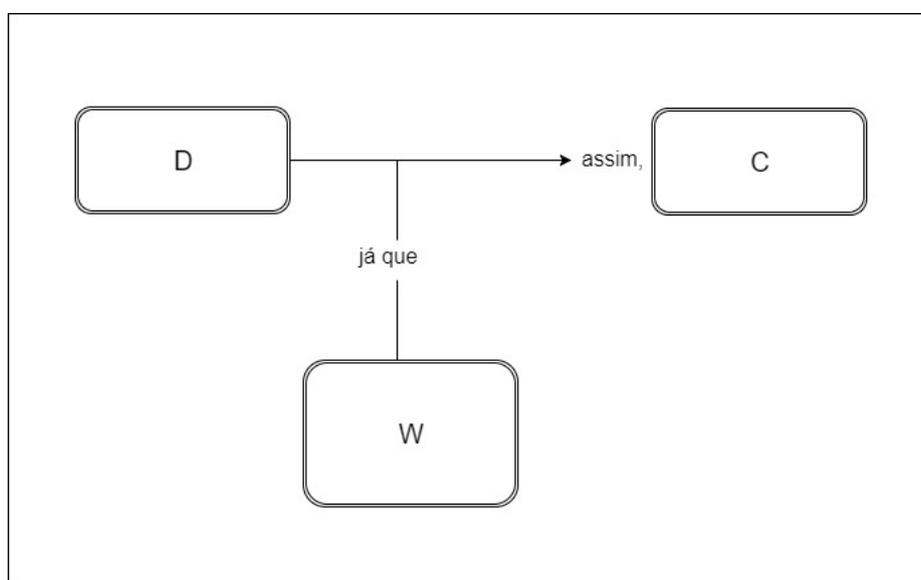


Figura 4: Modelo de Toulmin com os componentes Dados (D), Garantias (W) e Alegação (C). (TOULMIN, 2003)

Na Figura 4, a seta simboliza a relação entre os dados e a alegação, onde os dados servem de apoiadores para a alegação. Já a garantia sob a seta serve como uma ponte para passar dos dados para a alegação.

Podem existir diferentes tipos de garantias, as quais podem proporcionar distintos graus de força para uma alegação. Dessa forma, se torna necessário qualificar o nível de força que os dados conferem a alegação em virtude da garantia. Considerando a necessidade de qualificar um dado, se acrescenta ao modelo dois novos componentes, estes chamados de qualificadores (Q - *Qualifier*) e refutação (R - *Rebuttal*).

No modelo de Toulmin, os qualificadores se posicionam ao lado da alegação que eles qualificam, indicando a força que a garantia possui naquela alegação. Já a refutação, a qual tem a função de refutar ou invalidar a alegação tem seu posicionamento abaixo do qualificador.

Nas Figuras 6 e 7, seguindo o exemplo é ilustrado o funcionamento do argumento

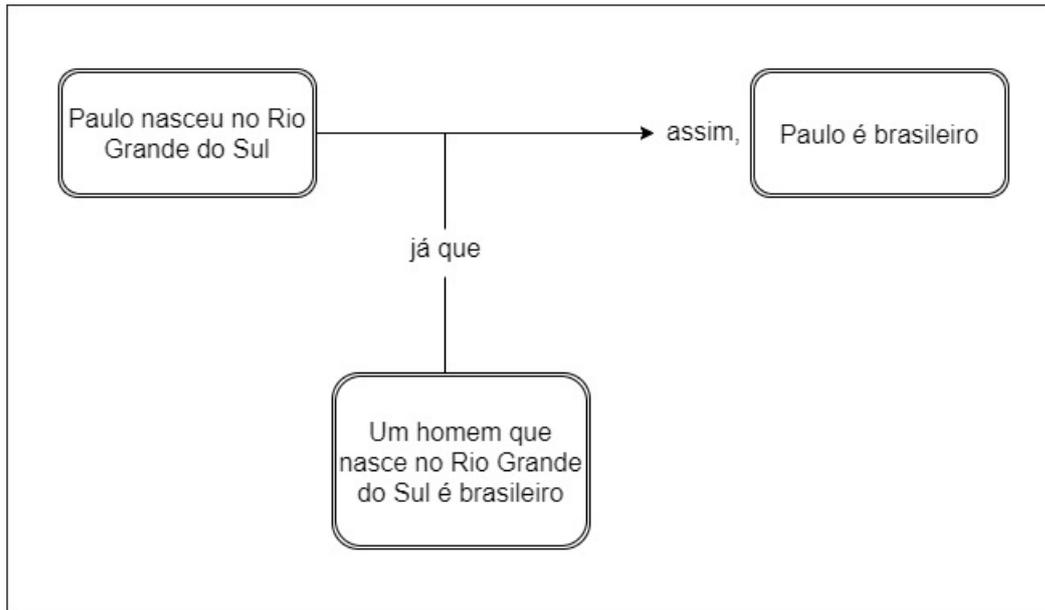


Figura 5: Exemplo do modelo de Toulmin com os componentes Dados (D), Garantias (W) e alegação (C)

com os componeres Q e R adicionados no modelo.

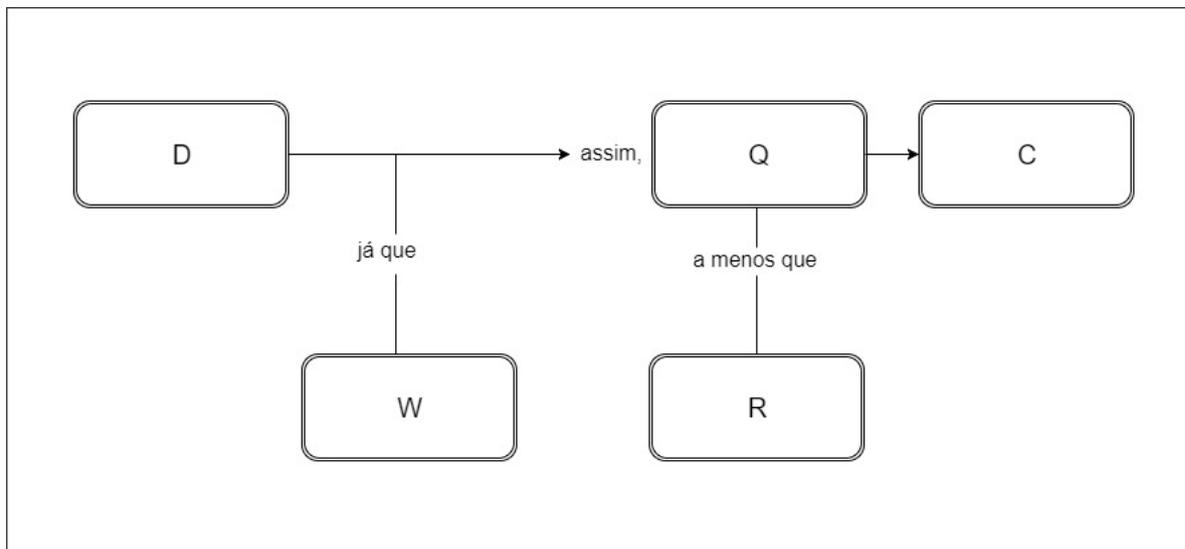


Figura 6: Modelo de Toulmin com a adição dos componentes Refutação (R) e Qualificador (Q) (TOULMIN, 2003). (TOULMIN, 2003)

De modo que, após apresentar os dados, as garantias, qualificadores, refutadores ou desafiante pode ainda questionar o porque dessas garantias. No exemplo apresentado foi presumido que uma pessoa nascida no Rio Grande do Sul é brasileiro, porém o desafiante da alegação pode questionar: Mas porque você pensa assim?

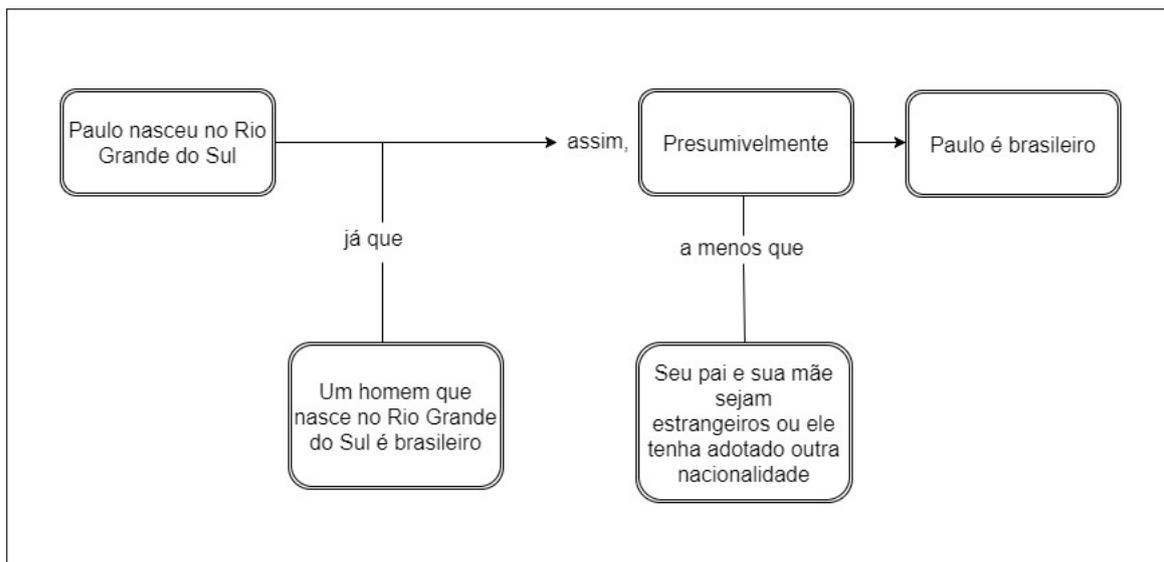


Figura 7: Exemplo do Modelo de Toulmin com a adição dos componentes Refutação (R) e Qualificador (Q)

Dessa forma, além das garantias pode haver outros avais, os quais servem como apoio para as garantias. Logo, é acrescentado no modelo mais um componente, o de apoio a garantia (B - *Backing*). Nas Figuras 8 e 9 é possível visualizar a versão completa do modelo proposto por Toulmin.

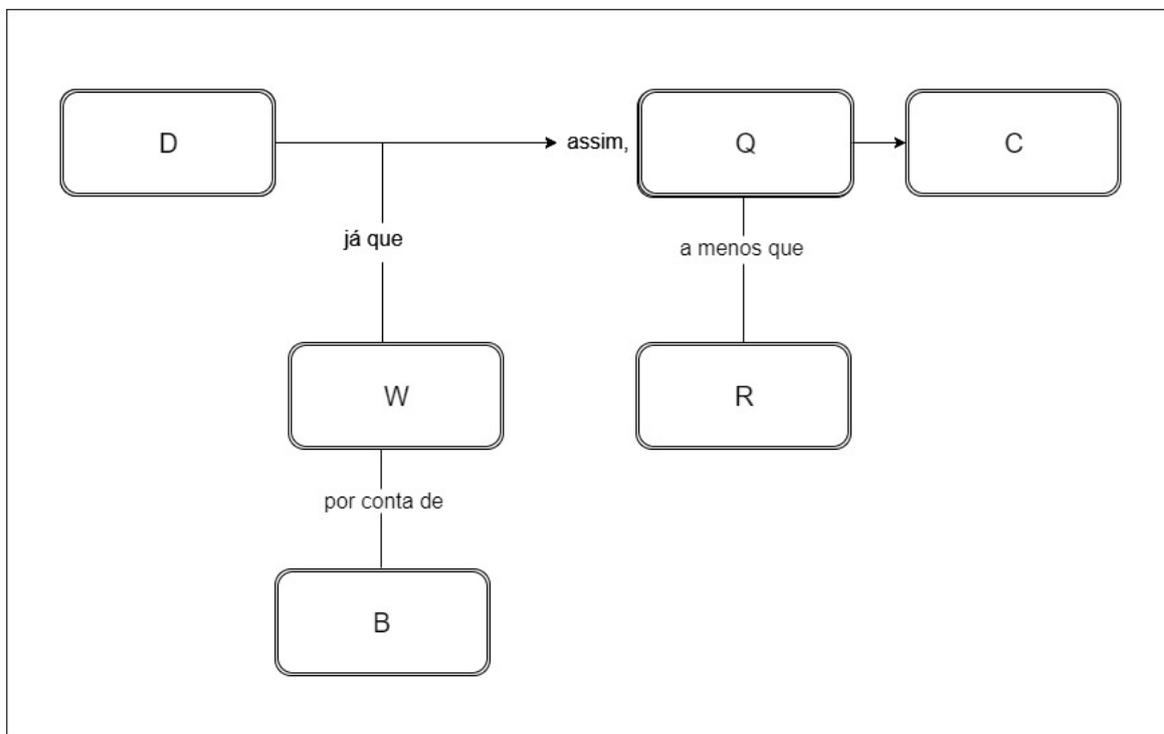


Figura 8: Modelo de Toulmin com a adição do componente de Apoio (B) (TOULMIN, 2003)

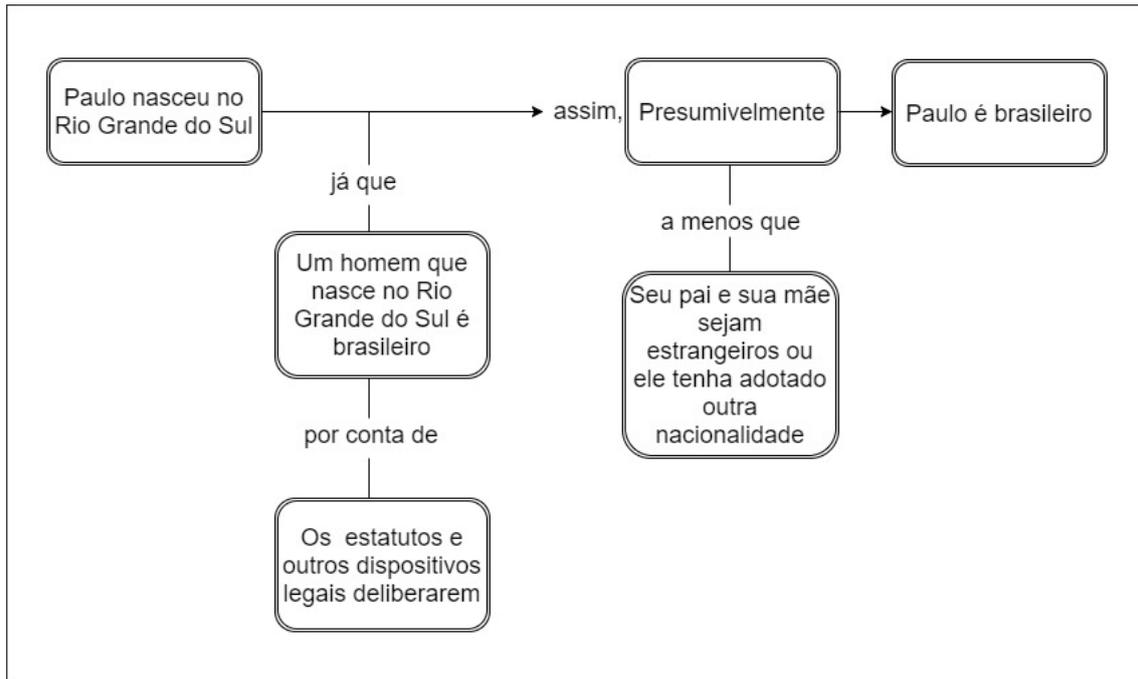


Figura 9: Exemplo do Modelo de Toulmin com a adição do componente de Apoio (B)

2.8 Considerações Finais do Capítulo

Na primeira parte desse capítulo foram apresentados os conceitos de agentes, sistemas multiagente, arquitetura BDI, *AgentSpeak(L)* e Plataforma Jason. Na segunda parte foi introduzido o conceito de argumentação, onde foram citados pesquisadores que se destacaram em suas áreas do conhecimento e foi discutido o papel da argumentação em sistemas multiagente. A terceira parte deste capítulo descreveu o modelo proposto por Toulmin e apresentou exemplos de seu funcionamento.

No próximo capítulo, dedicado à apresentar os trabalhos relacionados, serão descritos trabalhos que apresentam alguma similariedade com o que este estudo propõe.

3 TRABALHOS RELACIONADOS

Neste capítulo são descritos trabalhos que abordam propostas similares a proposta apresentada neste trabalho. Para um melhor entendimento, este capítulo foi dividido em duas seções: Na seção 1 “Mecanismos de raciocínio baseados em argumentação” são apresentados trabalhos que propõe teorias e implementações de modelos para o raciocínio de agentes. Na seção 2 deste capítulo "Trabalhos envolvendo o modelo Toulmin na computação" são descritos trabalhos que utilizaram o modelo proposto por TOULMIN (2003) para soluções computacionais.

3.1 Mecanismos de raciocínio baseados em argumentação

Neste trabalho, as pesquisas foram direcionadas para trabalhos envolvendo mecanismo de raciocínio baseados em argumentação. Diferentes abordagens para o raciocínio baseado em argumentação podem ser encontrados na literatura (BONDARENKO et al., 1997), (KAKAS; MORAITIS, 2003), (RAHWAN et al., 2003), (ATKINSON; BENCH-CAPON; MCBURNEY, 2005a), (ATKINSON; BENCH-CAPON; MCBURNEY, 2005b), (ATKINSON; BENCH-CAPON, 2007), (ROTSTEIN; GARCÍA; SIMARI, 2007), (BERARIU, 2014), (PANISSON, 2015), (VELMOVITSKY et al., 2017). Dentre os trabalhos encontrados, os mais similares a proposta deste trabalho são descritos a seguir.

Em (KAKAS; MORAITIS, 2003) é apresentada uma arquitetura baseada em argumentos para apoiar a tomada de decisão dentro de uma arquitetura modular para agentes. Neste trabalho, os autores adotaram uma arquitetura modular, de forma que cada módulo é dedicado a uma das capacidades do agente. Neste contexto, cada módulo pode argumentar de forma independente sobre assuntos de seu conhecimento e sugerir a melhor ação para estes assuntos. O comportamento geral de um agente é resultado da interação entre os módulos e suas decisões individuais. Os autores abordam que os processos de liberação de cada módulo podem ser diferentes, porém pode-se considerar que a maioria possui semelhanças na tomada de decisão para escolher entre distintas opções possíveis.

Segundo os autores, a estrutura apresentada no trabalho fornece um alto nível de adaptação nas tomadas de decisão do agente quando seu ambiente se altera. Além de pos-

sibilita ao agente raciocinar em um ambiente aberto, onde as informações disponíveis podem estar incompletas através da abdução integrada dentro do quadro de argumentação proposto. Os autores ainda, para prover aos agentes uma dimensão de individualidade enquanto interagem em uma sociedade, adicionaram um módulo adicional de personalidade a arquitetura proposta.

A arquitetura proposta neste trabalho apresenta características similares ao trabalho de KAKAS; MORAITIS (2003), uma vez que ambos buscam desenvolver uma arquitetura que propicie a um agente tomar decisões. Porém, o trabalho proposto é direcionado ao desenvolvimento de uma arquitetura baseada em argumentos exclusivamente para agentes de arquitetura BDI. Além de nossa arquitetura ser fundamentada sobre o modelo de TOULMIN (2003), diferentemente do trabalho de KAKAS; MORAITIS (2003), que utiliza outra abordagem.

Nos trabalhos (ATKINSON; BENCH-CAPON; MCBURNEY, 2005a) e (ATKINSON; BENCH-CAPON; MCBURNEY, 2005b), os autores introduzem um formalismo baseado em raciocínio prático, o qual especifica como os agentes devem argumentar sobre ações para determinar uma situação. O formalismo apresentado pelos autores fornece um conjunto de pré-condições que servem para especificar quando um agente pode construir uma posição, tendo como base, suas crenças, o conjunto de ações disponíveis para desempenho, seus desejos e seus valores. Devido à necessidade de existir o componente valores na proposta, os autores ampliaram a arquitetura BDI para incluir a noção de valores, a qual serve para fornecer justificativas para a seleção das intenções do agente com base em suas crenças e desejos, e especificam um conjunto de pré-condições que assim que satisfeitas permitem a agentes atacar uma justificativa de ação. Além disso, após todas as justificativas persuasivas e suas críticas serem identificadas, um curso defensivo de ações pode ser fundamentado com o uso de quadro de argumentação baseado em valores, o qual é uma extensão da estrutura abstrata de (DUNG, 1995).

Na proposta inicial deste trabalho, o agente utiliza o seu raciocínio para gerar crenças com qualificadores e não para gerar ações, como nos trabalhos (ATKINSON; BENCH-CAPON; MCBURNEY, 2005a,b). Contudo, futuramente pode ser estudada essa possibilidade. Neste trabalho, a base teórica da arquitetura é fundamentada sobre o modelo de TOULMIN (2003), diferente dos trabalhos (ATKINSON; BENCH-CAPON; MCBURNEY, 2005a) e (ATKINSON; BENCH-CAPON; MCBURNEY, 2005b) que utilizam uma argumentação baseada em valores, que estende a estrutura abstrata de (DUNG, 1995).

Já o trabalho (BERARIU, 2014) utiliza uma abordagem prática para a construção de agentes BDI argumentativos e baseia-se na estrutura abstrata de DUNG (1995), a qual é utilizada para o desenvolvimento de um módulo para ser integrado na plataforma Jason. O autor desenvolveu um módulo desacoplado do modelo BDI tradicional, uma vez que só opera na base de crenças dos agentes e não existe nenhuma interferência na execução dos planos, criação de objetivos ou compromissos do agente. O principal objetivo do trabalho

é permitir que os agentes consultem o módulo de argumentação para obter sugestões de ataques ou justificativas baseadas em argumentos para crenças aceitas ou rejeitadas.

Em (PANISSON, 2015), o autor desenvolveu um mecanismo de raciocínio baseado em argumentação em uma linguagem de programação orientada a agentes, a qual tem suas propriedades definidas de acordo com o modelo BDI. O mecanismo baseado em argumentação desenvolvido pelo autor tem sua fundamentação no mecanismo de *defeasible logic*, e possibilita aos agentes raciocinar sobre incertezas e argumentar, para dar suporte as suas alegações nas trocas de diálogos com outros agentes. Os autores utilizarão linguagem *AgentSpeak(L)* para implementar sua abordagem usando regras compatíveis com o Prolog, adaptando o d-Prolog, que é uma implementação do formalismo do *defeasible logic*. O autor descreve que em sua abordagem os agentes podem consultar sobre a existência de argumentos e sua aceitabilidade em seu próprio raciocínio durante a execução de um plano e durante um diálogo.

O trabalho (VELMOVITSKY et al., 2017) apresenta a implementação de um sistema de argumentação para o gerenciamento participativo de áreas ambientais protegidas, especificamente modelar um agente gerente de um parque. A implementação do trabalho baseou-se na arquitetura BDI, utilizando a linguagem *AgentSpeak(L)*. Os autores modelaram um sistema de argumentação por meio diferentes camadas de base de conhecimento e a relação de ataques entre argumentos, de forma, a gerar uma base para selecionar melhores argumentos viáveis. Os autores, apresentam um estudo de caso, onde a arquitetura proposta foi testada, modelando um agente como gerente de parque em um jogo sério para gerenciamento participativo de áreas protegidas. O agente gerente do parque tem como objetivo tomar decisões sobre os tipos de conservação examinando e argumentando fatos e argumentos sobre a situação e as preocupações da área protegida.

Os trabalhos de BERARIU (2014), PANISSON (2015) e (VELMOVITSKY et al., 2017) são os que apresentam mais características e objetivos similares ao trabalho aqui proposto. Esses trabalhos tem por objetivo desenvolver um mecanismo de raciocínio baseado em argumentação para agentes BDI. Além de adotarem a linguagem *AgentSpeak(L)* para a implementação de suas propostas. O que difere o trabalho aqui proposto dos três anteriormente citados são as abordagens teóricas utilizadas como base para o desenvolvimento e implementação.

- O trabalho de BERARIU (2014) utiliza como base teórica a última instanciação do formalismo abstrato de Dung apresentado por PRAKKEN (2010) .
- O trabalho de PANISSON (2015) utiliza como base teórica o formalismo do *defeasible logic*.
- No trabalho de VELMOVITSKY et al. (2017), os autores modelaram um sistema de argumentação que utiliza diferentes camadas de base de conhecimento e a relação

de ataques entre argumentos gera uma base para selecionar melhores argumentos viáveis.

- Neste trabalho utiliza-se como base teórica o modelo proposto por TOULMIN (2003).

O diferencial da arquitetura proposta em relação aos trabalhos de BERARIU (2014), PANISSON (2015) e (VELMOVITSKY et al., 2017), caracteriza-se por usar o modelo de Toulmin para fazer inferência em agentes BDI, de forma que possam ser analisados os componentes de forma independente e com isso justificar uma determinada alegação. Essa arquitetura divide uma alegação em componentes, os quais permitem uma análise mais profunda de como foi realizado o raciocínio e por isso argumentar sobre o mesmo.

As Tabelas 1 e 2 apresentam um resumo dos trabalhos relacionados aos mecanismos de raciocínio baseados em argumentação.

Tabela 1: Trabalhos Relacionados

Trabalho	Proposta
(KAKAS; MORAITIS, 2003)	Desenvolver uma arquitetura baseada em argumentos para apoiar a tomada de decisão dentro de uma arquitetura modular para agentes.
(ATKINSON; BENCH-CAPON; MCBURNEY, 2005a) e (ATKINSON; BENCH-CAPON; MCBURNEY, 2005b)	Introduzir um formalismo baseado em raciocínio prático, o qual especifica como os agentes devem argumentar sobre ações para determinar uma situação.

Tabela 2: Trabalhos Relacionados - Mecanismos de raciocínio baseados em Argumentação

Trabalho	Proposta
(BERARIU, 2014)	O trabalho utiliza uma abordagem prática para a construção de agentes BDI argumentativos e se baseia na estrutura abstrata de DUNG (1995), a qual é utilizada para o desenvolvimento de um módulo para ser integrado na plataforma Jason.
(PANISSON, 2015)	Os autores desenvolveram um mecanismo de raciocínio baseado em argumentação na linguagem <i>AgentSpeak(L)</i> . O mecanismo baseado em argumentação desenvolvido pelos autores tem sua fundamentação no mecanismo de <i>defeasible logic</i> , e possibilita os agentes raciocinarem sobre incertezas e argumentar para dar suporte as suas alegações nas trocas de diálogos com outros agentes.
(VELMOVITSKY et al., 2017)	O trabalho apresenta a implementação de um sistema de argumentação para o gerenciamento participativo de áreas ambientais protegidas, especificamente modelar um agente gerente de um parque. A implementação do trabalho baseou-se na arquitetura BDI, utilizando a linguagem <i>AgentSpeak(L)</i> .
Este trabalho	Aborda o desenvolvimento de uma arquitetura baseada no modelo de Toulmin para fazer inferência em agentes BDI, de forma que possam ser analisados os componentes de forma independente. Além de implementar o modelo teórico da arquitetura na linguagem <i>AgentSpeak(L)</i> .

3.2 Modelo Toulmin na Computação

O trabalho (BENCH-CAPON, 1998) apresenta o desenvolvimento de um jogo de diálogo fundamentado sobre o modelo de (TOULMIN, 2003). O autor apresenta uma es-

pecificação completa do jogo e uma implementação, de forma a permitir que o jogo seja jogado por jogadores reais (humanos). O autor ilustra o funcionamento de sua proposta mostrando um exemplo de diálogo e aponta que o modelo de Toulmin é o mais adequado para a realização de diálogos legais e não um modelo lógico.

No trabalho (GORDON; KARACAPILIDIS, 1997), os autores apresentaram o sistema ZENO, o qual baseia seu formalismo para a argumentação no modelo de TOULMIN (2003) e no modelo IBIS BIS – *Issue Based Information System* (RITTEL; WEBBER, 1973). O sistema ZENO foi desenvolvido pelo GMD (*German National Research Center for Information Technology*) com a finalidade de ser utilizado no planejamento urbano e foi desenvolvido para dar apoio a resolução de conflitos na regulação do planejamento urbano e dar suporte na realização de acordos no mundo da política.

Em (VERHEIJ, 2005a), o autor aborda como proposta uma elaboração formal do modelo de TOULMIN (2003) e estende o modelo para um tratamento formal de argumentos, onde os argumentos são avaliados em termos de uma chamada interpretação dialética de seus pressupostos. Para a construção de seu trabalho, o autor se baseou principalmente em seu trabalho sobre lógica dialética DEFLOG, o qual está relacionado ao software de assistência de argumento ARGUMED apresentado pelo autor em VERHEIJ (2003) e VERHEIJ (2005b).

No trabalho (ROWE; REED, 2014) aborda-se que argumentos formais, utilizados na ciência, medicina e lei, geralmente para estabelecer uma conclusão, são representados por diagramas no formato de árvores e gráficos. Dessa forma, os autores apresentam um pacote de software de código aberto chamado Araucária, disponível e desenvolvido na universidade de Dundee. A Araucária permite que argumentos de texto sejam marcados e representados no formato de diagramas padrão.

Já no trabalho aqui proposto, o modelo Toulmin foi utilizado como base teórica para o desenvolvimento da arquitetura proposta. Este modelo foi escolhido por apresentar uma metodologia que estrutura um argumento em seis componentes, os quais são essenciais para criar uma alegação. Portanto, utilizando este modelo, torna-se possível analisar em partes individuais um argumento e justificar uma alegação, proporcionando a representação de incertezas através da geração de quantificadores de força.

A Tabela 3 apresenta um resumo dos trabalhos relacionados ao modelo de Toulmin na computação.

3.3 Considerações Finais do Capítulo

Nosso objetivo neste capítulo foi apresentar trabalhos que abordam mecanismos de raciocínio baseados em argumentação e trabalhos envolvendo o uso do modelo Toulmin aplicado à computação.

Todos trabalhos descritos no capítulo apresentam propostas interessantes e válidas, todavia, este trabalho se difere dos trabalhos citados e contribui cientificamente por apresentar uma arquitetura em agentes BDI que permite analisar os componentes de forma independente e com isso justificar uma determinada alegação.

Tabela 3: Trabalhos Relacionados - Modelo Toulmin

Trabalho	Proposta
(BENCH-CAPON, 1998)	O trabalho utiliza o modelo de Toulmin como base para o desenvolvimento de um jogo de diálogo. O autor apresenta uma especificação completa do jogo e uma implementação, de forma a permitir que o jogo seja jogado por jogadores reais (humanos).
(GORDON; KARACAPILIDIS, 1997)	O trabalho apresenta o sistema ZENO, o qual baseia seu formalismo para a argumentação nos modelos de TOULMIN (2003) e no modelo IBIS BIS - Issue Based Information System (RITTEL; WEBBER, 1973), o qual foi desenvolvido para dar apoio a resolução de conflitos, com a finalidade de ser utilizado no planejamento urbano.
(VERHEIJ, 2005a)	O autor aborda como proposta uma elaboração formal do modelo de TOULMIN (2003) e estende o modelo para um tratamento formal de argumentos, onde os argumentos são avaliados em termos de uma chamada interpretação dialética de seus pressupostos.
ROWE; REED (2014)	O trabalho utiliza o modelo de Toulmin para representar argumentos de textos em formas de diagrama.
Este trabalho	Utiliza o modelo de Toulmin como base para o desenvolvimento de uma arquitetura para inferência baseada em argumentos em agentes BDI. A arquitetura desenvolvida possibilita analisar e justificar uma alegação, proporcionando a representação de incertezas através da geração de quantificadores de força.

4 ARQUITETURA PROPOSTA

Este capítulo tem como objetivo descrever como a arquitetura foi desenvolvida e modelada, apresentando a adaptação dos componentes encontrados no modelo proposto por Toulmin para o modelo BDI.

O capítulo está dividido em duas seções. Na primeira seção, é descrito o desenvolvimento do modelo de raciocínio baseado em argumentação, o qual foi fundamentado sobre o modelo Toulmin para fornecer uma estrutura para o raciocínio de agentes BDI. Na segunda seção é apresentada a implementação da arquitetura para o raciocínio de agentes BDI, que foi desenvolvida na linguagem de programação orientada a agentes *AgentSpeak(L)* disponível na plataforma Jason.

4.1 Arquitetura Baseada no Modelo de Toulmin para o Raciocínio Baseado em Argumentos entre Agentes BDI

A arquitetura desenvolvida visa tornar o raciocínio do agente autônomo mais robusto, reduzindo conflitos de crenças e auxiliando o agente a tomar uma decisão. A tomada de decisão do agente autônomo se torna possível, uma vez que, o mesmo possui componentes adaptados do modelo de Toulmin que irão auxiliá-lo a tomar uma decisão tendo como base um conjunto de crenças de entrada.

A arquitetura proposta é composta por seis componentes: Dados (D), Alegação (C), Garantia (W), Apoio (B), Refutação (R) e Qualificador (Q). Estes componentes são utilizados como base para quantificar a autenticidade de uma alegação. Os componentes da arquitetura estão descritos na Tabela 4.

Tabela 4: Componentes da Arquitetura

Componente	Descrição
Dados	Crenças que o agente pode ter no momento da sua criação, ou adquiridas do ambiente, ou recebidas através de diálogos com outros agentes, ou através de diálogos com usuários reais.
Garantias	Crenças que apoiam a alegação.
Refutações	Crenças que refutam a alegação.
Apoio	Documentos legais que apoiam as garantias e refutações.
Alegação	Uma nova crença gerada pelo raciocínio de um agente sobre as crenças de dados, garantias e refutações.
Qualificador	É um peso que quantifica o valor da sustentabilidade que uma alegação possui.

Com as definições dos componentes adaptados do modelo de Toulmin, tornou-se possível desenvolver uma arquitetura para inferência baseada em argumentos em agentes BDI. A Figura 10 apresenta o sistema de raciocínio processual (PSR) e a Figura 11 apresenta os componentes adaptados do modelo toulmin inseridos no PSR.

Na Figura 11 percebe-se que os componentes de Dados, Garantias, Refutação e Alegação são crenças e a função qualificador está localizada na biblioteca de planos.

A Figura 12 ilustra o funcionamento do modelo desenvolvido. Inicialmente, existem dados, garantias e refutações, as quais são crenças que o agente adquiriu através de diálogos ou percepções no ambiente. Estas crenças entram em um plano que busca tomar uma decisão. Este plano possui uma função chamada qualificador, a qual recebe os conjuntos de crenças de dados, garantias e refutações e retorna um peso quantificador para a alegação que será criada pelo plano. Por fim, o plano cria uma alegação com seu respectivo qualificador que acaba se tornando uma nova crença na sua base de crenças.

O algoritmo 1 apresenta o pseudo-código do processo descrito sobre a Figura 12.

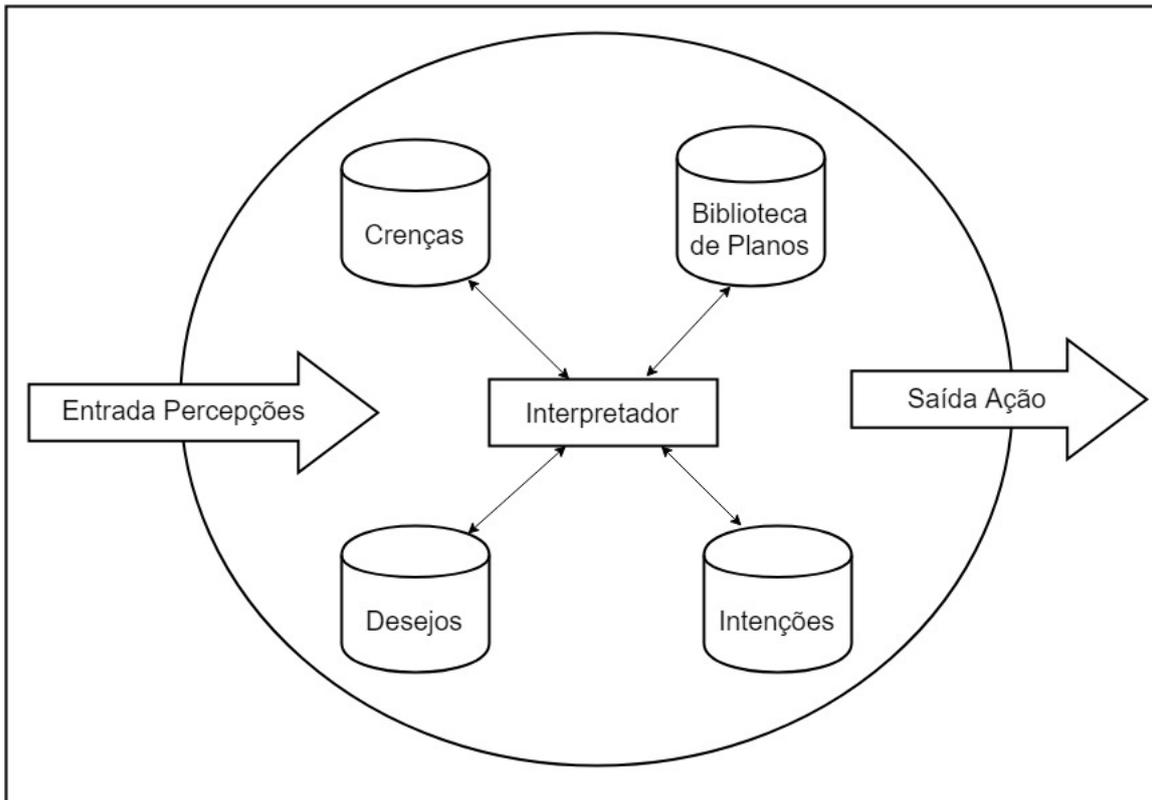


Figura 10: Sistema de Raciocínio Processual (PSR) (BORDINI; HÜBNER; WOOLDRIDGE, 2007)

Algoritmo 1: Arquitetura Toulmin

- 1: **entrada:** D[]: lista de Crenças; W[]: lista de Crenças; R[]: lista de Crenças
 - 2: **saída:** (Alegação, Qualificador): Tupla
 - 3:
 - 4: **procedimento** ARQUITETURA TOULMIN (D[], W[], R[])
 - 5: (Alegação, Qualificador) ← ComputaAlegação(D[], W[], R[])
 - 6: **retorna** (Alegação, Qualificador)
 - 7: **fimprocedimento**
-

4.1.1 Função Qualificador

A função qualificador tem objetivo de receber as crenças de dados, garantias e refutações, e com essas crenças gerar um peso quantificador para a crença alegação que o plano irá criar. Esta função qualificador, pode ser implementada conforme a necessidade do usuário, adaptando-se a diversos problemas. A função genérica do qualificador é dada a seguir:

$$F(D, W, R) \rightarrow (C, Q)$$

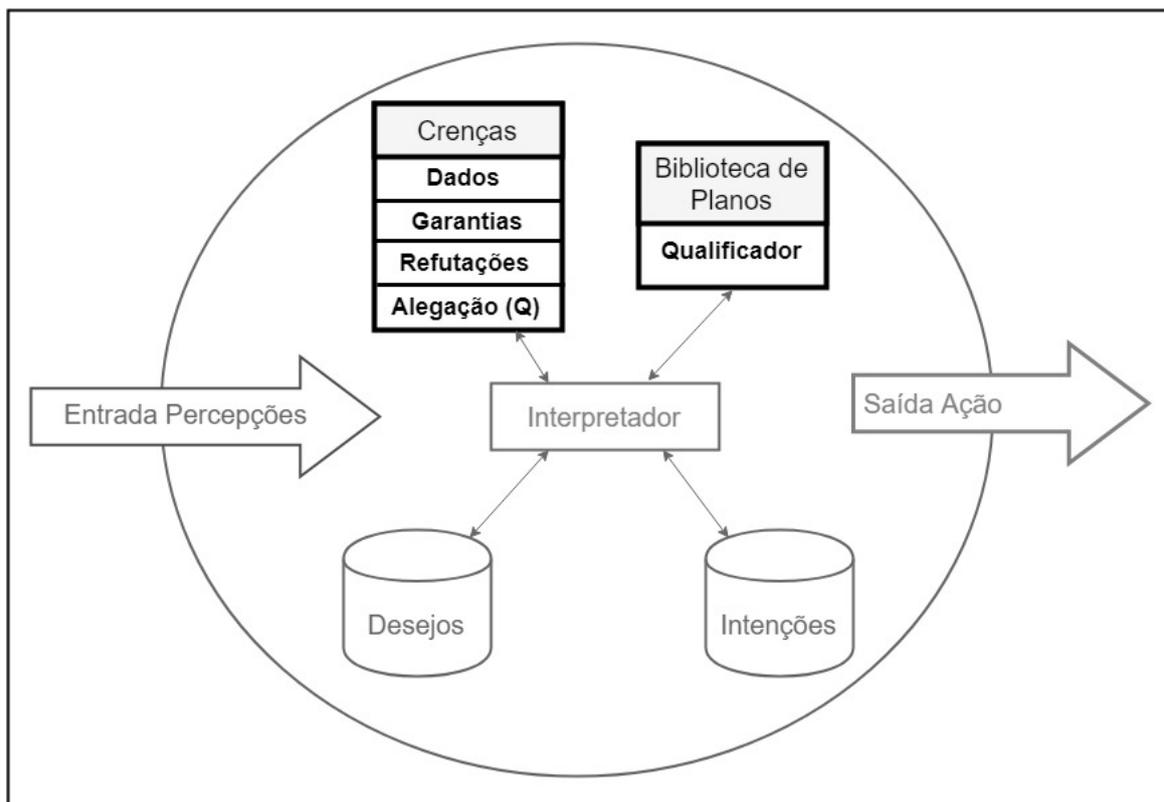


Figura 11: Modelo Toulmin em Agentes BDI (Arquitetura Toulmin)

4.2

A próxima seção deste capítulo descreve como o modelo apresentado nesta seção foi implementado na linguagem AgentSpeak(L) utilizando a plataforma Jason.

4.3 Implementação da Arquitetura

Para demonstrar o funcionamento da arquitetura desenvolvida, foi implementado o modelo teórico apresentado na seção anterior na plataforma Jason (BORDINI; HÜBNER; WOOLDRIDGE, 2007). Como já mencionado Jason é uma plataforma que possibilita o desenvolvimento de sistemas multiagente usando as linguagens Java e *AgentSpeak(L)*. A plataforma Jason permite a criação de ações internas, que ampliam a linguagem de programação com operações que não estão disponíveis ou implementadas. Para a implementação da arquitetura foi desenvolvida uma ação interna chamada *Qualifier* (Qualificador), a qual será descrita com mais detalhes na seção 4.2.2.

4.3.1 Crenças

No modelo desenvolvido considera-se os componentes dados, garantias e refutações como crenças. Dessa forma, torna-se necessário implementar estes componentes como representações de crenças na linguagem *AgentSpeak(L)*. A representação dos componen-

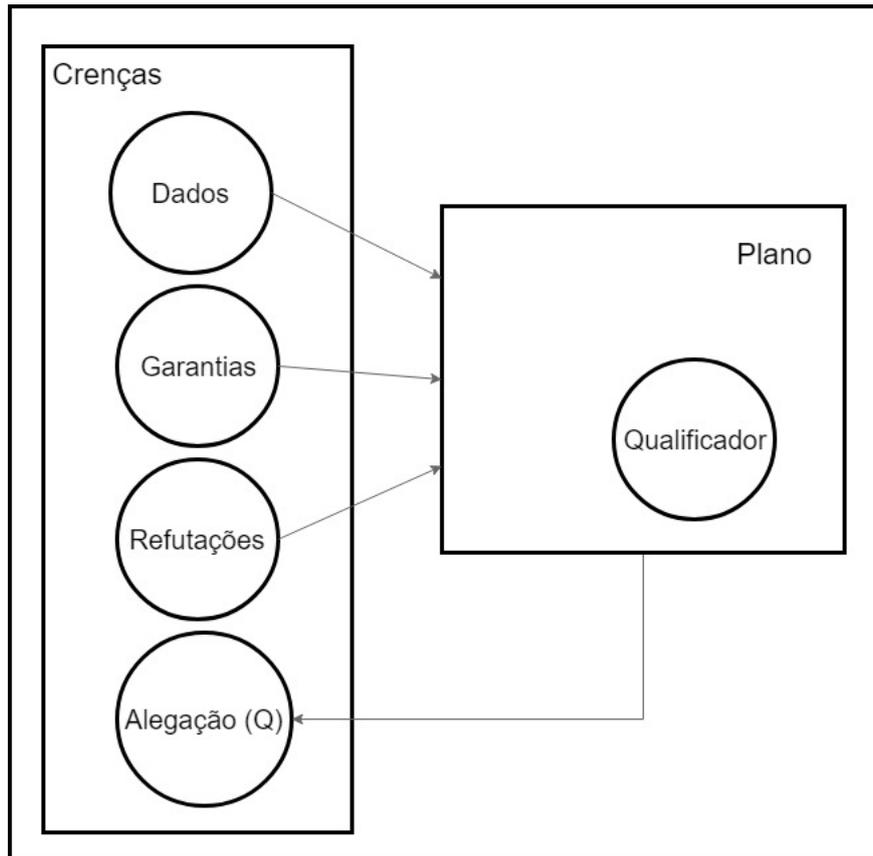


Figura 12: Modelo Toulmin em Agentes BDI (Funcionamento)

tes dados, garantias e refutações como crenças na linguagem *AgentSpeak(L)* dá-se da seguinte forma:

- `data(data1).`
- `warrant(warrant1, weight1).`
- `rebuttal(rebuttal1, weight1).`

As crenças de dados são informações que o agente pode possuir no momento de sua criação ou adquirir através de percepções do ambiente ou interações com outros agentes através diálogos. Os dados são o conhecimento que um agente possui para implicar em uma decisão que será tomada.

As crenças de garantias são informações que apoiam as crenças de dados em uma justificativa que fortalece como verdade a alegação que será gerada (ex. certamente Paulo é Brasileiro). Cada garantia possui um peso, o qual varia entre 0.0 e 1.0. Este valor aumenta conforme o nível de certeza que as garantias possuem sobre um determinado dado.

Já as crenças de refutações são informações que reduzem ou invalidam os dados de entrada (ex. certamente Paulo "não" é Brasileiro). Cada refutação também possui um

peso, o qual varia entre 0.0 e 1.0 e aumenta conforme o nível de certeza que as refutações possuem sobre um determinado dado.

Em uma simulação podem existir mais de um dado, mais de uma garantia e mais de uma refutação (Figura 13). Dessa forma, existe a necessidade de usar uma função para capturar todos os dados, garantias e refutações e colocar em listas. A plataforma Jason possui uma ação interna desenvolvida que acopla termos que possuem um mesmo nome em uma lista. Esta ação interna estará inserida no plano e será explicada na próxima seção.

```

01 /* beliefs */
02 data(data1).
03 data(data2).
04 data(data3).
05
06 warrant(warrant1, weight1).
07 warrant(warrant2, weight2).
08 warrant(warrant3, weight3).
09
10 rebuttal(rebuttal1, weight1).
11 rebuttal(rebuttal2, weight2).
12 rebuttal(rebuttal3, weight3).

```

Figura 13: Crenças em *AgentSpeak(L)* na plataforma Jason

4.3.2 Planos

Para a implementação de nosso modelo teórico existe a necessidade da criação de um plano, o qual busca as crenças dados, garantias e refutações e gera uma nova crença com seu respectivo qualificador. Dentro de um plano existe uma função de qualificação que recebe como parâmetros de entrada as crenças dados, garantias e refutações, computa as entradas e retorna um qualificador. Ao final do processo da função qualificador, o plano gera uma nova crença com um quantificador de verdade gerado pela função qualificador.

Como mencionado anteriormente (subseção 4.2.1), um plano precisa de uma função que busque todas as crenças que possuem um mesmo nome, uma vez que no modelo proposto existem "*N*" dados, "*N*" garantias e "*N*" refutações. A plataforma Jason possui por padrão uma ação interna chamada `.findall`, a qual realiza a tarefa de buscar "*N*" termos que possuem o mesmo nome e inserir em uma lista. O plano implementado na linguagem *AgentSpeak(L)* pode ser visualizado na Figura 14.

A seguir será descrito como foi implementada a ação interna `qualifier` inserida no plano implementado.

```

10 /* Plans */
11 !+plan:true
12 <- .findall(X, data(X), ListData);
13     .findall(Y, warrant(Y), ListWarrant);
14     .findall(Z, rebuttal(Z), ListRebuttal);
15     .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
                        Qualifier);
15     +claim("plan", Qualifier).

```

Figura 14: Planos em *AgentSpeak(L)* na plataforma Jason

4.3.3 Função Qualifier

Neste primeiro modelo¹, os pesos que quantificam cada garantia e cada refutação são valores entre 0.0 e 1.0. O valor dos pesos aumenta conforme o nível de certeza que as garantias e refutações influenciam sobre a alegação que será gerada.

A função de qualificação foi definida para calcular a soma dos pesos das garantias que apoiam as informações de entrada (Dados) (*SPG* – Soma dos Pesos das Garantias) e calcular a soma dos pesos das refutações que refutam os dados de entrada (*SPR* – Soma dos Pesos das Refutações). Para apoiar ou refutar um dado, as garantias e/ou as refutações devem possuir o dado em suas bases, ou seja, deve existir uma garantia ou refutação igual ao dado de entrada (ex. Dado: “Paulo nasceu no Rio Grande do Sul”, Garantia: “Paulo nasceu no Rio Grande do Sul, 1.0”). Dessa forma, podem existir N garantias e N refutações, porém só serão somados os pesos das garantias e refutações que são iguais aos dados de entrada.

Entretanto, se houver algum dado de entrada contido nas garantias e/ou refutações com peso máximo (1.0), a função de qualificação irá considerar apenas a garantia e/ou a refutação com esse peso, a partir da seguinte condição:

$$Q_{Final} = \begin{cases} 1.0 & \text{Se } \exists \text{ } PGV_i = 1.0 \\ 0 & \text{Se } \exists \text{ } PRV_i = 1.0 \\ 0.5 & \text{Se } \exists \text{ } PGV_i = 1.0 \text{ e } PRV_i = 1.0 \end{cases}$$

Caso exista algum dado de entrada que seja compatível com alguma garantia de valor 1.0, o peso que quantifica a alegação será 1.0. Dessa forma, gerando um quantificador simbólico de "Certamente sim"².

Caso exista algum dado de entrada que seja compatível com alguma refutação de valor 1.0, o peso que quantifica a alegação será 0. Dessa forma gerando um quantificador simbólico de "Certamente não".

¹Outras métricas podem ser utilizadas, dependendo do problema a ser modelado.

²Abaixo, nessa mesma seção, são apresentados todos os quantificadores simbólicos definidos.

Caso existam dados que são compatíveis com garantias e refutações e possuam valores 1.0, o peso que quantifica a alegação será 0.5. Dessa forma, gerando um quantificador simbólico de "Talvez".

Caso contrário, as equações 1 e 2 serão executadas.

$$SPG = \sum_{i=0}^{NGV} PGV_i \quad (1)$$

$$SPR = \sum_{i=0}^{NRV} PRV_i \quad (2)$$

Na equação 1, existe o número de garantias válidas (NGV) e o peso de cada garantia válida (PGV_i). Dessa forma, a equação apresenta um somatório, onde existe a variável i , que inicia com o valor 1 e vai até NGV somando o PGV de cada garantia que possui um dado compatível. A equação 2 realiza o mesmo processo que a equação 1, porém realizando o somatório dos dados compatíveis com informações contidas na lista de refutações. Na equação 2, NRV é o número de refutações válidas e PRV é o peso de refutações válidas.

Posteriormente a realização das somas, os valores são normalizados. O cálculo (equação 3) subtrai o $SPR_{normalizado}$ do valor do $SPG_{normalizado}$ gerando o valor (Q).

$$Q = SPG_{normalizado} - SPR_{normalizado} \quad (3)$$

Após o cálculo da equação 3, será gerando o valor Q_{Final} , que é o quantificador de verdade que a alegação gerada possui, baseado na seguinte condição:

$$Q_{Final} = \begin{cases} |Q| & \text{Se } SPG > SPR \\ 1 - |Q| & \text{Se } SPR > SPG \end{cases}$$

O peso que quantifica o valor da alegação é um valor entre 0.0 e 1.0, o qual é transformado em quantificador simbólico. Existem cinco possíveis quantificadores simbólicos:

- "Certamente não." → $[0, 0.2)$;
- "Provavelmente não." → $[0.2, 0.4)$;
- "Talvez." → $[0.4, 0.6]$;
- "Provavelmente sim." → $(0.6, 0.8]$;
- "Certamente sim." → $(0.8, 1.0]$.

4.3.4 Exemplo do processo de raciocínio

Para exemplificar o processo de raciocínio de um agente modelado com a arquitetura proposta, apresenta-se na Figura 15 um exemplo simples do processo de raciocínio. Na Figura 15, existem as seguintes crenças de entrada (Dados):

- “Paulo tem um filho brasileiro”
- "Os pais de Paulo são estrangeiros"
- "Paulo vive em Londres"
- "Paulo tem uma irmã que nasceu em Londres"

As Garantias que fortalecem a alegação de que Paulo é brasileiro são:

- Paulo nasceu no Rio Grande do Sul, 1.0.
- Paulo tem um filho brasileiro, 0.3.
- A esposa de Paulo é brasileira, 0.1.

As Refutações que diminuem a força da alegação são:

- Paulo possui outra nacionalidade, 1.0.
- Os pais de Paulo são estrangeiros, 0.3.
- Paulo vive em Londres, 0.2.
- Paulo tem uma irmã que nasceu em Londres, 0.3.

No exemplo, os dados de entrada entram no plano que tem como objetivo responder se Paulo é brasileiro.

Inicialmente, o processo verifica se existe algum dado de entrada de peso máximo (1.0) contido nas listas de garantias e/ou refutações. No exemplo, não existem dados que possuem um valor máximo, pois a garantia "Paulo nasceu no Rio Grande do Sul" com peso 1.0 e a refutação "Paulo possui outra nacionalidade" com peso 1.0, não tem dados de entrada que corroborem o uso destas duas informações (garantia e refutação). Os dados de entrada são compatíveis com uma garantia de peso 0.3 ("Paulo tem um filho brasileiro") e com três refutações de pesos 0.3, 0.2 e 0.3 ("Os pais de Paulo são estrangeiros", "Paulo vive em Londres" e "Paulo tem uma irmã que nasceu em Londres"). Após o processo de verificação, são aplicadas as equações 1 e 2, sendo os resultados:

$$SPG = 0.3$$

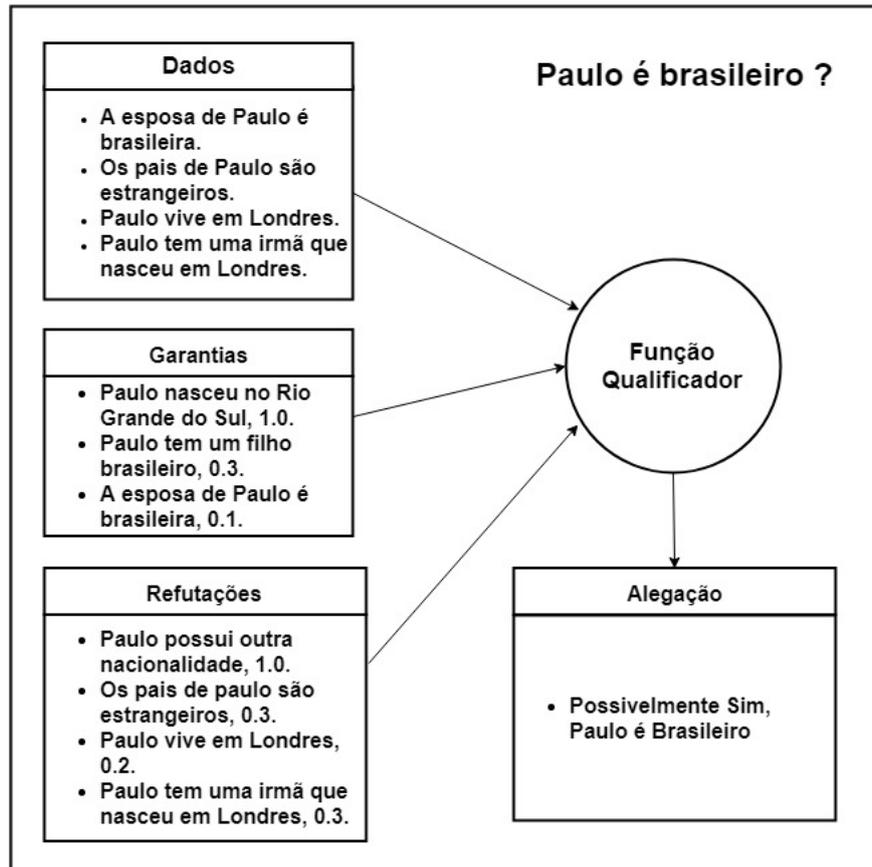


Figura 15: Exemplo Arquitetura Toulmin

$$SPR = (0.3 + 0.2 + 0.3) = 0.8$$

Posteriormente, aplica-se a equação 3 com os valores de SPG e SPR normalizados:

$$Q = (0.37 - 1.0) = -0.63$$

Após o cálculo da equação 3, será gerando o valor Q_{Final} , que é o quantificador de verdade que a alegação gerada possui, baseado na condição mostrada anteriormente.

$$SPR > SPG = Q_{Final} = 1 - | -0.63 | = 0.37$$

O peso gerado que quantifica o valor da alegação foi 0.37, uma vez que, os dados de entrada obtiveram um peso mais elevado nas listas de refutação. O valor 0.37 encaixa-se entre os valores 0.2 e 0.4 dos qualificadores simbólicos. Dessa forma, o qualificador simbólico da alegação é "Provavelmente não".

Ao final do processo executado pela função qualificador, será gerada uma alegação "Paulo **Provavelmente não** é brasileiro".

4.3.5 Implementação da Estrutura da Arquitetura em AgentSpeak(L)

A Figura 16 ilustra a implementação da arquitetura proposta na linguagem *AgentSpeak(L)*. Existe um plano que geralmente visa responder a uma pergunta, por exemplo: "Paulo é brasileiro?". O plano criado busca na base de crenças com o uso da ação interna `.findall` crenças de dados, garantias e refutações e cria as listas *ListData*, *ListWarrant* e *ListRebuttal*. Este plano também possui a ação interna *Qualifier* que recebe como entrada as listas *ListData*, *ListWarrant*, *ListRebuttal* e retorna o qualificador na variável *Qualifier*.

Inicialmente o plano busca as crenças de dados, garantias, refutações e cria listas. Após o processo de busca das crenças e criação de listas é possível executar a ação interna *Qualifier* tendo como parâmetro de entrada as listas *ListData*, *ListWarrant*, *ListRebuttal*. A ação interna do qualificador verificará se os dados inferem a questão como verdadeira ("Paulo é brasileiro?"). Caso contrário, os dados refutam e definem a questão como falsa. Após esse processo, o qualificador gera uma alegação com seu qualificador. Nesse caso, se Paulo é brasileiro ou não.

```

01 /* Belief */
02 data(data1).
03 data(data2).
04
05 warrant(warrant1).
06 warrant(warrant2).
07
08 rebuttal(rebuttal1).
09 rebuttal(rebuttal2).
10
11 /* Goal */
12 !modelToulmin
13
14 /* Plan */
15+!modelToulmin : true
16 <- .findall(X, data(X), ListData);
17   .findall(Y, warrant(Y), ListWarrant);
18   .findall(Z, rebuttal(Z), ListRebuttal);
19   .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
                       Qualifier);
20   +claim("Claim", Qualifier).

```

Figura 16: Modelo implementado na linguagem AgentSpeak(L)

4.4 Considerações Finais do Capítulo

Este capítulo apresentou o desenvolvimento da arquitetura fundamentada sobre o modelo de Toulmin para a inferência baseada em argumentos em agentes BDI. O capítulo

foi dividido em duas seções, inicialmente foram descritos os componentes teóricos da arquitetura proposta, sua estrutura e seu funcionamento. Em seguida, apresentou-se a implementação do modelo teórico da arquitetura na linguagem *AgentSpeak(L)* disponível na plataforma Jason.

Posteriormente a descrição da implementação da arquitetura, apresentou-se um exemplo teórico demonstrando o funcionamento geral da arquitetura.

Uma vez desenvolvida a arquitetura, o próximo capítulo apresenta quatro estudos de caso para analisar o funcionamento e adaptação da arquitetura em distintas situações.

5 ESTUDOS DE CASO

Nesta seção serão apresentados estudos de caso para avaliar o funcionamento da arquitetura desenvolvida. Quatro estudos de casos foram utilizados para avaliar a arquitetura, os quais foram implementados na plataforma Jason. O primeiro estudo de caso foi adaptado de TOULMIN (2003), que visa verificar a nacionalidade de um indivíduo. O segundo verifica se um indivíduo deve ir a um parque. O terceiro foi adaptado de PANISSON et al. (2014), o qual aborda como tema a compra de uma passagem para uma conferência em Paris. O quarto é mais complexo e visa classificar um paciente em um grupo de risco referente a sintomas da dengue.

5.1 Estudo de Caso 1 - Nacionalidade

Este estudo de caso foi adaptado de TOULMIN (2003) e busca responder se um indivíduo chamado Paulo é brasileiro. No estudo, as garantias atuam como apoiadores da alegação que "Paulo é brasileiro" e as refutações atuam como redutores de força desta alegação. A tabela 5 apresenta as crenças de dados, garantias e refutações usadas neste estudo de caso.

Tabela 5: Componentes Estudo de Caso 1 - Nacionalidade

Componente	Informações
Dados	<ul style="list-style-type: none"> ● Paulo vive no Rio Grande do Sul. ● Os pais de Paulo são brasileiros. ● A esposa de Paulo nasceu no Rio Grande do Sul. ● Paulo tem uma irmã que vive em Londres. ● Paulo tem um filho brasileiro. ● Paulo trabalha no Rio Grande do Sul. ● Paulo estudou em Londres.
Garantias	<ul style="list-style-type: none"> ● Paulo nasceu no Rio Grande do Sul, 1.0 . ● Paulo vive no Rio Grande do Sul, 0.2. ● Os pais de Paulo são brasileiros, 0.2. ● Paulo tem um filho brasileiro, 0.3. ● Paulo trabalha no Rio Grande do Sul, 0.2. ● A esposa de Paulo nasceu no Rio Grande do Sul, 0.1. ● Paulo tem uma irmã que vive no Rio Grande do Sul, 0.2.
Refutação	<ul style="list-style-type: none"> ● Paulo nasceu em Londres, 1.0. ● Paulo vive em Londres, 0.2. ● A esposa de Paulo nasceu em Londres, 0.1. ● Paulo tem uma irmã que vive em Londres, 0.2. ● Paulo estudou em Londres, 2.0.

A Figura 17 apresenta as informações contidas na Tabela 5 inseridas na arquitetura desenvolvida.

Na Tabela 5 e na Figura 17 percebe-se que existem cinco garantias e duas refutações que são compatíveis com os dados de entrada. Portanto, os únicos valores utilizados para gerar um quantificador neste estudo de caso são os pesos das garantias e refutações que corroborem os dados de entrada. A seguir, são apresentados os passos seguidos para realizar o cálculo que gera a alegação deste estudo de caso.

```

01
02 data("Paulo vive no Rio Grande do Sul").
03 data("Os pais de Paulo são brasileiros").
04 data("A esposa de Paulo nasceu no Rio Grande do Sul").
05 data("Paulo tem um filho brasileiro").
06 data("Paulo trabalha no Rio Grande do Sul").
07 data("Paulo tem uma irmã que vive em Londres").
08 data("Paulo estudou em Londres").
09
10 warrant("Paulo nasceu no Rio Grande do Sul", 1.0).
11 warrant("Paulo vive no Rio Grande do Sul", 0.2).
12 warrant("Os pais de Paulo são brasileiros", 0.2).
13 warrant("Paulo tem uma filho brasileiro", 0.3).
14 warrant("Paulo trabalha no Rio Grande do Sul", 0.2).
15 warrant("A esposa de Paulo nasceu no Rio Grande
      do Sul", 0.1).
16 warrant("Paulo tem uma irmã que vive no Rio Grande
      do Sul", 0.2).
17
18 rebuttal("Paulo nasceu em Londres", 1.0.).
19 rebuttal("Paulo vive em Londres", 0.2).
20 rebuttal("A esposa de Paulo nasceu em Londres", 0.1).
21 rebuttal("Paulo tem uma irmã que vive em Londres, 0.2").
22 rebuttal("Paulo estudou em Londres", 0.2).
23
24 !pauloBrasileiro.
25
26 +!pauloBrasileiro : true
27 <- .findall(X, data(X), ListData);
28     .findall(Y, warrant(Y), ListWarrant);
29     .findall(Z, rebuttal(Z), ListRebuttal);
30     .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
      Qualifier);
31     +claim(Qualifier, "Paulo é brasileiro").
32

```

Figura 17: Estudo de Caso 1 - Nacionalidade em *AgentSpeak(L)*

Inicialmente, a função verifica se existem dados de entrada que possuem garantias e/ou refutações de peso máximo (1.0). Não há nenhuma garantia ou refutação com peso 1.0 que seja compatível com os dados de entra. Nesse caso, a função utiliza as equações 1 e 2:

$$SPG = (0.2 + 0.2 + 0.3 + 0.2 + 0.1) = 1.0$$

$$SPR = (0.2 + 0.2) = 0.4$$

Posteriormente, aplica-se a equação 3 com os valores de *SPG* e *SPR* normalizados:

$$Q = (1.0 - 0.4) = 0.6$$

Após o cálculo da equação 3, é gerado o valor Q_{Final} , que é o quantificador de verdade que a alegação gerada possui, baseado na condição apresentada na seção 4.

$$SPG > SPR- > Q_{Final} = | 0.6 | = 0.6$$

O qualificador gerado possui o valor 0.6, uma vez que, o valor das garantias, que servem como apoiadores da alegação, apresentaram um valor maior. O valor 0.6 encaixa-se entre os valores 0.6 e 0.8 dos qualificadores simbólicos. Dessa forma, o qualificador simbólico da alegação é "Provavelmente Sim". Este valor simbólico poderia ter sido outro caso as entradas (dados) fossem diferentes, por exemplo:

- Entrada "Paulo nasceu no Rio Grande do Sul": O qualificador simbólico seria "Certamente Sim", uma vez que, o dado Paulo nasceu no Rio Grande do Sul é compatível com uma garantia de peso máximo (1.0).
- Entrada "Paulo nasceu em Londres": O qualificador simbólico seria "Certamente Não", uma vez que, o dado Paulo nasceu em Londres é compatível com uma refutação de peso máximo (1.0).
- Modificar o dado de entrada "Paulo vive no Rio Grande do Sul" para "Paulo vive em Londres": o qualificador simbólico seria "Talvez", uma vez que os valores compatíveis com as refutações e com as garantias teriam um valor muito próximo.

Neste estudo de caso, ao final do processo executado pela ação interna *toulmin.qualifier* () (Figura 17, linha 30), a alegação "Provavelmente sim, Paulo é brasileiro" é gerada e torna-se uma nova crença na base de conhecimento do agente.

5.2 Estudo de Caso 2 - Ir ao parque

Neste estudo de caso, existe um agente que precisa decidir se vai ao parque ou não. A Tabela 6 apresenta as crenças de dados, garantias e refutações deste estudo.

A Figura 18 apresenta o modelo desenvolvido para este exemplo em *AgentSpeak(L)*. Na Tabela 6 e na Figura 18 percebe-se que existem uma garantia que apoia e duas refutações que não apoiam os dados de entrada. Portanto, os únicos valores utilizados para gerar um quantificador neste estudo de caso são os pesos das garantias e refutações que corroboram os dados de entrada em sua base. A seguir são apresentados os passos seguidos para realizar o cálculo que gera a alegação deste estudo de caso.

Inicialmente, a função verifica se existe dados de entrada que possuem garantias e/ou refutações de peso máximo (1.0). Como não há, a função utiliza as equações 1 e 2:

$$SPG = (0.2) = 0.2$$

$$SPR = (0.7 + 0.5) = 1.2$$

Tabela 6: Componentes Estudo de Caso 2 - Ir ao parque

Componente	Informações
Dados	<ul style="list-style-type: none"> • O parque não é seguro. • Está chovendo. • O parque é perto.
Garantias	<ul style="list-style-type: none"> • O parque está aberto, 0.4. • O parque é seguro, 0.5. • Não está chovendo, 0.4. • O parque é perto, 0.2.
Refutação	<ul style="list-style-type: none"> • O parque não está aberto, 1.0. • O parque não é seguro, 0.7. • Está chovendo, 0.5. • O parque é longe, 0.2.

Posteriormente, aplica-se a equação 3 com os valores de SPG e SPR normalizados:

$$Q = (0.16 - 1.0) = -0.84$$

Após o cálculo da equação 3, é gerado o valor Q_{Final} , que é o quantificador de verdade que a alegação gerada possui, baseado na condição apresentada na seção 4.

$$SPR > SPG \rightarrow Q_{Final} = 1 - |-0.84| = 0.16$$

O qualificador gerado possui o valor 0.16, uma vez que, o valor das refutações da alegação apresentaram um valor maior. O valor 0.16 encaixa-se entre os valores 0.0 e 0.2 dos qualificadores simbólicos. Dessa forma, o qualificador simbólico da alegação é "Certamente não". Este valor simbólico poderia ter sido outro, caso as entradas (dados) fossem diferentes, por exemplo:

- Entrada "Não está chovendo" no lugar da entrada "Está chovendo": O qualificador simbólico seria "Talvez", uma vez que, com a troca por esse novo dado, os pesos de garantias e refutações seriam similares.

Neste estudo de caso, ao final do processo executado pela ação interna *toulmin.qualifier* () (Figura 18, linha 23), a alegação "Certamente não ir ao parque" é gerada e torna-se uma nova crença na base de conhecimento do agente.

```

01
02 data("O parque não é seguro").
03 data("O parque não é seguro").
04 data("o parque é perto").
05
06 warrant("O parque esta aberto", 0.4).
07 warrant("O parque esta aberto", 0.4).
08 warrant("O parque é seguro", 0.5).
09 warrant("Não esta chovendo", 0.4).
10 warrant("O parque é perto", 0.2)].
11
12 rebuttal("O parque não esta aberto", 1.0).
13 rebuttal("O parque não é seguro", 0.7).
14 rebuttal("Esta chovendo", 0.5).
15 rebuttal("O parque é longe", 0.5).
16
17 !irParque.
18
19 +!irParque : true
20 <- .findall(X, data(X), ListData);
21   .findall(Y, warrant(Y), ListWarrant);
22   .findall(Z, rebuttal(Z), ListRebuttal);
23   .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
24                       Qualifier);
25   +claim(Qualifier, "Irei ao parque").

```

Figura 18: Estudo de Caso 2 - Ir ao parque em *AgentSpeak(L)*

5.3 Estudo de Caso 3 - Comprar passagem para conferência em Paris

Este exemplo foi adaptado de (PANISSON et al., 2014), e apresenta como objetivo comprar uma passagem para uma conferência em Paris. O exemplo contextualiza o envio de um artigo para conferência AAMAS e que pelo fato do agente acreditar que o artigo será aceito, o mesmo fica na dúvida se compra ou não uma passagem para Paris, local onde será realizada a conferência.

Adaptando o exemplo para a estrutura da arquitetura proposta neste trabalho, tem-se as seguintes crenças de dados, garantias e refutações (Tabela 7).

A Figura 19 apresenta o modelo desenvolvido para este exemplo em *AgentSpeak(L)*.

Na Tabela 7 e na Figura 19 percebe-se que existem dois dados contidos nas garantias que apoiam e um dado contido nas refutações que refuta a alegação "comprar passagem para Paris". Neste exemplo, não existem dados que possuem garantias ou refutações de peso máximo. Dessa forma, a função qualificador utilizada as equações 1 e 2:

$$SPG = (0.7 + 0.6) = 1.3$$

$$SPR = (0.3) = 0.3$$

Tabela 7: Componentes Estudo de Caso 3 - Comprar passagem para conferência em Paris

Componente	Informações
Dados	<ul style="list-style-type: none"> • Artigo aceito. • Auxílio integral para a viagem. • Artigo com correções.
Garantias	<ul style="list-style-type: none"> • Artigo aceito, 0.7. • Auxílio Integral para viagem, 0.6. • Auxílio Parcial para viagem, 0.3.
Refutação	<ul style="list-style-type: none"> • Artigo não aceito, 1.0. • Artigo submetido de forma incorreta, 1.0. • Sem auxílio, 0.5. • Artigo com correções, 0.3.

Posteriormente, aplica-se a equação 3 com os valores de SPG e SPR normalizados:

$$Q = (1.0 - 0.23) = 0.77$$

Após o cálculo da equação 3, é gerado o valor Q_{Final} , que é o quantificador de verdade que a alegação gerada possui, baseado na condição apresentada na seção 4.

$$SPG > SPR \rightarrow Q_{Final} = |0.77| = 0.77$$

O qualificador gerado possui o valor 0.77, uma vez que, o valor das garantias que servem como apoiadores da alegação apresentaram um valor maior. O valor 0.77 encaixa-se entre os valores 0.6 e 0.8 dos qualificadores simbólicos. Dessa forma, o qualificador simbólico da alegação é "Provavelmente Sim".

O valor simbólico gerado poderia ter sido outro, caso as crenças dados fossem diferentes, por exemplo:

- Entrada "Artigo não aceito" ou "Artigo submetido de forma incorreta": O qualificador simbólico seria "Certamente não", visto que, os dados "Artigo não aceito" e "Artigo submetido de forma incorreta" são compatíveis com refutações de peso máximo (1.0).

Neste estudo de caso, a alegação "Provavelmente sim, comprar passagem para Paris" é gerada e torna-se uma nova crença na base de conhecimento do agente.

```

01
02 data("Artigo aceito").
03 data("Auxílio Integral para viagem").
04 data("Artigo com correções").
05
06 warrant("Artigo aceito", 0.7).
07 warrant("Auxílio Integral para viagem", 0.6).
08 warrant("Auxílio parcial para viagem", 0.3).
09
10 rebuttal("Artigo não aceito", 1.0).
11 rebuttal("Artigo submetido de forma incorreta", 1.0).
12 rebuttal("Artigo com correções", 0.3).
13
14 !comprarPassagemParis.
15
16 +!comprarPassagemParis : true
17 <- .findall(X, data(X), ListData);
18     .findall(Y, warrant(Y), ListWarrant);
19     .findall(Z, rebuttal(Z), ListRebuttal);
20     .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
21                         Qualifier);
21     +claim(Qualifier, "Comprar passagem para Paris").
22

```

Figura 19: Estudo de Caso 3 - Comprar passagem para conferência em Paris em *AgentSpeak(L)*.

5.4 Estudo de Caso 4 - Classificação de Risco de Paciente com Dengue

Este estudo de caso aborda a classificação de um paciente em nível de gravidade de Dengue. O objetivo da classificação de risco de um paciente com Dengue é reduzir o tempo de espera em um serviço de saúde. Para a classificação de risco foram utilizados os critérios da Política Nacional de Humanização do Ministério da Saúde e o estadiamento da doença ¹. A Tabela 8 apresenta a classificação de risco de acordo com os sinais e sintomas.

Este estudo de caso baseou-se no documento criado pelo Ministério da Saúde - Departamento de Vigilância de Doenças Transmissíveis (DOENÇAS TRANSMISSÍVEIS, 2013), que indica que um paciente, ao chegar em um hospital, pode ser classificado em quatro níveis de dengue: A, B, C ou D.

Segundo o documento DOENÇAS TRANSMISSÍVEIS (2013), os tipos de dengue são caracterizados por apresentar os seguintes sintomas em cada grupo:

¹Estadiamento é o termo usado avaliar a extensão de uma doença em relação ao órgão de origem, isto é, avaliar qual o grau de comprometimento do órgão. Muito usado na área de oncologia.

Tabela 8: Classificação de risco de acordo com os sinais e sintomas

Classificação de risco de acordo com os sinais e sintomas	
Azul - Grupo A	Atendimento de acordo com o horário de chegada
Verde - Grupo B	Prioridade não-urgente
Amarelo - Grupo C	Urgência, atendimento o mais rápido possível
Vermelho - Grupo D	Emergência, paciente com necessidade de atendimento imediato

- Grupo A

- Febre por até sete dias, acompanhada de pelo menos dois sinais e sintomas inespecíficos (cefaléia, prostração, dor retro-orbitária, exantema, mialgias e artralgias), e história epidemiológica compatível; lembrar que, nos lactentes, alguma irritabilidade e choro persistente podem ser a expressão de sintomas como cefaléia e alergias.
- Ausência de sinais de alarme.
- Prova do laço negativo e ausência de manifestações hemorrágicas espontâneas.
- Sem comorbidades, grupo de risco ou condições clínicas especiais.

- Grupo B

- Febre por até sete dias, acompanhada de pelo menos dois sinais e sintomas inespecíficos (cefaléia, prostração, dor retro-orbitária, exantema, mialgias, artralgias) e história epidemiológica compatível.
- Ausência de sinais de alarme.
- Com sangramento de pele espontâneo (petéquias) ou induzido (prova do laço +).

- Condições clínicas especiais e/ou de risco social ou comorbidades: lactentes (menores de 2 anos), gestantes, adultos com idade acima de 65 anos, com hipertensão arterial ou outras doenças cardiovasculares graves, diabetes mellitus, DPOC, doenças hematológicas crônicas (principalmente anemia falciforme e púrpuras), doença renal crônica, doença ácido péptica, hepatopatias e doenças auto-imunes.

- Grupo C

- Febre por até sete dias, acompanhada de pelo menos dois sinais e sintomas inespecíficos (cefaléia, prostração, dor retro-orbitária, exantema, mialgias, artralgias) e história epidemiológica compatível.
- Presença de algum sinal de alarme.
 - * dor abdominal intensa e contínua;
 - * vômitos persistentes;
 - * hipotensão postural e/ou lipotímia;
 - * hepatomegalia dolorosa;
 - * sangramento de mucosa ou hemorragias importantes (hematêmese e/ou melena);
 - * sonolência e/ou irritabilidade;
 - * diminuição da diurese;
 - * diminuição repentina da temperatura
 - * corpórea ou hipotermia;
 - * aumento repentino do hematócrito;
 - * queda abrupta de plaquetas;
 - * desconforto respiratório
- Manifestações hemorrágicas presentes ou ausentes

- Grupo D

- Febre por até sete dias, acompanhada de pelo menos dois sinais e sintomas inespecíficos (cefaléia, prostração, dor retro-orbitária, exantema, mialgias, artralgias) e história epidemiológica compatível.
- Presença de sinais de choque, desconforto respiratório ou disfunção grave de órgãos.
 - * choque;
 - * pressão arterial convergente (PA diferencial $\leq 20mmHg$);
 - * extremidades frias, cianose;

- * pulso rápido e fino;
 - * enchimento capilar lento (> 2 segundos)
 - * hipotensão arterial.
- Manifestações hemorrágicas presentes ou ausentes.

5.4.1 Modelagem do Estudo de Caso

Este estudo de caso é mais complexo que os anteriores, visto que, o mesmo utiliza um número maior de informações e gera mais de uma alegação, pois ele analisa todos os grupos de dengue e gera um qualificador simbólico para cada grupo.

As Tabelas 9, 10, 11 e 12 descrevem a modelagem do estudo de caso, adaptando as informações contidas em cada grupo como garantias e refutações. Por exemplo, as garantias do grupo A, que não são garantias do grupo C, torna-se refutações no grupo C.

Neste estudo de caso, existem quatro planos, onde cada plano irá gerar uma alegação com um qualificador de força para cada grupo da doença (Dengue A, B, C e D). A estrutura deste exemplo na linguagem *AgentSpeak(L)* pode ser visualizada na Figura 20.

5.4.2 Possíveis Resultados Gerados

Nesta subseção serão apresentados alguns dos possíveis resultados que podem ser gerados pelos planos $+!dengueGrupo_A$, $+!dengueGrupo_B$, $+!dengueGrupo_C$ e $+!dengueGrupo_D$.

Um exemplo das informações contidas na Tabela 9, adaptadas a arquitetura implementada em *AgentSpeak(L)* (plano $+!dengueGrupo_A$), pode ser visualizado na Figura 21. As informações contidas nas Tabelas restantes (10, 11 e 12) seguem o mesmo formato da Figura 21 para adaptação a arquitetura proposta.

A seguir, são descritos alguns exemplos quando informações de entrada específicas são informadas, e assim, são geradas saídas (quantificadores simbólicos) distintos. Apenas no último exemplo, os cálculos completos da função qualificador são apresentados.

- Dados de Entrada: "Febre por até sete dias + dois sintomas", "Prova do laço negativo e ausência de manifestações hemorrágicas espontâneas":
 - Grupo A: "Certamente Sim", uma vez que, os dados de entrada são compatíveis com as garantias deste grupo e não existiam dados de refutação que corroborem com as entradas.
 - Grupo B: "Provavelmente não", uma vez que contém algumas informações compatíveis com suas garantias, porém o peso das refutações é mais alto.
 - Grupo C: "Certamente não", uma vez que os dados de entrada tem um peso muito forte nas refutações.

Tabela 9: Garantias e Refutações - Grupo A

Grupo A	
Garantias	<ul style="list-style-type: none"> ● Febre por até sete dias + cefaleia e/ou prostração e/ou dor retro-orbitária e/ou exantema e/ou mialgias e/ou artralguas, 0.3 ● Ausência de sinais de alarme, 0.8 ● Prova do laço negativo e ausência de manifestações hemorrágicas espontâneas, 0.8 ● Sem comorbidades, grupo de risco ou condições clínicas especiais, 0.9
Refutações	<ul style="list-style-type: none"> ● Sangramento de Pele ou induzido por prova de laço, 0.9 ● Condições clínicas especiais, 0.8 ● Presença de algum sinal de alarme: dor abdominal intensa e contínua; vômitos persistentes; hipotensão postural e/ou lipotímia; hepatomegalia dolorosa; sangramento de mucosa ou hemorragias; importantes (hematêmese e/ou melena); sonolência e/ou irritabilidade; diminuição da diurese; diminuição repentina da temperatura; corpórea ou hipotermia; aumento repentino do hematócrito; queda abrupta de plaquetas; desconforto respiratório, 1.0 ● Presença de sinais de choque, desconforto respiratório ou disfunção grave de órgãos: choque; pressão arterial convergente (PA diferencial $\leq 20mmHg$); extremidades frias, cianose; pulso rápido e fino; enchimento capilar lento (> 2 segundos); hipotensão arterial, 1.0 ● Manifestações hemorrágicas presentes ou ausentes, 1.0

- Grupo D: "Certamente não", uma vez que os dados de entrada tem um peso muito forte nas refutações.
- Dados de Entrada: quaisquer sintomas referentes "Presença de algum sinal de alarme" (ver descrição completa na definição de cada grupo da dengue):
 - Grupo A: "Certamente não", uma vez que os dados de entrada são compatíveis com um peso elevado nas refutações.

Tabela 10: Garantias e Refutações - Grupo B

Grupo B	
Garantias	<ul style="list-style-type: none"> ● Febre por até sete dias + cefaleia e/ou prostração e/ou dor retro-orbitária e/ou exantema e/ou mialgias e/ou artralguas, 0.4 ● Sangramento de Pele ou induzido por prova de laço, 0.8 ● Condições clínicas especiais, 0.9
Refutações	<ul style="list-style-type: none"> ● Ausência de sinais de alarme. 0.9 ● Prova do laço negativo e ausência de manifestações hemorrágicas espontâneas., 0.8 ● Sem comorbidades, grupo de risco ou condições clínicas especiais, 0.8 ● Presença de algum sinal de alarme: dor abdominal intensa e contínua; vômitos persistentes; hipotensão postural e/ou lipotímia; hepatomegalia dolorosa; sangramento de mucosa ou hemorragias; importantes (hematêmese e/ou melena); sonolência e/ou irritabilidade; diminuição da diurese; diminuição repentina da temperatura; corpórea ou hipotermia; aumento repentino do hematócrito; queda abrupta de plaquetas; desconforto respiratório, 1.0 ● Presença de sinais de choque, desconforto respiratório ou disfunção grave de órgãos: choque; pressão arterial convergente (PA diferencial $\leq 20mmHg$); extremidades frias, cianose; pulso rápido e fino; enchimento capilar lento (> 2 segundos); hipotensão arterial, 1.0 ● Manifestações hemorrágicas presentes ou ausentes, 1.0

- Grupo B: "Certamente não", uma vez que os dados de entrada são compatíveis com um peso elevado nas refutações.
- Grupo C: "Certamente Sim", uma vez que os dados de entrada tem um peso muito forte nas garantias.
- Grupo D: "Certamente não", uma vez que os dados de entrada tem um peso muito forte nas refutações.

Tabela 11: Garantias e Refutações - Grupo C

Grupo C	
Garantias	<ul style="list-style-type: none"> ● Febre por até sete dias + cefaleia e/ou prostração e/ou dor retro-orbitária e/ou exantema e/ou mialgias e/ou artralguas, 0.2 ● Presença de algum sinal de alarme: dor abdominal intensa e contínua; vômitos persistentes; hipotensão postural e/ou lipotímia; hepatomegalia dolorosa; sangramento de mucosa ou hemorragias; importantes (hematêmese e/ou melena); sonolência e/ou irritabilidade; diminuição da diurese; diminuição repentina da temperatura; corpórea ou hipotermia; aumento repentino do hematócrito; queda abrupta de plaquetas; desconforto respiratório, 0.9 ● Manifestações hemorrágicas presentes ou ausentes, 0.3
Refutações	<ul style="list-style-type: none"> ● Ausência de sinais de alarme., 1.0 ● Prova do laço negativo e ausência de manifestações hemorrágicas espontâneas, 1.0 ● Sem comorbidades, grupo de risco ou condições clínicas especiais, 1.0 ● Sangramento de Pele ou induzido por prova de laço 0.8 ● Condições clínicas especiais 0.8 ● Presença de sinais de choque, desconforto respiratório ou disfunção grave de órgãos: choque; pressão arterial convergente (PA diferencial $\leq 20mmHg$); extremidades frias, cianose; pulso rápido e fino; enchimento capilar lento (> 2 segundos); hipotensão arterial. 1.0

- Dados de Entrada: quaisquer sintomas referentes "Presença de sinais de choque" (ver descrição completa na definição de cada grupo da dengue):
 - Grupo A: "Certamente não", uma vez que os dados de entrada são compatíveis com um peso elevado nas refutações.
 - Grupo B: "Certamente não", uma vez que os dados de entrada são compatíveis

Tabela 12: Garantias e Refutações - Grupo D

Grupo D	
Garantias	<ul style="list-style-type: none"> ● Febre por até sete dias + cefaleia e/ou prostração e/ou dor retro-orbitária e/ou exantema e/ou mialgias e/ou artralgias), 0.2 ● Presença de sinais de choque, desconforto respiratório ou disfunção grave de órgãos: choque; pressão arterial convergente (PA diferencial $\leq 20mmHg$); extremidades frias, cianose; pulso rápido e fino; enchimento capilar lento (> 2 segundos); hipotensão arterial. 0.9 ● Manifestações hemorrágicas presentes ou ausentes, 0.3
Refutações	<ul style="list-style-type: none"> ● Ausência de sinais de alarme, 1.0 ● Prova do laço negativo e ausência de manifestações hemorrágicas espontâneas, 1.0 ● Sem comorbidades, grupo de risco ou condições clínicas especiais, 1.0 ● Sangramento de Pele ou induzido por prova de laço, 0.8 ● Condições clínicas especiais, 0.8 ● Presença de algum sinal de alarme: dor abdominal intensa e contínua; vômitos persistentes; hipotensão postural e/ou lipotímia; hepatomegalia dolorosa; sangramento de mucosa ou hemorragias; importantes (hematêmese e/ou melena); sonolência e/ou irritabilidade; diminuição da diurese; diminuição repentina da temperatura; corpórea ou hipotermia; aumento repentino do hematócrito; queda abrupta de plaquetas; desconforto respiratório, 0.9

com um peso elevado nas refutações.

– Grupo C: "Certamente não", uma vez que os dados de entrada tem um peso muito forte nas refutações.

– Grupo D: "Certamente sim", uma vez que os dados de entrada tem um peso muito forte nas garantias.

- Entrada com os seguintes sintomas: "Febre por até sete dias + cefaleia e/ou prostra-

```

...
10 +!dengueGrupo_A : true
11 <- .findall(X, data(X), ListData);
11     .findall(Y, warrantDengueGrupo_A(Y), ListWarrant);
13     .findall(Z, rebuttalDengueGrupo_A(Z), ListRebuttal);
14     .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
                          Qualifier);
15     +claim(Qualifier", "Dengue Grupo A")
16
17 +!dengueGrupo_B : true
18 <- .findall(X, data(X), ListData);
19     .findall(Y, warrantDengueGrupo_B(Y), ListWarrant);
20     .findall(Z, rebuttalDengueGrupo_B(Z), ListRebuttal);
21     .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
                          Qualifier);
22     +claim(Qualifier", "Dengue Grupo B")
23
24 +!dengueGrupo_C : true
25 <- .findall(X, data(X), ListData);
26     .findall(Y, warrantDengueGrupo_C(Y), ListWarrant);
27     .findall(Z, rebuttalDengueGrupo_C(Z), ListRebuttal);
28     .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
                          Qualifier);
29     +claim(Qualifier", "Dengue Grupo C").
30
31 +!dengueGrupo_D : true
32 <- .findall(X, data(X), ListData);
33     .findall(Y, warrantDengueGrupo_D(Y), ListWarrant);
34     .findall(Z, rebuttalDengueGrupo_D(Z), ListRebuttal);
35     .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
                          Qualifier);
36     +claim(Qualifier", "Dengue Grupo D").
...

```

Figura 20: Estudo de Caso 4 - Diagnóstico da Dengue em AgentSpeak(L)

ção e/ou dor retro-orbitária e/ou exantema e/ou mialgias e/ou artralgias”, ”Presença de algum sinal de alarme” e ”Manifestações hemorrágicas presentes ou ausentes”.

Com as entradas apresentadas, cada plano terá as seguintes saídas:

– Grupo A:

Inicialmente a função verifica se existem dados de entrada compatíveis com garantias e/ou refutações de peso máximo (1.0). Neste grupo, existe um dado compatível com uma refutação de peso 1.0, logo a seguinte condição é executada:

```

...
02 warrantDengueGrupo_A(Febre por até sete dias + dois
      sintomas", 0.2).
03 warrantDengueGrupo_A("Ausência de sinais de alarme", 0.2).
04 warrantDengueGrupo_A("Prova do laço negativo e ausência de
      espontâneas, 0.4").
05 warrantDengueGrupo_A("Sem comorbidades, grupo de risco ou
      condições clínicas especiais", 1.0).
06
07 rebuttalDengueGrupo_A("Sangramento de pele ou induzido por
      prova de laço", 0.4).
08 rebuttalDengueGrupo_A("Condições clínicas especiais", 0.4).
09 rebuttalDengueGrupo_A("Presença de algum sinal de alarme",
      1.0).
10 rebuttalDengueGrupo_A("Presença de sinais de choque", 1.0).
11 rebuttalDengueGrupo_A("Manifestações hemorrágicas presentes
      ou ausentes", 1.0).
12
...
30 +!dengueGrupo_A : true
31 <- .findall(X, data(X), ListData);
32   .findall(Y, warrantDengueGrupo_A(Y), ListWarrant);
33   .findall(Z, rebuttalDengueGrupo_A(Z), ListRebuttal);
34   .toulmin.qualifier(ListData, ListWarrant, ListRebuttal,
      Qualifier);
35   +claim(Qualifier", "Dengue Grupo A")
...

```

Figura 21: Parte do código do Plano $+!DengueGrupo_A$ em AgentSpeak(L)

$$Q_{Final} = \begin{cases} 0 & \text{Se } \exists PRV_i = 1.0 \end{cases}$$

Portanto, a alegação gerada com seu respectivo qualificador é "Certamente não Grupo A".

– Grupo B:

Inicialmente, a função verifica se existem dados de entrada compatíveis com garantias e/ou refutações de peso máximo (1.0). Neste grupo, também existe um dado compatível com uma refutação de peso 1.0, logo a seguinte condição é executada:

$$Q_{Final} = \begin{cases} 0 & \text{Se } \exists PRV_i = 1.0 \end{cases}$$

Portanto, a alegação gerada com seu respectivo qualificador é "Certamente não Grupo B".

– Grupo C:

Inicialmente, a função verifica se existem dados de entrada compatíveis com garantias e/ou refutações de peso máximo (1.0). Neste grupo, não existem garantias ou refutações de peso máximo. Logo, função qualificador utiliza as equações 1 e 2:

$$SPG = (0.2 + 0.9 + 0.3) = 1.4$$

$$SPR = (0.0) = 0.0$$

Posteriormente, aplica-se a equação 3 com os valores de SPG e SPR normalizados:

$$Q = (1.0 - 0.0) = 1.0$$

Após o cálculo da equação 3, é gerado o valor Q_{Final} , que é o quantificador de verdade que a alegação gerada possui, baseado na condição apresentada na seção 4.

$$SPG > SPR \rightarrow Q_{Final} = |1.0| = 1.0$$

O qualificador gerado possui o valor 1.0, uma vez que, o valor das garantias que servem como apoiadores da alegação apresentaram um valor maior. O valor 1.0 encaixa-se entre os valores 0.8 e 1.0 dos qualificadores simbólicos. Dessa forma, o qualificador simbólico da alegação é "Certamente sim". Portanto, a alegação gerada com seu respectivo qualificador é "Certamente sim Grupo C".

– Grupo D:

Inicialmente, a função verifica se existem dados de entrada compatíveis com garantias e/ou refutações de peso máximo (1.0). Neste grupo, não existem garantias nem refutações de peso máximo. Logo, função qualificador utiliza as equações 1 e 2:

$$SPG = (0.2 + 0.3) = 0.5$$

$$SPR = (0.9) = 0.9$$

Posteriormente, aplica-se a equação 3 com os valores de SPG e SPR normalizados:

$$Q = (0.55 - 1.0) = -0.45$$

Após o cálculo da equação 3, é gerado o valor Q_{Final} , que é o quantificador de verdade que a alegação gerada possui, baseado na condição apresentada na seção 4.

$$SPG > SPR \rightarrow Q_{Final} = 1 - |-0.45| = 0.55$$

O qualificador gerado possui o valor 0.55, uma vez que, o valor das garantias e refutações apresentaram um valor próximo. O valor 0.55 encaixa-se entre os valores 0.4 e 0.6 dos qualificadores simbólicos. Dessa forma, o qualificador simbólico da alegação é "Talvez". Portanto, a alegação gerada com seu respectivo qualificador é "Talvez Grupo D".

Este estudo de caso pode gerar "N" outras alegações com "N" outros qualificadores, dependendo apenas dos dados de entrada.

5.5 Considerações Finais do Capítulo

Esta seção descreveu quatro estudos de caso. O primeiro foi adaptado de TOULMIN (2003), o qual tem o objetivo de verificar a nacionalidade de um indivíduo. O segundo analisa se um indivíduo deve ir ao parque. O terceiro foi adaptado de (PANISSON, 2015) e aborda a compra de uma passagem para uma conferência em Paris. Por fim, o quarto estudo de caso abordou a classificação de um paciente em grupo de risco referente a sintomas de dengue.

Estes estudos apresentaram diferentes situações, onde foi possível analisar o funcionamento da arquitetura e seu comportamento em cada situação. Ao final dos estudos de caso, tornou-se possível comprovar sua adaptabilidade em distintas situações. No próximo capítulo, dedicado a apresentar a conclusão deste trabalho, serão descritas as contribuições relaziadas neste trabalho e sugestões de trabalhos futuros.

6 CONCLUSÃO

Sistemas Multiagente são um conjunto de agentes autônomos, que atuam de forma organizada em um ambiente compartilhado, colaborando entre si e buscando de uma forma conjunta atingir um objetivo satisfatório para o grupo.

Em sistemas multiagente, existem distintas arquiteturas que tem a finalidade de especificar a estrutura interna de um agente, neste trabalho foi utilizada a arquitetura racional chamada BDI. A arquitetura BDI é uma das arquiteturas mais conhecidas para modelar o comportamento de agentes racionais e possibilita construir um agente com algumas atitudes mentais, sendo estas: crenças, desejos e intenções.

Prover técnicas e métodos para proporcionar o raciocínio dos agentes, evitando conflito de crenças e possibilitando tomadas de decisões bem fundamentadas é muito importante. Neste contexto, surge a argumentação como uma ferramenta que possibilita os agentes utilizarem um raciocínio baseado em argumentação para tomarem decisões e reduzirem conflitos de informações.

A teoria da argumentação abrange diversas áreas do conhecimento e tem ganhado um espaço significativo na comunidade de sistemas multiagente, uma vez que, pode possibilitar o agente a raciocinar sobre suas decisões próprias, decisões adquiridas através de percepções sobre o ambiente ou através da comunicação com outros agentes.

Toulmin, no campo da filosofia, desenvolveu um modelo, o qual indica seis componentes para montar a estrutura de um argumento, sendo eles: dados, garantias, alegação, apoio, qualificador e refutação.

O objetivo central deste estudo foi o desenvolvimento de uma arquitetura para inferência baseada em argumentos. A arquitetura desenvolvida neste trabalho teve sua fundamentação sobre o modelo apresentado por Toulmin, o qual é caracterizado por apresentar um modelo de argumento alternativo ao modelo racional tradicional.

A arquitetura desenvolvida possibilita não só a geração de alegações, mas também, quantificar o peso dessa alegação como uma verdade. Dessa forma, torna-se possível examinar porque uma determinada alegação é falsa ou verdadeira.

A arquitetura possibilita analisar e justificar uma alegação, uma vez que, a arquitetura é composta por componentes explicitamente separados. Além de possibilitar a represen-

tação de incertezas, visto que, a arquitetura gera alegações com quantificadores.

A arquitetura desenvolvida torna-se robusta pelo fato de possibilitar a implementação de diferentes funções qualificadoras, as quais podem ser desenvolvidas com características específicas para cada situação.

A arquitetura foi implementada na linguagem orientada a agentes *AgentSpeak(L)* disponível na plataforma Jason. Dessa forma, tornando possível realizar estudos de casos práticos para avaliar o comportamento da arquitetura desenvolvida em distintas situações.

Para implementar o modelo foi necessário criar um plano, o qual possui uma ação interna disponível na plataforma Jason para buscar as crenças dados, garantias, refutações e colocar em listas. Ainda, foi necessário desenvolver uma ação interna denominada *Qualifier* que recebe como entrada as listas e realiza uma busca que verifica se existem dados nas listas de dados compatíveis com garantias e refutações contidas nas listas de garantias e refutações. Ao final dessa busca, a função realiza um cálculo com os pesos das crenças de garantias e refutações que foram compatíveis com alguma crença de dado e gera uma alegação com o seu quantificador simbólico.

Ao final da implementação da arquitetura, foram realizados quatro estudos de caso, com temáticas distintas, sendo elas: a descoberta da nacionalidade de um indivíduo; um indivíduo decidir se vai ao parque ou não; tomar a decisão de comprar ou não uma passagem para uma conferência em Paris; e diagnosticar um paciente com um tipo de Dengue.

Futuramente, como mencionado na Seção 4, a função de qualificação pode ser otimizada acrescentando novas funções, mais complexas ou específicas para cada domínio, podendo criar novas ações internas, com novos parâmetros, permitindo escolher qual função mais adequada para o cálculo do qualificador.

A arquitetura implementada já possibilita aos usuários inserirem dados (crenças) para alimentar a base de informações que a arquitetura utiliza para comparar com as garantias e refutações. Porém, torna-se interessante melhorar graficamente a interface de inserção dos dados para deixar o *layout* visualmente mais convidativo.

Além das melhorias citadas, futuramente seria interessante realizar análises comparativas usando um mesmo exemplo, com os modelos para o raciocínio por argumentos implementados por BERARIU (2014) e por PANISSON et al. (2014). Estas análises podem ser realizadas sobre os resultados de saída dos modelos implementados e sobre as dificuldades para a modelagem do exemplo. Pode ser feita também uma avaliação através de um questionário, tendo com público-alvo usuários especialistas sem conhecimento sobre programação, buscando analisar sua aceitação sobre os modelos.

REFERÊNCIAS

ALVARES, L. O.; SICHMAN, J. S. Introdução aos sistemas multiagentes. In: XVII CONGRESSO DA SBC-ANAIS JAI'97, 1997. **Anais...** [S.l.: s.n.], 1997.

AMANDI, A. **Programação de Agentes Orientada a Objetos. Tese de Doutorado. Universidade Federal do Rio Grande do Sul.** Porto Alegre, RS, Brasil: [s.n.], 1997.

AMGOUD, L.; CAYROL, C. A reasoning model based on the production of acceptable arguments. **Annals of Mathematics and Artificial Intelligence**, [S.l.], v.34, n.1, p.197–215, 2002.

AMGOUD, L.; MAUDET, N.; PARSONS, S. Modelling dialogues using argumentation. In: MULTIAGENT SYSTEMS, 2000. PROCEEDINGS. FOURTH INTERNATIONAL CONFERENCE ON, 2000. **Anais...** [S.l.: s.n.], 2000. p.31–38.

AMGOUD, L.; PARSONS, S.; MAUDET, N. Arguments, dialogue, and negotiation. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 14., 2000. **Proceedings...** [S.l.: s.n.], 2000. p.338–342.

ATKINSON, K.; BENCH-CAPON, T. Practical reasoning as presumptive argumentation using action based alternating transition systems. **Artificial Intelligence**, [S.l.], v.171, n.10-15, p.855–874, 2007.

ATKINSON, K.; BENCH-CAPON, T.; MCBURNEY, P. A dialogue game protocol for multi-agent argument over proposals for action. **Autonomous Agents and Multi-Agent Systems**, [S.l.], v.11, n.2, p.153–171, 2005.

ATKINSON, K.; BENCH-CAPON, T.; MCBURNEY, P. Generating intentions through argumentation. In: AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 2005. **Proceedings...** [S.l.: s.n.], 2005. p.1261–1262.

BENCH-CAPON, T. J. Specification and implementation of Toulmin dialogue game. In: JURIX, 1998. **Proceedings...** [S.l.: s.n.], 1998. v.98, p.5–20.

BERARIU, T. An argumentation framework for bdi agents. In: **Intelligent Distributed Computing VII**. [S.l.]: Springer, 2014. p.343–354.

BIRNBAUM, L.; FLOWERS, M.; MCGUIRE, R. Towards an AI model of argumentation. In: FIRST AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1980. **Proceedings...** [S.l.: s.n.], 1980. p.313–315.

BONDARENKO, A.; DUNG, P. M.; KOWALSKI, R. A.; TONI, F. An abstract, argumentation-theoretic approach to default reasoning. **Artificial intelligence**, [S.l.], v.93, n.1-2, p.63–101, 1997.

BORDINI, R. H.; HÜBNER, J. F.; WOOLDRIDGE, M. **Programming multi-agent systems in AgentSpeak using Jason**. [S.l.]: John Wiley & Sons, 2007. v.8.

BORDINI, R. H.; VIEIRA, R. Linguagens de Programação Orientadas a Agentes: uma introdução baseada em AgentSpeak (L). **Revista de informática teórica e aplicada. Porto Alegre. Vol. 10, n. 1 (2003), p. 7-38**, [S.l.], 2003.

BRATMAN, M. **Intention, plans, and practical reason**. [S.l.: s.n.], 1987.

DOENÇAS TRANSMISSÍVEIS, M. da Saúde (BR). Secretaria de Vigilância em Saúde. Departamento de Vigilância das. **Dengue: diagnóstico e manejo clínico: adulto e criança**. [S.l.]: Ministério da Saúde Brasília, 2013.

DUNG, P. M. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. **Artificial intelligence**, [S.l.], v.77, n.2, p.321–357, 1995.

FERBER, J.; GASSER, L. **Intelligence artificielle distribuée**. France: Tutorial notes of the 11th conference on expert systems and their applications, 1991.

FLOWERS, M.; MCGUIRE, R.; BIRNBAUM, L. Adversary arguments and the logic of personal attacks. **Strategies for natural language processing**, [S.l.], p.275–294, 1982.

FRANKLIN, S.; GRAESSER, A. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. **Intelligent agents III agent theories, architectures, and languages**, [S.l.], p.21–35, 1997.

GORDON, T. F.; KARACAPILIDIS, N. The Zeno argumentation framework. In: ARTIFICIAL INTELLIGENCE AND LAW, 6., 1997. **Proceedings...** [S.l.: s.n.], 1997. p.10–18.

HITCHCOCK, D. Good reasoning on the Toulmin model. **Argumentation**, [S.l.], v.19, n.3, p.373–391, 2005.

HÜBNER, J. F.; BORDINI, R. H.; VIEIRA, R. Introdução ao desenvolvimento de sistemas multiagentes com Jason. **XII Escola de Informática da SBC**, [S.l.], v.2, p.51–89, 2004.

KAKAS, A.; MORAITIS, P. Argumentation based decision making for autonomous agents. In: AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 2003. **Proceedings...** [S.l.: s.n.], 2003. p.883–890.

KRAUS, S.; SYCARA, K.; EVENCHIK, A. Reaching agreements through argumentation: a logical model and implementation. **Artificial Intelligence**, [S.l.], v.104, n.1-2, p.1–69, 1998.

KRAUSE, P.; AMBLER, S.; ELVANG-GORANSSON, M.; FOX, J. A logic of argumentation for reasoning under uncertainty. **Computational Intelligence**, [S.l.], v.11, n.1, p.113–131, 1995.

MACHADO, R.; BORDINI, R. H. Running AgentSpeak (L) agents on SIM_AGENT. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, 2001. **Anais...** [S.l.: s.n.], 2001. p.158–174.

MAUDET, N.; PARSONS, S.; RAHWAN, I. Argumentation in multi-agent systems: Context and recent developments. In: INTERNATIONAL WORKSHOP ON ARGUMENTATION IN MULTI-AGENT SYSTEMS, 2006. **Anais...** [S.l.: s.n.], 2006. p.1–16.

MOREIRA, Á. F.; BORDINI, R. H. An operational semantics for a BDI agent-oriented programming language. In: WORKSHOP ON LOGICS FOR AGENT-BASED SYSTEMS (LABS-02), HELD IN CONJUNCTION WITH THE EIGHTH INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING (KR2002), APRIL, 2002. **Proceedings...** [S.l.: s.n.], 2002. v.22, n.25, p.45–59.

MOULIN, B.; IRANDOUST, H.; BÉLANGER, M.; DESBORDES, G. Explanation and argumentation capabilities: Towards the creation of more persuasive agents. **Artificial Intelligence Review**, [S.l.], v.17, n.3, p.169–222, 2002.

NUNES, I. O. de. Implementação do Modelo e da Arquitetura BDI. **Monografias em Ciência da Computação**, [S.l.], v.1, 2007.

OLIVA, E.; MCBURNEY, P.; OMICINI, A. Co-argumentation artifact for agent societies. In: INTERNATIONAL WORKSHOP ON ARGUMENTATION IN MULTI-AGENT SYSTEMS, 2007. **Anais...** [S.l.: s.n.], 2007. p.31–46.

- PANISSON, A. R. **Argumentation-based dialogues for task reallocation among rational agents**. 2015. Dissertação (Mestrado em Ciência da Computação) — Pontifícia Universidade Católica do Rio Grande do Sul.
- PANISSON, A. R.; MENEGUZZI, F.; VIEIRA, R.; BORDINI, R. H. An Approach for Argumentation-based Reasoning Using Defeasible Logic in Multi-Agent Programming Languages. In: INTERNATIONAL WORKSHOP ON ARGUMENTATION IN MULTI-AGENT SYSTEMS, 11., 2014. **Anais...** [S.l.: s.n.], 2014.
- PERELMAN, C. The new rhetoric. In: **Pragmatics of natural languages**. [S.l.]: Springer, 1971. p.145–149.
- PRAKKEN, H. An abstract framework for argumentation with structured arguments. **Argument and Computation**, [S.l.], v.1, n.2, p.93–124, 2010.
- PRAKKEN, H.; SARTOR, G. Argument-based extended logic programming with defeasible priorities. **Journal of applied non-classical logics**, [S.l.], v.7, n.1-2, p.25–75, 1997.
- RAHWAN, I.; RAMCHURN, S. D.; JENNINGS, N. R.; MCBURNEY, P.; PARSONS, S.; SONENBERG, L. Argumentation-based negotiation. **The Knowledge Engineering Review**, [S.l.], v.18, n.4, p.343–375, 2003.
- RAO, A. AgentSpeak (L): BDI agents speak out in a logical computable language. **Agents breaking away**, [S.l.], p.42–55, 1996.
- RAO, A. S.; GEORGEFF, M. P. et al. BDI agents: From theory to practice. In: ICMAS, 1995. **Anais...** [S.l.: s.n.], 1995. v.95, p.312–319.
- REZENDE, S. O. **Sistemas Inteligentes: fundamentos e aplicações**. Barueri,SP: [s.n.], 2003.
- RITTEL, H.; WEBBER, M. **Dilemmas in a General Theory of Planning.. Policy Sciences**4. 2 (1973): 155–69.
- ROTSTEIN, N. D.; GARCÍA, A. J.; SIMARI, G. R. Defeasible argumentation support for an extended BDI architecture. In: INTERNATIONAL WORKSHOP ON ARGUMENTATION IN MULTI-AGENT SYSTEMS, 2007. **Anais...** [S.l.: s.n.], 2007. p.145–163.
- ROWE, G.; REED, C. Argument diagramming: The araucaria project. In: **Knowledge Cartography**. [S.l.]: Springer, 2014. p.173–191.
- RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.: s.n.], 1995.
- SIERRA, C.; JENNINGS, N.; NORIEGA, P.; PARSONS, S. A framework for argumentation-based negotiation. **Intelligent agents IV agent theories, architectures, and languages**, [S.l.], p.177–192, 1998.

- TOULMIN, S. E. **The uses of argument**. [S.l.]: Cambridge university press, 2003.
- VAN EEMEREN, F. H.; GROOTENDORST, R.; JOHNSON, R. H.; PLANTIN, C.; WILLARD, C. A. **Fundamentals of argumentation theory: A handbook of historical backgrounds and contemporary developments**. [S.l.]: Routledge, 2013.
- VELMOVITSKY, P. E.; BRIOT, J.-P.; VIANA, M.; LUCENA, C. J. P. de. Practical reasoning in an argumentation-based decision BDI agent: a case study for participatory management of protected areas. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING & KNOWLEDGE ENGINEERING (SEKE'2017), 29., 2017. **Anais...** [S.l.: s.n.], 2017. p.527–530.
- VERHEIJ, B. Artificial argument assistants for defeasible argumentation. **Artificial intelligence**, [S.l.], v.150, n.1-2, p.291–324, 2003.
- VERHEIJ, B. Evaluating arguments based on Toulmin's scheme. **Argumentation**, [S.l.], v.19, n.3, p.347–371, 2005.
- VERHEIJ, B. **Virtual arguments: on the design of argument assistants for lawyers and other arguers**. [S.l.]: Cambridge University Press, 2005. v.6.
- VREESWIJK, G. A. Abstract argumentation systems. **Artificial intelligence**, [S.l.], v.90, n.1, p.225–279, 1997.
- WALTON, D.; REED, C.; MACAGNO, F. **Argumentation schemes**. [S.l.]: Cambridge University Press, 2008.
- WEISS, G. **Multiagent systems: a modern approach to distributed artificial intelligence**. [S.l.]: MIT press, 1999.
- WOOLDRIDGE, M.; JENNINGS, N. R. Agent theories, architectures, and languages: a survey. In: **Intelligent agents**. [S.l.]: Springer, 1994. p.1–39.
- WOOLDRIDGE, M. **Introduction to Multiagent Systems**. New York, NY, USA: John Wiley & Sons, Inc., 2001.