

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL**

NOME DO AUTOR

TÍTULO EM PORTUGUÊS

DISSERTAÇÃO

CURITIBA

2019

NOME DO AUTOR

TÍTULO EM PORTUGUÊS

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Ciências” – Área de Concentração: Informática Industrial.

Orientador: Nome do Orientador

CURITIBA

2019

Dados Internacionais de Catalogação na Publicação

T137 Sobrenome, Nome

 Título em português/ Nome do Autor. – 2019.

 36 f. : il. ; 30 cm

 Orientador: Nome do Orientador.

 Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. Curitiba, 2019.

 Bibliografia: f. 33-34.

 1. Teoria do controle. 2. Redes de comutação. 3. TCP/IP (Protocolo de rede de computação), ...

CDD (22. ed.) 621.3

Biblioteca xxxxxx

Título da Dissertação Nº 596:

**“Esquema de Controle de Congestionamento para
TCP Baseado na Banda Disponível”.**

por

Marcos Talau

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM CIÊNCIAS – Área de Concentração: Telemática, pelo Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Curitiba, às 09h30min. do dia 04 de maio de 2012. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores:



Visto da coordenação:



Texto da dedicatória.

AGRADECIMENTOS

Texto dos agradecimentos.

Texto da epígrafe.

RESUMO

SOBRENOME, Nome. TÍTULO EM PORTUGUÊS. 36 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

Texto do resumo (máximo de 500 palavras).

Palavras-chave: Palavra-chave 1, Palavra-chave 2, ...

ABSTRACT

SOBRENOME, Nome. TITLE IN ENGLISH. 36 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

Abstract text (maximum of 500 words).

Keywords: Keyword 1, Keyword 2, ...

LISTA DE FIGURAS

| | | |
|----------|---|----|
| FIGURA 1 | – Arvore de objetivos definido pelo modelo Moise (HÜBNER et al., 2002) | 24 |
| FIGURA 2 | – Linguagem para descrever um programa de multiagentes normativos com a possibilidade de violações e sanções na notação EBNF segundo o texto (DASTANI et al., 2009). Nesta notação, $\langle ident \rangle$ é usado para denotar uma <i>string</i> e $\langle int \rangle$ inteiros. Os termos $\langle b-prop \rangle$ e $\langle i-prop \rangle$ são usados para designar dois tipos de conjuntos de proposições que são disjuntos entre si | 26 |
| FIGURA 3 | – Um programa descrito na linguagem proposta neste estudo onde um agente representa um passageiro em uma estação de trem que pode entrar com ou sem um <i>ticket</i> na plataforma e no trem (DASTANI et al., 2009). | 27 |
| FIGURA 4 | – Ontologia que descreve <i>Domain Model</i> no model V3S (BAROT et al., 2013) | 28 |

LISTA DE TABELAS

| | | |
|----------|--|----|
| TABELA 1 | – Construtores da linguagem <i>ACTIVITY-DL</i> (BAROT et al., 2013) | 29 |
| TABELA 2 | – As pré-condições possíveis para as atividades (BAROT et al., 2013) | 30 |

LISTA DE QUADROS

LISTA DE SIGLAS

LISTA DE SÍMBOLOS

SUMÁRIO

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO | 13 |
| 1.1 | MOTIVAÇÃO | 13 |
| 1.2 | RELEVÂNCIA | 13 |
| 1.3 | OBJETIVOS | 13 |
| 1.3.1 | Objetivo Geral | 13 |
| 1.3.2 | Objetivos Específicos | 13 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 14 |
| 2.1 | AGENTES | 14 |
| 2.2 | ARTEFATOS | 16 |
| 2.3 | SISTEMA MULTIAGENTE | 18 |
| 2.4 | NORMAS | 18 |
| 2.5 | RISCOS | 19 |
| 2.6 | POSSIBILIDADES | 19 |
| 3 | METODOLOGIA | 20 |
| 3.1 | ANÁLISE DOS MODELOS | 20 |
| 3.2 | ACOMPANHAMENTO DE PROFISSIONAIS EM ATIVIDADE DE RISCO | 20 |
| 3.3 | CONSTRUÇÃO DO MODELO | 20 |
| 3.4 | IMPLEMENTAÇÃO | 20 |
| 4 | RESULTADOS | 21 |
| 4.1 | ESTRUTURA CONCEITUAL | 21 |
| 4.1.1 | Módulos | 21 |
| 4.1.2 | Conjuntos | 21 |
| 4.1.3 | Predicados | 21 |
| 4.1.4 | Regras | 21 |
| 4.2 | UML | 21 |
| 4.2.1 | Diagrama de Classes | 21 |
| 4.2.2 | Diagrama de Atividades | 21 |
| 4.3 | CASO DE ESTUDO | 21 |
| 4.4 | RACIOCÍNIO | 21 |
| 4.5 | VALIDAÇÃO | 21 |
| 5 | ANÁLISE COMPARATIVA | 22 |
| 5.1 | MOISE+ | 22 |
| 5.1.1 | Estrutura | 22 |
| 5.1.2 | Análise comparativa | 26 |
| 5.2 | DASTANI | 26 |
| 5.2.1 | Estrutura | 26 |
| 5.2.2 | Análise comparativa | 28 |
| 5.3 | V3S | 28 |
| 5.3.1 | Estrutura | 28 |
| 5.3.2 | Análise comparativa | 30 |
| 5.4 | NORMMAS | 30 |

| | |
|--|-----------|
| 5.4.1 Estrutura | 30 |
| 5.4.2 Análise comparativa | 31 |
| 5.5 PERSPECTIVA GENÉRICA | 31 |
| 6 CONCLUSÃO | 32 |
| 6.1 AVALIAÇÃO DOS OBJETIVOS | 32 |
| 6.2 TRABALHOS FUTUROS | 32 |
| REFERÊNCIAS | 33 |
| Apêndice A – NOME DO APÊNDICE | 35 |
| Anexo A – NOME DO ANEXO | 36 |

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

1.2 RELEVÂNCIA

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

1.3.2 OBJETIVOS ESPECÍFICOS

2 FUNDAMENTAÇÃO TEÓRICA

2.1 AGENTES

Não existe uma definição universal para tratar o conceito de agente sendo que esse tópico se encontra em meio a debates e controvérsias. Contudo, existe um entendimento generalizado de que um comportamento *autonomo* é o cerne de noção que se tem por agente (JENNINGS; LESPÉRANCE, 2000). Apesar disso, a construção de um modelo computacional não pode ser feito sem uma definição. Assim sendo, nesse texto um agente é um sistema computacional que está situado em um dado ambiente e que apresenta comportamento autonomo (JENNINGS; LESPÉRANCE, 2000) (JENNINGS; LESPÉRANCE, 2000). Não apenas isso, mas um agente faz uso do seu comportamento autonomo com o propósito de atingir objetivos que a ele é designado (JENNINGS; LESPÉRANCE, 2000) (JENNINGS; LESPÉRANCE, 2000).

Como a definição de agente faz uso do conceito de ambiente, não é possível tratar esse tópico de forma consistente sem considerar a semântica deste termo. Assim sendo, ambiente é aquilo que apresenta as propriedades listadas a seguir (RUSSELL; NORVIG, 2003) (JENNINGS; LESPÉRANCE, 2000);

- *Acessibilidade vs Inacessibilidade*; Um ambiente acessível é aquele onde um agente consegue ter informações claras, precisas e atualizadas no que tange a característica do ambiente.
- *Determinístico vs Não-Determinístico*; Um comportamento determinístico é aquele onde uma ação possui um efeito claro e garantido, sem incertezas sobre o estado que irá resultar.
- *Episódico vs Não-Episódico*; Um ambiente tende a ser o mais episódico possível tanto quando o desempenho do agente estiver associado a um episódio discreto e específico no ambiente.

- *Estático vs Dinâmico*; Um ambiente é estático se não houver outros processos em paralelo aos eventos associados ao agente.
- *Discreto vs Contínuo*; Um ambiente é discreto se existe um número finito de ações e percepções.

Outro aspecto que está contido na definição consiste no conceito expresso pelo termo *autônomo*. Assim sendo, a pesquisa retratada neste texto trabalha esse termo da seguinte forma; Uma entidade que possui essa natureza tem a capacidade agir por si mesmo. Essa entidade não precisa de nenhuma outra entidade externa (ex. ser humano) para realizar decisões (JENNINGS; LESPÉRANCE, 2000) (JENNINGS; LESPÉRANCE, 2000).

Há uma série de exemplos que se enquadram dentro dessa definição. Sistemas de controle é um desses exemplos. Um *termostato* é um sistema de controle que está em um dado ambiente (como um quarto ou uma sala) (JENNINGS; LESPÉRANCE, 2000), gera dois sinais de saída (um desses sinais indica que a temperatura está baixa demais (ou alta demais dependendo da aplicação) e o outro sinal demonstra que a temperatura está no nível aceitável). O termostato tem o seu comportamento autônomo baseado em duas regras (JENNINGS; LESPÉRANCE, 2000):

- Se a temperatura estiver abaixo (ou acima) do nível de temperatura definido, então ligar o atuador.
- Se a temperatura estiver dentro do nível estabelecido, então desligar o atuador.

Outro exemplo de agentes consiste nos programas *Daemons* em sistemas *UNIX*. Esses algoritmos trabalham em segundo plano e monitoram um dado ambiente de *software*. Com base em certas regras, na ocorrência de um dado evento no ambiente, esses programas realizam uma dada atuação (JENNINGS; LESPÉRANCE, 2000).

Os exemplos presentes neste texto são apropriados dentro do conceito de agentes. Contudo, esses exemplos não se enquadram dentro da definição de agentes inteligentes (JENNINGS; LESPÉRANCE, 2000). Uma entidade que se enquadra dentro das características de um agente inteligente deve necessariamente respeitar a definição já apresentada e deve apresentar as seguintes propriedades; reatividade (capaz de perceber as mudanças que ocorrem no ambiente e responder a elas de maneira apropriada no que condiz aos objetivos do agente), pro-atividade (apresenta comportamento orientado a objetivos sendo que o agente toma as decisões a fim de satisfazer os objetivos em interesse) e habilidades sociais (capacidade de

interagir com outros agentes (e possivelmente humanos) a fim de poder satisfazer os próprios objetivos) (JENNINGS; LESPÉRANCE, 2000) (RUSSELL; NORVIG, 2003).

Os agentes podem ser definidos em categorias. Um desses são agentes puramente reativos. Esses agentes tomam decisões considerando apenas informações que estão no instante presente. Por consequência, o comportamento deles ocorre por respostas diretas ao ambiente (JENNINGS; LESPÉRANCE, 2000).

Outra categoria de agentes são aqueles que possuem estados. Essas entidades possuem uma dada estrutura de dados interna que são considerados quando agente toma uma certa decisão (JENNINGS; LESPÉRANCE, 2000).

Uma outra maneira de analisar os agentes se dá por meio das arquiteturas e modelos disponíveis para representar os agentes (tomada de decisão, estado interno). Para o propósito do estudo que está sendo apresentando neste texto, é o suficiente considerar de forma sucinta quatro dessas arquiteturas. A primeira consiste nos agentes baseados em lógica onde os agentes realizam deduções lógicas para tomar uma decisão (GENESERETH; NILSSON, 1987), a segunda arquitetura consiste nos agentes reativos que tomam decisões com base em um dado mapeamento de uma certa situação em uma dada ação (BONASSO et al., 1995), a terceira arquitetura é *BDI* cuja decisão ocorre da manipulação de estruturas de dados que representam crenças, desejos e intenções do agente (RAO; GEORGEFF, 1991) e a quarta arquitetura consiste em uma estrutura em camadas onde a tomada de decisão acontece por intermédio de diversas camadas abstração a cerca do ambiente (WOOLDRIDGE; JENNINGS, 1995b) (JENNINGS; LESPÉRANCE, 2000).

2.2 ARTEFATOS

A definição de um artefato pode ser feita por analisar, antes, o comportamento de um agente. De maneira genérica, existe duas formas que podem ser usadas para caracterizar o comportamento de um agente, e essas são *goal-governed* e *goal-oriented* (CASTELFRANCHI et al., 2018) (RICCI et al., 2006).

Um agente que é caracterizado como *goal-governed* são aqueles que apresentam capacidades cognitivas, que podem representar os seus respectivos objetivos e, portanto, são capazes de definir seus objetivos em interesse (CASTELFRANCHI et al., 2018) (RICCI et al., 2006). Em contraste com isso, *goal-oriented* consiste nos agentes que são programados com a finalidade de alcançar um determinado objetivo (CASTELFRANCHI et al., 2018) (RICCI et al., 2006). Em um sistema multiagente muitas vezes uma dada entidade não é adequadamente

representada por nenhuma dessas duas categorias.

Assim sendo, artefatos são entidades que não se enquadram em *goal-governed*, não se enquadram em *goal-oriented*, são explicitamente projetados com o propósito de serem explorados pelos agentes para que possam alcançar seus objetivos individuais e sociais (RICCI et al., 2006) (RICCI et al., 2007).

Para ilustrar usando como referência a sociedade humana; se agentes (entidades autônomas) estão para seres humanos então artefatos são as ferramentas (não autônomas) que são usadas para uma determinada finalidade (ex. um marceneiro (agente), usa um martelo (artefato) para pregar um prego (artefato)) (RICCI et al., 2006).

Uma outra distinção entre agentes e artefatos se torna claramente explícita quando verificar sobre o ponto de vista conceitual e filosófico. Isso, pois agentes apresentam a capacidade de se comunicar com outros agentes, em contraste com isso artefatos não apresentam essa condição (RICCI et al., 2006).

Existem quatro elementos que podem ser usados para caracterizar um artefato (RICCI et al., 2006), que são; *interface de uso*, *instruções de operação*, *funcionalidade* e *estrutura-comportamento*.

A *interface de uso* consiste no conjunto de operações que podem ser invocadas pelo agente a fim explorar as suas funcionalidades (RICCI et al., 2006). A *instruções de operação* consiste na descrição de como fazer o uso das funcionalidades do artefato. Isso implica protocolos de uso das operações que devem ser invocadas pelo agente (RICCI et al., 2006). A *Funcionalidade* do artefato consiste no propósito definido pelo programador do sistema (RICCI et al., 2006). A *estrutura-comportamento* consiste nos aspectos internos do artefato, que é como o artefato é implementado a fim de providenciar as suas funcionalidades (RICCI et al., 2006).

Cartago (*Common "Artefacts for Agents" Open Framework*) é um framework usado para especificar as relações entre agentes e artefatos. Tendo em vista o fato de ser um dos principais *frameworks* na descrição das interações entre agentes e artefatos, não é possível falar no tema de artefatos em SMA sem ao menos comentar a estrutura conceitual presente no Cartago. Esse modelo é composto de três blocos, que são; *agent bodies*, *artifact* e *workspace* (RICCI et al., 2007).

Agent bodies é o que possibilita a inteligência do agente se relacionar com o ambiente. Para cada agente criado, há um *agent bodies* construído. Um *agent bodies* possui *effectors* (efetores) que tem o propósito de executar ações sobre o ambiente de trabalho e possui sensores (captam sinais do ambiente em sua volta). A relação completa entre agente-*agent*

bodies-artefato, no Cartago, é dada da seguinte forma; eventos observáveis são gerados pelos artefatos, sensores (componentes presentes no *agent bodies*) são sensibilizados, essa informação é transmitida para a inteligência do agente (esta, por sua vez, realiza os raciocínios e toma as decisões necessárias), o agente comunica ao *agent bodies* a ação que deve ser feita ao meio e, por último, através do *effectors* uma dada ação é produzida no ambiente de trabalho.

Artifacts (artefatos) são os tijolos básicos na gerência do Cartago. Cada artefato apresenta um nome lógico e número de identificação (id) único que são definidos pelo *artefat creator* no momento da instanciação. O nome lógico consiste num caminho ágil para o agente poder referenciar e compartilhar o respectivo artefato com os demais agentes. O id é requisitado na identificação dos artefatos quando uma dada ação é executada. Os artefatos também apresentam nome completo que inclui o nome do(s) *workspace(s)* onde está logicamente localizados (RICCI et al., 2007).

Workspaces a localização lógica dos artefatos ocorre dentro de *workspaces*, que pode ser usado para definir a topologia do ambiente de trabalho. Neste âmbito, o *workspaces* são feitos com a finalidade de especificar nome lógico e id. Está dentro do escopo deste item definir uma topologia de ambiente possibilitando que uma estrutura de interação entre agentes e artefatos. Assim sendo o agente só pode usar um artefato que está no mesmo *workspace* onde ele está contido.

2.3 SISTEMA MULTIAGENTE

(WEISS, 2000) (WOOLDRIDGE, 2002) (ABBAS et al., 2015) (LÓPEZ; LUCK, 2004) (HÜBNER et al., 2002) (CASTELFRANCHI, 1998)

- Apresentar definições de SMA
- Aprestar conceito de objetivo
- Apresentar conceito de papel
- Apresentar os conceitos de relações deonticas

2.4 NORMAS

(LÓPEZ; LUCK, 2004) (LOPEZ; LUCK, 2003) (DASTANI et al., 2009)

- Definir o que é norma

- Tratar os tipos de normas
- Tratar o conceito de agentes normativos
- Tratar o conceito de sistemas multiagentes normativos
- Apresentar o modelo do Dastani
- Enfatizar como ele apresenta os conceitos de violação e sanção

2.5 RISCOS

(RASMUSSEN, 1997) (FADIER et al., 2003)

- Comentar as observacoes de Rasmussen
- Definir o risco dentro da estrutura do framework (FADIER et al., 2003)

2.6 POSSIBILIDADES

(DUBOIS et al., 1994)

- Comentar a ocorrência das possibilidades dentro do contexto da lógica fuzzy
- Descrever a ocorrência da lógica no contexto modal (encontrar paper?).

3 METODOLOGIA

3.1 ANÁLISE DOS MODELOS

3.2 ACOMPANHAMENTO DE PROFISSIONAIS EM ATIVIDADE DE RISCO

3.3 CONSTRUÇÃO DO MODELO

3.4 IMPLEMENTAÇÃO

4 RESULTADOS

4.1 ESTRUTURA CONCEITUAL

4.1.1 MÓDULOS

4.1.2 CONJUNTOS

4.1.3 PREDICADOS

4.1.4 REGRAS

4.2 UML

4.2.1 DIAGRAMA DE CLASSES

4.2.2 DIAGRAMA DE ATIVIDADES

4.3 CASO DE ESTUDO

4.4 RACIOCÍNIO

4.5 VALIDAÇÃO

5 ANÁLISE COMPARATIVA

5.1 MOISE+

O Moise+ é usado para realizar a especificação de sistemas multiagentes. Para cumprir com essa finalidade, existe três tipos de especificação que são; Estrutural, Funcional, e Deôntica.

5.1.1 ESTRUTURA

A especificação estrutural acontece em três níveis, individual, social e coletivo. O nível individual trata de definir os papéis ρ dos agentes. Uma possível entre os papéis acontece por intermédio da hereditariedade em que se ρ' é filho de ρ . Isso implica afirmar que ρ' é uma especialização de ρ . Um exemplo apropriado para isso é o jogo de futebol onde existe o papel jogador dado por ρ e existe o papel atacante dado por ρ' (HÜBNER et al., 2002). Em termos formais, essa relação é dada por;

$$\rho_a \sqsubset \rho_b$$

O nível social estabelece relações de ligação dado pelo predicado $link(\rho_s, \rho_d, t)$. Existe três possíveis valores para t , os quais são $t = \{aut, com, acq\}$. O valor *auth* significa autoridade (neste caso ρ_s exerce autoridade sobre ρ_d), o valor *com* significa comunicação (neste caso ρ_s pode se comunicar com ρ_d) e o valor *acq* significa conhecimento (ρ_s tem conhecimento da existência de ρ_d) (HÜBNER et al., 2002). O MOISE+ define as seguintes relações de implicabilidade

$$\begin{aligned} link(\rho_s, \rho_d, auth) &\rightarrow link(\rho_s, \rho_d, com) \\ link(\rho_s, \rho_d, com) &\rightarrow link(\rho_s, \rho_d, acq) \end{aligned} \tag{1}$$

O modelo também determina como se dá as relações de hereditariedade para o predicado de *link*, é dado por (HÜBNER et al., 2002);

$$\begin{aligned} link(\rho_s, \rho_d, t) \wedge \rho'_s \sqsubset \rho'_s &\rightarrow link(\rho'_s, \rho_d, t) \\ link(\rho_s, \rho_d, t) \wedge \rho'_d \sqsubset \rho'_d &\rightarrow link(\rho_s, \rho'_d, t) \end{aligned} \quad (2)$$

O nível coletivo determina a existência de compatibilidade entre os papéis (HÜBNER et al., 2002). Essa é uma relação reflexiva e transitiva de determina que se um papel ρ_a possui a capacidade de realizar um determinado objetivo, então o papel ρ_b também tem essa capacidade. Em termos formais, essa relação se dá da seguinte forma (HÜBNER et al., 2002).;

$$\rho_a \bowtie \rho_b \wedge \rho_a \neq \rho_b \wedge \rho_a \sqsubset \rho' \rightarrow \rho' \bowtie \rho_b$$

O nível coletivo também apresenta o conceito de grupo dado por *gt* e constituído por;

$$gt = \langle R, SG, L^{intra}, L^{inter}, C^{intra}, C^{inter}, np, ng \rangle$$

Em que R é o conjunto dos papéis não abstratos, SG são subgrupos que estão contidos neste grupo, L^{intra} consiste dos *links* intra-grupos, L^{inter} dos links inter-grupos, C^{intra} das relações de compatibilidade intra-grupos e C^{inter} das relações de compatibilidade inter-grupos. O símbolo np denota a cardinalidade mínima e máxima para uma dada função e o símbolo ng realiza o mesmo para os subgrupos (HÜBNER et al., 2002).

A Especificação Funcional tem como por finalidade descrever os objetivos a serem atingidos dentro de uma estrutura de árvore. A figura a seguir define como se dá esse tipo de especificação;

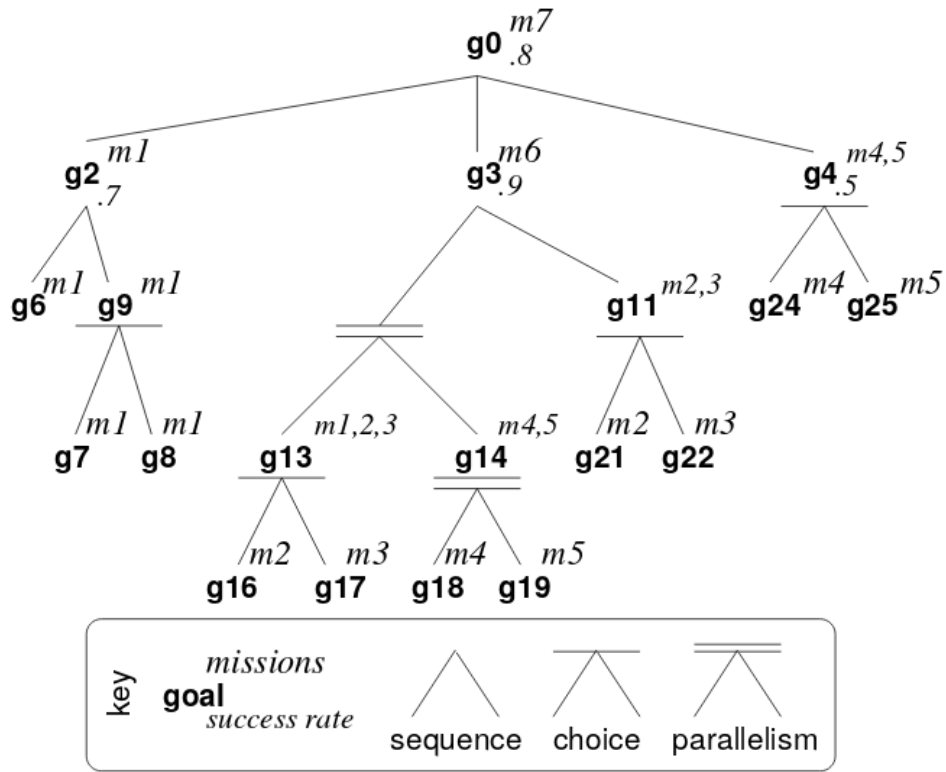


Figura 1: Arvore de objetivos definido pelo modelo Moise (HÜBNER et al., 2002)

A figura 1 define três tipos de relação de subobjetivos; *sequence* onde todos os subobjetivos devem necessariamente ser concluídos em sequência, *choice* onde o agente tem a possibilidade de escolher qual objetivo ele deseja seguir e *parallelism* onde todos os objetivos devem ser concluídos, contudo sem uma sequência definida. Como é possível observar na figura, os objetivos são agrupados em conjuntos de missões m . A relação a seguir define isso melhor;

$$m_k = \{g_n, \dots, g_m\}$$

A Especificação Deontica define predicados para estabelecer permissões e obrigações entre os papéis e as missões. Toda obrigação implica necessariamente em uma permissão. A relação a seguir estabelece isso;

$$obl(\rho, m, tc) \rightarrow per(\rho, m, tc)$$

$$obl(\rho, m, tc) \wedge \rho \sqsubseteq \rho' \rightarrow obl(\rho', m, tc) \quad (3)$$

$$per(\rho, m, tc) \wedge \rho \sqsubseteq \rho' \rightarrow per(\rho', m, tc) \quad (4)$$

$$(5)$$

Onde o predicado *obl* define uma obrigação e o predicado *per* define permissão. O argumento *tc* define uma periodicidade de tempo para o qual a relação deôntica é válida.

Existe muitos ponto similares entre o modelo proposto neste estudo e o Moise. Ambos os modelos apresentam papeis ρ , apresentam objetivos g e apresentam relações deônticas de obrigação e permissão. Isso se deve ao fato, em partes, que o modelo proposto neste estudo importou muitos dos conceitos presentes no Moise tendo em vista a aplicabilidade para o domínio em interesse. Esses conceitos são importantes para o modelo deste estudo pois são grande relevância para descrever situações onde uma equipe de pessoas devem atingir certos objetivos trabalhando em cooperação.

Contudo, ambos modelos apresentam diferenças significativas. Uma dessas diferenças reside em como as relações deônticas são atribuídas, pois no Moise a relação é feita entre o papel e a missão e no modelo proposto neste estudo a relação é feita entre o papel e o objetivo. Isso se deve ao fato de que uma das finalidades deste estudo consiste na realização de uma análise de sanções e violações. Como essas análises trabalham no nível de atividades, objetos e condições (dentro do contexto deste estudo), trabalhar na ordem de objetivos trás uma análise mais aprimorada no estudo das violações e sanções.

O Moise não apresenta suporte a tratativa de sanções e violações. Assim sendo, para conseguir atingir um modelo onde fosse possível analisar certos tipos de sanções e violações de interesse, foi necessário introduzir predicados que não fazem parte do vocabulário e da sintaxe do Moise.

O Moise trabalha uma estrutura lógica de interesse a descrição de grupos, *links* e compatibilidades. Esses conceitos não são trabalhados no modelo em interesse por não serem necessários ao domínio de interesse deste estudo. Assim sendo, esses conceitos trariam complexidades adicionais sem justificativa válida para isso.

5.1.2 ANÁLISE COMPARATIVA

5.2 DASTANI

O texto (DASTANI et al., 2009) apresenta um modelo com a finalidade de descrever agentes normativos com a capacidade de cometer uma certa violação. O modelo define sanções para as violações. O estudo estrutura o modelo como uma linguagem de programação (usando a notação EBNF) onde os problemas de sistemas multiagentes são tratados como programas escritos neste linguagem (DASTANI et al., 2009). Assim sendo, um programa de sistemas multi-agentes é descrito como sendo;

```

N-MAS_Prog  := "Agents: " (<agentName> <agentProg> [<nr>])+ ;
              "Facts: " <bruteFacts>
              "Effects: " <effects>
              "Counts-as rules: " <counts-as>
              "Sanction rules: " <sanctions>;

<agentName> := <ident>;
<agentProg> := <ident>;
<nr>        := <int>;
<bruteFacts>:= <b-literals>;
<effects>   := ({<b-literals>} <actionName> {<b-literals>})+;
<counts-as>:= ( <literals> ⇒ <i-literals> )+;
<sanctions>:= ( <i-literals> ⇒ <b-literals> )+;
<actionName>:= <ident>;
<b-literals>:= <b-literal> {"," <b-literal>};
<i-literals>:= <i-literal> {"," <i-literal>};
<literals>  := <literal> {"," <literal>};
<literal>   := <b-literal> | <i-literal>;
<b-literal> := <b-prop> | "not" <b-prop>;
<i-literal> := <i-prop> | "not" <i-prop>;

```

Figura 2: Linguagem para descrever um programa de multiagentes normativos com a possibilidade de violações e sanções na notação EBNF segundo o texto (DASTANI et al., 2009). Nesta notação, *< ident >* é usado para denotar uma *string* e *< int >* inteiros. Os termos *< b – prop >* e *< i – prop >* são usados para designar dois tipos de conjuntos de proposições que são disjuntos entre si

5.2.1 ESTRUTURA

Com base no proposto por esse modelo, um programa de sistemas multiagentes é descrito por *Agents* (agentes), *Facts* (fatos), *Effects* (efeitos), (Count-as rules) (regras que determinam o adequado comportamento do agente), *Sanction rules* (regras de sanção) (DASTANI et al., 2009).

Os *Agents* são definidos em termos de duas *strings* e um inteiro. A primeira *string* é

agentName e é usado para definir o nome do agente e a segunda *string* é *agentProg* e define o nome do arquivo onde se encontra especificações do respectivo agente. O inteiro *nr* é usado para definir a quantia do agente (DASTANI et al., 2009).

Os *Facts* são compostos por conjuntos de literais denominados de *bruteFacts*. Esses são fatos onde ocorre uma violação que desencadeia em uma dada sanção (DASTANI et al., 2009).

Os *Effects* são compostos por *effects*. Estes, por sua vez, são estruturados em termos de *b-literals* (conjuntos que contem literais onde estes, por sua vez, representam um dado estado de mundo), e *actionName* onde este, por sua vez, descreve ações que geram transições de um estado de mundo para outro estado (DASTANI et al., 2009).

Os *Count-as rules* são compostos por *counts-as*. Esses, por sua vez, definem regras normativas. Isso implica em relações de implicabilidade que resultam em uma dada violação (DASTANI et al., 2009).

Os *Sanction rules* são estruturados por *sanctions*. Esses, por sua vez, definem regras de implicabilidade que competem a uma tratativa da violação (DASTANI et al., 2009).

A figura a seguir ilustra um exemplo de um programa de sistema multiagente escrito nesta linguagem;

```

Agents:      passenger  PassProg  1
Facts:       {-at_platform, -in_train, -ticket}
Effects:     {-at_platform} enter {at_platform},
             {-ticket} buy_ticket {ticket},
             {at_platform, -in_train} embark {-at_platform, in_train}
Counts-as rules: {at_platform, -ticket} ⇒ {viol1},
                 {in_train, -ticket} ⇒ {viol⊥}
Sanction rules: {viol1} ⇒ {fined10}

```

Figura 3: Um programa descrito na linguagem proposta neste estudo onde um agente representa um passageiro em uma estação de trem que pode entrar com ou sem um *ticket* na plataforma e no trem (DASTANI et al., 2009).

A figura 3 apresenta um programa que contem um agente com nome *passenger*. O agente pode estar ou não na plataforma e no trem, sem ou com *ticket*. Se o agente entrar na plataforma ou no trem sem o *ticket*, então esse agente cometeu uma violação. Para este programa, a sanção da violação que ocorre por entrar na plataforma sem o *ticket* resulta em uma punição onde o agente deve pagar 10 Euros pelo ocorrido (DASTANI et al., 2009).

5.2.2 ANÁLISE COMPARATIVA

5.3 V3S

V3S é um modelo com a finalidade de gerar ambientes para desenvolver treinamentos complexos em ambiente de realidade virtual visando atividades de risco e de emergência. O modelo é composto por três submodelos; *Domain Model*, *Activity Model* e *Risk Model* (BAROT et al., 2013). O *Domain Model* é o núcleo do sistema. Todos os objetos, ações e relações são descritos por uma ontologia. A figura 4 exibe a estrutura de classe desta ontologia.

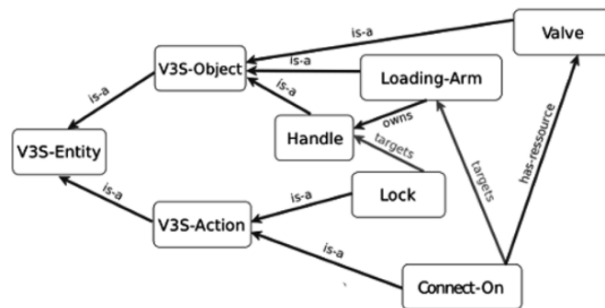


Figura 4: Ontologia que descreve *Domain Model* no model V3S (BAROT et al., 2013)

5.3.1 ESTRUTURA

Activity Model é estruturado sobre uma linguagem de descrição conhecido por *ACTIVITY-DL*. Essa linguagem usa álgebra de Allen's que tem como finalidade definir raciocínios temporais (ALLEN, 1983). As relações definidas por essa álgebra é dada por;

1. $X < Y$ onde X : ocorre antes de Y
2. $XmY, YmiX$: X encontra Y
3. $XoY, XoiY$: X sobrepõem a Y
4. $XsY, YsiX$: X começa Y
5. $XdY, YdiX$: X ocorre durante Y
6. $XfY, YfiX$: X termina junto com Y
7. $X = Y$ X é igual a Y

| Construtor | Nome | Relações de Allen |
|------------|--------------|--|
| IND | Independent | $A\{<, >, m, mi, o, oi, s, si, d, di, f, fi, =\}B$ |
| SEQ | Sequential | $A\{<, >, m, mi\}B$ |
| SEQ-ORD | Ordered | $A\{<, >, m\}B$ |
| PAR | Parallel | $A\{o, oi, s, si, d, di, f, fi, =\}B$ |
| PAR-SIM | Simultaneous | $A\{=\}B$ |
| PAR-START | Start | $A\{s, si, =\}B$ |
| PAR-END | End | $A\{f, fi, =\}B$ |

Tabela 1: Construtores da linguagem *ACTIVITY-DL* (BAROT et al., 2013)

A linguagem define construtores que são semanticamente equivalente a certos operadores da álgebra de Allen's. Esses construtores (atuantes sobre atividades) são definidos pela tabela 1

No que tange a questões referentes a segurança e violação, a linguagem *ACTIVITY-DL* define *tags* no estudo (FADIER et al., 2003). Essas *tags* são BCTUs, BATUs. BCTUs representam situações informais entre os atores de um determinado campo, por exemplo; trabalhar com produtos químicos sem usar equipamentos de proteção individual. BATUs corresponde a práticas de risco tolerado. Existe dois tipos de BATUs, primeiro - quando realizado por obrigação ou interesse de conforto (quando os operadores acham impossível realizar a operação respeitando as instruções), segundo - ocorre durante a operação do sistema com o propósito de evitar que o sistema deixe de funcionar.

Activity Model apresenta uma categorização de pré-condições. A tabela 2 detalhe essa categorização.

Risk Models é a parte do modelo que define a análise de risco. Existe duas categorias; risco de análise clássico e método de análise de confiabilidade humana. A primeira categoria permite definir uma análise quantitativa de risco, contudo falha ao definir a complexidade dos resultados frente a fatores humanos. Em contrapartida, a segunda categoria considera fatores humanos, contudo falha em definir medidas objetivas sobre questões de segurança (BAROT et al., 2013).

O V3S combina ambas situações usando a abordagem MELISSA (CAMUS et al., 2012) (BAROT et al., 2013). Essa abordagem é baseada em três pontos (1) atividades relacionadas em cenários de acidentes, (2) descrição das tarefas de representação e (3) fatores influentes em potencial nas atividades.

| Categoria | Pré-condição | Descrição |
|-------------------------|---------------------|---|
| Condições para perceber | Nomológico | Descreve o estado do mundo necessário para que a tarefa seja fisicamente realizável. Condições dependem diretamente das regras de ação definidas no modelo de domínio. Exemplo: Abre a porta se estiver fechada. |
| Condições para perceber | Regulamentar | Descreve o estado do mundo necessário para uma boa realização da atividade de acordo com prescrito em procedimento. Exemplo: Para desconectar o tubo, a proteções devem ser desgastado. |
| Condições para Examinar | Contextual | Descreve o estado de mundo em que a atividade é relevante. Quando essa condição é falsa, então a atividade deve ser ignorada. Exemplo: Limpar o tubo é relevante apenas se o tubo estiver sujo. |
| Condições para Examinar | Favorável | Descreve o estado de mundo onde a tarefa é preferencial sobre as demais. Essas condições ajudam a escolher entre várias tarefas quando existe uma alternativa para a realização de uma tarefa decomposta. Exemplo: se o parafuso estiver enferrujado, desarmar. |

Tabela 2: As pré-condições possíveis para as atividades (BAROT et al., 2013)

5.3.2 ANÁLISE COMPARATIVA

5.4 NORMMAS

NormMAS é um modelo usado para definir comportamento normativo de sistemas multiagentes (CHANG; MENEGUZZI, 2016). No que tange questões referentes ao comportamento normativo, o modelo trabalha com duas definições (CHANG; MENEGUZZI, 2016).

5.4.1 ESTRUTURA

Definição 1. *Um norma é definida por meio de uma tupla $N = \langle \mu, \kappa, \chi, \tau, \rho \rangle$*

- $\mu \in \{obligation, prohibition\}$ representa as modalidades de norma.
- $\kappa \in \{action, state\}$ representa o tipo de *trigger* da condição.
- χ representa o conjunto de estados em que uma norma se aplica.
- τ representa a norma da condição de *trigger*
- ρ representa a sanção aplicava pela violação do agente.

A definição 1 pode ser compreendida sobre o seguinte exemplo;

Todos os imigrantes que possuem passaporte válido, devem ser aceitos. A falha resulta na perda de 5 créditos.

Dentro da definição 1, o exemplo fica;

$$\langle obligation, action, valid(Passport), accept(Passport), loss(5) \rangle \quad (6)$$

NormMAS define um *Registro de ação* que é dado pela definição 2.

Definição 2. *Um Registro de Ação é definido por meio de uma tupla $R = \langle \gamma, \alpha, \beta \rangle$*

- γ representa o agente executando uma ação;
- α representa a ação sendo executada pelo agente γ
- β representa os estados internos do agente γ no momento da execução.

5.4.2 ANÁLISE COMPARATIVA

5.5 PERSPECTIVA GENÉRICA

6 CONCLUSÃO

6.1 AVALIAÇÃO DOS OBJETIVOS

6.2 TRABALHOS FUTUROS

REFERÊNCIAS

ABBAS, H.; SHAHEEN, S.; AMIN, M. H. Organization of multi-agent systems: An overview. **International Journal of Intelligent Information Systems**, 06 2015.

ALLEN, J. F. Maintaining knowledge about temporal intervals. **Commun. ACM**, ACM, New York, NY, USA, v. 26, n. 11, p. 832–843, nov. 1983. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/182.358434>>.

BAROT, C. et al. V3S: A virtual environment for risk-management training based on human-activity models. **Presence**, v. 22, n. 1, p. 1–19, 2013. Disponível em: <http://www.mitpressjournals.org/doi/abs/10.1162/PRES.a_00134>.

BONASSO, R. P. et al. Experiences with an architecture for intelligent, reactive agents. In: . [S.l.: s.n.], 1995. v. 9, p. 187–202.

CAMUS, F.; LENNE, D.; PLOT, E. Designing virtual environments for risk prevention: the melissa approach. **International Journal on Interactive Design and Manufacturing (IJIDeM)**, v. 6, n. 1, p. 55–63, Feb 2012. ISSN 1955-2505. Disponível em: <<https://doi.org/10.1007/s12008-011-0138-4>>.

CASTELFRANCHI, C. Modelling social action for ai agents. **Artificial Intelligence**, v. 103, n. 1, p. 157 – 182, 1998. ISSN 0004-3702. Artificial Intelligence 40 years later. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0004370298000563>>.

CASTELFRANCHI, C. et al. Social trust: A cognitive approach. 12 2018.

CHANG, S.; MENEGUZZI, F. Simulating normative behaviour in multi-agent environments using monitoring artefacts. In: DIGNUM, V. et al. (Ed.). **Coordination, Organizations, Institutions, and Norms in Agent Systems XI**. Cham: Springer International Publishing, 2016. p. 59–77. ISBN 978-3-319-42691-4.

DASTANI, M. et al. Normative multi-agent programs and their logics. In: MEYER, J.-J. C.; BROERSEN, J. (Ed.). **Knowledge Representation for Agents and Multi-Agent Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 16–31. ISBN 978-3-642-05301-6.

DUBOIS, D.; LANG, J.; PRADE, H. Possibilistic logic. In: _____. [S.l.: s.n.], 1994. p. 439–513.

FADIER, E.; GARZA, C. D. L.; DIDELOT, A. Safe design and human activity: construction of a theoretical framework from an analysis of a printing sector. **Safety Science**, v. 41, n. 9, p. 759 – 789, 2003. ISSN 0925-7535. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S092575350200022X>>.

GENESERETH, M. R.; NILSSON, N. J. **Logical Foundations of Artificial Intelligence**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987. ISBN 0-934613-31-1.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In: BITTENCOURT, G.; RAMALHO, G. L. (Ed.). **Advances in Artificial Intelligence**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 118–128. ISBN 978-3-540-36127-5.

JENNINGS, N. R.; LESPÉRANCE, Y. **Intelligent Agents VI. Agent Theories, Architectures, and Languages: 6th International Workshop, ATAL'99, Orlando, Florida, USA, July 15-17, 1999. Proceedings**. [S.l.: s.n.], 2000. ISBN 978-3-540-67200-5.

LOPEZ, F.; LUCK, M. Modelling norms for autonomous agents. In: . [S.l.: s.n.], 2003. p. 238 – 245. ISBN 0-7695-1915-6.

LÓPEZ, F. López y; LUCK, M. A model of normative multi-agent systems and dynamic relationships. In: LINDEMANN, G.; MOLDT, D.; PAOLUCCI, M. (Ed.). **Regulated Agent-Based Social Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 259–280. ISBN 978-3-540-25867-4.

RAO, A. S.; GEORGEFF, M. P. Modeling rational agents within a bdi-architecture. In: **Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991. (KR'91), p. 473–484. ISBN 1-55860-165-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=3087158.3087205>>.

RASMUSSEN, J. Risk management in a dynamic society: a modelling problem. **Safety Science**, v. 27, n. 2, p. 183 – 213, 1997. ISSN 0925-7535. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925753597000520>>.

RICCI, A.; VIROLI, M.; OMICINI, A. Programming mas with artifacts. In: BORDINI, R. H. et al. (Ed.). **Programming Multi-Agent Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 206–221. ISBN 978-3-540-32617-5.

RICCI, A.; VIROLI, M.; OMICINI, A. Cartago: A framework for prototyping artifact-based environments in mas. In: WEYNS, D.; PARUNAK, H. V. D.; MICHEL, F. (Ed.). **Environments for Multi-Agent Systems III**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 67–86. ISBN 978-3-540-71103-2.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence - a modern approach, 2nd Edition**. Prentice Hall, 2003. (Prentice Hall series in artificial intelligence). ISBN 0130803022. Disponível em: <<http://www.worldcat.org/oclc/314283679>>.

WEISS, G. Multiagent systems: A modern approach to distributed artificial intelligence. In: _____. [S.l.: s.n.], 2000. v. 1, p. –648. ISBN 0262731312.

WOOLDRIDGE, M. An introduction to multiagent systems. In: _____. [S.l.: s.n.], 2002. p. –348. ISBN 047149691X.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: theory and practice. **The Knowledge Engineering Review**, Cambridge University Press, v. 10, n. 2, p. 115–152, 1995.

APÊNDICE A – NOME DO APÊNDICE

ANEXO A – NOME DO ANEXO