

**Anais do XIV Workshop-Escola de Sistemas de
Agentes, seus Ambientes e aplicações**

— WESAAC 2020—

Organizado por

**Gleifer Vaz Alves
Gustavo Guiménez Lugo
André Pinz Borges
Carlos Eduardo Pantoja**

**Ponta Grossa, 07 - 09 de julho de 2020.
(Realizado no formato online)**

Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações –XIV WESAAC / Alves, G. V.; Lugo, G. G.; Borges, A. P.; Pantoja, C. E. (Org). ANAIS.— — Ponta Grossa, 2020.

273p. : il.

ISSN 2177-2096 [DOI 10.5281/zenodo.4037413](https://doi.org/10.5281/zenodo.4037413)

1. Agentes Inteligentes. 2. Sistemas de Agentes de Software. 3. Ambientes para Agentes. 4. Aplicações de Agentes. I. Alves, G. V. II. Lugo, G. G. III. Borges, A. P. IV. Pantoja, C. E.

PREFÁCIO

Este documento contém os trabalhos apresentados na Décima Quarta Edição do WESAAC (*Workshop Escola de Sistemas de Agentes, seus Ambientes e aplicações*), realizado no formato online com o apoio da Universidade Tecnológica Federal do Paraná (UTFPR) e Centro Federal de Educação e Tecnológica Celso Suckow (CEFET-RJ), entre os dias 07 e 09 de Julho de 2020.

Essa edição estava originalmente programa para ocorrer presencialmente na cidade de Ponta Grossa-PR, no Câmpus da UTFPR. Porém, devido a pandemia da Covid-19 o evento foi realizado no formato online.

Com todas diferenças e particularidades da realização de um evento, sem talvez o seu principal aspecto que é o contato direto entre os pesquisadores, foi necessário repensar e reorganizar vários aspectos do evento.

De toda forma, o evento procurou preservar suas principais características, isso é a apresentação de artigos aceitos (após o processo de revisão), participação de palestrantes (inclusive internacionais) e oferecimento de oficinas voltadas especialmente aos alunos. Assim, buscou-se preservar a premissa principal do WESAAC: integrar pesquisadores e estudantes de todos os níveis na área de Agentes e Sistemas de Agentes e divulgar as atividades dos diversos grupos de pesquisa do Brasil, bem como promover o intercâmbio de conhecimentos e experiências.

Gostaríamos de agradecer aos pesquisadores convidados, Viviana Mascardi, Paulo Leitão e Rocha Costa, os quais aceitaram o convite em participar do WESAAC, mesmo no formato online. Igualmente agradecemos à Diana Adamatti, Míriam Blank Born, Giovanni Parente Farias, Lucas Castro, Fabian C. P. B. Manoel e Vinicius Souza de Jesus pelas oficinas ministradas. Não obstante agradecemos ao valioso trabalho de todo comitê de programa na revisão dos artigos e igualmente ao comitê consultivo que foi fundamental durante todos os estágios do evento. Naturalmente, ainda agradecemos aos apresentadores, que igualmente aceitaram o desafio de apresentações por meio de vídeos e agradecemos à todos participantes do Workshop. Por fim, agradecemos ao apoio da Comissão de Voluntários formada por alunos da graduação em Computação na UTFPR, Ponta Grossa.

Ponta Grossa, 2020.

Gleifer Vaz Alves
Gustavo Guiménez Lugo
André Pinz Borges
Carlos Eduardo Pantoja

Organização e Realização



Organização

Organização Geral

Gleifer Vaz Alves (UTFPR)

Gustavo Giménez Lugo (UTFPR)

Coordenação do Comitê do Programa

Carlos Eduardo Pantoja (CEFET-RJ)

Organização Local

André Pinz Borges (UTFPR)

Comitê Consultivo

Anarosa Alves Franco Brandão (USP)

Diana Francisca Adamatti (FURG)

Gustavo Alberto Giménez Lugo (UTFPR)

Jerusa Marchi (UFSC)

Jaime Sichman (USP)

João Luis Tavares da Silva (UCS)

Jomi Fred Hubner (UFSC)

Mariela Inés Cortés (UECE)

Rafael Heitor Bordini (PUCRS)

Rejane Frozza (UNISC)

Ricardo Choren (IME)

Viviane Torres da Silva (UFF)

Comitê do Programa

Alessandro Ricci (Universidade de Bolonha, Itália)

Álvaro Moreira (UFRGS)

Ana Paula Lemke (IFRS)

Anarosa Alves Franco Brandão (USP)

André Pinz Borges (UTFPR)

Antonio Carlos Rocha Costa (PPGComp / FURG - PPGFIL / PUCRS)

Carlos Pantoja (CEFET-RJ)

Cesar Tacla (UTFPR)

Clarissa Xavier (UFRGS)

Cleo Billa (FURG)

Diana Francisca Adamatti (FURG)

Eder Mateus Nunes Gonçalves (FURG)

Fernanda Alencar (UFPE)

Fernando de La Prieta (Universidade de Salamanca)

Fernando dos Santos (UDESC)

Francisca Raquel de Vasconcelos Silveira (IFCE)

Gabriel Ramos (UNISINOS)

Gleifer Vaz Alves (UTFPR)

Graçaliz Dimuro (FURG)

Guillermo Simari (Universidade do Sul - Bahia Blanca, Argentina)

Gustavo Augusto Lima de Campos (UECE)

Gustavo Giménez Lugo (UTFPR)

Jaime Sichman (USP)

Jerusa Marchi (UFSC)

João Luis Tavares da Silva (UniFTEC)
Jomi Fred Hubner (UFSC)
Luis Gustavo Nardin (Faculdade Nacional da Irlanda)
Maicon Zatelli (UFSC)
Maiquel de Brito (UFSC)
Mariela Cortés (UECE)
Marilton Sanchotene de Aguiar (UFPel)
Olivier Boissier (Minas Saint-Etienne, Instituto Henri Fayol, Laboratório Hubert Curien UMR CNRS 5516, França)
Rafael Cardoso (Universidade de Liverpool, Inglaterra)
Patricia Tedesco (UFPE)
Paulo Trigo (ISEL, Portugal)
Paulo Leitão (IPB, Portugal)
Rafael Bordini (PUCRS)
Raquel Barbosa (IFRS)
Rejane Frozza (UNISC)
Ricardo Choren (IME)
Ricardo Grunitzki (Instituto de Ciência e Tecnologia SIDIA Samsung)
Ricardo Azambuja Silveira (UFSC)
Sara Casare (USP)
Túlio Lima Basílio (IFRS)
Vera Maria Benjamim Werneck (UERJ)
Viviane Torres da Silva (IBM)

SUMÁRIO

ARTIGOS COMPLETOS

Symbolic Environments of Agent Societies Antonio Carlos da Rocha Costa	12
The Role of Social Image as a Behavior Filter: A Case Study Based on Cognitive Agents Jeferson J. Baqueta, Mariela Morveli-Espinoza, Gustavo A. Gimenez-Lugo and Cesar A. Tacla	25
Modelo Ontologico para Significar Status-Functions em Instituições Artificiais Rafhael R. Cunha, Jomi F. Hubner, Maiquel de Brito	37
Agregação de Julgamento usando Argumentação Abstrata para JaCaMo: Especificação e Estudo de Caso Elton M. Sato, Cesar A. Tacla, Mariela Morveli-Espinoza	49
Estudo comparativo em GAMA e Google Earth Engine: possibilidades para a área de sistemas multiagente Míriam Born, Fernanda P. Mota, Giovanni Farias, Matheus Gonçalves, Bruna Leitzke, Marilton Aguiar, Diana F. Adamatti	61
Sistemas Multiagente e Jogos de Papeis Aplicados a Gerenciamento de Recursos Naturais Vinicius Borges Martins, Raquel de Miranda Barbosa, Diana Francisca Adamatti	74
Aspectos teóricos e aplicados para o desenvolvimento de uma simulação social baseada em agentes para análise dos desequilíbrios de gênero na tecnologia Giulia Bordignon Silveira, Diana F. Adamatti	86
Modelagem da Organização de um Sistema Multiagente no Contexto de Recursos Hídricos Míriam Blank Born, Diana F. Adamatti, Marilton Sanhotene de Aguiar	93
Agent Programming for Industrial Applications: Some Advantages and Drawbacks Otavio A. Matoso, Luis P. A. Lampert, Jomi F. Hubner, Mateus Conceicao, Sergio P. Bernardes, Cleber J. Amaral, Maicon R. Zатели, Marcelo L. de Lima	104
Farmer: Sistema Multiagente para a Manutenção Autônoma em uma Agricultura Utilizando IoT Middleware Alice Trinta, Fabian C. B. Manoel, Carlos Eduardo Pantoja	116
A Systematic Review of Literature on Recommendation Systems and Machine Learning Applied to Multiagent Systems Karine Pestana Ramos and Marilton Sanhotene de Aguiar	128

Towards Jacamo-rest: A Resource-Oriented Abstraction for Managing Multi-Agent Systems	140
Cleber J. Amaral, Jomi F. Hubner and Timotheus Kampik	
Implantação de Precificação Dinâmica em um Estacionamento Inteligente utilizando Agentes	152
Alexandre L. L. Mellado, Gleifer Vaz Alves, Paulo Leitão, André Pinz Borges	
Integrando Sistemas Multi-Agentes Embarcados, Simulação Urbana e Aplicações de IoT	165
Lucas Fernando Souza de Castro, Fabian Cesar P. B. Manoel, Vinícius Souza de Jesus, Carlos Eduardo Pantoja, André Pinz Borges, Gleifer Vaz Alves	
Simulação do Comportamento de um Agente Inteligente Guiado por Regras de Trânsito	177
Vithor Tozetto Ferreira, Gleifer Vaz Alves	
Desenvolvimento de Aplicação para um Estacionamento Inteligente via Computação em Nuvem baseada em Agentes e Sistema Ciber-Físico	185
Pedro W. Botelho, Gleifer Vaz Alves, Paulo Leitão, André Pinz Borges	
Gerenciamento de Tráfego Urbano em uma Cidade Inteligente baseado em Agentes Autônomos e Descentralizados	195
Gleifer V. Alves, André P. Borges, Paulo Leitão	
Agile O-MaSE: Framework Ágil Orientado a Agentes	206
Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck	
Ambiente Virtual Personalizado baseado em Estilos de Aprendizagem: uma abordagem avaliativa com as ferramentas Face Reader e Eye Tracker	219
Danielli Cossul, Brunno José Fagundes, Guilherme Ferreira, Rejane Frozza, William Andrei Nascimento da Silveira	
RESUMOS ESTENDIDOS	
Implementando Agentes Baseados no Modelo de Processamento de Objetivos Baseado em Crenças	232
Henrique M. R. Jasinski, Mariela Morveli-Espinoza, Cesar A. Tacla	
Desenvolvimento de um Framework para Modelagem e Simulação de Redes Regulatórias Genéticas usando Sistemas Multiagente	238
Nilzair B. Agostinho, Adriano V. Werhli, Diana F. Adamatti	
Simulação baseada em agentes para a gestão de recursos hídricos	244
Julia P. C. Pereira, Diana F. Adamatti, Raquel M. Barbosa	
Um estudo do Myers-Briggs Type Indicator (MBTI) para modelagem de sistemas multiagentes no apoio a processos de	250

recrutamento e seleção nas empresas

Luiz Fernando Braz, Jaime Simão Sichman

Modelagem para processamento e coleta de dados de trajetórias temporais na sociologia 256

Matheus Mattos de Almeida Cruz

Modelagem de Emoções para Tomada de Decisão de Agentes 262

Thiago Dantas, Patrícia Padula, Diana F. Adamatti, Cleo Z. Billa

Influência da Dor no Ritmo Circadiano: uma Modelagem Baseada em Sistemas Multiagente 268

Angelica T. Santos, Catia M. Machado, Diana F. Adamatti

ARTIGOS COMPLETOS

Symbolic Environments of Agent Societies *

Antônio Carlos da Rocha Costa¹

¹Programa de Pós-Graduação em Filosofia, PUCRS, Brazil

ac.rocha.costa@gmail.com

Abstract. *This paper analyzes the concept of symbolic environment of agent societies and examines how the symbolic systems that constitute a symbolic environments represent the cultural formations (understood in a wide sense: morality, legality, art, science, politics, religion, ethnicity, etc.) of such societies. It also considers the general notion of social actor and illustrates, through a case study, how symbolic environments may support the culture-driven conducts of the social actors of agent societies.*

Resumo. *Este artigo analisa o conceito de ambiente simbólico de sociedades de agentes e examina como os sistemas simbólicos presentes em ambientes simbólicos representam formações culturais (entendidas em um sentido amplo: moralidade, legalidade, arte, ciência, política, religião, etc.) em tais sociedades. Considera, também, a noção geral de ator social e ilustra, através de um estudo de caso, como sistemas simbólicos podem suportar condutas dirigidas pela cultura em atores sociais de sociedades de agentes.*

1. Introduction


1.1. Computational and Cultural Symbolic Environments

The word “symbol” is ambiguous when applied in the context of the computational modeling of social systems.

On one hand, it has the special meaning that “symbol” has in Computer Science: a *physical mark* representing some concept or entity present either in a computational system or in the external world modeled by that the computational system. This meaning is derived from the meaning of “symbol” in Logics and Mathematics.

On the other hand, the word “symbol” has the general cultural sense of *anything* that has a meaning of any kind for someone. For instance: a *person*, taken as a symbol of some sportive game; a *tower*, taken as the symbol of a country or city; a *gesture*, taken as symbol of a political movement; a *god*, taken as a symbol of a civilization; etc.

When attempting to model computationally the social and cultural aspects of some social system, that ambiguity extends to the notion of *symbolic environment*: one comes to deal both with *symbolic environments* in the computational sense, and with *symbolic environments* in the cultural sense.

*  O trabalho Symbolic Environments of Agent Societies de Antônio Carlos da Rocha Costa está licenciado com uma Licença Creative Commons - Atribuição-CompartilhaIgual 4.0 Internacional. <http://creativecommons.org/licenses/by-sa/4.0/>

Of course, one can take computational symbols to be a particular kind of cultural symbols, as computer languages are taken as cultural objects.

In this paper, we introduce an approach to making use of *computational symbolic environments* to allow for the explicit embedding of *cultural symbolic environments* in agent societies, by taking computational symbols as *computational representations* of cultural symbols.

1.2. The Structure of the Paper

The paper is structured as follows.

In Sect. 2, we discuss a particular model of multiagent system that we call *agent society*. Also, we discuss a general concept of *social actor*, allowing for a uniform reference to social actions performed by agents, groups of agents, organizations, groups of organizations, etc., in the context of agent societies.

In Sect. 3, we introduce the formal concept of *symbolic environment of agent society*, to serve as the *computational* sense of the term “symbolic environment”. *Symbolic systems* are introduced as the *main components* of symbolic environments of agent societies. Regarding the *cultural* sense of “symbolic environment”, we discuss the way symbolic environments of agent societies may be used to reify the *cultural systems* and the *cultural symbolic environments* of agent societies¹.

In Sect. 5, we informally describe and analyze a hypothetical *social model* that makes use of symbolic environments to computationally represent the core cultural aspects of a particular historical moment of a society (colonial Brazil) where a *syncretic religious formation* was established, and became important for most of its population.

We conclude the paper, in Sect. 7, referring related works and some prospects for the use of *agent societies* and their *symbolic environments* in the effort to develop the notion of *agent culture*.

2. Agent Societies and Social Actors

In our work (see, e.g., [3]), we have been defining in the following way the concept of agent society:

An *agent society* is a multiagent system that is:

- *open*, meaning that agents can enter and leave the system freely;
- *organized*, meaning that the working of the society is constituted essentially by *collective processes* (i.e., processes realized on a collective basis, by *groups of agents*, possibly structured in a hierarchy of nested groups), so that one can determine, in principle, the part that any *individual process* (i.e., process realized on an individual basis, by a single agent) has in any of those collective processes;
- *persistent*, meaning that the system of collective processes that support the organization of the society persists in time, independently

¹Notice that, regarding implementation issues, the concept of *symbolic environment of agent society* can lead to the concept of *cultural artifact*, an extension to all the cultural aspects of agent societies, of the concepts of *organizational* [1] *legal artifact* [2].

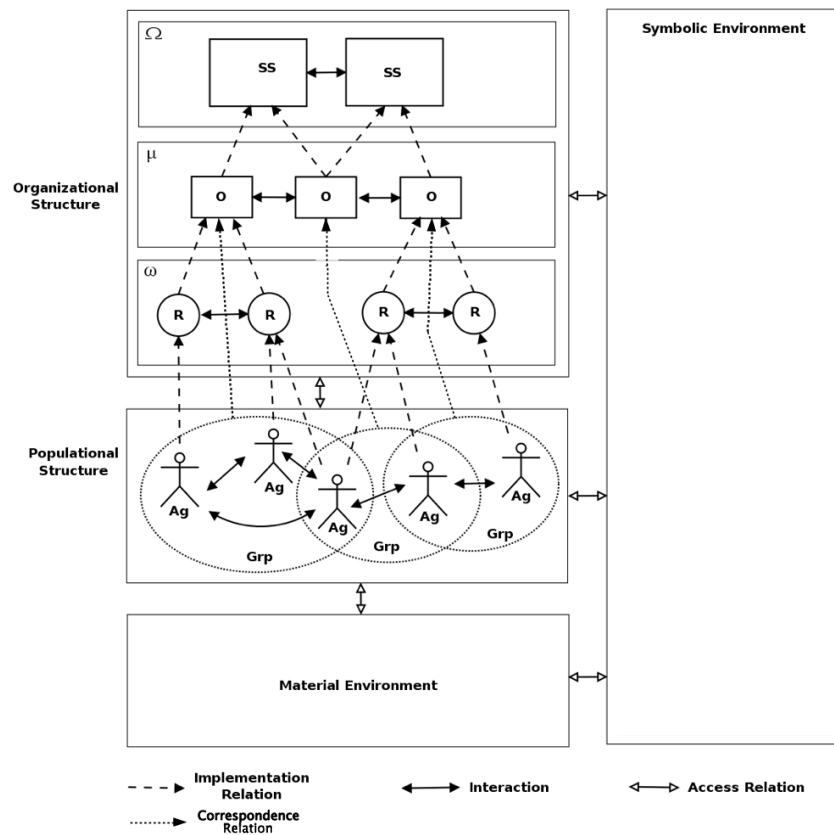


Figure 1. Architectural Overview of the Agent Societies

of which agent enters or leaves the society (up to a certain minimal size of the *population* of the society);

- *situated*, meaning that the agents operate in material and symbolic environments, possibly involving material and/or symbolic objects of those environments in their (individual or collective) processes.

Figure 1 gives a schematic, partial view of the architecture of an agent society (see [3, 4], for more details):

- the figure gives no detail of the *material environment*;
- the *populational structure* is constituted by the agents of the society;
- the *individual* and the *collective* processes are assumed to be articulated in an *organizational structure*, structured in three levels of organization:
 - the *micro-organizational level* (ω), encompassing all the individual and collective processes that agents realize when implementing the *social roles* (R) existent in the society;
 - the *meso-organizational level* (μ), encompassing all the individual and collective processes that groups of agents realize when implementing the *organizations* (O) (i.e., associations, institutions, corporations, etc.) existent in the society;

- the *macro-organizational level* (Ω), encompassing all the individual and collective processes that groups of agents realize when implementing the *social sub-systems* (SS) (i.e., the *inter-organizational* systems) existent in the society.
- we call *social actor* any agent, social role, organization, social sub-system (or groups thereof) which is capable of both driving an individual process of its own, and participating in collective processes involving other social actors;
- of the *social actors*, the figure illustrates individual agents (Ag) and the *groups of agents* (Grp);
- the *symbolic environment* is constituted by the set of *symbolic systems* produced by the social actors of the society, each symbolic system concerning a specific *cultural domain* (moral, legal, political, scientific, religious, etc.) of the society, and by a set of *environment management operations*, such that:
 - each *symbolic system* is constituted by a set of *symbolic formations*, and by a set of *formation operations*;
 - each *symbolic formation* is constituted by a set of *symbolic frameworks*, each symbolic framework encompassing an articulated set of *symbolic items*, and by a set of *framework operations*;
 - the *formation operations* and the *framework operations* are used by the social actors to transform symbolic formations and frameworks, respectively;
 - the *environment management operations* are used by the social actors to create and delete symbolic systems, formations, and frameworks, as well as for other purposes, as explained in Sect. 3;

We say that a conduct of a social actor is a *symbolic conduct* whenever it involves an operation that refers to any of the components of the symbolic environment $SEnv^t$.

Formally, we state:

Definition 1 *An agent society is a time-indexed structure*

$$AgSoc^t = (Pop^t, Org^t, MEnv^t, SEnv^t)$$

where, at each time $t \in T$:

- Pop^t is the current populational structure;
- Org^t is the current organizational structure;
- $MEnv^t$ is the current material environment;
- $SEnv^t$ is the current symbolic environment;

For the sake of space, we leave undefined the details of much of such structure (see [3, 4]). What interests us, here, is the symbolic environment $SEnv^t$ and the symbolic systems that compose it, which we formally detail in the next section.

3. Symbolic Environments of Agent Societies

The concept of *physical symbol system* was introduced by Allen Newel and Hebert Simon, in [5], to capture in an abstract way the idea of the necessary physical substrate of symbol-based systems endowed with the property of “general intelligence” (in other words, the

necessary physical substrate of minds that can perform general problem-solving processes, see also [6]).

Later, to functionally account for the logical and epistemological relations between the contents represented in the physical symbols systems and the constraints involved in rational problem-solving processes, Newell construed, in [7], a higher conceptual level, the so-called *knowledge level*.

The notion of *symbolic environment of agent society*, which we are detailing in this section, should be taken as an extension to the notion of physical symbol system. It is conceived as the architectural component of agent societies that can give, to articulated groups of social actors (so, even to the society as a whole) the possibility of *collective rational intervention* in the solving of the problems that may be affecting them.

The *cultural level* [8] of the society can then be understood to correspond to the “knowledge level” of the society, realized for a collective purpose through the society’s *symbolic environment* (see [9], for instance, for the idea of the rooting of the culture of a society in its collective symbolization of its collective knowledge).

The formal concept of a symbolic environment that is *operant* in an agent society assumes that:

- its *symbolic systems* are endowed with a *physical substrate*;
- its *symbols* are physical patterns that can occur on the physical substrates of the symbolic systems; their universe is denoted by **Symb**;
- the structures of *symbolic formations*, *symbolic frameworks* and *symbolic items*, which are kept formally undefined here are denoted by **SForm**, **SFrmwk** and **SItem**, respectively, with²:
 - **SForm** $\in \wp(\mathbf{Frmwk})$;
 - **SFrmwk** $\in \wp(\mathbf{SItem})$;
 - **SItem** $\in \wp(\mathbf{Symb})$;
- **Env** denotes the universe of possible *environments* (material or symbolic);
- **Symb** is assumed to have enough symbols to *refer to all the material objects* that may occur in the material environments; that is, if **MObj[Env]** denotes the set of all the *material objects* that may occur in the material environments existent in **Env**, then it holds that **MObj[Env]** $\in \wp(\mathbf{Obj})$;
- $T = 0, 1, 2, \dots$ denotes the *time structure*, linearly ordered.

With such assumptions, we let (see Fig. 1, cf. [5]):

Definition 2 A symbolic environment is a *time-indexed structure*:

$$SEnv^t = (Symb^t, SSys^t, FormOp^t, FrmwkOp^t, dsynt^t, EnvMngOp)$$

where, for each $t \in T$:

- $Symb^t \in \wp(\mathbf{Symb})$ is the *current set of symbols present in the symbolic environment*;
- $SSys^t = (SForm^t, SFrmwk^t, SItem^t)$ is the *current set of symbolic systems*, where:

² $\wp(X)$ is the powerset of the set X .

- $SForm^t \in \wp(SFrmwk^t)$ is the current set of symbolic formations;
- $SFrmwk^t \in \wp(SItem^t)$ is the current set of symbolic frameworks;
- $SItem^t \in \wp(Symb^t)$ is the current set of symbolic items;
- $FormOp^t \subseteq (SForm^t)^n$ (for $n = 0, 1, 2, \dots$) is the set of formation operations that can be realized in $CSSys^t$, so that if $op = (s_1, s_2, \dots, s_n) \in FormOp^t$ we say that s_n is the symbolic formation that results from the realization of the formation operation op on the tuple of symbolic formations $(s_1, s_2, \dots, s_{n-1})$, at the time t (and so that if $n = 0$ then op is a constant value: $op = s_0$);
- $FrmwkOp^t \subseteq (SFrmwk^t)^n$ (for $n = 0, 1, 2, \dots$) is the set of framework operations that can be realized in $SEnv^t$, so that if $op = (s_1, s_2, \dots, s_n) \in FrmwkOp^t$ we say that s_n is the symbolic framework that results from the realization of the framework operation op on the tuple of symbolic frameworks $(s_1, s_2, \dots, s_{n-1})$, at the time t (and so that if $n = 0$ then op is a constant value: $op = s_0$);
- $dsgnt^t \subseteq \mathbf{Symb} \times SComp^t$, where $SComp^t$ is the set of symbolic components of the symbolic environment $SEnv^t$:

$$SComp^t = SSys^t \cup \{SForm^t \mid SForm^t \text{ belongs to some } S \in SSys^t\}$$

$$\cup \{SFrmwk^t \mid SFrmwk^t \text{ belongs to some } SForm^t \text{ of } SSys^t\}$$

$$\cup \{SItem^t \mid SItem^t \text{ belongs to some } SFrmwk^t \text{ of } SFrmwk^t\}$$
 is the current designation relation, such that if $(s, C) \in dsgnt^t$, we say that the symbol s designates, at the time t , the symbolic component C ;
- $EnvMngOp$ is the fixed set of management operations that can be performed on the set of symbolic systems $Symb^t$ (such as the creation and deletion of symbolic systems).

4. The Ideological Frameworks of Agent Societies

Ideological framework is the term we have used in [10] to refer to the set of *social views, values, norms, and behavioral patterns* that social actors use, in an agent society, to guide the behaviors, decisions and social assessments they perform in that society.

To formally present ideological frameworks, we have specified in [11] the elements of TinyIML, a toy language for the formal modeling of the *ideological aspects* of agent societies.

In this section, we show how *ideological frameworks* specified in TinyIML can be embedded in the *symbolic environments* of such societies.

4.1. The Ideology Modeling Language TinyIML

The toy language TinyIML was introduced in [10, 11] to illustrate a way to represent the ideological aspects of societies that are computationally modeled in terms of agent societies.

The following are the main components of a TinyIML ideological model:

- *ideological envisagement*: a particular way to view, in an ideological way, a particular aspect of an agent society (e.g., a *segmenting envisagement* determines a particular ideological way of segmenting the population of the society into a set of *groups* or *classes*);
- *ideological framework*: an articulated combination of ideological envisagements, capable of providing a behavioral orientation, regarding social issues, to the social actors that adopt it;

- *ideological system*: the set of all ideological frameworks present in a given agent society;
- *ideology*: the set of ideological envisagements common to all the ideological frameworks adopted by a social actor, characterizing the general orientation of the typical social behaviors of that social actor.

The concrete syntax of TinyIML was informally presented in [11]. It is also informally presented and used in this paper (in section 5).

4.2. Embedding Ideological Frameworks in the Symbolic Environments of Agent Societies

The following establishes the correspondence between the *ideological aspects* of an agent society and the *components* of its symbolic environment:

- *symbolic items* are used to represent *ideological envisagements*;
- *symbolic frameworks* are used to represent *ideological frameworks*;
- *symbolic formations* are used to represent *ideologies*;
- *symbolic systems* are used to represent *ideological systems*;
- the whole *cultural level* of the agent society is represented by the whole *symbolic environment*.

5. Case Study: A Symbolic Environment Embedding an Elementary Ideological System in an Agent Society

The case study in this section concerns the issue of the *religious syncretism* that happened between Catholicism and African Cults, among the Afro-Brazilian slaves in Colonial Brazil (see slides presented in [12]).

The case study intends to present, in a more concrete way, the abstract concept of *symbolic environment* of agent societies that was introduced above.

5.1. The Social-Economic Context of Slavery in Colonial Brazil

Slavery (as a relation of production) was first established in Brazil early in the 16th century, against the indigenous peoples. Slavery of the Africans, supported by intense Portuguese and British slave traffic across the Atlantic ocean, began in the 17th century. The slavery of indigenous people was officially abolished in the 18th century. Slavery of Africans and Afro-Brazilians rested legal until the end of the 19th century.

The main type of social-economic organization that made use of slave labor, in Colonial Brazil, was sugar cane plantations [13]. Each plantation, in general, formed a mostly closed small “society”, due to the usual geographical distance that they kept from each other and from the villages and cities with which they negotiate [14], and which served as brokers for the commerce with Lisbon, the royal metropolis.

There are many aspects of the daily life of such *plantation societies* that one could choose to formally model. As Gilberto Freyre has shown, in his classical study of the societies of the Brazilian colonial plantations [15], strong issues such as identity, genre, and moral sexuality, were present in the daily interaction between white masters and black slaves, in those plantation societies. And the ways those issues were handled by the slaves,

at the time, became historically important because they forged a *black culture* that, later, infused the culture of most of the Brazilian society.

As indicated by the title of this case study, we have chosen to look, here, at the *religious syncretism* that the slaves developed there, integrating Catholicism and their own African cults. The main question we shall face in this case study is this:

If we model sugar cane plantation societies as agent societies, what sort of information can be represented about the possible religious beliefs and attitudes of the social actors of the plantation societies in the symbolic environments of those agent societies?

5.2. The Religious Syncretism of the Afro-Brazilian Slaves

Africans and Afro-Brazilians that were submitted to slavery, developed a whole repertoire of expedients in attempting to make their original cultures survive slavery. Among them (one that eventually developed a particularly strong impact on the overall culture of the country) was the *religious syncretism* between the Catholicism of their masters, and their own original African Cults.

The general central feature of such syncretism was the identification of each of the deities of the African cults (called *orishas*, in the widely spread *Candomblé* cult) with a saint of the Catholic Church. The most important reason, commonly reported [16, 17], for that identification was that it allowed the slaves to formally disguise the workshop of the African deities under the appearance of a workshop of Catholic saints.

In the following, we make use of the structure of the *symbolic environment of agent society*, introduced in Sect. 3, to model the basic ideas forming that religious syncretism. This should allow for both a concrete view of what we mean by a *symbolic framework*, a *symbolic formation*, and a *symbolic system*, and the *ideological system* of an agent society.

5.3. Outline of the Formal Model of the Religious Formation of Typical Plantation Societies of Colonial Brazil

In this section, we assume as the object of the case study the *religious formation* of typical plantations of Colonial Brazil. Figure 2 illustrates the overall plan of the typical plantations, with: the master's house (M), the senzala (S), the slaves overseer's house (O), the road to the plantation area (P), the road to the village (V), the backyard of the senzala (B).

Figure 3 shows the result of the formal modeling, making use the TinyIML notation [11], of the *religious formation* of that plantation. Two *ideological frameworks*, already mentioned in Sect. 5.2, constitute together the religious formation: the religious framework of the typical *slaves* (denoted by RFrm_Slaves) and the religious framework of the typical *masters* (denoted by RFrm_Masters). We have not modeled, however, the religious framework of the typical *slave overseers* (SOverseers) of the plantations.

A few general comments are in order about Fig. 3. About TinyIML in general:

- the following denote *set operations and relations*:
 - “= \subset ” denotes the relation of *set inclusion*;
 - “ \wedge ” denotes the operation of *set intersection*;
 - “{ }” denotes the *empty set*;
- the following are types of ideological envisagements:

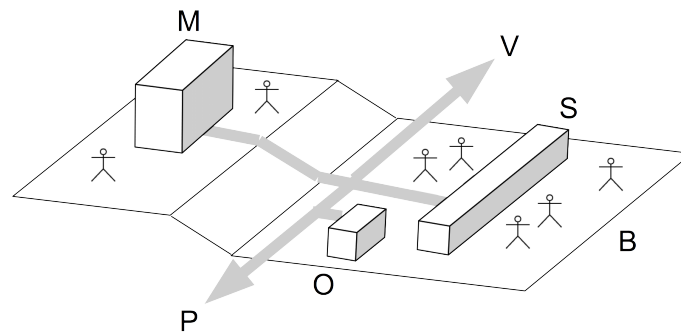


Figure 2. Basic plan of a typical Brazilian sugar cane plantation, with the master's house (M) overseeing the whole area.

- "SegmEnvis": segmentation envisagements;
- "NormEnvis": normative envisagements;
- "ValuatEnvis": valuation envisagements;
- the following are specific notations:
 - "TypeCndts": the keyword for a list of *types of conducts*;
 - "<": the order relation *less than*;
 - "<<": the order relation *much less than*;
 - "<_C": the comparison relation regarding the competence "C";

About the religious model itself:

- ACFollowers are the followers of African cults, SocActors are the social actors of the plantation society;
- Masters and Slaves coincide in the way they segment the set of social actors, including the way Masters and Slaves relate to Catholics and ACFollowers;
- but masters and slaves diverge about the possibility of entities that are both Saints and Orishas;
- in consequence, they diverge about the possibility of someone being both a Catholic and an ACFollower;
- Slaves consider Overseers as Masters, while Masters themselves do not;
- for Slaves, who work, *religious practices* are more important than *work*;
- for Masters, who don't work, *work* is more important than *religious practices*;
- Slaves are much more liberal than Masters, regarding *norms* that determine admissible *religious practices*;
- while Slaves see themselves as more competent than Masters in religious matters (i.e., being more faithful), regarding both Catholicism and African cults, Masters see themselves as better Catholics than Slaves, while acknowledging that Slaves are better ACFollowers than themselves;
- the *religious frameworks* of each type of social actor, RF_Slaves and RF_Masters, simply aggregate their respective *religious envisagements*;
- the *religious formation* of the plantation society is modeled by the set of those two *religious frameworks*.

Notice that we have considered here that *religious frameworks* are composed of four *religious envisagements*:

```

%-----
SegmEnvis SE_Slaves:
  Slaves, Masters, SOverseers, Saints,
  Orishas, Catholics, ACFollowers
    =< SocActors
  Slaves =< ACFollowers
  Masters =< Catholics
  SOverseers =< Masters
  Saints = Orishas
  Catholics ^ ACFollowers = {}
SegmEnvis SE_Masters:
  Slaves, Masters, SOverseers, Saints,
  Orishas, Catholics, ACFollowers
    =< SocActors
  Slaves =< ACFollowers
  Masters =< Catholics
  Saints ^ Orishas = {}
  Catholics ^ ACFollowers = {}
%-----
NormEnvis NE_Slaves:
  TypeCndts: Pray-for(_), Dance-for(_)
  permit (ACFollowers, Pray-for(Orishas))
  permit (ACFollowers, Dance-for(Orishas))
  permit (ACFollowers, Pray-for(Saints))
  permit (ACFollowers, Dance-for(Saints))
%-----
NormEnvis NE_Masters:
  TypeCndts: Pray-for(_), Dance-for(_)
  permit (Catholics, Pray-for(Saints))
  prohib (Catholics, Pray-for(Orishas))
  prohib (Catholics, Dance-for(Saints))
  prohib (Catholics, Dance-for(Orishas))
%-----
ValuatEnvis VE_Slaves:
  TypeCndts: Pray-for, Dance-for, Work
  Work < Pray-for
  Work < Dance-for
ValuatEnvis VE_Masters:
  TypeCndts: Pray-for, Dance-for, Work
  Pray-for << Work
  Dance-for << Work
%-----
QualifEnvis QE_Slaves:
  QualifRel: <_Work
  Masters <_Work Slaves
  QualifRel: <_Pray-for(_)
  Masters <_Pray-for(Orishas) Slaves
  Masters <_Pray-for(Saints) Slaves
  QualifRel: <_Dance-for(_)
  Masters <_Dance-for(Orishas) Slaves
  Masters <_Dance-for(Saints) Slaves
QualifEnvis QE_Masters:
  QualifRel: <_Work
  Masters <_Work Slaves
  QualifRel: <_Pray-for(_)
  Slaves <_Pray-for(Saints) Masters
  Masters <_Pray-for(Orishas) Slaves
  QualifRel: <_Dance-for(_)
  Masters <_Dance-for(Orishas) Slaves
%-----
IdeoFrm RFrm_Slaves:
  SegmEnvis: SE_Slaves
  NormEnvis: NE_Slaves
  ValuatEnvis: VE_Slaves
  QualifEnvis: QE_Slaves
IdeoFrm RFrm_Masters:
  SegmEnvis: SE_Masters
  NormEnvis: NE_Masters
  ValuatEnvis: VE_Masters
  QualifEnvis: QE_Masters
%-----
IdeoSys Plantation:
  IdeoFrm RFrm_Slaves
  IdeoFrm RFrm_Masters

```

Figure 3. Some of the religious frameworks of the social game about colonial Brazil.

- a *religious segmenting envisagement*, which determines a *religious segmentation* of the social actors of the agent society in question;
- a *religious normative envisagement*, which determines *religious norms of conducts* for the religious segments determined by the religious segmenting envisagement;
- a *religious valuating envisagement*, which determines *religious valuation relations* among the possible conducts of the religious segments, establishing a valuating hierarchy among them;
- a *religious qualifying envisagement*, which determines, for the religious segments, their relative qualification (or competence) to perform the various conducts valuated by the given valuating envisagement.

In [11], the religious aspects of the social situation represented in the theatrical play *O Pagador de Promessas (The Keeper of Promises)* were analyzed in terms that are similar to the ones just mentioned, identifying analogous *religious frameworks* and *religious formation* in that situation.

5.4. An Aside: Running the Model in a Hypothetical Simulation Game

Imagine that you, together with several other users, have access to a *cooperative simulation system* concerned with the daily life of a Brazilian colonial sugar cane plantation.

Let the game put the group of users in a participatorily simulated sugar cane plantation, where they can be avatared by some simulated *people* (the population of the plantation) and make use of some *things* (like houses, slave shelters (*senzalas*), animals, farmstead, work tools, etc.). Let the *population* of the sugar cane plantation be composed of the (white) *owner* of the plantation and his family, the *free employees* (including the *slaves overseer*) and their families, and the (black) *slaves* and their families (also slaves).

Consider now that you are playing the game in the role of a *slaves overseer*, and that the *owner* of the simulated plantation is (apparently) a religious man, a faithful catholic, that has commanded you both to impose the Catholic faith and rituals on the *slaves*, and to preclude, using any available means, the realization of non-catholic religious rituals from their part.

Assume that, late in a certain night, you find that all slaves secretly gathered in the backyard of the *senzala*. You go there and you find them dancing in front of the image of a Catholic saint.

You interrogate some of them and you get as their unanimous response that they are worshipping that Catholic saint. You know that the slave owner's religious framework does not acknowledge dancing as a way to worship Catholic saints. But the statue in front of you is the statue of a saint.

The question, then, is: Do you decide to accept the response you got (even knowing that the slaves identified Catholic saints and orishas) and don't punish the slaves, thus effectively taking their side in their religious dispute with the slave owner, or do you stick to the side of the slave owner and punish the slaves for participating in the dancing ritual, with which they were clearly (from the point of view of their owner's religious framework) worshipping an orisha?

Naturally, it is not important, for the case study, which decision you take, in the given situation. The important thing is that the hypothetical simulation software, making use of the *symbolic environment* to model the *cultural level* of the plantation society, is able to store, process, and provide to you, if requested, detailed information about the religious formation of that society, about the religious beliefs and conducts of each of its social actors and, eventually, the possible reactions of the master and of the slaves to the decisions you make and to the actions you perform.

6. Related Work

Classical papers on the modeling and simulation of ideologies are, for instance, Carbonell [18] and Schank and Carbonell [19], which reduce ideologies and ideological frameworks to the individual social actors' systems of goals. More recently, Ophir[20] presented an alternative modeling, where ideologies are immediately tied to the social actors' interactive behaviors.

Ana Paiva and collaborators [21, 22] introduced a *cultural model*, based on a set of *cultural dimensions*, to culturally assess the effects of cultures on organizations and nations, but the set of such ordinally defined cultural dimensions is, as in [20], effectively used to assess the behaviors of individual social actors.

None of those works consider ideologies as socially constructed collective symbolic constructs, which can be variously interpreted by different social actors.

7. Conclusion

Clearly, *ideological systems* can be seen to inhabit both the *minds* of the social actors and the *symbolic environments* of agent societies. In this paper, we have focused on such symbolic environments and how ideological systems can be embedded in them.

We first introduced the notion of *symbolic environment* as a computational means for representing the *cultural level* of agent societies [8] (see also [10] and [11]). Then, we examined a case study that exemplified the use of the symbolic environment of a typical plantation society to computationally support the religious system of that society.

Notice that, since social actors are often not demanded to be *objective* and *rational* (in the epistemological meaning of these words), but only to be *sensible*, the applicability of the concepts of *cultural level* of agent societies (and its realization in *symbolic environments*) seems to depend on the determination of fruitful ways to computationally realize the forms of reasonings (e.g., the *artistic* or the *religious* reasonings) that are employed in areas of culture where objectivity and rationality are not the *sole* (possibly, not even the *primary*) characteristics of the reasoning of social actors.

Finally, we remark that the classical sociological study of the Brazilian colonial sugar cane plantation is Gilberto Freyre's *Casa Grande & Senzala* [15], first published in 1933. For more recent reviews of religious syncretism see, e.g., [16, 17, 23], which also consider other cases of religious syncretism in Brazil.

References

- [1] F.Hübner, J., Boissier, O., Kitio, R., Ricci, A.: Instrumenting multi-agent organisations with organisational artifacts and agents: Giving the organisational power back to the agents. *Journal of Autonomous Agents and Multi-Agent Systems* **20** (2010) 369–400
- [2] Costa, A.C.R.: Situated legal systems and their operational semantics. *Artificial Intelligence & Law* **43** (2015) 43–102
- [3] Costa, A.C.R.: *A Variational Basis for the Regulation and Structuration Mechanisms of Agent Societies*. Springer, Cham (2019)
- [4] Costa, A.C.R., Dimuro, G.P.: A minimal dynamical organization model. In Dignum, V., ed.: *Handbook of Multi-Agent Systems: Semantics and Dynamics of Organizational Models*. IGI Global, Hershey (2009) 419–445
- [5] Newell, A., Simon, H.: Computer science as empirical inquiry: Symbols and search. *Comm. of the ACM* **19** (1976) 113–126
- [6] Newell, A.: Physical symbol systems. *Cognitive Science* **4** (1980) 135–183
- [7] Newell, A.: The knowledge level. *Artificial Intelligence* **18** (1982) 87–127
- [8] Costa, A.C.R.: The cultural level of agent societies. Invited talk at WESAAC 2011 - 5o. Workshop-School of Agent Systems, their Environments, and Applications. Curitiba, Brazil. *Proceedings* (2011) (In Portuguese).
- [9] White, L.A., Dillingham, B.: *The Concept of Culture*. Burgess, Minneapolis (1973)
- [10] Costa, A.C.R.: Situated ideological systems: A core formal concept, some computational notation, some applications. *Axiomathes* **27** (2015) 15–78

- [11] Costa, A.C.R.: Elements for an ideology modeling language. (Submitted for publication) (2015)
- [12] Costa, A.C.R.: Symbolic environments and the cultural aspects of augmented worlds. Slides presented at ECAI-ALAW 2016: 1st International Meeting on Agents Living in Augmented Worlds (2016)
- [13] Wikipedia: Slavery in brazil. Available at: https://en.wikipedia.org/wiki/Slavery_in_Brazil (2010)
- [14] Vainfas, R.: Ideologia e Escravidão. Vozes, Petrópolis (1986)
- [15] Freyre, G.: Casa Grande & Senzala - Formação da Família Brasileira sob o Regime da Economia Patriarcal. Global, São Paulo (2003)
- [16] Turner, N.: Religious Syncretism in Brazil: Catholicism, Evangelicalism and Candomblé. GRIN Verlag (2013) Available at www.grin.com.
- [17] Ferretti, S.F.: Repensando o Sincretismo. Edusp, São Paulo (1995)
- [18] Carbonell, J.G.: POLITICS: Automated ideological reasoning. *Cognitive Science* **2** (1978) 27–51
- [19] Schank, R., Carbonell, J.: Re: The Gettysburgh Address - Representing social and political acts. Yale Univ., Dept. Comp. Sci. (1978) Research Report #127.
- [20] Ophir, S.: Simulating ideologies. *Journal of Artificial Societies and Social Simulation* **1** (1998) Available at: <http://jasss.soc.surrey.ac.uk/1/4/5.html>.
- [21] Mascarenhas, S., Enz, S., Paiva, A.: Using rituals to express cultural differences in synthetic characters. In Decker, K., Sichman, J., Sierra, C., Castelfranchi, C., eds.: AAMAS 8th Int. Conf. on Autonomous Agents and Multiagent Systems,, Budapest (2009) 305–312
- [22] Hofstede, G.: Culture's Consequences: comparing values, behaviors, institutions, and organizations across nations. SAGE Publications, Thousand Oaks (2001)
- [23] Nogueira, J.G.: Sincretismo religioso no brasil em Casa Grande & Senzala: Influências na religiosidade brasileira. *História e-História* (2009) Online at: <http://www.historiaehistoria.com.br/materia.cfm?tb=alunos&id=205>.

The Role of Social Image as a Behavior Filter: A Case Study Based on Cognitive Agents *


Jeferson J. Baqueta, Mariela Morveli-Espinoza, Gustavo A. Gimenez-Lugo and Cesar A. Tacla

¹ Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial
Universidade Tecnológica Federal do Parana (UTFPR)
Curitiba – PR – Brazil

{jefersonbaqueta, morveli.espinoza}@gmail.com,
gustavo@dainf.ct.utfpr.edu.br, tacla@utfpr.edu.br

Abstract. *In multi-agent systems (MAS), computational reputation models have been adopted as an important solution in order to ensure security and efficiency. The evaluation mechanisms, offered by these models, can be used to punish inappropriate behaviors of agents and improve the partner selection process in uncertain situations. However, as the reputation is a shared opinion, in some cases, a given agent can be penalized excessively and no longer be chosen as a negotiation partner, even when he is not a cheater. In this context, in this work, we investigate how the social image, a social evaluation about the characteristics of someone, can be used to avoid very severe punishments, as well, its effects on the decisions taken by an agent about its negotiation partners, assuming that these decisions are initially made based only on reputation of other agents. Our results demonstrate that, in some scenarios, social image may lead an agent to be more flexible during the selection of his negotiation partners.*

Resumo. *Nos sistemas multi-agente (SMA), os modelos de reputação computacionais têm sido adotados como solução importante para garantir segurança e eficiência. Os mecanismos de avaliação, oferecidos por esses modelos, podem ser usados para punir comportamentos inapropriados dos agentes e melhorar o processo de seleção de parceiros em situações de incertezas. No entanto, como a reputação é uma opinião compartilhada, em alguns casos, um determinado agente pode ser penalizado excessivamente e não ser mais escolhido como parceiro de negociação, mesmo esse não sendo um trapaceiro. Nesse contexto, neste trabalho, investigamos como a imagem social, uma avaliação social sobre as características de alguém, pode ser usada para evitar punições demasiadamente severas, assim como os efeitos da imagem social nas decisões tomadas por um dado agente sobre seus parceiros de negociação, assumindo que tais decisões são inicialmente feitas considerando somente a reputação dos demais agentes. Nossos resultados demonstram que, em alguns cenários, a imagem social pode levar o agente a ser mais flexível durante a seleção de seus parceiros de negociação.*

*  O trabalho The Role of Social Image as a Behavior Filter: A Case Study Based on Cognitive Agents de Jeferson J. Baqueta, Mariela Morveli-Espinoza, Gustavo A. Gimenez-Lugo e Cesar A. Tacla está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial 4.0 Internacional. <http://creativecommons.org/licenses/by-nc/4.0/>

1. Introduction

Reputation is a fundamental aspect for human relations, generally it is associated to how much reliable someone is. According to [Sabater and Sierra 2001], reputation can be defined as an opinion of a person about something. In particular, this opinion is formed and updated along the time, either from direct interactions or through information provided by other members of the society. There are several reputation models that translate some reputation concepts from human world to virtual societies [Sabater and Sierra 2001], [Huyhn et al. 2004], [Sabater et al. 2006], [Pinyol et al. 2012] and [Borges et al. 2015].

Reputation models are fundamental to design and implement agent systems [Luck et al. 2005]. According to [Pinyol and Sabater-Mir 2013] a reputation model aggregates important information that agents can use to select their partners. In general, the choices performed by an agent are based on the social behavior of other agents in the society. Hence, the most severe punishment for bad behavior is the social rejection. On the one hand, as discussed in [Sabater and Sierra 2005], this approach contributes to search trustful partners and to avoid cheaters and frauds in virtual societies. On the other hand, according to severity of punishment applied for bad behavior, an agent can be socially excluded without necessarily to be a cheater. For instance, consider a situation, where a given agent has received a series of bad evaluations for failing to perform a task as promised. Thus, a bad reputation about the agent's behavior will spread in the society, and due to this, such agent will be refused as partner in future negotiations. However, if the agent's failures occurred due to the unavailability of resources, in the future, according to the severity of evaluations, the agent may continue being refused, even when he gets the necessary resources to perform the task very well.

In particular, the unfair punishment problem can be minimized by using of social image, since the selection of partners is based on the roles or functions performed by agents in the society. As presented in [Sabater et al. 2006], social image can be defined as a social evaluation consisting of a set of evaluations about the characteristics of a target (someone or something). In this approach, a rating (*e.g.*, good or bad), is associated to a target according to the functions performed by it. The difference between reputation and social image can be exemplified through a commercial scenario, where several customers consider and share each other the information that a given seller is a cheater (reputation), because great part of his products are low quality products. However, for some products, besides good quality, the seller offers the lowest prices in the market. Thus, even if the customers know about the seller's reputation, for some products, the purchase is advantageous due to good image that the customers have about the seller. Notice that, in this case, the seller performs different functions when sells low or good quality products.

In [Sabater et al. 2006] and [Pinyol et al. 2012], the social image and reputation are used simultaneously to provide evaluations on potential negotiation partners in a multi-agent context. In particular, the authors discuss the effects of social image on reputation, and vice-versa, considering the social relations existing among agents. An important consideration adopted by authors is that, both image and reputation can be shared among agents. However, in the case of social image, the speaker agent must commit himself to the veracity of the information transmitted. Furthermore, according to [Conte and Paolucci 2002], social image is a belief usually produced from the direct experiences of the agent, and hence, expressing a more personal opinion about a target. On

the other hand, reputation is defined as meta-belief, since it is produced based on third party opinions that circulate in the society. In this way, reputation can be seen as a general opinion shared by the majority of the member of the society. In this way, reputation can be seen as a general opinion shared by the majority of the member of the society.

In this work, we investigate how the social image can be used to avoid very severe punishments resultant of reputation evaluations, as well, the effects of social image on the decisions taken by agents about their negotiation partners, considering the reputation information spread on the society. Differently of the approach adopted in [Sabater et al. 2006] and [Pinyol et al. 2012], in which image and reputation are applied in a cognitive model, where the agents can join to trust groups and decide when it is advantageous spread or omit the reputation or social image information about someone, we are using the social image as a filter for identify good behaviors, and therefore, good negotiation partners. In particular, in our approach, each agent has its own image about its candidate partners, which can be used to reward or punish such candidates based on the functions performed by them. In our experiments, we notice that when social image and reputation are used together, variations on the behavior of a given agent are more easily noted by other members of the society. Hence, agents penalized in initial negotiations due to bad behavior, may be selected more easily in future negotiations if they improve their behaviors.

The rest of this paper is organized as follow. Section II presents the modeling considerations adopted to compute the reputation and social image values. Section III discusses the details of implementation and the case study used in our experiments. Section IV describes the experimental results and discusses the result obtained. Finally, the conclusions and the future works are summarized in Section V.

2. Reputation and Social Image Modeling

This section reviews some definitions and the main concepts regarding the modeling of the reputation and social image adopted in this work. This modeling was based on Regret [Sabater and Sierra 2001] and FIRE [Huynh et al. 2004], since they also adopt a commercial transactions scenario in their experiments. In our case, as case study, we implement a commercial transactions system, where agents interact each other making use of behavior evaluations that are circulating on the society (reputation), as well as through their own evaluations about the characteristics of a given agent (social image). In particular, a commercial transaction can be seen as an agreement, which is performed between two agents and defines the terms of a contract. Similarly to the Regret and FIRE models, we consider that in a commercial transaction an agent X buys a certain product from an agent Y and after that the agent X rates the agent Y according to the conditions of a contract of service (*e.g.*, product price, product quality, and delivery time).

2.1. Ratings

As defined in [Sabater and Sierra 2001], a commercial transaction is an agreement between two agents that specify the terms and conditions of a commercial contract (*e.g.*, the expected quality and the delivery date of a certain product purchased by someone). Thus, when an agent X buys a product from an agent Y through a commercial transaction, an outcome is created, which represents the contract conditions agreed by both agents, X

and Y . In our case study, an outcome is defined in terms of price, quality, and delivery time of a product. For instance, the following outcome O_{xy} could be the result of a commercial transaction between agent X and agent Y :

$$O_{xy} : product_{price} == 100 \wedge product_{quality} == 80\% \wedge product_{delivery\ time} == 10$$

this means that agent X expects paying 100 dollars by a given product with quality of 80%. Moreover, such product must be delivered within 10 days by agent Y . Notice that when this outcome is produced, an expectation is created by agent X in relation to agent Y . Thus, as discussed in [Castelfranchi and Guerini 2007], agent X believes that agent Y will commit to the terms of contract. Therefore, when the product is delivered by agent Y , agent X , based on its expectations, can evaluate agent Y . In particular, to make such evaluation, agent X must consider its satisfaction with respect to the service provided by agent X [Castelfranchi and Falcone 2010]. Thus, as a result of the evaluation, the appraiser agent, in this case agent X , is going to produce a rating $r : (x, y, i, c, v)$, where x represents the appraiser agent, y the appraised agent, i the time instant in which the rating was produced, c the contract term considered in evaluation (price, quality, or delivery time) and v the value associated to the contract term, which belongs to the interval $[0, 1]$, where 0 means a rating absolutely negative, 1.0 means a rating absolutely positive, and 0.5 means neutral rating.

In conclusion, all ratings produced by a given appraiser agent are stored in its belief base. Thus, the appraiser can memorize its impressions about other agents. In general way, an impression is compound of the ratings given for each contract term agreed in a certain commercial transaction. Therefore, the impression $imp_y^x : (r_{c=price}, r_{c=quality}, r_{c=delivery})$ stores a set of ratings (r) made by the agent x , in a certain instant i , about an agent y , in which a value v is associated to a each one of contract terms c (price, quality, and delivery time). Furthermore, impressions memorized by an appraiser can be shared with other agents in order to spread the reputation of a given target. Therefore, the reputation of a certain agent is built according to impressions about it that are produced and shared among members of the society.

2.2. Reputation

As performed in [Sabater and Sierra 2001] and [Huynh et al. 2004], for computing the reputation, we will use the weighted mean of the impressions. This approach groups a set of impressions together in order to form a single summary value, giving more relevance to impressions received most recently. The equation used to compute the reputation is:

$$R^t(IBD_f^x) = \sum_{\iota_i \in IBD_f^x} p(t, t_i) \cdot W_i \quad (1)$$

where R^t , represents a reputation value in a given time instant t , IBD_f^x indicates all impressions from belief base of agent x filtered by one or more terms of contract (c), which are selected according to a given filter f . For instance, this filter could select all the impressions of an agent x about the agent y , considering as contract terms, the product price and delivery time. Furthermore, $p(t, t_i)$ represents a time dependent function that prioritizes impressions produced closer to time instant t , (e.g., $\frac{t_i}{t}$), and W_i indicates a value (v), within interval $[0, 1]$, assigned to selected contract term (c). A complete discussion and specific details about this equation can be found in [Sabater and Sierra 2001].

Besides reputation value (R^f), we calculate the reliability of reputation, as specified in [Sabater and Sierra 2001]. In particular, from reliability of reputation, it is possible to establish how much truthful the reputation of someone or something is, since the impressions used for computing reputation may come from unreliable sources. Basically, the reliability value is computed taking into account two factors: (i) the number of impressions adopted to calculate the reputation, isolated experience is not enough to make a correct judgement of someone; and (ii) subjective reputation deviation, which aims measure the variability of impressions used in reputation computing. According to [Sabater and Sierra 2001], high variability indicates a low credibility reputation.

2.3. Social image

As discussed previously, image is a social evaluation associated to functions or roles that a given target plays. Furthermore, different members of a society may have different images about the same target. For instance, in a commerce scenario, agent x may have the social image that the prices of books sold by agent y are fair (agent x has a good social image of agent y as book seller). On the other hand, at the same time, agent x may believe that the magazines sold by agent y are very expensive (agent x has a bad social image of agent y as magazine seller). Therefore, for agent x , agent y can be seen from different perspectives, according to the type of product that agent x wants buying. However, notice that, independently of agent x 's image about agent y , in the society, agent y may have a bad reputation as a seller in general way. Besides, it is important to notice that social image is produced through own impressions of an agent with respect to the observed characteristics of a target. Due to this detail, social image as social evaluation, tends to be more sensitive to direct interactions performed among agents than reputation.

In our experiments, the social image concept is employed as a product filter used to select good sellers based on experiences of buyers. In particular, when a buyer purchases a product of a seller, the impression resulting from this transaction is stored in the belief base of the buyer as part of the social image of the seller. However, besides contract terms, this impression has a reference to the product sold by seller. Such detail allows the buyer to filter the sellers by product, instead of just using contract terms. Thus, whether a buyer buys a specific product of a seller, and has a good impression about it, in the future, case this buyer needs buying a similar product, he may select the same seller of last time, independently of reputation of such seller. Therefore, in this approach, sellers who sell well only some products, are only penalized for the bad sales of other products.

The social image is computed from an aggregation of impressions, similar to reputation. Thus, in our experiments, we also use the Equation 1 to compute the social image value. However, differently of reputation, the social image value is calculated using only the own impressions of agent, which are obtained from direct interactions (e.g., the filter f in the Equation 1, could be configured to select just the impressions produced by own agent, this effect is obtained setting the appraiser agent (x) as "self" before to perform a search for impressions on the belief base of the appraiser agent.). Additionally, in this case, the impressions are filtered based on the product desired by the appraiser agent, since every purchased products and their respective sellers are stored in the belief base of buyer agents. Thus, besides contract terms (product price, product quality, and delivery time), in the social image computing, the type of product is also considered.

3. Commercial Transaction System

In this section, we present the implementation details about the commercial transaction system implemented as our case study. In general way, the system consist of two distinct groups of agents, Buyers, represented by set $\mathbf{B} : \{b_1, b_2, \dots, b_n\}$ and Sellers, represented by set $\mathbf{S} : \{s_1, s_2, \dots, s_m\}$. These agent groups interact with each other through buying requests, which are produced from buyers and consumed by sellers. Each buying request results in a commercial transaction, which in turn, generates an impression that is used to compute the social image and reputation values, as appears in Figure 1. Notice that a tuple $R^t(s_i, price, quality, delivery)$ is associated to reputation of the seller s_i , which is computed in the instant t , considering each evaluation criterion (product price, product quality, and delivery time). In turn, a tuple $Image(s_i, product, imp_{s_i}^{b_i})$, associates the seller s_i to a given impression, which is produced in a certain time instant, from a purchase of a specific product. Notice also that the sellers reputations are shared by all buyers, differently of social image, which is particular for each buyer. Moreover, all agents were implemented in Jason [Bordini and Hübner 2005], an interpreter for an extended version of AgentSpeak [Rao 1996].

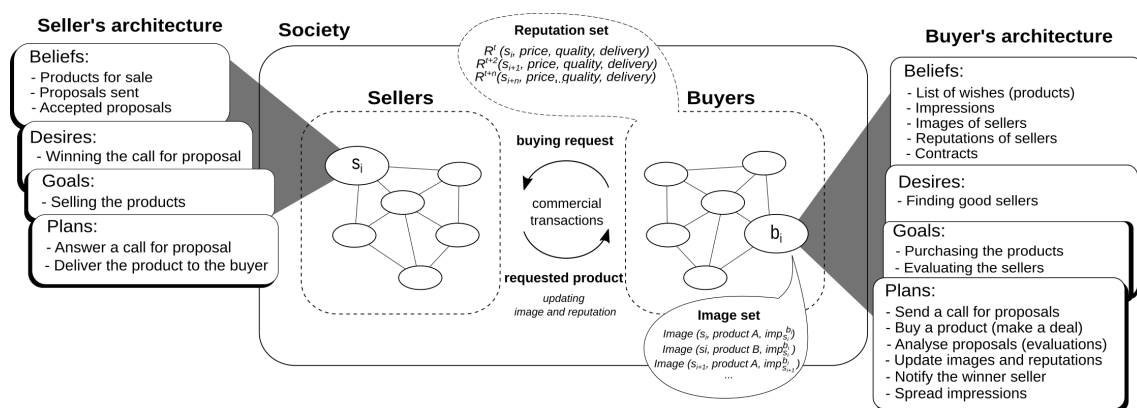


Figure 1. Commercial transaction system, buying request cycle and architectures of the agents

3.1. Buyer Agents

Buyers start the commercial transactions asking for a product. Each buyer $b \in \mathbf{B}$ has its own list of wishes, which consists of products that may or not be acquired as a result of a commercial transaction. In some cases, when there are not sellers who sell the product specified by the buyer, the current transaction is ended and defined as aborted. After that, a new transaction is started and this process is repeated until the list of wishes of the buyer is empty. In particular, the buyers are divided into two types, (i) one-condition oriented, which is focused on just one contract conditions, as product price, product quality, or delivery time. Thus, buyers oriented to price just care about low cost products, ignoring the others contract terms. (ii) multi-conditions oriented, in this case, it is possible to customize the relevance of each contract term. For instance, the following relevance tuple $p_{b_i} : (price = 1, quality = 0, delivery\ time = 0.5)$, indicates that for buyer b_i the product price term must be defined as the highest priority, the product quality is not important, and the delivery time is partially relevant.

3.2. Seller Agents

Each seller $s \in \mathbf{S}$ has its own list of products for sale, which may or not be exclusive. The main goal of a seller is to try to satisfy the buying requests that come from buyers. However, it is possible that more than one seller can meet a given buying request. In this case, a race condition occurs, which can be solved considering the buying orientations of the buyer (price oriented, quality oriented, or delivery time oriented), or ranking the sellers based on their reputation and social image. In general, sellers may lie, in this case, the seller changes the contract conditions agreed with buyer and deliver the product out of initial specifications. The severity of the lie depends on the profile of the seller and on the product's type, because there are products that are easier to sell than others. In this way, the more difficult it is to sell the product, the greater is the chance of seller lying. With respect to sellers profiles, there are three types of sellers, (i) good seller, which never lies. Independently of product's type, a good seller always fulfils the contract conditions; (ii) bad seller, which always lies, indifferently of product's type; and (iii) general seller: this seller has its behavior defined according to type of product added in its list of products.

3.3. Social Interactions

In order to ensure the correct functioning of commercial transactions, the contract net protocol (CNP) was implemented. The CNP was proposed by [Smith 1980] and is used to allocate tasks among autonomous agents. In particular, the protocol starts when a buyer $b \in \mathbf{B}$ receives a perception from the environment about the product that must be purchased. In this moment, a call-for-proposal (*cfp*) is sent to all sellers in the society. In this call, the buyer b_i specifies the product that want to buy and the identifier of call (id), since several *cfp* may be triggered at the same time by different buyers. When a seller $s \in \mathbf{S}$ receives a *cfp*, and has the requested product, such seller sends a proposal to the respective buyer; otherwise, he does not answer the *cfp*. The sent proposal contains the contract conditions defined by the seller, which may be changed in the future without consent of the buyer. In turn, upon receiving all proposals, the buyer b analyzes each one and selects the best one based on its profile, the seller's reputation and the seller's social image with respect to requested product. Finally, after the seller delivers the product, based on its expectations, the buyer evaluates the seller and produces its impression, which is shared with the others buyers.

4. Experiments and Results

In this section we present our experimental study. In particular, all experiments presented herein are based on buying and selling scenarios, where buyers interact with sellers in order to purchase some specific products. In our analysis, we are considering two distinct situations. In the first one, the purchases performed by buyers are guided the buying requirements (*e.g.*, low price, good quality, and short delivery time) and based on reputations of sellers. In turn, in the second situation, besides buying requirements and reputation information, we are also considering the social image influence on the decisions taken by buyers about their partners.

4.1. Sale Patterns

As discussed previously, the social behavior of the sellers is influenced by the type of products sold by them. Thus, in order to simulate different sale patterns, behavioral functions are associated to the products sold by sellers. Above all, a behavioral function

defines how much a seller must change the original contract conditions of a given product before deliver it to the buyer. Therefore, according to the behavioral function adopted, the commitment of sellers with the original contract conditions may be changed along the time. Moreover, notice that the behavioral functions affect directly the reputation of sellers, since a seller may become a cheater due to lack of commitment with its promises. The behavioral functions used in our experiments are described below:

- *Constant*: products associated to this kind of function are easy to sell. Thus, the seller does not need to lie. Therefore, sellers fulfilment the contract conditions, delivering the product to buyers as promised.
- *Increase-function*: this function represents the situation where a seller starts his business very well (not lying), but along the time, the seller needs lying in order to keep himself on the market. In this case, sellers tend increasing the value of one or more contract conditions before deliver the product to buyer, since the resulting value of equation is multiplied by the contract conditions. The equation adopted to simulate this behaviour is the following:

$$f_{INC}(i) = \frac{\log_2 i}{n} \quad (2)$$

where n represents the number of buying requests performed by buyers. In particular, every buyers perform the same number of buying requests. Moreover, each buying request has an identifier (i) and may be won by only one seller. Therefore, the identifier i belongs to the interval $[1, n]$, where 1 means the first buying request performed by a buyer and n means the last one.

- *Decrease-function*: this function represents a situation where the seller starts lying and then he tries improving his behaviour in the next transactions. The equation adopted is the following:

$$f_{DEC}(i) = C - \frac{\log_2 i}{n} \quad (3)$$

where, similar to the Equation 2, i represents the current transaction and n represents the total of transactions that will be performed. In turn, the constant C indicates the initial value of lie told by the seller in the beginning of experiment. Notice that this value tends to decrease over time. In particular, in our experiments, the values obtained from the functions described by Equation 2 and Equation 3 are translated to the interval $[0, 1]$, where 0 represents no changes on the contract conditions, and 1 represents the maximum change value (doubles the original value of one or more contract conditions).

4.2. Single Seller

In this experiment, a buyer has a list of wishes compounds of 10 products. These products are divided into products of type A and type B , which are purchased in a specific sequence, as shown in Figure 2. In particular, the buyer has a price oriented profile, and hence searches for low price products. Moreover, the buyer negotiates with a single seller. On the other hand, the seller is a general seller that sells products of type A , which is associated to a constant behavioral function, and also, products of type B , which is associated to a high-growth behavioral function.

In Figure 2 (a), all purchases performed by buyer are just based on seller's reputation. Thus, the commercial transactions are established only when the seller's reputation is above of the acceptable reputation limit (0.8). Thus, when the seller's reputation is below of reputation limit, the commercial transaction is canceled and the sale is not performed (aborted sale). In Figure 2 (a), when a product of type B is sold in iteration 4, the seller's reputation, in regard to product price, becomes lower than reputation limit, and this way, no sales are made from this point forward. In turn, when the social image filter is applied, the decisions of buyer become more flexible, as presented in Figure 2 (b). Notice that in iteration 5, the buyer purchases the product B, despite the seller's low reputation. In this case, as the social image of the buyer about the seller, regarding product B, is acceptable (above 0.6), the purchase is performed, as it is shown in Figure 2 (b) and Table 1 (c). In particular, as the impressions used for social image computing come from the own agent, presenting high reliability, we are assuming that the acceptable social image limit is lower than the reputation limit. Another important observation about the use of social image filter, can be notice in iterations 8 and 10, where the sales of the products of type A are made due to high image values.

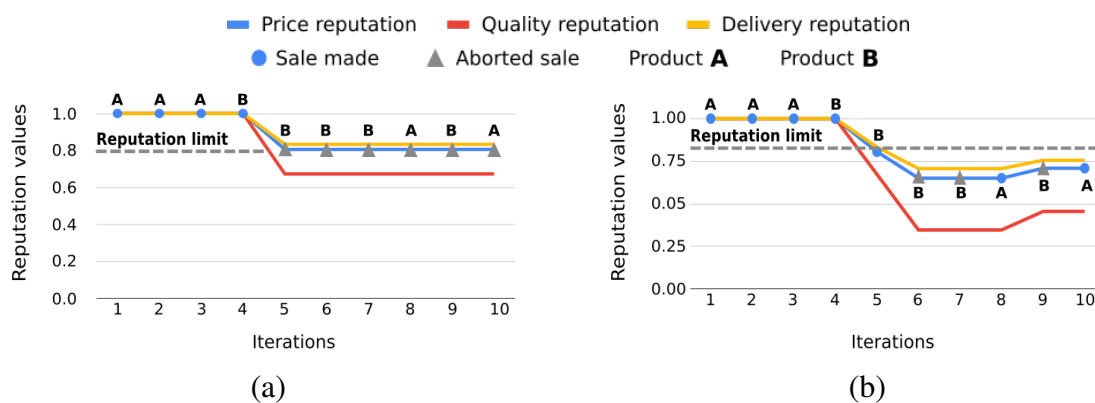


Figure 2. Influence of social image filter on the seller's reputation: (a) sales made by seller taking into account just the reputation values, and (b) sales performed by a seller considering the social image filter.

Iteration	Product	Price image	Quality image	Delivery image
1	A	-	-	-
2	A	1.00	1.00	1.00
3	A	1.00	1.00	1.00
4	B	-	-	-
5	B	0.61	0.35	0.67
6	B	0.56	0.18	0.63
7	B	0.52	0.12	0.58
8	A	1.00	1.00	1.00
9	B	0.47	0.09	0.53
10	A	1.00	1.00	1.00

Table 1. Seller's social image considering the single seller experiment.

4.3. Seller Competitions

In this experiment, a buyer needs to purchase 20 products, which are divided into products of type A and type B, which must be purchased in a specific sequence, similar the previous

experiment. Moreover, the buyer is a general buyer, searching for products with a good trade-off between price and quality. In turn, there are two sellers, *seller 1* and *seller 2*, as presented in Figure 3. Both sellers are general sellers, and this way, the social behavior these sellers are defined according to products sold by them. In particular, both sellers sell the same products, of type *A* and type *B*. In the case of *seller 1*, a behavioral function of type semi-constant is associated to the products of type *A* as well to the products of type *B*, whereas for *seller 2*, a behavioral function of type constant is associated to the products of type *A* and a high-growth function is associated to the products of type *B*. Furthermore, the products sold by *seller 2* are 20% cheaper and with quality 20% better than products sold by *seller 1*.

Notice that when the experiment starts, in Figure 3 (a), the buyer selects the *seller 2* as negotiation partner, since its products are better and cheaper than products sold by *seller 1*. However, when the reputation of *seller 2* becomes lower than reputation limit, between the iterations 7 and 8, the buyer selects the *seller 1* as its new negotiation partner, which is kept until the end of experiment. On the other hand, in Figure 3 (b), when the social image filter is applied, the *seller 2* gets to sell more products of type *B* before the buyer selects the *seller 1* as partner. Moreover, in iterations 18, 19 and 20, the buyer selects the *seller 2* again as negotiation partner due to the high image value of products of type *A*, Table 2 (c). Notice that the sales of products of type *A* increase the reputation of *seller 2*, since good impressions are produced from these sales.

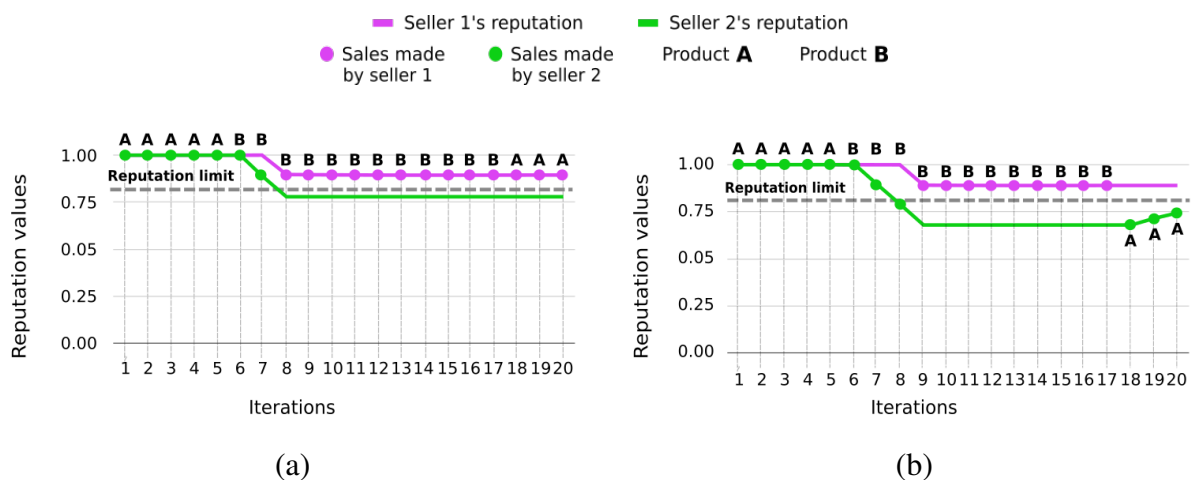


Figure 3. Agents competing for sales: (a) using just reputation values to select the best seller, and (b) influence of social image on partner selection.

Iteration	Product	Seller1's image	Seller2's image
6	B	-	1.00
7	B	-	0.65
8	B	-	0.61
9	B	-	0.57
10	B	0.95	0.57
18	A	-	1.00
19	A	-	1.00
20	A	-	1.00

Table 2. Sellers' social image considering the seller competitions experiment.

4.4. Results Analysis

In our experiments, we remark the use of social image, as behavior filter, makes the agents decisions making, with respect to the selection of their negotiation partners, more flexible than when the agents just consider the information about reputation. Thus, punishments for bad behavior tend to be fairer, since, besides agent's reputation (shared opinions by agents), the functions performed by agent in the society are also considered when such agent is punished.

On the other hand, we also notice that the social image may have a great influence on the reputation of agents, since the impressions produced from decision taken based on social image are shared in the society. This influence may be either positive or negative, depending on the expectations of appraiser agent with respect to service provided by the provider agent. In particular, as presented in Figure 2 (b) and Figure 3 (b), when the buyer selects a seller based on the social image, this decision affects the seller's reputation, because the impressions produced are used to update the reputation status of seller. Thus, whether a sequence of good impressions are produced from decisions based on social image, the seller's reputation can be improved, even if such seller is considered a cheater by other agents (from from a point of view). In turn, in our experiments, as the sellers are filtered by products, the image becomes more sensitive than reputation, since the social image needs of a smaller number of impressions to be computed.

5. Conclusions

In this paper we presented a particular study case, where agents interact each other from commercial transactions. In particular, buyer agents perform buying requests of certain products, and seller agents try to provide the requested product. Sellers are evaluated after the product delivery, where they are rated by the respective buyers. A rating express how much reliable the seller was with respect to fulfilment some contract (product price, product quality and delivery time). For a buyer, a rating is an impression, which is used to update the social image and the reputation of some seller.

Our experiments have demonstrated that variations on the behavior of agents are more easily noted when the social image and reputation are used together. In particular, we notice that social image has a great influence on an agent's reputation, since impressions produced from good experiences associated to social image are used to compute reputation. Therefore, whether a satisfactory number of good impressions is produced based on image of a given agent, the chances of this agent to be chosen as a negotiation partner in the future improve, independently of its reputation. Notice that in this situation, the evaluations based on social image consider the functions performed by agent in the society, besides its reputation.

As future work, we intend to explore the concept of trust in the agent relationships. In particular, when an agent uses the social image to select a given negotiation partner, a trust relationship is defined between the agents, in which an agent delegates a task to its partner and creates expectations about the fulfillment this task. Therefore, it is important that there is a minimum degree of commitment to the part of negotiation partner in order to meet the expectation of appraiser agent.

Referências

- Bordini, R. H. and Hübner, J. F. (2005). Bdi agent programming in agentspeak using jason. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 143–164. Springer.
- Borges, A. P., Botêlho, V., Dordal, O. B., Avila, B. C., and Scalabrin, E. E. (2015). Safety in multi-agent systems: Reputation based on dossier. In *The Twenty-Eighth International Flairs Conference*.
- Castelfranchi, C. and Falcone, R. (2010). *Trust theory: A socio-cognitive and computational model*, volume 18. John Wiley & Sons.
- Castelfranchi, C. and Guerini, M. (2007). Is it a promise or a threat? *Pragmatics & Cognition*, 15(2):277–311.
- Conte, R. and Paolucci, M. (2002). *Reputation in artificial societies: Social beliefs for social order*, volume 6. Springer Science & Business Media.
- Huynh, T. D., Jennings, N. R., and Shadbolt, N. (2004). Fire: An integrated trust and reputation model for open multi-agent systems.
- Luck, M., McBurney, P., Shehory, O., and Willmott, S. (2005). *Agent technology: computing as interaction (a roadmap for agent based computing)*. University of Southampton.
- Pinyol, I. and Sabater-Mir, J. (2013). Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, 40(1):1–25.
- Pinyol, I., Sabater-Mir, J., Dellunde, P., and Paolucci, M. (2012). Reputation-based decisions for logic-based cognitive agents. *Autonomous Agents and Multi-Agent Systems*, 24(1):175–216.
- Rao, A. S. (1996). Agentspeak (1): Bdi agents speak out in a logical computable language. In *European workshop on modelling autonomous agents in a multi-agent world*, pages 42–55. Springer.
- Sabater, J., Paolucci, M., and Conte, R. (2006). Repute: Reputation and image among limited autonomous partners. *Journal of artificial societies and social simulation*, 9(2).
- Sabater, J. and Sierra, C. (2001). Regret: reputation in gregarious societies. In *Proceedings of the fifth international conference on Autonomous agents*, pages 194–195.
- Sabater, J. and Sierra, C. (2005). Review on computational trust and reputation models. *Artificial intelligence review*, 24(1):33–60.
- Smith, R. G. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, (12):1104–1113.

Modelo Ontológico para Significar *Status-Functions* em Instituições Artificiais*

Rafhael R. Cunha¹, Jomi F. Hübner¹, Maiquel de Brito²

¹PGEAS – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

²Departamento de Engenharia de Controle, Automação e Computação
Universidade Federal de Santa Catarina (UFSC)
Blumenau – SC – Brasil


rafael.cunha@posgrad.ufsc.br, jomi.hubner@ufsc.br, maiquel.b@ufsc.br

Resumo. *Em sistemas multiagente, instituições artificiais conectam conceitos abstratos, pertencentes à realidade institucional, aos elementos concretos que compõem o sistema. Neste sentido, a realidade institucional é composta por um conjunto de conceitos abstratos, denominados Status-Functions, e uma série de regras constitutivas que realizam tal conexão. No atual estado da arte, a única finalidade do processo de constituição é relacionar elementos constituídos com a respectiva dinâmica normativa, limitando o raciocínio dos agentes em relação à satisfação de seus objetivos. Diante do exposto, este artigo apresenta um modelo ontológico com o objetivo de definir o significado das Status-Functions de modo a superar essa limitação. Como resultado, são elencadas algumas vantagens que a utilização do modelo oferece e limitações que ainda precisam ser superadas como direções para trabalhos futuros.*

1. Introdução

Sistemas multiagente (SMA) inspiram-se na sociedade humana para tratar de aspectos sociais das interações entre os agentes. Um destes aspectos é a *realidade institucional*, que é a porção da realidade composta de elementos abstratos (ex. compradores, dinheiro, presidentes), constituídos a partir de elementos concretos, tais como pessoas, pedaços de papel, etc. que passam a ter significados e funções que não são inerentes exclusivamente às suas características físicas. Por exemplo, indivíduos compram mercadorias porque algum pedaço de papel recebe o significado e é interpretado (conta como) dinheiro e seguem alguma pessoa porque ela conta como presidente. *O valor está na noção de dinheiro ao invés de estar no papel, a liderança está na posição de presidente e não no indivíduo.* Em outras palavras, o valor está relacionado ao **significado** do elemento abstrato que geralmente é descrito como um conjunto de funções atribuídas ao elemento concreto - **significante** - após ser interpretado como um elemento abstrato.

A noção de realidade institucional tem sido adaptada para ser utilizada em SMA como um meio de interpretação do ambiente, sendo constituída a partir de el-

* O trabalho Modelo Ontológico para Significar Status-Functions em Instituições Artificiais de Rafael Rodrigues Cunha, Jomi Fred Hübner e Maiquel de Brito está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional.<http://creativecommons.org/licenses/by-nc-sa/4.0/>

elementos lá colocados [Brito et al. 2016b]. Por exemplo, agentes agindo no ambiente podem constituir (ou contar como) *compradores* na realidade institucional enquanto as ações desses agentes, realizadas no ambiente, podem contar como *pagamento*. A existência de uma realidade institucional traz ao SMA a noção de *instituição artificial* (ou simplesmente *instituição*) e é utilizada em diversos trabalhos [Cliffe et al. 2006a, Cardoso and Oliveira 2007, Brito et al. 2016a, Fornara 2011] para representar a interpretação social dos elementos presentes no ambiente e servir como base para a especificação e avaliação normativa. Neste momento, é importante introduzir duas noções que compõem a realidade institucional. A primeira é chamada de *Status-Functions*, utilizada para fazer referência aos *status* atribuídos aos elementos concretos e às funções associadas a esses *status*. A segunda, refere-se à noção de *constituição*, que é o processo de constituir a *Status-Function* a partir do elemento concreto.

Ao utilizar uma instituição, é possível escrever normas utilizando *Status-Functions* tais como *o comprador é obrigado a realizar um pagamento*, cobrindo elementos que a instituição considera como *comprador* e ações que são consideradas como *pagamento*. A instituição é responsável por definir quais condições fazem com que um agente seja considerado *comprador* ou uma ação seja considerada *pagamento*. Os trabalhos atuais tratam apenas da identificação das *Status-Functions* e de sua constituição, *sem tratar do significado da Status-Function*. Considere, por exemplo, que o agente *Bob* quer comprar um livro que está sendo vendido por *Tom*. Para isso, (i) *Bob* precisa executar uma ação que signifique *dar um valor em troca de um bem* e (ii) *Tom* espera que *Bob* realize uma ação com esse significado para, então lhe entregar o livro. Os objetivos dos agentes dependem de uma interpretação comum a respeito de certos fatos e, por isso, são chamados *objetivos sociais*. Se um deles não tiver essa interpretação, nenhum deles atingirá seu objetivo. Neste mesmo sistema, uma regra constitutiva define que a *Ação transfer conta como pay*. O significado de *pay* depende exclusivamente da interpretação particular de cada agente. Os agentes podem inferir que *pay* significa *dar um valor em troca de um bem*. No entanto, é arriscado depender dos agentes para tal interpretação. Por exemplo, se no lugar de *pay*, houvesse qualquer outro identificador (ex. *abc*), os agentes não conseguiriam interpretar o significado do *status* atribuído à ação *transfer*. Em outras palavras, não está explícito aos agentes que a função de *pay* significa *dar um valor em troca de um bem* e pode satisfazer os objetivos de possuir um livro e vender um livro respectivamente. A instituição, da forma que é concebida atualmente, não fornece instrumentos para que *Bob* e *Tom* interpretem as ações a partir das consequências que elas produzem no sistema e, portanto, possam concluir também que satisfazem seus objetivos.

Propostas de instituições em SMA foram analisadas, por exemplo em [Fornara and Colombetti 2009b, Brito et al. 2016a, Cardoso and Oliveira 2007, Dastani et al. 2009, Campos et al. 2009] considerando as implementações efetivadas de instituições em sistemas computacionais. Contudo, o aspecto apresentado acima — limitação no raciocínio do agente quanto à satisfação de seus objetivos sociais — permanece sem soluções. Portanto, o objetivo deste trabalho é desenvolver um modelo para expressar o significado das *Status-Functions* que compõem as instituições artificiais presentes no SMA. Para isso, utilizamos como inspiração a teoria da *Documentality* proposta pelo filósofo Maurizio Ferraris [Condello 2018, Condello et al. 2019] que considera a existência de um documento pragmático — no caso deste trabalho um modelo ontológico — que antecede e produz o valor dos conceitos sociais presentes na

realidade institucional.

O artigo está organizado da seguinte maneira: A seção 2 apresenta breves introduções sobre teorias que explicam a realidade social. Na seção 3 é discutido sobre o atual estado da arte desta área de conhecimento. A seção 4 descreve um modelo de ontologia para definir o significado de *Status-Functions*. A seção 5 ilustra um breve exemplo da utilização de SMA ao considerar o modelo proposto. Por fim, na seção 6 são feitas as conclusões sobre a pesquisa e indicações para trabalhos futuros.

2. Teorias Filosóficas

Realidade Institucional é parte de um amplo conceito de realidade social proposto por John Searle [Searle 1995, Searle 2010]. Searle argumenta que há fatos que são explicados pela ciência básica, independente de qualquer atitude mental dos agentes (ex. a água ser composta de hidrogênio e oxigênio). No entanto, há fatos que também são objetivos mas que existem somente porque nós acreditamos em sua existência. Por exemplo, um pedaço de papel e um indivíduo não são considerados, respectivamente, *dinheiro*, e *presidente* em função apenas de suas características físicas. Mais do que isso, esses elementos precisam de *Status-Functions*, que são status que os habilitam a desempenhar funções que não podem ser explicadas por suas virtudes físicas (ex. ser um meio de troca, no caso do *dinheiro*). A origem desses status está relacionada à intencionalidade coletiva dos indivíduos em uma sociedade. Para atribuir a *Status-Function* ao elemento concreto, Searle propõe regras constitutivas com a seguinte fórmula: *X count-as Y in C* (ex. um pedaço de papel *count-as* dinheiro), onde *X* representa o elemento concreto, *Y* a *Status-Function* e *C* o contexto onde aquela atribuição é válida. Para um elemento possuir um status (e desempenhar a função correspondente) é necessário que a sociedade concorde a respeito disso, ou seja, estabeleça um acordo coletivo [Searle 1995, p. 44].

A teoria de Searle, entretanto, não detalha a origem de elementos como *Status-Functions*. Embora sugira que é consequência da intencionalidade coletiva, algumas coisas ainda permanecem sem explicação. Por exemplo, no caso do dinheiro, ao longo da história, concordou-se em dar funções a um pedaço de papel, uma concha ou um saco de sal. Contudo, *a função parece não ter gênese*. Ao refletir sobre dinheiro ou qualquer outro objeto social, dificilmente será possível determinar quando e como ele foi inventado, além da dificuldade ainda maior de explicar a natureza da intencionalidade coletiva que motiva as pessoas a agir de diferentes formas ao ter contato com o elemento concreto constituído com o status. Para resolver esses problemas, Ferraris [Condello 2018, Condello et al. 2019] propõe a observação da realidade social em um nível profundo, menos intuitivo, em que a realidade é formada por estruturas chamadas de *Documentality* — estruturas de documentos que armazenam atos de fala que não apenas descrevem ou prescrevem, mas na verdade constroem objetos sociais. Por exemplo, o dinheiro exerce suas funções nas intenções individuais apenas baseado na recordação (e conseqüentemente no conjunto de funções) *que os objetos sociais resgatam aos indivíduos com base na gravação*. Em outras palavras, os objetos sociais são utilizados para exteriorizar o conjunto de gravações que permitem ao indivíduo recordar as funcionalidades que o *status* (ex. dinheiro) disponibiliza ao estar atribuído em um objeto social (ex. nota de papel). Segundo Ferraris, é o próprio objeto social — através da intencionalidade individual, resgatando as recordações armazenadas em documentos que remetem as funções que ele desempenha — que determina seu valor.

A conclusão através das teorias é que deve haver um sistema adicional de elementos na construção da realidade social para que as funções, valores e status possam persistir e ter um valor reconhecível ao longo do tempo [Condello 2018]. Esta conclusão também parece trazer benefícios no cenário de SMA, em que o sistema adicional indica um caminho útil para compreender as *Status-Functions* que compõem a realidade institucional, permitindo aprimorar o raciocínio de agentes em relação a satisfação de seus objetivos e superar a dificuldade que motiva a realização deste trabalho.

3. Trabalhos relacionados

A ideia principal de usar instituições artificiais como uma contrapartida de instituições humanas em sistemas computacionais tem inspirado trabalhos em SMA. De diferentes maneiras esses trabalhos utilizam a relação *count-as*, estabelecida através das regras constitutivas propostas por Searle, como um componente a ser considerado nas especificações institucionais. A finalidade desta seção é descrever o estado da arte nesta área de conhecimento. Para tanto, a questão a ser respondida nesta pesquisa foi: De que forma as instituições artificiais possuem meios para representar os elementos abstratos que compõem a realidade institucional?

Table 1: Extração de dados dos artigos selecionados.

Papers	Critérios			
	O	F	X	Y
[Fornara 2011]	✓		Ações ou fatos institucionais	Classe <i>InstAction</i> com propriedades como: papel que é responsável pela ação, tempo para executá-la, condição para execução, etc.
[Boella and van der Torre 2004]	✓		Ações e fatos naturais (ex. cerca)	Label para rotular X
[Campos et al. 2009]		✓	Propriedades de agentes, ambiente e Instituição	Valor que representa um objetivo institucional desejável.
[Dastani et al. 2009]		✓	Ações	Condição (ações) e uma consequência normativa.
[Vázquez-Salceda et al. 2008]	✓		Ações	Classifica a ação realizada como um sub-conceito de algum elemento presente no estado constitutivo.
[Cliffe et al. 2006a]	✓		Ações ou eventos institucionais	Label para rotular X
[Aldewereld et al. 2010]	✓		Ações	Label para rotular X

Continued on next page

Table 1 – Continued from previous page

Papers	O	F	X	Y
[Cardoso and Oliveira 2007]	✓		Troca de mensagem entre agentes (Ações)	Commitment
[Viganò and Colombetti 2008]	✓		Ações	Função de status com informações sobre agente, ação e condição normativa.
[Grossi et al. 2006]	✓		Ações	Label para rotular X
[Muntaner and Esteva 2007]		✓	Objetivos dos agentes	Especificação Institucional
[Brito et al. 2016a]	✓		Agentes, eventos (ações) e estados	Label para rotular X

O: Classificação Ontológica

F: Classificação Funcional

X: Elementos Concretos considerados para ser o X da regra constitutiva (*x count-as y in c*)

Y: Significado de Y na regra constitutiva (*x count-as y in c*)

Todos os trabalhos analisados são inspirados na teoria do filósofo John Searle [Searle 1995, Searle 2010] e são marcados por um propósito exclusivamente normativo. Alguns trabalhos apresentam abordagens funcionais, isto é, relacionam fatos brutos a estados normativos (ex. uma determinada ação conta como uma violação de norma). Outros, exibem abordagens ontológicas, em que fatos brutos constituem conceitos abstratos que posteriormente são utilizados na especificação de normas (ex. o envio de uma mensagem conta como um lance em um leilão). Tal predisposição traz algumas consequências discutidas a seguir.

Conforme a Tabela 1, há soluções [Cliffe et al. 2006b, Fornara 2011] que permitem aos agentes conhecerem e raciocinarem sobre regras constitutivas e normativas. De fato algumas especificações institucionais propiciam que agentes interpretem regras a fim de satisfazer condições normativas. Contudo, geralmente a *Status-Function* (*Y*) é somente um *label* atribuído ao elemento concreto (*X*) e utilizado na especificação das normas regulativas. Assim, o *Y* não parece ter outra finalidade além de servir como base para o desenvolvimento de normas regulativas estáveis. Neste sentido, o agente é capaz de raciocinar e entender quais ações podem ser executadas no ambiente para satisfazer as especificações normativas. Entretanto, por não existir um modelo que explicita o significado dessas ações no contexto institucional, o agente pode ter dificuldades em compreender que as ações executadas também podem satisfazer seus objetivos sociais. Por exemplo, o significado *ser um meio de troca* é uma das funções relacionadas a *Status-Function comprar*. Considerando o exemplo anteriormente descrito de comercialização de livros, atualmente não há meios de *Bob* e *Tom* compreenderem que as ações executadas, se interpretadas através de suas *functions*, podem satisfazer também seus objetivos.

A limitação discutida acima parece indicar a necessidade do desenvolvimento de um modelo que explicita o significado das *Status-Functions* pertencentes a realidade in-

stitucional. Deste modo, as limitações apontadas podem ser superadas. Além disso, Aguilar et.al [Rodriguez-Aguilar et al. 2015] corroboram com tal conclusão ao afirmar que instituições ainda não consideraram como ajudar os agentes na tomada de decisões para atingir seus próprios objetivos. Instituições que dão suporte aos raciocínio dos agentes, com o propósito de atingir seus objetivos, é uma questão em aberto para futuras especificações institucionais.

4. Modelo ontológico para significar *Status-Functions* em instituições artificiais

Inspirado na teoria da *Documentality*, considera-se que o significado das *Status-Functions* é expressado através de documentos que, neste caso, são *ontologias*. Em computação, ontologias são representações formais dos conceitos e relações entre eles em um determinado domínio [Baader et al. 2003]. Ontologias contêm três componentes: *Tbox*, *Rbox* e *Abox* [Thomas 2018]. A *Tbox* introduz as classes pertencentes ao domínio da aplicação. A *Rbox* descreve propriedades entre classes (chamadas de propriedades de objeto) e propriedades entre classes e valores de dados (chamadas de propriedades de dados). Por exemplo, a primeira pode definir a relação entre a classe *dinheiro* e a classe *função*. A última, a relação entre a classe *dinheiro* e o tipo primitivo de dados *inteiro*. Por fim, *Abox* contém uma coleção de afirmações que descrevem indivíduos (instâncias de classes) utilizados para representar elementos específicos [Fornara and Colombetti 2009a]. Existem diferentes níveis de abrangência para as ontologias e elas são classificadas em função disso. Neste trabalho desenvolvemos uma ontologia de nível superior. Esta descreve conceitos mais gerais como espaço, tempo, etc. que são independentes de um problema ou domínio particular [Guarino 1998]. Ontologias de nível superior servem de base para ontologias mais específicas, dependentes do domínio da aplicação em que são utilizadas.

Assume-se, neste trabalho, que o significado das *Status-Functions* é fornecido através de um conjunto de funções (*functions*) que também são *Status-Functions*. Isto ocorre porque as *functions* também demandam uma interpretação comum, acordo coletivo, etc. que as caracterizam também como *Status-Functions*. A Figura 1 ilustra o modelo desenvolvido. Na Figura, a mesma classe *Status-Function* que representa o conceito *Status-Function* da realidade institucional, pode representar, juntamente com a relação *hasFunction*, o conjunto de funções de uma *Status-Function*. Por tratar-se de uma proposta de trabalho, estuda-se a viabilidade de considerar também funções associadas a *Status-Functions* (ou seja, seu significado) que não sejam *Status-Functions*. A formalização do significado das *Status-Functions*, inspirado em [Fornara and Colombetti 2009a], ocorre através da relação:

- Propriedade *hasFunction* : *Status-Function* \rightarrow_o *Status-Function*: Propriedade responsável por representar a relação que tem como ponto de origem a classe *Status-Function* e ponto de destino a classe *Status-Function*. Tal relação indica que uma *Status-Function* pode possuir funções associadas que, neste trabalho, também são consideradas *Status-Functions*. Portanto, trata-se de uma auto-relação, ou relação reflexiva, que indica que indivíduos de uma classe se relacionam com indivíduos da mesma classe (não consigo mesmos). As funções atribuídas ao elemento concreto assim que constituído com a *Status-Function*, permitem que o elemento realize ações que não podem ser explicadas por suas virtudes físicas.

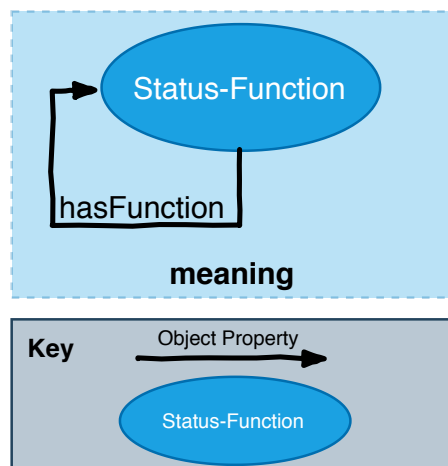


Figure 1. Representação gráfica de uma ontologia para significar *Status-Function*.

Com a definição do modelo, pode-se associar a solução proposta diretamente ao problema dos agentes não conseguirem raciocinar a respeito da satisfação de seus objetivos. Por exemplo, considere-se que a *Status-Function vender* tem um conjunto de funções associadas, dentre elas, a de *transferir um livro*. Considere também que o objetivo do agente *Tom* é *transferir um livro*. Para isso, *Tom* precisa executar uma ação concreta que é interpretada na instituição como *vender*. O conceito de *vender* não faz parte do mundo físico: trata-se da realização de uma ação e da interpretação comum dessa ação pelos demais indivíduos envolvidos na interação. Contudo, *Tom* interpretar que a ação realizada *conta como vender* não é suficiente para concluir que atingiu seu objetivo. Para isso, *Tom* precisa interpretar também as funções associadas, dadas pela relação *hasFunction*, ao conceito de *vender*, que no caso deste exemplo, podem ser resumidas na noção de *transferir um livro*. A partir desta interpretação, *Tom* é capaz de compreender que seu objetivo foi atingido. Cabe lembrar ainda que a mesma ação concreta com interpretação especial pela instituição pode ter diversos significados (*hasFunctions* associadas) que, também são *Status-Functions*, mas podem satisfazer distintos objetivos de um ou mais agentes. Por exemplo, a *Status-Function pagar* que pode ser constituída a partir da ação concreta *entregar uma nota de papel* pode significar *comprar um livro* e *vender um livro* respectivamente. Em um cenário de comércio, a mesma ação pode satisfazer o objetivos de quem está comprando, como também de quem está vendendo, desde que ela seja interpretada através de suas *functions*. A instrumentalização de *Functions* relacionadas à *Status-Functions* permite explicar o significado das *Status-Functions* no contexto das instituições artificiais e, a partir disso, auxiliar os agentes a entender quando seus objetivos são satisfeitos.

O modelo proposto é uma representação abstrata (formada apenas por conceitos e propriedades) que define o significado das *Status-Functions*. Esses conceitos e propriedades (relações) representam a *Tbox* e *Rbox* da ontologia. Para aplicar o modelo em SMA, é necessário desenvolver modelos específicos que incluam a definição de indivíduos e conceitos pertencentes a um domínio de aplicação (parte *Abox* da ontologia). Por exemplo as funções *pagar* e *vender* podem descrever as *Functions* da *Status-Functions dinheiro* em um contexto específico. Por depender do domínio da aplicação, o

desenvolvimento das ontologias específicas deve ser tratado pelo projetista da aplicação. Contudo, a próxima seção apresenta um exemplo de uso do modelo desenvolvido.

5. Exemplo de uso

Para exemplificar o modelo proposto, é considerado novamente o exemplo ilustrado na Seção 1. O agente *Bob* tem o objetivo de comprar um livro (*buy(book)*) vendido pelo agente *Tom*. Adicionalmente, *Tom* tem o objetivo de vender um livro (*sell(book)*). *Bob* e *Tom* atuam em um sistema que define a regra constitutiva *ação transfer count-as pay*. Essa regra é geralmente utilizada para apoiar a descrição normativa no SMA. Entretanto, conforme já discutido, seguir as normas regulativas ou entender quais ações tem um significado na instituição (sem entender também as *functions* vinculadas a essas ações) não é suficiente em alguns cenários para atingir também os objetivos dos agentes. No caso específico deste exemplo, um problema pode ser observado: os agentes não conseguem raciocinar que a ação *transfer*, que conta como *pay*, pode atingir seus objetivos.

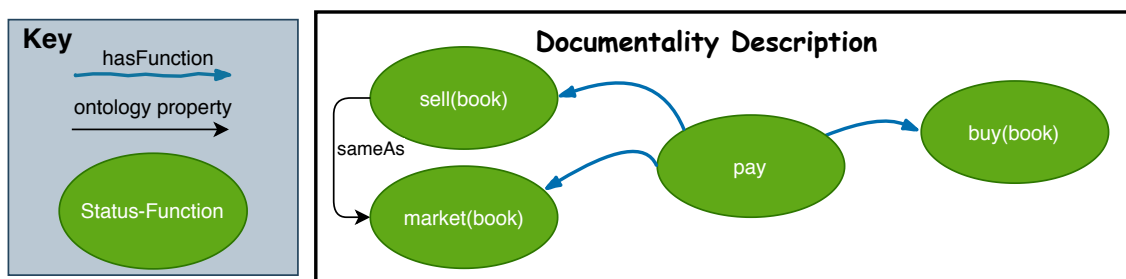


Figure 2. Modelo ontológico de *Status-Functions* no cenário de comércio de livro.

A fim de superar o problema introduzido no exemplo acima, desenvolveu-se uma ontologia específica, ilustrada na Figura 2, para o cenário de comércio de livro. Esta ontologia respeita os conceitos e relações definidas no modelo apresentado na Figura 1. Na Figura 2, a classe *pay* contém um conjunto de funções associadas (*buy(book)*, *sell(book)* e *market(book)*). Esta classe representa uma *Status-Function* e as funções vinculadas à *Status-Function pay* também são *Status-Functions*. A Figura 3 ilustra a especificação do SMA no cenário de comércio de livros. Alguns incrementos podem ser visualizados em relação ao exemplo descrito na Seção 1. A *Normative Description* foi adicionada ao exemplo original. Uma norma regulativa foi inserida com o objetivo de regular o sistema para que os agentes se comportem adequadamente. Por fim, a *Documentality Description* foi incluída, contendo uma descrição textual da representação ontológica ilustrada na Figura 2.

A Tabela 2 simula a execução deste exemplo. Não considerou-se sintaxe de qualquer modelo organizacional ou normativo em particular. O instante de tempo zero, representa o estado inicial do sistema antes de sua execução. A partir do instante de tempo um, o sistema inicia sua execução e as interações ocorrem. Com a introdução da *Documentality Description*, algumas vantagens podem ser elencadas:

- No instante $T=1$, *Bob* consegue compreender que a ação *transfer* é interpretada pela instituição como *pay*. *Bob* também consegue raciocinar a respeito das *hasFunctions* vinculadas à *Status-Function pay*. Neste caso, uma das *hasFunction*

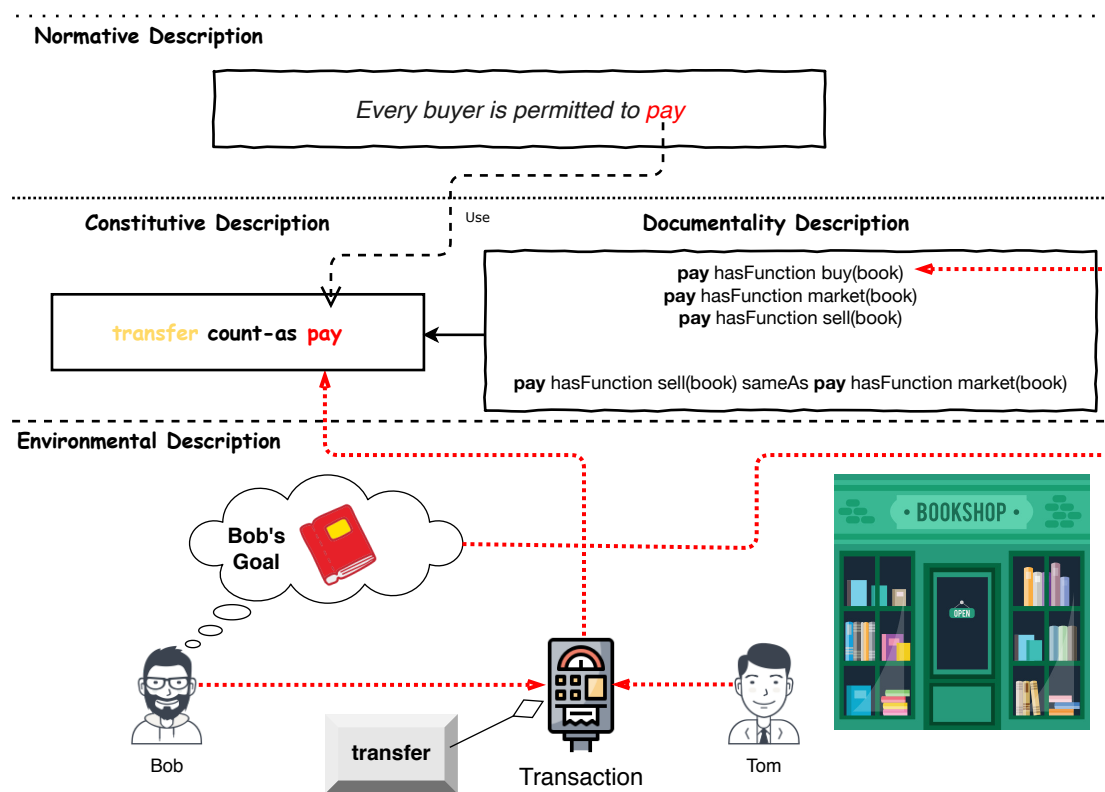


Figure 3. Especificação do SMA do cenário de comércio de livros.

vinculadas a *pay* é *buy(book)*. Portanto, *Bob* consegue compreender que, se executar a ação *transfer*, atingirá seu objetivo.

- No instante $T=2$, *Bob* executa a ação *transfer*, através do plano (*pay(book)*), que é interpretada pela instituição como *pay*. A execução do plano satisfaz a norma vigorando. Além disso, a interpretação de *pay* através de suas *hasFunctions*, permite que *Bob* compreenda que satisfaz seu objetivo, pois uma *hasFunction* de *pay* é *buy(book)*.
- No instante $T=1$, *Tom* consegue compreender que a ação *transfer* é interpretada pela instituição como *pay*.
- No instante $T=2$, *Tom* consegue compreender que a ação executada por *Bob* tem algumas funções associadas, dentre elas, a de *sell(book)*. Neste caso, *Tom* também consegue perceber que seu objetivo foi satisfeito.

A simulação de execução do exemplo acima, por intermédio da Tabela 2, permite perceber que a significação das *Status-Functions* através da definição de um conjunto de *functions* associadas possibilita que instituições artificiais não sejam utilizadas apenas para propósitos normativos. Alguns trabalhos [Cliffe et al. 2007, Vázquez-Salceda et al. 2008, Grossi et al. 2006] já apresentam definições de instituições artificiais com propósitos distintos a questões normativas. Observa-se duas vantagens ao explicitar o significado das *Status-Functions*: (i) agentes não precisam agir condicionados apenas a normas regulativas e (ii) a potencialidade do reuso. A primeira vantagem está relacionada ao aumento de capacidade do raciocínio dos agentes, que, a partir da significação, consegue entender as *functions* vinculadas a *Status-Functions* e tomar decisões baseadas nessas informações. A segunda vantagem está relacionada ao aumento da

Table 2. Simulação da execução do exemplo de comércio de livro.

Time	T = 0	T = 1	T = 2
Bob	Objetivo de: <i>buy(book)</i> ; Plano de: <i>pay(book)</i>	Bob verifica que a ação <i>transfer</i> conta como <i>pay</i> Bob conclui que <i>pay</i> <i>hasFunction buy(book)</i> e, como consequência, conclui que se executar a ação <i>transfer</i> (no tempo T = 2), ele atinge seu objetivo.	Bob executa o plano <i>pay(book)</i> e realização a ação <i>transfer</i> ; Norm <i>every buyer is permitted to pay(book)</i> é satisfeita. Bob conclui o objetivo de <i>buy(book)</i> .
Tom	Objetivo de: <i>sell(book)</i>	Tom verifica que a ação <i>transfer</i> conta como <i>pay</i>	A partir da informação apresentada na <i>Documentality</i> , Tom verifica que a ação realizada por Bob (<i>pay</i>) tem <i>hasFunction sell(book)</i> e portanto também atinge seu objetivo.

possibilidade de reuso. Por exemplo, considere o desenvolvimento de outro SMA também relacionado a um cenário de comercialização de produtos. Os conceitos e definições dos significados de *Seller*, *Buyer*, etc. parecem ser igualmente importantes em cenários distintos. Ao passo que exista uma definição do significado das *Status-Functions*, torna-se mais fácil e prático reutilizar especificações prontas a fim de promover a interoperabilidade e a agilidade no desenvolvimento de projetos de software.

6. Considerações finais e trabalhos futuros

O objetivo desta pesquisa foi desenvolver um modelo para explicitar o significado das *Status-Functions* que compõem as instituições artificiais em SMA. Este objetivo tem como finalidade superar o problema da *limitação no raciocínio do agente quanto à satisfação de seus objetivos sociais*. O problema é provocado pelo fato de as instituições não fornecerem meios para o agente entender o significado das *Status-Functions* que compõem a realidade institucional. Neste trabalho, este significado é dado por um conjunto de *functions* que são atribuídas juntamente com a *Status-Function* ao elemento concreto. Assume-se também que as *functions* relacionadas à *Status-Function* são consideradas *Status-Functions*. Essa rede de *Status-Functions* foi inspirada na teoria de Ferraris introduzida na Seção 2. O modelo proposto captura a atribuição do significado para as *Status-Functions* através da classe *Status-Function* e da relação *hasFunction* incorporadas dentro de uma ontologia descrita neste capítulo. Logo, a ontologia permite relacionar significados a *Status-Function*. Por intermédio das *functions*, os agentes são capazes de entender o significado que a *Status-Function* possui no sistema. Neste sentido, se o raciocínio dos agentes considerar a interpretação das *hasFunctions* associadas as *Status-Functions*, o problema relacionado a limitação de raciocínio dos agentes pode ser superado.

Algumas direções para trabalhos futuros podem ser sugeridas como (i) integração do modelo proposto com mecanismos já existente (JASDL [Klapiscak and Bordini 2008],

JAsA [Grandi 2011], etc), afim de disponibilizar os dados presentes na *Documentality Description*; (ii) desenvolvimento de um repositório WEB que armazene as descrições de *Status-Functions* já desenvolvidas, com a finalidade de permitir o reuso e (iii) a utilização do modelo proposto em outros cenários concretos para levantar questões não observadas pelos autores.

7. Agradecimentos

Os autores agradecem ao Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) pelo fomento a esta pesquisa.

References

- Aldewereld, H., Álvarez-Napagao, S., Dignum, F., and Vázquez-Salceda, J. (2010). Making norms concrete. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 807–814. International Foundation for Autonomous Agents and Multiagent Systems.
- Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., Nardi, D., et al. (2003). *The description logic handbook: Theory, implementation and applications*. Cambridge university press.
- Boella, G. and van der Torre, L. (2004). Regulative and constitutive norms in normative multiagent systems. *KR*, 4:255–265.
- Brito, M. d. et al. (2016a). *A model of institucional reality supporting the regulation in artificial institutions*. PhD thesis, Universidade Federal de Santa Catarina.
- Brito, M. D. E., Thévin, L., Garbay, C., Boissier, O., and Hübner, J. F. (2016b). Supporting flexible regulation of crisis management by means of situated artificial institution. 17(4):309–324.
- Campos, J., Sanchez, M., Aguilar, J., and Esteva, M. (2009). Formalising Situatedness and Adaptation in. pages 126–139.
- Cardoso, H. L. and Oliveira, E. (2007). Institutional Reality and Norms: Specifying and Monitoring Agent Organizations. *International Journal of Cooperative Information Systems*, 16(01):67–95.
- Cliffe, O., De Vos, M., and Padget, J. (2006a). Answer set programming for representing and reasoning about virtual institutions. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 60–79. Springer.
- Cliffe, O., De Vos, M., and Padget, J. (2006b). Specifying and reasoning about multiple institutions. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*, pages 67–85. Springer.
- Cliffe, O., De Vos, M., and Padget, J. (2007). Answer Set Programming for Representing and Reasoning About Virtual Institutions. pages 60–79.
- Condello, A. (2018). Two questions on the ontology of money. *Ardeth*, (03):181–191.
- Condello, A., Ferraris, M., and Searle, J. R. (2019). *Money, Social Ontology and Law*. Routledge.

- Dastani, M., Grossi, D., Meyer, J.-J. C., and Tinneimeier, N. (2009). Normative multi-agent programs and their logics. In *Knowledge Representation for Agents and Multi-Agent Systems*, pages 16–31. Springer.
- Fornara, N. (2011). Specifying and monitoring obligations in open multiagent systems using semantic web technology. In *Semantic agent systems*, pages 25–45. Springer.
- Fornara, N. and Colombetti, M. (2009a). Ontology and time evolution of obligations and prohibitions using semantic web technology. In *International Workshop on Declarative Agent Languages and Technologies*, pages 101–118. Springer.
- Fornara, N. and Colombetti, M. (2009b). Specifying and Enforcing Norms. (2204):1–17.
- Grandi, A. (2011). Jasa: Semantically aware agents to improve adaptivity and context-awareness of smart environments.
- Grossi, D., Aldewereld, H., Vázquez-Salceda, J., and Dignum, F. (2006). Ontological aspects of the implementation of norms in agent-based electronic institutions. *Computational and Mathematical Organization Theory*, 12(2-3 SPEC. ISS.):251–275.
- Guarino, N. (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press.
- Klapiscak, T. and Bordini, R. H. (2008). Jasdl: A practical programming approach combining agent and semantic web technologies. In *International Workshop on Declarative Agent Languages and Technologies*, pages 91–110. Springer.
- Muntaner, E. and Esteva, J. L. (2007). Towards a formalisation of dynamic electronic institutions. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*, pages 97–109. Springer.
- Rodríguez-Aguilar, J. A., Sierra, C., Arcos, J. L., Lopez-Sanchez, M., and Rodríguez, I. (2015). Towards next generation coordination infrastructures. *Knowledge Engineering Review*, 30(4):435–453.
- Searle, J. (2010). *Making the social world: The structure of human civilization*. Oxford University Press.
- Searle, J. R. (1995). *The construction of social reality*. Simon and Schuster.
- Thomas, C. (2018). *Ontology in Information Science*. BoD—Books on Demand.
- Vázquez-Salceda, J., Aldewereld, H., Grossi, D., and Dignum, F. (2008). From human regulations to regulated software agents' behavior. *Artificial Intelligence and Law*, 16(1):73–87.
- Viganò, F. and Colombetti, M. (2008). Model Checking Norms and Sanctions in Institutions. (ii):316–329.

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

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

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

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

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

1. Introdução

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

Tabela 1. Paradoxo Doutrinal

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

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



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

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

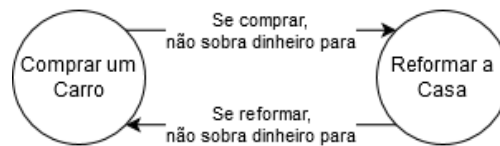


Figura 1. Problema de Decisão de Compra.

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

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

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

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

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

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

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

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

2. Fundamentação Teórica

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

2.1. Argumentação

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

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

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

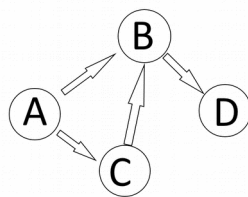


Figura 2. Exemplo de um *Framework* de Argumentação.

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

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

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

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

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

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

2.2. Agregação de Julgamento

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

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

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

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

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

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

2.3. JaCaMo

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

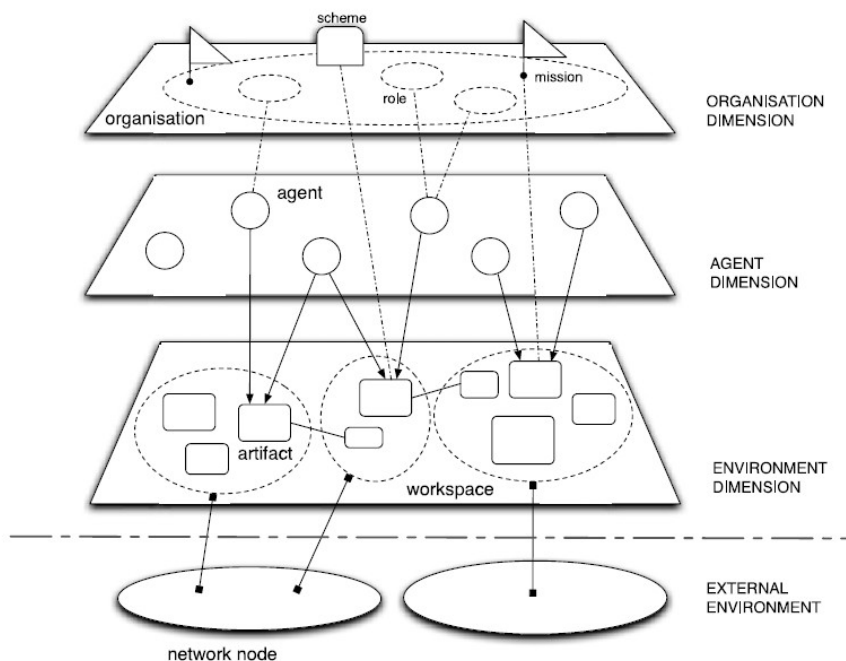


Figura 3. Visão Geral do JaCaMo. (Extraído de Boissier et al, 2013)

Jason

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

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

CArtAgO

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

Moise

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

3. Mapeamento

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

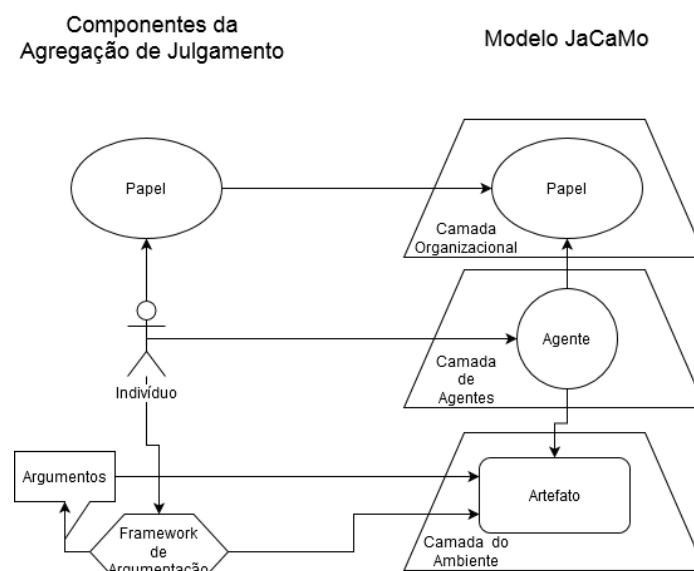


Figura 4. Mapeamento dos Modelos.

3.1. Mapeamento dos Indivíduos

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

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

3.2. Mapeamento do Framework de Argumentação

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

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

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

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

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

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

3.3. Semântica de Argumentação

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

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

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

3.4. Protocolo de Comunicação

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

Agregador (único):

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

Indivíduo (múltiplos):

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

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

4. Estudo de Caso

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

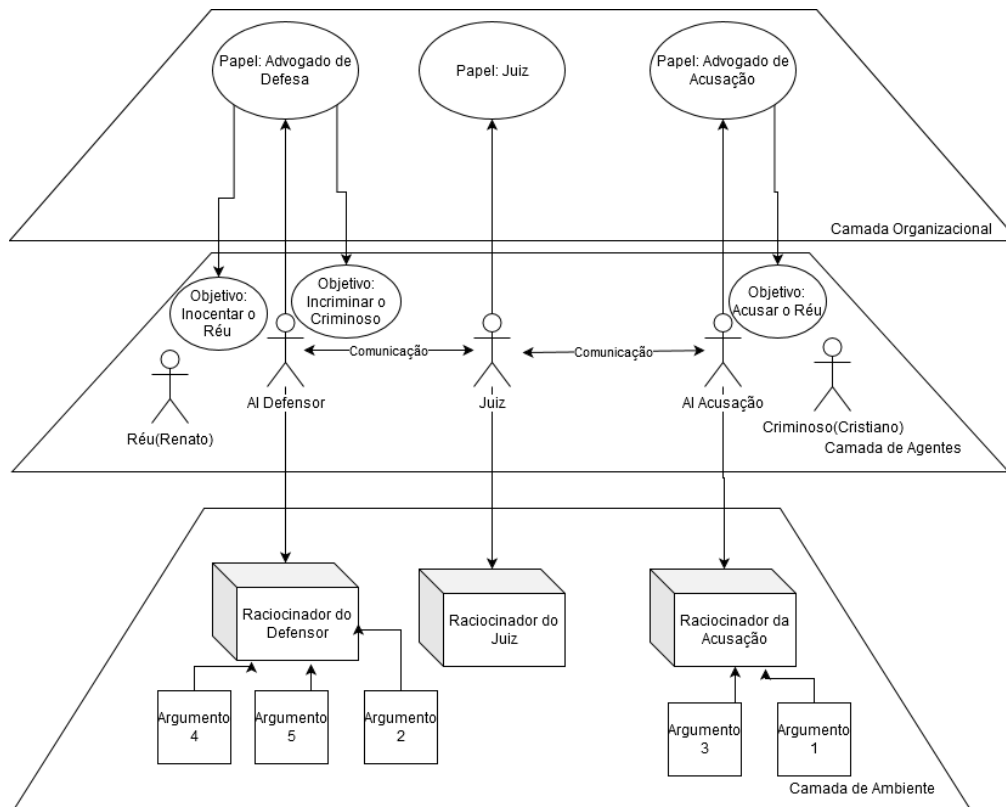


Figura 5. Visão Geral do Estudo de Caso.

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

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

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

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

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

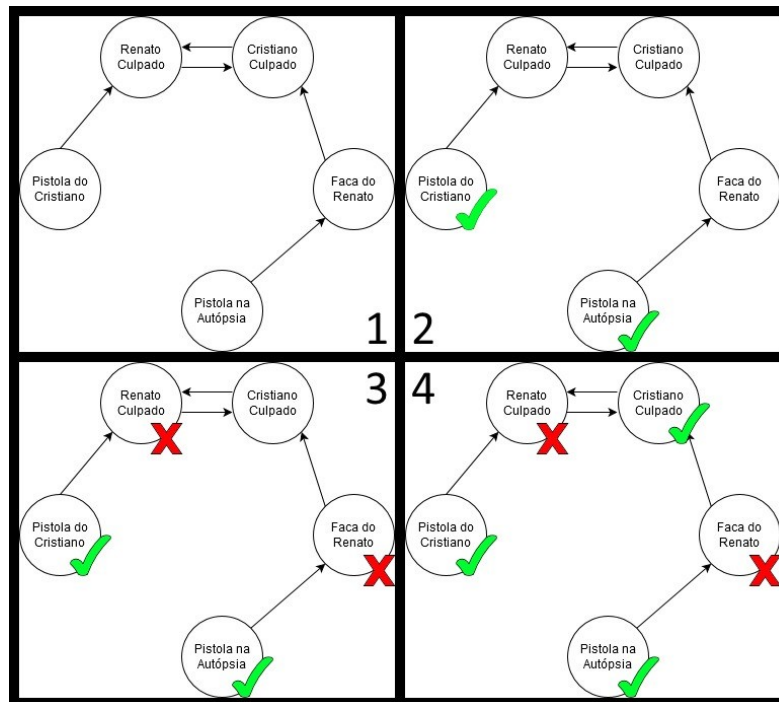


Figura 6. Sequência de Seleção dos Argumentos.

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

```

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

```

Figura 7. Resultado do Estudo de Caso.

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

5. Conclusão

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

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

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

Referências

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

[Arrow, K. \(1950\). "A Difficulty in the Concept of Social Welfare". *Journal of Political*](#)

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

Estudo comparativo em GAMA e Google Earth Engine: possibilidades para a área de sistemas multiagente*

Míriam Born¹, Fernanda P. Mota¹, Giovani Farias¹,
Matheus Gonçalves², Bruna Leitzke², Marilton Aguiar¹, Diana F. Adamatti²


¹Programa de Pós-Graduação em Computação
Universidade Federal de Pelotas (UFPEL) – Pelotas – RS – Brasil

²Centro de Ciências Computacionais (C3)
Universidade Federal do Rio Grande (FURG) – Rio Grande – RS – Brasil

{mborn,marilton}@inf.ufpel.edu.br, brunaleitzke@hotmail.com,
{nandapm2010,dianaada,giovanifarias,m2gonsalvez}@gmail.com

Abstract. *The recent advances in information and communication technologies, which provide an improvement in the collection and analysis of hydrological data, and in the understanding of the physical processes of water allow the implementation of simulation models closer to reality. This paper aims to present a comparison of the main features and functionality of Google Earth and GAMA tools. These two platforms provide the integration of technologies present in geographic information systems as multi-agent systems, which make them interesting for the development of applications in the scope of the environmental area and, more specifically in this work, in the management of water resources, with the study of the Lagoa Mirim and Canal São Gonçalo watershed. The result of this analysis will guide us to the definition of the most suitable platform for the future modeling of the system.*

Resumo. *Os recentes progressos nas tecnologias de informação e comunicação, que propiciam a melhora na coleta e na análise de dados hidrológicos, e no entendimento dos processos físicos da água permitem a implementação de modelos de simulação mais próximos da realidade. Este artigo tem como objetivo apresentar uma comparação das principais características e funcionalidades das ferramentas Google Earth e GAMA. Estas duas plataformas propiciam a integração das tecnologias presentes em sistemas de informação geográficas com sistemas multiagente, o que as tornam interessantes para o desenvolvimento de aplicações no âmbito da área ambiental e, mais especificamente neste trabalho, no gerenciamento de recursos hídricos, tendo como estudo de caso a bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo. O resultado desta análise nos guiará para a definição da plataforma mais adequada para a modelagem futura do sistema.*

* O trabalho Estudo comparativo em GAMA e Google Earth Engine: possibilidades para a área de sistemas multiagente de Míriam Born, Fernanda P. Mota, Giovani Farias, Matheus Gonçalves, Bruna Leitzke, Marilton Aguiar, Diana F. Adamatti está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial 4.0 Internacional.<http://creativecommons.org/licenses/by-nc/4.0/>

1. Introdução

Os avanços na hidro-informática auxiliam no entendimento dos processos físicos da água. Além disso, este progresso ajuda na melhoria da coleta e análise de dados hidrológicos, no desenvolvimento de tecnologias da informação e comunicação e de sistemas de informação geográfica [Simmonds et al. 2019].

O Modelo Baseado em Agentes (do inglês, *Agent-Based Model* – ABM) foi elaborado a partir da inteligência artificial [Macal and North 2005]. O ABM está sendo utilizado para simular decisões humanas ao modelar sistemas humanos e naturais complexos [An 2012] e sistemas socioecológicos [Filatova et al. 2013].

A maioria das aplicações dos modelos baseados em agentes no gerenciamento e planejamento de recursos hídricos contava com sistemas hipotéticos (construídos a partir de hipóteses) para teste de métodos e análise de políticas. Embora, estes estudos possam ajudar a fornecer orientações para avaliar ou projetar políticas de gerenciamento de água de forma exploratória, ainda existe uma lacuna na tradução dos resultados numéricos destes sistemas para a implementação de políticas de água no mundo real.

O trabalho de [Lin et al. 2020] teve como objetivo preencher essa lacuna, no qual foi desenvolvido um ABM para o sistema de alocação de água baseado em depósito de água. O ABM foi então calibrado com relação aos usos anuais de água registrados de 2007 a 2014 e foi utilizado para avaliar políticas de água e desenvolver estratégias eficazes de gerenciamento de água na região de Bakken.

A maioria das aplicações ABM na literatura atual se apoiaram em sistemas hipotéticos para teste de métodos e análise de políticas. Além disso, os modelos baseados em agentes são frequentemente criticados por não serem calibrados externamente (ou validados) com dados empíricos coletados no mundo real [Crooks et al. 2008, Berglund 2015]. [Vicuña et al. 2012] forneceu uma breve revisão bibliográfica sobre utilização do ABM no planejamento e gerenciamento de recursos hídricos nas últimas duas décadas.

[Kandiah et al. 2016, Tourigny and Fillion 2019] definiram os agentes como usuários finais da água urbana ao abordar problemas no suprimento e na demanda de água. [Ding et al. 2016] construíram agentes para a gestão da água das bacias hidrográficas. Embora a modelagem baseada em agentes possa ajudar a fornecer orientações para avaliar e projetar ferramentas e políticas de gerenciamento de água de forma exploratória [Noël and Cai 2017].

O principal objetivo deste trabalho é estabelecer um comparativo entre as funcionalidades da plataforma GAMA (*GIS Agent-based Modeling Architecture*)¹ e das ferramentas do Google Earth Engine². Os objetivos específicos visam identificar e avaliar qual plataforma seria mais adequada no contexto de análise de dados em recursos hídricos. Além disso, avaliamos a disponibilidade de dados proporcionada por cada uma e também analisamos a possibilidade de agregar o GAMA no GEE. Para análise deste comparativo realizamos um estudo de caso em ambas as plataformas.

Neste estudo, buscamos por ferramentas que possibilitam a análise de dados georreferenciados e que permitam a integração de simulação baseada em agentes. Embora,

¹<https://gama-platform.github.io/>

²<https://code.earthengine.google.com/>

não pareça justa, em uma primeira análise, comparar as plataformas GAMA e GEE, pelo fato do GAMA permitir a integração de simulação baseada em agentes. Acreditamos que o GEE permite uma análise muito mais aprimorada de dados georreferenciados que o GAMA o que justificaria nosso estudo, pois dependendo da aplicação, há necessidade de manipular estas informações de forma rápida e objetiva, que no GAMA exigiria maior esforço por não conter estes dados já presentes na plataforma.

O artigo está organizado como segue. Nas Seções 2.1 e 2.2, descrevemos as plataformas Google Earth e GAMA, respectivamente. Na Seção 3 descrevemos a metodologia para comparação das plataformas. Na Seção 4 descrevemos os resultados e as discussões. Por fim, na Seção 5 apresentamos as conclusões e os trabalhos futuros.

2. Comparação entre as Plataformas Google Earth Engine e GAMA

Nesta seção, comparamos as plataformas Google Earth Engine e GAMA a partir dos seguintes critérios: i) disponibilidade de dados georreferenciados na plataforma; ii) integração de informações georreferenciadas a plataforma; iii) manipulação de informações georreferenciadas; iv) integração de sistemas multiagente na plataforma; v) linguagem de programação utilizada na plataforma; vi) facilidade de utilização da Interface das plataformas; vii) utilização das plataformas no contexto dos recursos naturais; viii) integração da plataforma com outras ferramentas; ix) processamento grande volume de dados; e, x) processamento pequeno volume de dados.

2.1. Google Earth Engine

O *Google Earth Engine* (GEE) é uma plataforma de processamento geoespacial baseada em nuvem que pode acessar e analisar com eficiência numerosas fontes com grande volume de dados (petabytes) geoespaciais (como imagens de sensoriamento remoto Landsat e MODIS, entre outros) a partir de servidores do Google em um período de tempo menor e em computadores que não tem um grande processamento [Gorelick et al. 2017]. Um exemplo da interface do GEE pode ser visualizado na Figura 1.

O GEE utiliza um sistema de processamento paralelo para realizar o cálculo em um grande número de máquinas. Para habilitar esse processamento, o GEE tira proveito das técnicas padrão comumente usadas por linguagens funcionais, como transparência referencial e avaliação lenta, para ganhos significativos de otimização e eficiência [Gorelick et al. 2017].

Após a disponibilidade gratuita da série Landsat em 2008, o Google arquivou todos os conjuntos de dados e os vinculou ao mecanismo de computação em nuvem, para uso em código aberto. O arquivo atual de dados inclui os dados de outros satélites, bem como conjuntos de dados vetoriais baseados em sistemas de informações geográficas (*GIS – Geographic Information System*), modelos sociais, demográficos, climáticos, de elevação digital e camadas de dados climáticos [Gorelick et al. 2017].

Existem várias maneiras de interagir com a plataforma GEE. A linguagem de programação utilizada nesta ferramenta é o JavaScript, que facilita o desenvolvimento de aplicativos *mobile*, jogos e aplicações de aprendizagem de máquina. Esta plataforma é composta pelos seguintes módulos [Gorelick et al. 2017, Kumar and Mutanga 2019]: i) *Code Editor* é uma IDE para escrever e executar *scripts*; ii) *Explorer* é um aplicativo para

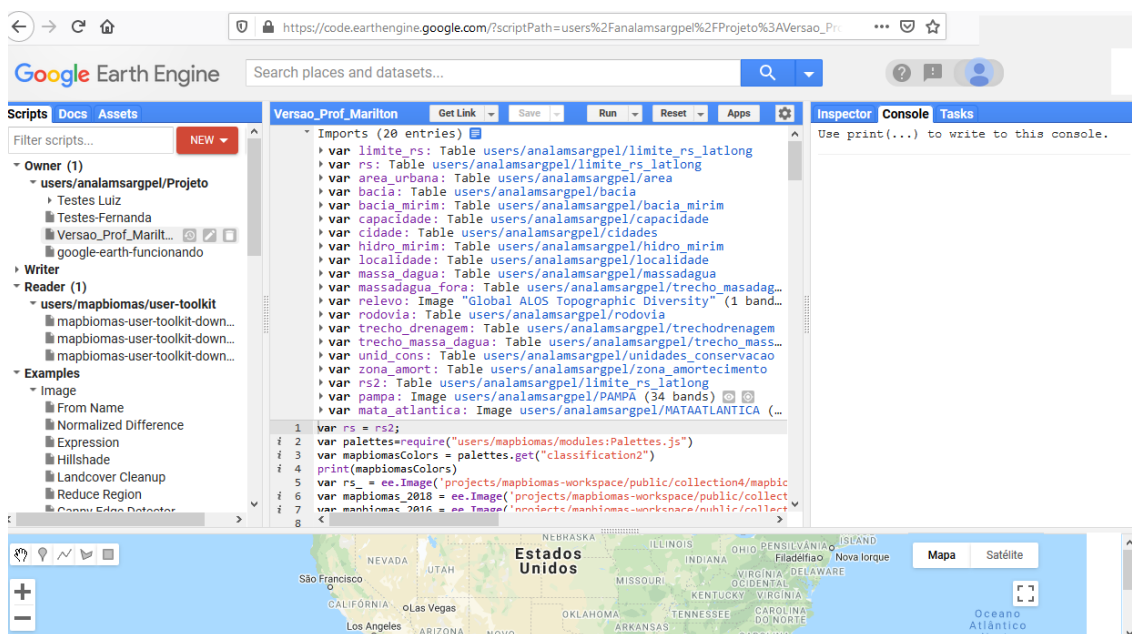


Figura 1. Plataforma GEE

explorar o catálogo de dados do GEE e executar análises simples; e, iii) bibliotecas cliente em Python e JavaScript para auxiliar os usuários no desenvolvimento de seus protótipos.

O *front-end* de fácil acesso fornece um ambiente conveniente para o desenvolvimento interativo de dados e algoritmos. Os usuários podem adicionar e selecionar seus próprios dados e coleções, enquanto usam os recursos de nuvem do Google para realizar todo o processamento. O GEE permite que cientistas, pesquisadores independentes e entusiastas explorem esse enorme banco de dados para detecção de mudanças, mapeamento de tendências e quantificação de recursos na superfície da Terra. A plataforma GEE não necessita de grande poder de processamento ou software muito complexo, o que permite a pesquisadores com poucos recursos, em países de terceiro mundo, terem a mesma capacidade de realizar análises que países desenvolvidos [Kumar and Mutanga 2019].

[Shami and Ghorbani 2019, Hakdaoui et al. 2020] investigaram e avaliaram a complementaridade dos dados do Landsat, Sentinel-2 e do Sentinel-1 em um ambiente no deserto de Imlili Sebkh. Os autores destacam a potencial aplicação do GEE no processamento de grandes quantidades de dados de satélite para observação de cavidades de água salgada permanentes, úmidas/secas, espaciais-temporais, livres e de longo prazo, além do monitoramento de umidade de Imlili Sebkh. Os resultados mostram que as imagens de radar não são apenas adequadas para o estudo de áreas desérticas, mas também para mapear as cavidades de água nas zonas úmidas do deserto.

[Ou et al. 2020] geraram mapas de estufa multitemporais a partir de imagens Landsat e do GEE da região de preservação natural de Shouguang (norte da China) no período de 1990 a 2018. Além disso, avaliaram e quantificaram a dinâmica espaço-temporal o aumento da agricultura nesta área de estudo. Os resultados demonstraram a vantagem de utilizar imagens do Landsat e o GEE para monitorar o desenvolvimento de estufas em um período de longo prazo e forneceram uma perspectiva mais intuitiva para entender o processo dessa abordagem especial de produção agrícola do que os estudos relevantes das

ciências sociais.

[Xia et al. 2019] investigaram as mudanças climáticas nas águas superficiais da bacia do rio Huai que apresentaram impactos significativos nas plantas agrícolas, no equilíbrio ecológico e no desenvolvimento socioeconômico. Para entender as mudanças nesta região, os autores utilizaram cenas disponíveis do Landsat TM, ETM+ e do OLI no período de 1989 a 2017 e processaram os dados na plataforma GEE. Segundo os autores, compreender as variações nas áreas de corpos d'água e os fatores de controle pode apoiar a designação e implementação de práticas sustentáveis de gerenciamento de água nos usos agrícola, industrial e doméstico.

[Shami and Ghorbani 2019] investigaram o armazenamento de água superficial e subterrânea na região do Irã que tem clima quente e árido. Os autores analisaram o armazenamento de água a partir dos dados do satélite GRACE e a precipitação anual total por meio do CHIRPS no GEE no período de 2003 a 2017. Os resultados obtidos indicam uma redução nos níveis de armazenamento de água no período entre 2008 e 2017. Além disso, os dados de precipitação anual mostram que a quantidade de chuva desde o ano de 2008 diminuiu nesta região.

Para esta revisão bibliográfica, realizamos uma busca por trabalhos que relacionassem a plataforma GEE, recursos hídricos e sistemas multiagente. No entanto, não foram encontrados trabalhos que envolvessem sistemas multiagentes.

2.2. Gama

GAMA (*GIS Agent-based Modeling Architecture*) é uma plataforma de desenvolvimento, modelagem e simulação que é constituída de ferramentas que auxiliam na concepção de modelos complexos. O GAMA é uma plataforma interessante de ser analisada por fornecer recursos como: a integração entre programação baseada em agentes, gerenciamento de dados geográficos, ferramentas de visualização e representação em vários níveis. Um exemplo da interface do GAMA pode ser visualizado na Figura 2.

A ferramenta é desenvolvida por diversas equipes da unidade internacional de pesquisa UMMISCO (*Unité de Modélisation Mathématique et Informatique des Systèmes Complexes*) no IRD (*Institut de Recherche pour le Développement*) da UPMC (*Université Pierre et Marie Curie*) desde o ano 2007 sendo um projeto *open source* [Born et al. 2019].

A plataforma foi desenvolvida na linguagem Java, que possibilita a incorporação de outras ferramentas baseadas em agentes como Jason, Jadex, Jade, entre outras [Born et al. 2019]. Um aspecto bastante interessante desta é que a mesma oferece uma linguagem de modelagem completa denominada GAML (*GAMA Modeling Language*).

Um editor é disponibilizado na plataforma com intuito de facilitar a modelagem e o desenvolvimento de projetos dos usuários. GAMA abrange ferramentas usuais de IDEs sendo algumas: coloração de sintaxe, formatação e adição de comentários nas linhas de código, compilação e preenchimento automático, e também, ampla documentação online para consulta, a qual possibilita ao usuário a pesquisa de informações sobre palavras-chave, operadores e declarações. E também, a partir deste ambiente de desenvolvimento integrado, a ferramenta propicia a formulação e construção de modelos baseados em agentes, com diferentes conjuntos de dados e possibilidade de integrar e manipular sistemas

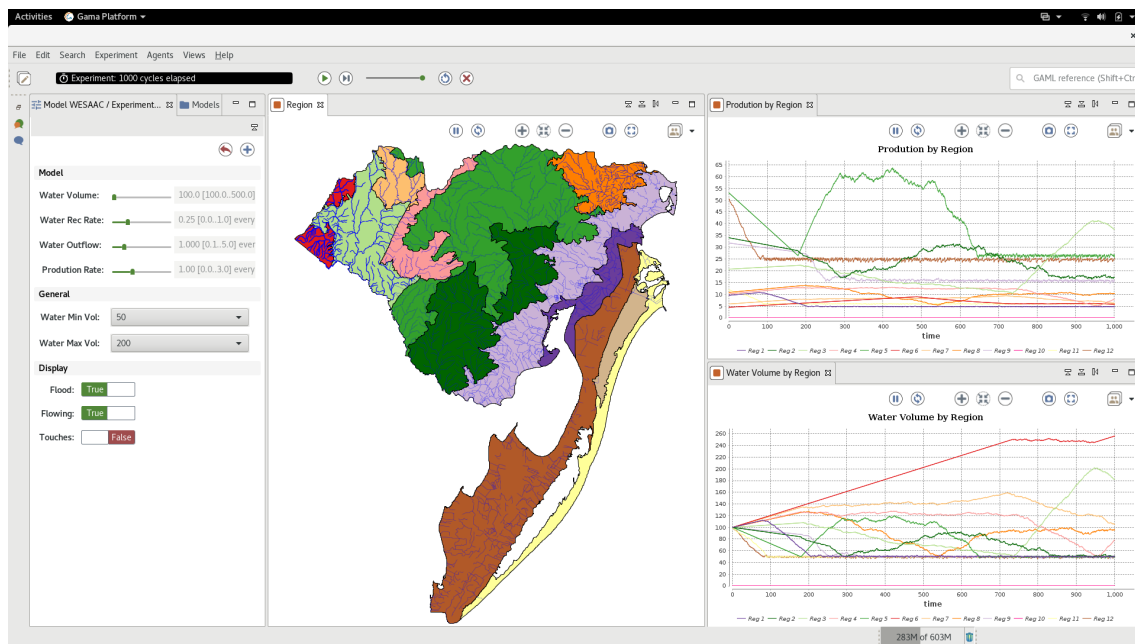


Figura 2. Interface da bacia Mirim-São Gonçalo na plataforma GAMA

de informações geográficas.

GAMA apresenta também flexibilidade e facilidade de interface permitindo ao usuário organizar e visualizar de maneira simples os painéis disponíveis para o desenvolvimento dos modelos. Na ferramenta de inspeção/verificação de agentes é possível obter informações sobre cada um destes ou de vários, ademais no mecanismo de busca da plataforma obtêm-se informações e exemplo de uso de diversos operadores.

GAMA é amplamente utilizada em diversos modelos da área de recursos naturais, como nas seguintes pesquisas [Nguyen Vu et al. 2009, Taillandier et al. 2012, Grignard et al. 2013, Drogoul et al. 2013]. Em [Chu et al. 2009] foi desenvolvido um sistema de apoio a decisões de alocação e planejamento de recursos para emergências de desastres naturais que ocorrem em áreas urbanas, sendo que o protótipo de interface com o usuário foi projetado na ferramenta GAMA.

No trabalho de [Amouroux et al. 2009], a plataforma GAMA foi utilizada com dois propósitos: i) na modelagem de propagação da gripe aviária para auxiliar na otimização da rede de monitoramento a partir de uma base de dados geolocalizados; e, ii) na utilização de simulação interativa que visa auxiliar na tomada de decisão durante situações de pós-desastre em área urbana.

Em [Thérond et al. 2014], os autores abordam a gestão sustentável de recursos hídricos como o uso da terra e sistemas de governança da água. O estudo utilizou o modelo de simulação MAELIA voltado à análise de questões políticas, desenvolvido na plataforma GAMA, o qual permite a realização de avaliações integradas no nível da bacia hidrográfica de estratégias relacionadas à gestão destes recursos.

O estudo de [Amouroux et al. 2008] apresenta a integração de um modelo baseado em agentes e um modelo ambiental baseado em GIS pode atuar como um laboratório virtual para epidemiologia. A construção do modelo foi implementada na plataforma

GAMA e adaptado aos requisitos epidemiológicos, contribuindo assim para a exploração das causas da doença, neste contexto, a influenza aviária.

[Gaudou et al. 2014] apresentam um estudo sobre a adaptação às mudanças climáticas ou mitigação de desastres. O projeto MAELIA, construído na plataforma GAMA, visa a simulação baseada em agentes para estudar os impactos ambientais. Este estudo de caso é aplicado na bacia hidrográfica francesa de Adour-Garonne, a mais afetada na França por escassez de água durante o período de baixa maré.

Contudo, observa-se que a plataforma GAMA oferece requisitos interessantes no desenvolvimento de modelos que utilizam SMA (Sistemas Multiagente) e dados georreferenciados de locais específicos, apresentando notável vantagem de sua utilização. Porém, uma das desvantagens é, dependendo do modelo, do número de agentes e dos dados a serem utilizados, o tempo de carregamento e processamento do modelo é bastante lento. Outra desvantagem é que, diferentemente do GEE, GAMA não carrega automaticamente informações georreferenciadas. Estes dados precisam ser manipulados em arquivo separado e serem carregados na plataforma.

3. Estudo de Caso

Neste trabalho, focamos a aplicação-piloto do trabalho no Comitê de Gerenciamento das Bacias Hidrográficas da Lagoa Mirim e do Canal São Gonçalo, que envolvem as cidades de Rio Grande e Pelotas. Modelamos a gestão da bacia hidrográfica a partir da base de dados do estado do Rio Grande do Sul (RS), por meio da plataforma GAMA e GEE, que permitem lidar facilmente com dados geoespaciais e vetoriais de sistemas de informação geográfica.

O modelo de dados GIS utilizado neste trabalho, representa uma base de dados com informações geográficas, que encontra-se em arquivos no formato *shapefile*, contendo informações geoespaciais da bacia hidrográfica Mirim-São Gonçalo, disponibilizados pela Secretaria Estadual de Meio Ambiente do RS.

Estes dados são relacionados a área urbana na região do RS, a bacia hidrográfica litorânea no RS, a bacia hidrográfica da Lagoa Mirim e do canal São Gonçalo, a capacidade da Lagoa Mirim, às cidades do RS próximas à Lagoa dos Patos e Mirim, ramificações da Lagoa Mirim e do Canal São Gonçalo e adjacências, às cidades do Uruguai banhadas pela Lagoa Mirim e do Canal São Gonçalo, a parte do oceano que cerca o Rio Grande do Sul, a Lagoa dos Patos e áreas de preservação ao redor da Lagoa dos Patos, Lagoa Mirim e Canal São Gonçalo.

Esses arquivos, descrevem espacialmente pontos, linhas e polígonos para representar as regiões, capacidades de uso do solo e os rios presentes na bacia. Além disso, também foram utilizados dados de satélites relacionados a chuva, relevo e vegetação. O modelo hidrográfico, neste caso, é composto por 12 regiões no GAMA e 37 no GEE, com características distintas.

A interface de simulação (apresentada na Figura 2) do modelo hidrográfico da bacia Mirim-São Gonçalo desenvolvido na plataforma GAMA, possibilita a representação gráfica (mapa) da variação dos volumes dos rios e alteração das cores das regiões que secam, bem como, a visualização em gráfico de linhas da variação da taxa de produção e do volume de água em cada região. Além disso, permite atribuir diferentes valores aos

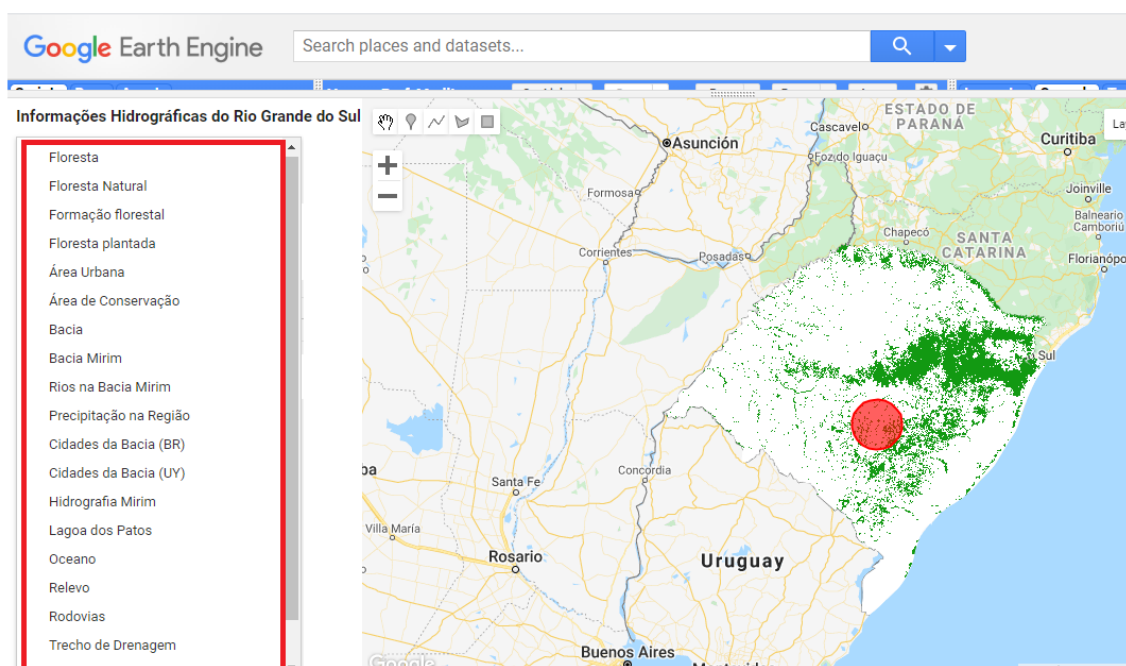


Figura 3. As camadas do modelo hidrográfico da bacia Mirim-São Gonçalo desenvolvida no GEE, destacadas pelo retângulo vermelho

diversos parâmetros de configuração do ambiente.

Já a simulação do modelo hidrográfico da bacia Mirim-São Gonçalo desenvolvida no GEE, possibilita a representação em mapa, gráficos da variação de precipitação, análise de cobertura e uso do solo, bem como podemos observar a vegetação da região (Figura 3). No entanto, a integração do GEE com os sistemas multiagente não foi desenvolvida neste trabalho, por encontrarmos dificuldade em integrar esta plataforma com PADE³.

4. Discussões

Este estudo encontra-se em fase de desenvolvimento, sendo assim, torna-se pertinente o estudo de ferramentas que possam auxiliar na pesquisa de sistemas multiagente aplicados a área de recursos hídricos e que possibilitem a integração de dados georreferenciados como apresentam a plataforma GAMA e as ferramentas GEE. Na Tabela 1 podemos observar um resumo comparativo dos critérios avaliados neste trabalho, no qual verificamos as vantagens e desvantagens das plataformas avaliadas neste estudo.

A plataforma GAMA consegue separar esses itens em diferentes camadas temáticas e representá-los de forma independente, permitindo trabalhar com eles de modo rápido e simples. Deste modo, cada região ou rio pode ser considerado um agente específico com suas próprias características e atributos. Na Figura 2, podemos visualizar o formato e localização de cada região. GAMA permite ao usuário relacionar as informações existentes através da posição e topologia dos objetos, gerando assim novas informações.

Já o GEE separa os itens em diferentes camadas o que permite trabalhar de modo rápido e simples com cada camada. Deste modo, cada região ou rio pode ser considerado

³https://pade.readthedocs.io/pt_BR/latest/

Tabela 1. Comparação entre as plataformas GEE e GAMA, no qual ✓ significa que a plataforma atende ao critério, X que a plataforma não atende ao critério.

Crítérios	GEE	GAMA
Disponibilidade de dados georreferenciados na plataforma	✓	✓
Integração de informações georreferenciadas a plataforma	✓	X
Manipulação de informações georreferenciadas	✓	✓
Integração de sistemas multiagente na plataforma	X	✓
Linguagem de programação utilizada na plataforma	JavaScript e Python	Java e GAML
Facilidade de utilização da Interface das plataformas	✓	✓
Utilização das plataformas no contexto dos recursos naturais	✓	✓
Integração da plataforma com outras ferramentas	✓	✓
Processamento grande volume de dados	✓	X
Processamento pequeno volume de dados	✓	✓

um elemento que foi construído a partir de uma imagem ou conjunto de dados georreferenciados. Na Figura 3, podemos visualizar as camadas no GEE, o qual, permite ao usuário relacionar as informações existentes através da sobreposição das camadas, gerando assim novas informações.

A partir deste estudo, podemos observar que a simulação multiagente é inerente a plataforma GAMA, enquanto que para construirmos a mesma simulação no GEE necessitamos desenvolver um modelo complexo através da integração entre programação baseada em agentes e dados georreferenciados. Para a utilização do GEE com sistemas multiagente, será preciso integrar o GEE a plataforma PADE⁴. Além disso, não poderíamos utilizar o poder de compartilhamento e auxílio ao desenvolvimento proporcionado pela plataforma GEE, pois necessitamos alterar a linguagem de programação para Python, que não está presente na ferramenta online e fazer toda a integração entre as diferentes plataformas: PADE, GEE.

É importante ressaltar que ambas plataformas oferecem recursos satisfatórios para a modelagem e implementação neste contexto da bacia hidrográfica. Analisando o aspecto de modelagem do sistema multiagente, GAMA permite melhor adaptação devido a programação de agentes incorporada à plataforma. No GEE este tipo de integração é mais complexo. Embora, o GEE é mais adequado para grandes volumes de dados, integração nativa de informações georreferenciadas e processamento de grandes volumes de dados, optamos neste estudo pela ferramenta GAMA por nos possibilitar a fácil integração de sistemas multiagente.

5. Conclusão

Esta pesquisa avaliou a utilização de Simulação Baseada em Multiagente (MABS – *Multiagent Based Simulation*) na gestão participativa dos recursos hídricos, mais especificamente, utilizando como base os dados do estado do Rio Grande do Sul e focando a aplicação-piloto do trabalho no Comitê de Gerenciamento das Bacias Hidrográficas da Lagoa Mirim e do Canal São Gonçalo, que envolvem as cidades de Rio Grande e Pelotas.

⁴https://pade.readthedocs.io/pt_BR/latest/

Nossa principal contribuição foi estabelecer um comparativo entre as funcionalidades da plataforma GAMA e as ferramentas GEE, tendo como objetivo principal identificar e avaliar qual ferramenta é mais adequada ao contexto de análise de dados em recursos hídricos. Além disso, avaliamos a disponibilidade de dados proporcionada por cada uma e também analisamos a possibilidade de integrá-las. Para análise deste comparativo realizamos um estudo de caso em ambas as plataformas.

Como resultado, podemos observar que as duas plataformas propiciam a integração de sistemas de dados georreferenciados, o que as torna interessantes no estudo de caso da bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo no âmbito da área de recursos hídricos. Para esta pesquisa, realizamos uma revisão bibliográfica em busca de trabalhos que relacionam a plataforma GEE com os recursos hídricos e sistemas multiagente. No entanto, não encontramos trabalhos na literatura.

Neste trabalho, modelamos a gestão da bacia hidrográfica a partir da base de dados do estado do Rio Grande do Sul, focando a aplicação-piloto do trabalho no Comitê de Gerenciamento das Bacias Hidrográficas da Lagoa Mirim e do Canal São Gonçalo, que envolvem as cidades de Rio Grande e Pelotas por meio da plataforma GAMA e das ferramentas GEE, que permite lidar facilmente com dados geoespaciais e vetoriais de sistemas de informação geográfica. Como resultado deste estudo podemos observar que a simulação multiagente é inerente a ferramenta GAMA, enquanto que para construirmos a mesma simulação no GEE necessitamos desenvolver um modelo complexo.

Neste estudo podemos observar que ambas plataformas oferecem recursos satisfatórios para a modelagem e implementação de problemas relacionados a recursos naturais. Embora, o GEE é mais adequado para grandes volumes de dados, integração nativa de informações georreferenciadas e processamento de grandes volumes de dados, optamos neste estudo pela ferramenta GAMA por nos possibilitar a fácil integração de sistemas multiagente.

Como trabalhos futuros pretendemos testar a integração entre as plataformas PADE e GEE. Além disso, gostaríamos de realizar um teste de interface e de funcionalidade, de forma a avaliar qual é a mais viável de utilizarmos e após a análise melhorar as funcionalidades da plataforma selecionada.

Agradecimentos

Os autores deste artigo agradecem a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES/Brasil) e a Agência Nacional de Águas (ANA/Brasil) – Pró-Recursos Hídricos Chamada N° 16/2017, pelo auxílio financeiro no desenvolvimento desta pesquisa.

Referências

- Amouroux, E., Chu, T.-Q., Boucher, A., and Drogoul, A. (2009). GAMA: An environment for implementing and running spatially explicit multi-agent simulations. In Ghose, A., Governatori, G., and Sadananda, R., editors, *Agent Computing and Multi-Agent Systems*, pages 359–371, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Amouroux, E., Desvaux, S., and Drogoul, A. (2008). Towards virtual epidemiology: An agent-based approach to the modeling of h5n1 propagation and persistence in north-

- vietnam. In Bui, T. D., Ho, T. V., and Ha, Q. T., editors, *Intelligent Agents and Multi-Agent Systems*, pages 26–33, Berlin, Heidelberg. Springer Berlin Heidelberg.
- An, L. (2012). Modeling human decisions in coupled human and natural systems: Review of agent-based models. *Ecological Modelling*, 229:25–36.
- Berglund, E. Z. (2015). Using agent-based modeling for water resources planning and management. *Journal of Water Resources Planning and Management*, 141(11):04015025.
- Born, M., Leitzke, B. S., Farias, G., Aguiar, M., and Adamatti, D. F. (2019). Modelagem baseada em agentes para análise de recursos hídricos. In *Anais do XIII Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações (WESAAC 2019)*, pages 107–118, Florianópolis/SC. wesaac.c3.furg.br.
- Chu, T.-Q., Drogoul, A., Boucher, A., and Zucker, J.-D. (2009). Interactive learning of independent experts' criteria for rescue simulations. *J. UCS*, 15:2701–2725.
- Crooks, A., Castle, C., and Batty, M. (2008). Key challenges in agent-based modelling for geo-spatial simulation. *Computers, Environment and Urban Systems*, 32(6):417–430.
- Ding, N., Erfani, R., Mokhtar, H., and Erfani, T. (2016). Agent based modelling for water resource allocation in the transboundary Nile river. *Water*, 8(4):139.
- Drogoul, A., Amouroux, E., Caillou, P., Gaudou, B., Grignard, A., Marilleau, N., Taillandier, P., Vavasseur, M., Vo, D.-A., and Zucker, J.-D. (2013). Gama: Multi-level and complex environment for agent-based models and simulations. In *Proceedings of the 2013 Int. Conf. on Aut. Agen. and Multi-agent Sys.*, AAMAS '13, pages 1361–1362, Richland, SC. Int. Fou. for Aut. Agen. and Mult. Sys.
- Filatova, T., Verburg, P. H., Parker, D. C., and Stannard, C. A. (2013). Spatial agent-based models for socio-ecological systems: challenges and prospects. *Environmental modelling & software*, 45:1–7.
- Gaudou, B., Sibertin-Blanc, C., Therond, O., Amblard, F., Auda, Y., Arcangeli, J.-P., Balestrat, M., Charron-Moirez, M.-H., Gondet, E., Hong, Y., Lardy, R., Louail, T., Mayor, E., Panzoli, D., Sauvage, S., Sánchez-Pérez, J.-M., Taillandier, P., Van Bai, N., Vavasseur, M., and Mazzega, P. (2014). The MAELIA multi-agent platform for integrated analysis of interactions between agricultural land-use and low-water management strategies. In Alam, S. J. and Parunak, H. V. D., editors, *Multi-Agent-Based Simulation XIV*, pages 85–100, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., and Moore, R. (2017). Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 202:18–27.
- Grignard, A., Taillandier, P., Gaudou, B., Vo, D. A., Huynh, N. Q., and Drogoul, A. (2013). Gama 1.6: Advancing the art of complex agent-based modeling and simulation. In Boella, G., Elkind, E., Savarimuthu, B. T. R., Dignum, F., and Purvis, M. K., editors, *PRIMA 2013: Principles and Practice of Multi-Agent Systems*, pages 117–131, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Hakdaoui, S., Emran, A., Pradhan, B., Qninba, A., Balla, T. E., Mfondoum, A. H. N., Lee, C.-W., and Alamri, A. M. (2020). Assessing the changes in the moisture/dryness

- of water cavity surfaces in imlili sebkha in southwestern morocco by using machine learning classification in google earth engine. *Remote Sensing*, 12(1):131.
- Kandiah, V. K., Berglund, E. Z., and Binder, A. R. (2016). Cellular automata modeling framework for urban water reuse planning and management. *Journal of Water Resources Planning and Management*, 142(12):04016054.
- Kumar, L. and Mutanga, O. (2019). *Google Earth Engine Applications*. MDPI.
- Lin, Z., Lim, S. H., Lin, T., and Borders, M. (2020). Using agent-based modeling for water resources management in the bakken region. *Journal of Water Resources Planning and Management*, 146(1):05019020.
- Macal, C. M. and North, M. J. (2005). Tutorial on agent-based modeling and simulation. In *Proceedings of the Winter Simulation Conference, 2005.*, pages 14–pp. IEEE.
- Nguyen Vu, Q. A., Gaudou, B., Canal, R., and Hassas, S. (2009). Coherence and robustness in a disturbed mas. In *2009 IEEE-RIVF International Conference on Computing and Communication Technologies*, pages 1–4.
- Noël, P. H. and Cai, X. (2017). On the role of individuals in models of coupled human and natural systems: Lessons from a case study in the republican river basin. *Environmental Modelling & Software*, 92:1–16.
- Ou, C., Yang, J., Du, Z., Liu, Y., Feng, Q., and Zhu, D. (2020). Long-term mapping of a greenhouse in a typical protected agricultural region using landsat imagery and the google earth engine. *Remote Sensing*, 12(1):55.
- Shami, S. and Ghorbani, Z. (2019). Investigating water storage changes in iran using grace and chirps data in the google earth engine system. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.
- Simmonds, J., Gómez, J. A., and Ledezma, A. (2019). The role of agent-based modeling and multi-agent systems in flood-based hydrological problems: a brief review. *Journal of Water and Climate Change*.
- Taillandier, P., Vo, D.-A., Amouroux, E., and Drogoul, A. (2012). Gama: A simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In Desai, N., Liu, A., and Winikoff, M., editors, *Principles and Practice of Multi-Agent Systems*, pages 242–258, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Thérond, O., Sibertin-Blanc, C., Lardy, R., Gaudou, B., Balestrat, M., Hong, Y., Louail, T., Nguyen, V. B., Panzoli, D., Sanchez-Perez, J.-M., Sauvage, S., Taillandier, P., Vavasseur, M., and Mazzega, P. (2014). Integrated modelling of social-ecological systems: The MAELIA high-resolution multi-agent platform to deal with water scarcity problems. In *7th International Environmental Modelling and Software Society (iEMSs 2014)*, page pp. 1, San Diego, California, United States.
- Tourigny, A. and Fillion, Y. (2019). Sensitivity analysis of an agent-based model used to simulate the spread of low-flow fixtures for residential water conservation and evaluate energy savings in a canadian water distribution system. *Journal of Water Resources Planning and Management*, 145(1):04018086.

- Vicuña, S., McPhee, J., and Garreaud, R. D. (2012). Agriculture vulnerability to climate change in a snowmelt-driven basin in semiarid chile. *Journal of Water Resources Planning and Management*, 138(5):431–441.
- Xia, H., Zhao, J., Qin, Y., Yang, J., Cui, Y., Song, H., Ma, L., Jin, N., and Meng, Q. (2019). Changes in water surface area during 1989–2017 in the huai river basin using landsat data and google earth engine. *Remote Sensing*, 11(15):1824.

Sistemas Multiagente e Jogos de Papéis Aplicados à Gerenciamento de Recursos Naturais *

Vinícius Borges Martins¹, Raquel de Miranda Barbosa², Diana Francisca Adamatti¹

¹Programa de Pós-Graduação em Computação
Universidade Federal do Rio Grande – Rio Grande, RS, Brasil

²Instituto Federal de Educação, Ciência e Tecnologia
do Rio Grande do Sul – Campus Rio Grande, Rio Grande, RS, Brasil

{vimiciusbormar27, raq.mbarbosa, dianaada}@gmail.com


Abstract. *This paper presents the theoretical basis and the creation of a game (role-playing game) that is going to be used with the Committee of the Mirim Lagoon and of the São Gonçalo Canal (Brazil), where each member will play the role of others members to better understand their actions and to make better decisions to the regional resources. Hence, the game was created physically, where, after every stage, the actions are transferred to be calculated by a JAVA software on a computer, returning the results to the players in printed papers. Further, will be created a graphical interface in the calculation module and the computational version of the game, besides that, will be implemented the smart agents to play randomly with the members of the Committee.*

Resumo. *Este trabalho apresenta o referencial teórico e a criação de um jogo (jogo de papéis) que será utilizado junto ao Comitê Regional da Lagoa Mirim e do Canal São Gonçalo, onde cada membro jogará no papel dos outros membros para melhor entender suas ações e para a melhor tomada de decisão para os recursos da região. Para isso, foi criado um jogo físico, onde, após cada etapa, as ações são transferidas para serem calculados por um programa em JAVA em um computador, retornando os resultados para os jogadores em papéis impressos. Futuramente, será criada uma interface gráfica no motor de cálculos e a versão computacional deste, além de serem implementados agentes inteligentes para jogar em papéis aleatórios com os membros do comitê.*

1. Introdução

O aumento de poluição no mundo, devido ao aumento de veículos, de lixo e de produtos químicos na agricultura, traz a expectativa de impureza em águas, ilhas de lixo, poluição no ar, etc, tornando claro a intervenção imediata com projetos que combatam a poluição [European Environment Agency 2018].

O Gerenciamento de Recursos Naturais (GRN) é uma área em expansão que visa o combate do problema apresentado, ou seja, o campo busca melhorar a qualidade dos

 O trabalho Sistemas Multiagente e Jogos de Papéis Aplicados à Gerenciamento de Recursos Naturais de Vinícius Borges Martins, Raquel de Miranda Barbosa e Diana Francisca Adamatti está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial 4.0 Internacional. <http://creativecommons.org/licenses/by-nc/4.0/>

recursos naturais do futuro com base na qualidade de vida das pessoas do presente, gerenciando terras (reservas naturais, propriedades públicas e privadas), água (bacias hidrográficas, oceanos), plantas (florestas, arborização em cidades) e animais (selvagens ou não) [Fuller et al. 2007].

Porém, o GRN pode trazer três desafios, segundo [Fuller et al. 2007]: (1) controle e otimização, (2) gerenciamento/comunicação, e (3) análise de dados. Portanto, a necessidade de ferramentas para suprir tais desafios são necessárias.

A Inteligência Artificial (IA), mais especificamente os Sistemas Multiagente (SMA), visam o uso de objetos inteligentes (chamados de *agentes*) que, com base nas informações que eles tem do ambiente, tomam a melhor decisão conhecida pra atingir um objetivo pessoal e, conseqüentemente, o objetivo do grupo, de maneira heterogênea e independente [Lesser 1999]. Os SMA, quando utilizados para simulações, são chamados de Simulações Baseadas em Agentes (MABS - *Multi-Agent-Based Simulations*). Estes, por sua vez, visam simular um ambiente onde os agentes interagem em prol de descobrir ou prever a solução de um problema, normalmente de forma interdisciplinar [Gilbert and Troitzsch 2005] [Le Page et al. 2015] [Le Page et al. 2016].

Tal interdisciplinaridade e ferramentas disponíveis nas MABS as torna um instrumento elegível para a estruturação e simulação dos problemas presentes no GRN.

Outra ferramenta que tem sido utilizada na área são os Jogos de Papéis (RPG - *Role-Playing Games*) [Perrotton et al. 2017]. No RPG, os participantes, no início do jogo, recebem um papel (personagem) do qual desempenhará até o final do jogo. Este tipo de jogo normalmente é dividido em turnos, de modo a todos os jogadores fazerem suas jogadas e essas jogadas influenciarem no ambiente antes de uma próxima jogada.

Um RPG pode ser jogado de forma impressa (com tabuleiros, cartas, etc), de forma oral (com um narrador e alternativas de jogadas orais) e de forma computacional. Há projetos ([Barnaud et al. 2007] [Adamatti et al. 2009] [Campo et al. 2009] [Le Page et al. 2016] [Le Page and Perrotton 2017]) que utilizam do RPG para a resolução de problemas para trabalhos em grupo, de modo a fazer cada jogador atuar, no jogo, no papel real de outro jogador, implicando na tomada de decisão com o ponto de vista do outro, tendendo a fazer os jogadores entenderem o porquê das decisões uns dos outros na vida real.

Desta forma, no GRN, o RPG torna-se outra ferramenta que auxiliaria a tomada de decisão por parte dos órgãos designados para este tipo de decisão, como os Comitês de Gerenciamento.

Este trabalho apresenta a base teórica reunida para criar um RPG computacional, denominado Gorim, que está sendo desenvolvido para ser usado com o Comitê da Lagoa Mirim e do Canal São Gonçalo (Brasil). O objetivo principal, portanto, é a explicação das áreas onde este trabalho está inserido, assim como a definição do jogo aqui apresentado. Espera-se que o jogo ajude os membros do Comitê nas suas tomadas de decisão, visto que eles estariam jogando nos papéis uns dos outros, vendo as possibilidades e necessidades dos outros membros, para, quando tomarem qualquer decisão, eles pensem na solução que traz o maior número de vantagens para todos, e, principalmente, para o meio ambiente. Além disso, agentes inteligentes serão inseridos para jogarem com os participantes como NPCs (*Non-Player Characters* - “personagens não-jogável”), para o caso de não haver o

número de jogadores necessários para o jogo.

2. Referencial Teórico

2.1. Sistemas Multiagente

Na programação convencional (orientada à objetos) são utilizados *objetos* na execução de programas. Porém, na programação orientada a agentes são utilizados *agentes*. A diferença entre estas duas estruturas é que, enquanto um objeto tem todas as funcionalidades pré-definidas, ou seja, tem exatamente o método a ser chamado para determinada entrada, os agentes têm um conjunto de opções a serem decididas por ele durante a execução do programa para tentar alcançar o objetivo traçado pelo usuário [Wooldridge 2002, p.26] [Moreno 2003]. Adicionalmente, um agente autônomo deve ser capaz de interagir ao longo do tempo com o ambiente em que foi inserido para importar as informações relevantes para as suas tomadas de decisões, a fim de atingir seus objetivos [Franklin and Graesser 1996].

Sistemas Multiagente são sistemas onde vários agentes trabalham em conjunto de forma independente para atingir suas metas, alcançando, então, o objetivo do sistema. Os agentes, dentro dos SMA, podem ser homogêneos (ou seja, aprendem da mesma forma) ou heterogêneos (de forma diferente), sendo que, em ambas as abordagens, trabalham de forma assíncrona das *threads* (linhas de execução) dos outros agentes [Lesser 1999]. Mesmo trabalhando de forma individual, os agentes devem ser capazes de agir organizadamente, pois a maioria dos problemas resolvidos por eles, são resolvidos de forma distribuída. Portanto, devem ser implementadas características como cooperação, coordenação, competição, negociação e comunicação [Alvares and Sichman 1997] [Frozza 1997] [Bordini et al. 2001] [Bousquet and Le Page 2004].

Simulações, por mostrarem detalhes com bastante precisão, são ferramentas úteis para a tomada de decisões [Frozza 1997]. Há uma área dos SMA que são unidos às simulações (formando as Simulações Baseadas em Agentes [MABS]) que tem como objetivo estudar as consequências de diferentes parâmetros em um ambiente unindo diversas áreas de estudo no processo (como psicologia e biologia social, ciência da computação, sociologia e economia) [Gilbert and Troitzsch 2005] [Le Page et al. 2015] [Le Page et al. 2016].

Resumidamente, o uso de SMA pode ser proveitoso, podendo-se citar: a rapidez na resolução de problemas devido à execução paralela das atividades dos agentes; a flexibilidade e escalabilidade conectando vários sistemas entre si; e ter um maior escopo de informações que retornam dos sistemas, tendo em vista que todos os recursos estão localizados no mesmo ambiente [Leitzke et al. 2019].

No caso deste trabalho, o uso de SMAs e MABS será feito ao serem inseridos jogadores virtuais (NPCs), para jogarem em conjunto com os jogadores reais no caso não haver o número suficiente desses para o início do jogo, e para simulações, possibilitando a previsão de consequências para o ambiente tendo como base as decisões tomadas durante o jogo.

2.2. Jogos de Papéis

Uma modalidade dos jogos são os chamados Jogos de Papéis, que estão localizados entre jogos e teatro, pois junta a interação entre pessoas em forma de jogo e a interpretação

destas acerca dos personagens disponíveis, tendo que agir em um ambiente com regras e comportamentos definidos anteriormente para completar-se a história dada [Pereira 2003] [Adamatti et al. 2007]. Deste modo, na maior parte das vezes, RPG não apresentam vencedores e perdedores, sendo uma forma colaborativa de jogos, revelando aspectos das relações sociais e permitindo a observação direta das interações entre os jogadores [Barreteau et al. 2003] [Adamatti et al. 2007].

O RPG tem sido frequentemente utilizado em treinamentos, pois tem uma abordagem de simulação de situações sem efeitos no mundo real, além do aprendizado obtido, tornando-o, deste modo, uma ferramenta lúdica, facilitando o aprendizado do assunto retratado [Perrotton et al. 2017].

[Barreteau et al. 2001] falam ainda que os RPGs e os SMA tem muito em comum: os agentes podem corresponder aos papéis, o ambiente ao tabuleiro, o tempo (*steps*) às rodadas do jogo, etc, sendo, assim, ferramentas eletíveis para uso em conjunto.

2.3. Gerenciamento de Recursos Naturais

A água é o recurso natural mais vital para o ecossistema e suas populações, pois ela é essencial para a vida e saúde de plantas e animais, além de ser importante em questões sociais e econômicas humanas [Ponte et al. 2016]. O campo de Gerenciamento de Recursos Naturais (GRN) é a área que estuda as formas de gerenciar água, animais, plantas e terras, baseando-se na qualidade do ambiente para as gerações futuras [Holzman 2009]. Segundo [Fuller et al. 2007], o GRN possui três desafios computacionais: (1) controle e otimização, (2) gerenciamento/comunicação, e (3) análise de dados.

Uma das áreas do GRN é a de Gestão de Recursos Hídricos (GRH), que discute melhores práticas de distribuição e uso da água, envolvendo organizações e diferentes grupos sociais. Além disso, a GRH é responsável pelas situações de risco ambiental e social, como enchentes e secas, situações essas que precisam de ações preventivas das autoridades [Tundisi 2006].

Bacias hidrográficas são áreas geográficas onde, devido ao relevo, a água da chuva ou de nascentes escorrem até o fundo do vale saindo por um único ponto (normalmente um rio) ou, se houver uma crista de relevo pelo caminho, separa-se a água entre duas bacias diferentes [Ahmad and Simonovic 2005].

O Rio Grande do Sul possui a Bacia Hidrográfica da Lagoa Mirim e do Canal São Gonçalo, tendo, como gestores destes recursos, o Comitê de Gerenciamento local. Dentre as cidades presentes, as bacias englobam as cidades de Rio Grande e Pelotas.

2.4. Integração das áreas

Levando em conta os desafios computacionais acerca do GRN, a interdisciplinaridade dos SMA os elegem como ferramenta para a modelagem dos problemas relacionados aos recursos naturais. Além disso, [Le Page and Perrotton 2017, Barreteau et al. 2001] afirmam a similaridade entre SMA e RPG, de forma a assegurar que, a partir de um ponto de vista formal, o RPG é um tipo de SMA devido a composição deste por entidades que estão imersos em um ambiente indo atrás de seus objetivos específicos. Por estas particularidades apresentadas, o projeto foi inserido dentre estas áreas, como mostrado na Figura 1.

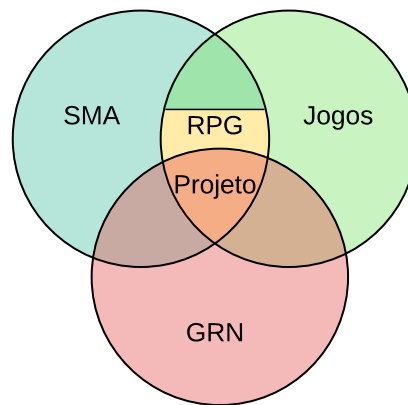


Figura 1. Diagrama de Venn que mostra onde o projeto está inserido.

2.5. Metodologias

Existem metodologias criadas para estudos utilizando as três áreas (SMA, RPG e GRN), apresentados nas subseções a seguir.

2.5.1. Companion Modelling (ComMod)

Esta metodologia foi proposta em [Bousquet et al. 1999] pelos membros do CIRAD¹ (*Centre de coopération Internationale en Recherche Agronomique pour le Développement*). Ela foi criada com o intuito de padronizar uma metodologia para a criação de sistemas/simulações a serem utilizadas como auxiliares nas tomadas de decisões de gestores. É dividida em três estágios:

- *Construção de um mundo artificial.* Este estágio envolve um ou mais pesquisadores e deve ser usado para reunir informações sobre o sistema em estudo, ou seja, identificar os diferentes atores e percepções envolvidos (as) e usar SMA para a modelagem.
- *Validação do modelo cognitivo.* Estágio de testes do modelo proposto. O teste deve ser realizado a fim de analisar as representações e interações dos agentes, utilizando um RPG.
- *Simulações.* Este estágio mostra como as dinâmicas do sistema surgem de interações entre os atores com diferentes pesos e representações. É dividida em dois sub-estágios:
 - *RPG.* Onde os atores usam suas interações e mostram diferentes cenários que são úteis para testes.
 - *SMA.* Utilizado para simular os cenários emergentes do sub-estágio anterior. Após as simulações, todos os resultados devem ser discutidos com os envolvidos.

2.5.2. Games and Multi-Agent-Based Simulation (GMABS)

A metodologia foi proposta em [Barreteau et al. 2001] e nomeada em [Adamatti et al. 2007] de GMABS (*Games and Multi-Agent-Based Simulation*),

¹www.cirad.fr

pois os autores principais não nomearam a abordagem. Nesta abordagem, o principal enfoque é na descrição dinâmica das técnicas de integração entre RPG e MABS. Esta metodologia esta dividida em cinco passos:

1. Os jogadores passam a conhecer as regras, os possíveis movimentos e o ambiente onde serão inseridos no RPG. É definido, então, com qual personagem cada um jogará.
2. Começa uma nova rodada e os jogadores tem um tempo para negociarem e decidirem suas ações da rodada.
3. As informações são repassadas para o simulador SMA.
4. Os dados são computados e o ambiente será atualizado em um novo cenário.
5. O ambiente atualizado é repassado aos jogadores e, se o jogo não tiver terminado, novas ações serão tomadas (passo 2).

O projeto utiliza uma junção das duas abordagens. Para a primeira versão do jogo (jogo de tabuleiro, estado atual do projeto), que o módulo de cálculos computacional é usado, foi implementado utilizando a metodologia GMABS. O projeto em geral, porém, para a melhor relação entre os envolvidos, usa a abordagem ComMod.

2.6. Trabalhos Relacionados

Uma Revisão Sistemática da Literatura (RSL) foi feita por [Farias et al. 2019], utilizando a metodologia proposta em [Mariano et al. 2017], para reunir todas as pesquisas nos principais repositórios considerando os três campos principais (RPG, SMA e GRN). Para tanto, eles utilizaram os parâmetros: artigos que foram publicados em menos de 10 anos contando do RSL, utilizar aos repositórios ACM², IEEE³, ScienceDirect⁴, Scopus (Elsevier)⁵ e SpringerLink⁶, e a questão principal “*What are the main works addressing natural resource management using multiagent systems and role-playing games?* - Quais são os principais trabalhos que abordam gerenciamento de recursos naturais usando sistemas multiagente e jogos de papéis?” [Farias et al. 2019]. Foram selecionados 10 artigos, que foram divididos em 4 grupos:

- *RPG* → *SMA*. Este grupo se caracteriza pelos trabalhos onde o RPG é jogado pelos envolvidos em papel ou cartas e, no final do jogo, as informações são transformadas em uma simulação multiagente.
- *SMA* → *RPG*. O SMA é desenvolvido pelos pesquisadores e, então, um RPG é criado com as métricas usadas no SMA para serem validadas pelos *stakeholders*.
- *RPG* + *SMA*. Neste caso, o RPG é feito em papel com um módulo de cálculos computacional, pela complexidade dos cálculos a serem realizados, ou seja, o RPG e o SMA trabalham em conjunto.
- *RPG* ++ *SMA*. Por último, essa forma de integração é definida pelo uso de um RPG computacional ligado com o SMA.

Os trabalhos pertencentes aos últimos dois grupos foram escolhidos como trabalhos relacionados devido ao estado atual do projeto (RPG + SMA) e o objetivo (RPG ++

²<https://dl.acm.org>

³<https://ieeexplore.ieee.org/Xplore/home.jsp>

⁴<https://www.sciencedirect.com>

⁵<https://www.elsevier.com/pt-br/solutions/scopus>

⁶<https://link.springer.com>

SMA). Em [Campo et al. 2009], é desenvolvido um ambiente que facilita a comunicação e negociação entre diferentes envolvidos sobre seus interesses, consequências de estratégias e identificação de intervenção de áreas. Os autores optaram por utilizar a abordagem ComMod.

O projeto proposto em [Le Page and Perrotton 2017] cria uma nova metodologia para modelagem de ambientes. Para o teste desta nova abordagem, os autores utilizaram o ComMod, criando um SMA e, conseqüentemente, um RPG, onde era estudada a tomada de decisão de uma vila quanto à possíveis incidentes em suas plantações.

[Adamatti et al. 2009] descreve um projeto que os autores usam uma ferramenta modelada por eles para os jogadores determinarem a qualidade e quantidade de água em um sistema de captação presente em um perímetro urbano. Foram feitos dois protótipos, onde, no primeiro, os participantes jogaram com outros participantes e, no segundo, eles jogaram randomicamente com agentes inteligentes. A abordagem utilizada para esse jogo foi a GMABS.

3. RPG Proposto

O jogo, denominado Gorim, consiste em cinco classes de personagens (Empresário, Agricultor, Fiscal Ambiental, Prefeito e Vereador) e duas cidades (Atlantis e Cidadela). Em cada cidade existe dois empresários, três agricultores, um fiscal, um prefeito e um vereador. A classe Empresário, porém, é dividida em quatro tipos: Semente, Máquina, Fertilizante e Agrotóxico, onde os empresários dos dois primeiros tipos respondem à Atlantis e, os outros, à Cidadela. Os personagens dessa classe são responsáveis por venderem seus produtos aos personagens agricultores, que fazem as próprias escolhas de combinações de produtos para plantar em suas terras, visto que cada combinação gera uma quantidade diferente de poluição e de produtividade. Os membros da classe Fiscal Ambiental, por sua vez, devem observar a poluição de cada jogador anterior para aplicar multa, se necessário. Além disso, são responsáveis por dar licença de Selo Verde aos agricultores, dando a eles desconto no imposto. Prefeitos são responsáveis por aplicar o dinheiro dos impostos e multas em soluções ambientais, que diminuem a poluição global, e, também, em aumentar ou diminuir os impostos. Os Vereadores, porém, devem dar ideias aos prefeitos quanto aos impostos e as ações ambientais.

Cada tipo de empresário tem tipos diferentes de produtos à oferecer aos agricultores. O empresário de Semente vende sementes de Hortaliça, Arroz ou Soja. O de Fertilizantes, por sua vez, vende fertilizantes Comum, Premium ou Super Premium, assim como os empresários de Agrotóxico. O empresário de Máquinas, porém, pode *alugar* três pacotes de máquinas, onde uma contém somente uma Semeadora, a segunda contém uma Semeadora e uma Colheitadeira, e a terceira uma Semeadora, uma Colheitadeira e um Drone. Este empresário também pode alugar um Pulverizador, que não provê nenhum aumento na produtividade, mas diminui a poluição produzida pela metade.

Os Agricultores têm em seu poder seis unidades de terra, onde é possível a plantação de uma semente com um fertilizante e um agrotóxico ou um pacote de máquina, ou seja, não é possível pôr um agrotóxico e um pacote de máquina no mesmo espaço de terra. Porém, é possível colocar um Pulverizador com um agrotóxico ou um pacote qualquer de máquina. É possível a não plantação também, ou seja, deixar a terra vazia. Cada produto tem um preço e cada combinação de produtos gera uma produtividade diferente,

que, por sua vez, dependendo do montante, cai em uma das classes de imposto, aumentando ou diminuindo a quantidade de imposto a ser paga.

Cada rodada do jogo é dividida em duas etapas, onde na primeira etapa jogam os empresários e os agricultores e, na segunda etapa, jogam os fiscais, os prefeitos e os vereadores. Na situação atual do projeto, jogam 10 jogadores de cada vez, ou seja, os papéis elegíveis (fiscal, prefeito e vereador) são jogados pelos jogadores da primeira etapa da rodada, que são eleitos de duas em duas rodadas pelos atores da mesma cidade. A Figura 2 ilustra a integração entre os papéis do jogo.

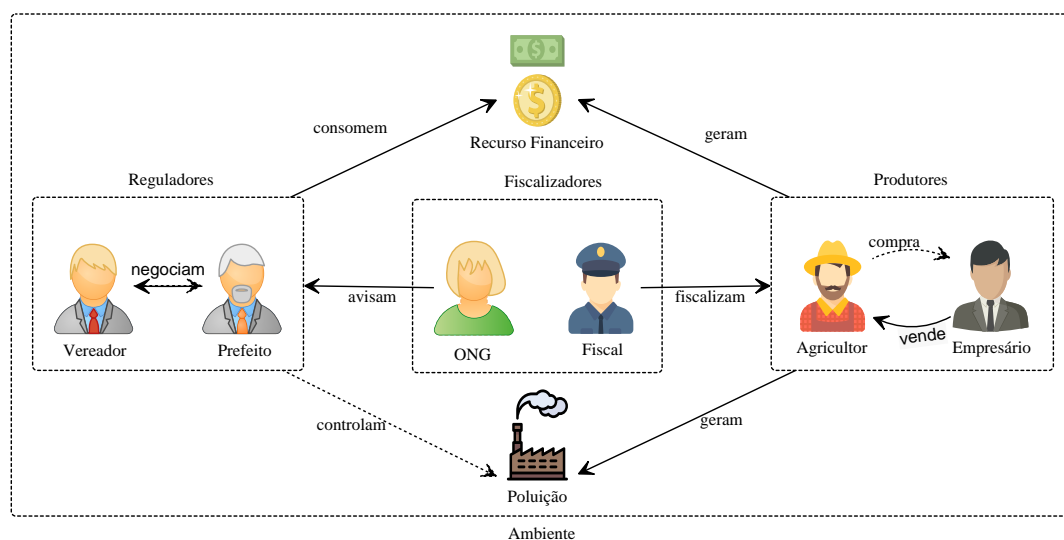


Figura 2. Diagrama de Integração entre os agentes do jogo.

A Figura 3a mostra estudantes e professores jogando no quarto teste-simulação do RPG, e a Figura 3b apresenta os broches feitos para os jogadores se identificarem durante o jogo. Na realização destes primeiros testes, foi possível verificar a integração entre os papéis, verificar regras e valores, e de fato consolidar o jogo. Também foi possível perceber que os jogadores, além de entenderem o objetivo do jogo, que é sobre a poluição da água, também se divertiram e criaram estratégias complexas.

3.1. Motor de Cálculos

Mesmo começando com valores redondos de “Dinheiros” (moeda do jogo) para cada jogador, a partir da cobrança de impostos e após compras/vendas, torna-se inviável o cálculo à mão do total de cada jogador, da poluição global e individual de cada um, o que tornou necessário a criação de um módulo computacional para fazer isso. Este motor de cálculos foi programado em JAVA estruturada na programação orientada a agentes. O diagrama de classes da estrutura do motor está apresentada na Figura 4. Além das classes apresentadas no diagrama, há a classe Mundo, que recebe as entradas da interface e chama os métodos de cada classe. Ao final de cada etapa, o motor ainda tem como saída os arquivos de resumo de cada jogador, que mostra o que foi feito na etapa. Além disso, durante todo o jogo existem outros arquivos: o arquivo onde todas as entradas são salvas, caso acontece alguma falha, se possa voltar ao lugar onde estava na execução, e

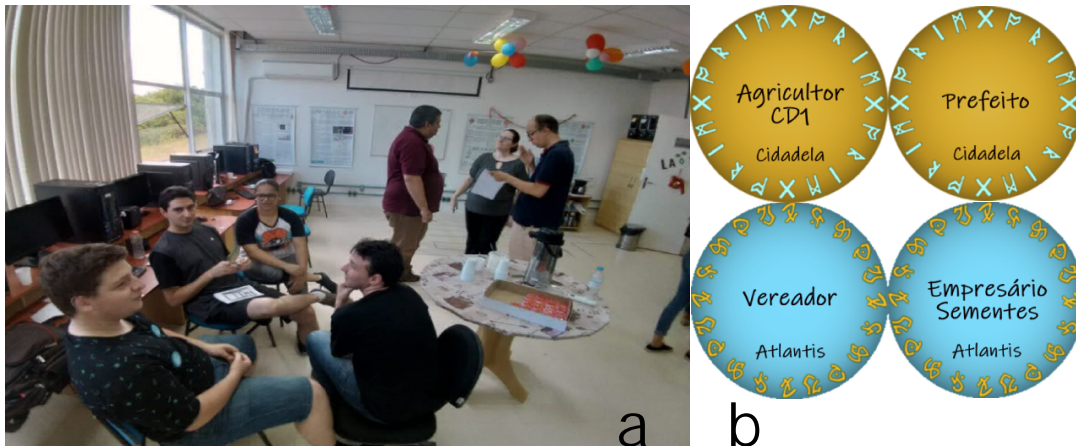


Figura 3. (a) Fotografia tirada em um teste-simulação do RPG. (b) Quatro dos 12 broches criados para identificar os papéis durante o jogo.

os arquivos de log, um textual que mostra de forma explícita todas as movimentações do jogo e um CSV, que mostra de forma resumida para a execução de uma mineração de dados, por exemplo.

4. Conclusões e Trabalhos Futuros

Este trabalho mostrou a base teórica, os trabalhos relacionados e o estado atual de um projeto que envolve as áreas de Sistemas Multiagente, Jogos de Papéis e Gerenciamento de Recursos Naturais. O projeto busca o desenvolvimento de um RPG computacional para ajudar a tomada de decisão do Comitê da Lagoa Mirim e do Canal São Gonçalo.

Atualmente, o projeto está em desenvolvimento, onde o jogo de tabuleiro está sendo refinado. O motor de cálculos está desenvolvido em JAVA. A etapa atual do projeto é a implementação da interface gráfica, que está sendo feita na linguagem TypeScript, com o uso do *framework* Angular. Para ligar ambas as partes, está sendo usado o *framework* Spring, para JAVA. Após, tudo será colocado em um servidor, para o jogo ser acessado remotamente. Junto a isso, os *logs* de dados de simulações foram usados em mineração de dados, traçando estratégias de jogadores para serem criados os NPCs. Estes jogarão com jogadores reais, quando não houver o número suficiente de jogadores necessário para começar o jogo, e será utilizada a linguagem AgentSpeak, da qual é possível a comunicação com o JAVA por meio do Jason.

5. Agradecimentos

Gostaríamos de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e à Agência Nacional das Águas (ANA) pelo apoio financeiro via Edital nº 16/2017.

Referências

Adamatti, D. F., Sichman, J. S., and Coelho, H. (2007). Utilização de rpg e mabs no desenvolvimento de sistemas de apoio a decisão em grupos. *Anais do IV Simpósio Brasileiro de Sistemas Colaborativos SBSC 2007*, page 15.

- Adamatti, D. F., Sichman, J. S. a., and Coelho, H. (2009). An analysis of the insertion of virtual players in gmaps methodology using the vip-jogoman prototype. *Journal of Artificial Societies and Social Simulation*, 12(3):7.
- Ahmad, S. and Simonovic, S. P. (2005). An artificial neural network model for generating hydrograph from hydro-meteorological parameters. *Journal of Hydrology*, 315(1-4):236–251.
- Alvares, L. O. and Sichman, J. S. (1997). Introdução aos sistemas multiagentes. In *XVII Congresso da SBC-Anais JAI'97*.
- Barnaud, C., Promburom, T., Trébuil, G., and Bousquet, F. (2007). An evolving simulation/gaming process to facilitate adaptive watershed management in northern mountainous thailand. *Simulation & gaming*, 38(3):398–420.
- Barreteau, O., Bousquet, F., and Attonaty, J.-M. (2001). Role-playing games for opening the black box of multi-agent systems: Method and lessons of its application to senegal river valley irrigated systems. *Journal of Artificial Societies and Social Simulation*, 4.
- Barreteau, O., Le Page, C., and D'aquino, P. (2003). Role-playing games, models and negotiation processes. *Journal of Artificial Societies and Social Simulation*, 6(2).
- Bordini, R. H., Vieira, R., and Moreira, A. F. (2001). Fundamentos de sistemas multiagentes. In *Anais do XXI Congresso da Sociedade Brasileira de Computação (SBC2001)*, volume 2, pages 3–41.
- Bousquet, F., Barreteau, O., Le Page, C., Mullon, C., and Weber, J. (1999). An environmental modelling approach: the use of multi-agent simulations. *Advances in environmental and ecological modelling*, 113(122).
- Bousquet, F. and Le Page, C. (2004). Multi-agent simulations and ecosystem management: a review. *Ecological modelling*, 176(3-4):313–332.
- Campo, P. C., Mendoza, G. A., Guizol, P., Villanueva, T. R., and Bousquet, F. (2009). Exploring management strategies for community-based forests using multi-agent systems: A case study in palawan, philippines. *Journal of Environmental Management*, 90(11):3607 – 3615.
- European Environment Agency (2018). Increasing environmental pollution (gmt 10). Online. Disponible at: <https://www.eea.europa.eu/soer-2015/global/pollution>. Accessed: november 15, 2019.
- Farias, G., Leitzke, B., Born, M., Aguiar, M., and Adamatti, D. (2019). Systematic review of natural resource management using multiagent systems and role-playing games. In *18th Mexican International Conference on Artificial Intelligence - MICAI*, Xalapa. Springer.
- Franklin, S. and Graesser, A. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 21–35. Springer.
- Frozza, R. (1997). Simula: Ambiente para desenvolvimento de sistemas multiagentes reativos. Master's thesis, Universidade Federal do Rio Grande do Sul (UFRGS).
- Fuller, M. M., Wang, D., Gross, L. J., and Berry, M. W. (2007). Computational science for natural resource management. *Computing in Science & Engineering*, 9(4):40.

- Gilbert, N. and Troitzsch, K. (2005). *Simulation for the social scientist*. McGraw-Hill Education (UK).
- Holzman, B. (2009). Natural resource management. [Online; accessed 30 out. 2019] <http://online.sfsu.edu/bholzman/courses/GEOG%20657/>.
- Le Page, C., Bobo Kadiri, S., Towa, K., William, O., Ngahane Bobo, F., and Waltert, M. (2015). Interactive simulations with a stylized scale model to codesign with villagers an agent-based model of bushmeat hunting in the periphery of korup national park (cameroon). *Journal of Artificial Societies and Social Simulation*, 18(1).
- Le Page, C., Dray, A., Perez, P., and Garcia, C. (2016). Exploring how knowledge and communication influence natural resources management with rehab. *Simulation & Gaming*, 47(2):257–284.
- Le Page, C. and Perrotton, A. (2017). Kilt: A modelling approach based on participatory agent-based simulation of stylized socio-ecosystems to stimulate social learning with local stakeholders. In Sukthankar, G. and Rodriguez-Aguilar, J. A., editors, *Autonomous Agents and Multiagent Systems*, pages 31–44, Cham. Springer International Publishing.
- Leitzke, B., Farias, G., Melo, M., Gonçalves, M., Born, M., Rodrigues, P., Martins, V., Barbosa, R., Aguiar, M., and Adamatti, D. (2019). Sistema multiagente para gestão de recursos hídricos: Modelagem da bacia do são gonçalo e da lagoa mirim. In *Anais do X Workshop de Computação Aplicada a Gestão do Meio Ambiente e Recursos Naturais*, Belém, Pará, Brasil.
- Lesser, V. R. (1999). Cooperative multiagent systems: a personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):133–142.
- Mariano, D. C. B., Leite, C., Santos, L. H. S., Rocha, R. E. O., and de Melo-Minardi, R. C. (2017). A guide to performing systematic literature reviews in bioinformatics. *arXiv preprint arXiv:1707.05813*.
- Moreno, A. (2003). Medical applications of multi-agent systems. *Computer Science and Mathematics Department, University of Rovira, Spain*.
- Pereira, C. E. K. (2003). Construção de personagem & aquisição de linguagem: O desafio do rpg no ines. In *vol. 10, (jul/dez) Rio de Janeiro INES, 2004 Semestral ISSN 1518-2509 1–Forum–Instituto Nacional de Educação de Surdos*, page 7.
- Perrotton, A., Garine-Wichatitsky, D., Fox, H. V., and Le Page, C. (2017). My cattle and your park: codesigning a role-playing game with rural communities to promote multistakeholder dialogue at the edge of protected areas. *Ecology and Society*, 22(1).
- Ponte, B., De la Fuente, D., Parreño, J., and Pino, R. (2016). Intelligent decision support system for real-time water demand management. *International Journal of Computational Intelligence Systems*, 9(1):168–183.
- Tundisi, J. G. (2006). Novas perspectivas para a gestão de recursos hídricos. *Revista USP*, 1(70):24–35.
- Wooldridge, M. (2002). *Introduction to MultiAgent Systems*. John Wiley and Sons.

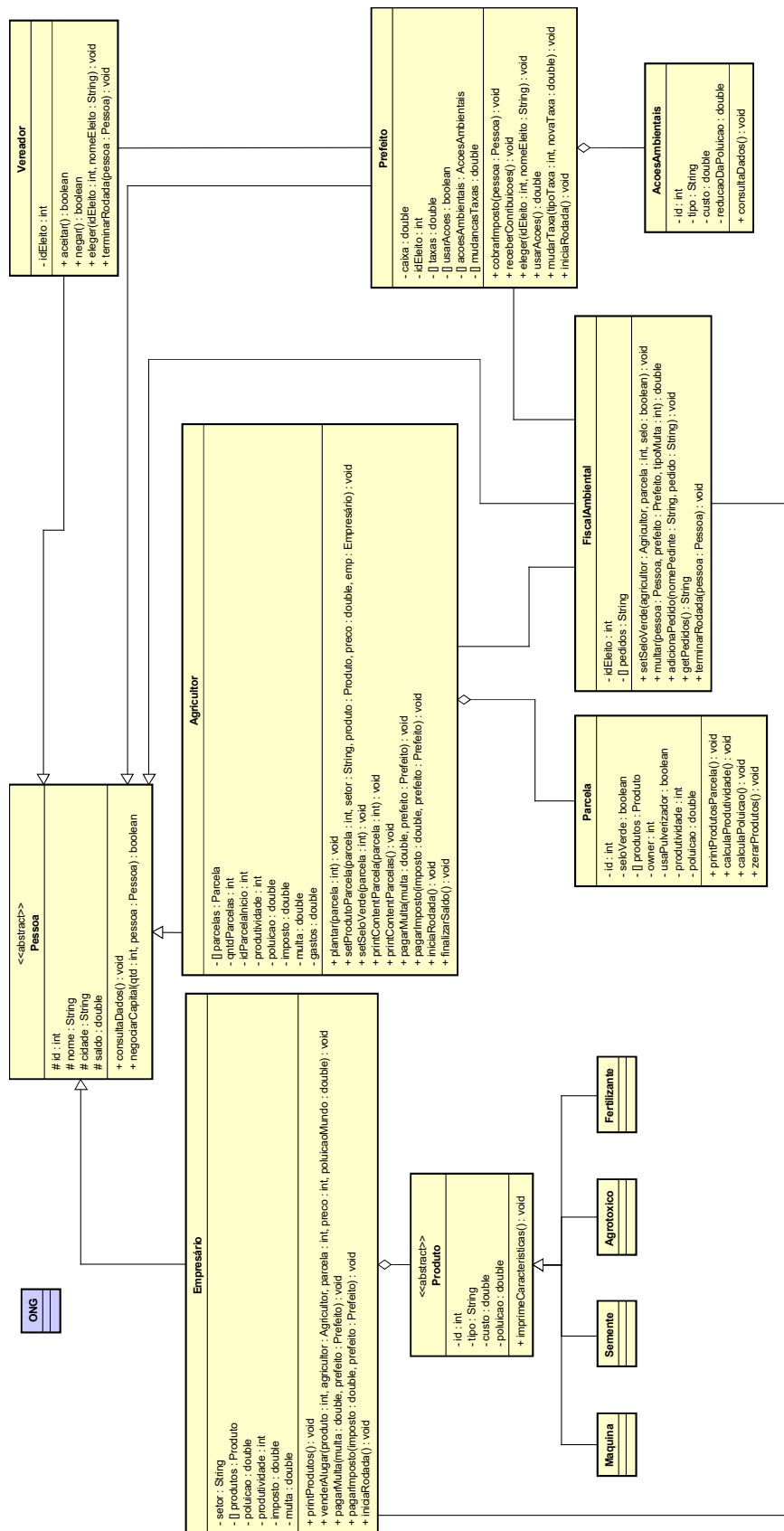


Figura 4. Diagrama de Classes que mostra a estrutura do Motor de cálculos.

Aspectos teóricos e aplicados para o desenvolvimento de uma simulação social baseada em agentes para análise dos desequilíbrios de gênero na tecnologia*

Gúlia Bordignon Silveira¹, Diana F. Adamatti¹

¹Programa de Pós-Graduação em Computação
Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Rio Grande – RS – Brasil


{gi, dianaadamatti}@furg.br

Resumo. *Conforme subimos a análise na escala de poder e prestígio na carreira científica e tecnológica, menos vemos rostos femininos. Isso é explicado pelo fato da computação ser construída entorno da masculinidade normativa, androcêntrica, branca, ocidental e heterossexual, que exclui qualquer indivíduo que não se enquadre nestes padrões. Pensar nas mulheres como sujeito da ciência e tecnologia faz com que surjam perguntas importantes na compreensão de fenômenos que ressignificaram sua participação na área. Uma abordagem pertinente para a compreensão destes fenômenos é a simulação social baseada em agentes, que possibilita a visão de questões centrais de um problema e estratégias de intervenção, através de complexos processos de interação e simulações computacionais. Desta forma, este trabalho apresenta aspectos teóricos e aplicados para que seja possível o desenvolvimento de uma simulação social baseada em agentes que aborde as desigualdades de gênero e barreiras enfrentadas pelas mulheres nas ciências tecnológicas.*

1. Introdução

A participação das mulheres na tecnologia já foi considerada um direito e um ponto importante para combater a naturalização dos papéis de gênero e estimular ambientes de trabalho mais diversos [Programaria 2015]. Hoje, para além de um direito, também configura-se como uma necessidade para o desenvolvimento da área científica e tecnológica do país.

Na última década, a produção científica nacional sobre o tema recebeu um grande impulso com pesquisas desde o processo de reconhecimento histórico da participação feminina, à identificação de barreiras e formulação de indicadores de gênero no eixo tecnológico [Costa 2000]. No trabalho de [Lima 2008] a exclusão das mulheres da ciência é mapeada de duas formas; a horizontal ou territorial, que trata da divisão de áreas conhecimento caracterizadas em femininas ou masculinas nas ciências, e a vertical ou hierárquica que aborda as exclusões a nível de carreira científica, como o fenômeno do “teto de vidro”. Ambas segregações se combinam para garantia da perpetuação das desigualdades de gênero.

* O trabalho Aspectos teóricos e aplicados para o desenvolvimento de uma simulação social baseada em agentes para análise dos desequilíbrios de gênero na tecnologia de Gúlia Bordignon Silveira e Diana F. Adamatti está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial 4.0 Internacional. <http://creativecommons.org/licenses/by-nc/4.0>

Indicadores de educação e trabalho segregados por gênero vêm evidenciando que as barreiras estão longe de serem superadas, por isso, considerando a relevância e necessidade de visibilidade para o tema, o objetivo deste trabalho é levantar os aspectos teóricos e metodológicos para o desenvolvimento de uma simulação social baseada em agentes (SSBA) que contenha uma emulação das dinâmicas das desigualdades de gênero mais marcantes na tecnologia.

2. Desigualdades de gênero na tecnologia

De forma geral, as mulheres são maioria na graduação representando 55% dos estudantes ingressantes, 57% dos matriculados e 61% dos concluintes. Na licenciatura, por exemplo, 70,6% das matrículas são do sexo feminino [INEP 2018]. Conseguimos observar as barreiras horizontais ao observar os cursos de graduação relacionados às ciências tecnológicas¹, apesar do número de cursos terem crescido 586% nos últimos 24 anos no Brasil, o percentual de mulheres matriculadas nos mesmos reduziu, passando de 34,8% para 15,5% [INEP 2018]. Segundo dados gerais da SBC, a participação nestes cursos é de 13% de alunas matriculadas e 15% de alunas formadas nas turmas do ano de 2017. Considerando isoladamente os cursos de Engenharia da Computação e de Software, a desigualdade entre homens e mulheres é ainda maior, dos 2500 formados no país, apenas 13% são do sexo feminino. Em contraste no curso de licenciatura em computação temos 42% de mulheres graduadas, indicando que os estereótipos de profissões ainda estão refletidos na educação. Nos cursos de Ciência da Computação e Sistemas de Informação as desigualdades ainda continuam marcantes com 16% de mulheres concluintes em 2017 [Nunes 2018].

Além da sub-representação, Dados da Pesquisa Nacional por Amostra de Domicílios evidenciam que no primeiro ano da faculdade, oito em cada dez delas desistem [PNAD 2018]. Esse fenômeno é fortemente justificado com utilização de hipóteses fortes sobre as preferências individuais e os objetivos da família [Madalozzo et al. 2010].

A respeito das mulheres produzindo ciência, o estudo [Elsevier 2017] analisou o desempenho de pesquisas sobre últimos 20 anos, em várias regiões do mundo no período entre 2011 e 2015 no Brasil, quase metade (49%) dos estudos científicos foram produzidos por mulheres, fator positivo visto que contrasta fortemente com o período de 1996 a 2000, quando as mulheres contribuíram com apenas 38% da produção científica brasileira. Apesar de ativas nas produções de estudos, as mulheres possuem apenas 25% de representatividade na categoria mais alta do CNPq o que nos leva a pensar nas barreiras verticais.

Encontra-se na revisão sistemática de [Cardoso et al. 2017], metáforas para descrever algumas barreiras identificadas pelas mulheres na ascensão profissional, onde encontramos situações com antecedentes e consequentes da discriminação e preconceitos enfrentados por profissionais do sexo feminino, tais como o Teto de vidro². Pe-

¹Ciência da Computação, Engenharia de Computação, Sistemas de Informação, Licenciatura em Computação, Cursos de Tecnologia, Engenharia de Software e Outros Cursos

²Onde o “vidro” representa as barreiras invisíveis, e o “teto” às dificuldades de ascensão a níveis empresariais mais elevados

nhasco de Vidro³ [Ryan and Haslam 2005], Labirinto⁴ [Eagly et al. 2007]. Encontram-se também metáforas individuais, alheias às organizações que contribuem para baixa representação feminina em posições de prestígio, tais como o Muro da Maternidade⁵ [Crosby et al. 2004], Barreiras no Canal⁶ [Kekelis et al. 2005].

Uma abordagem que também pode ser aplicada para entendimento das discriminações de gênero, dentro de complexos processos de interação, são as simulações sociais baseadas em agentes. Este ramo investiga o uso de tecnologia de agentes para simular fenômenos sociais em um computador, possibilitando a visão de questões centrais e estratégias de intervenção [Davidsson 2002].

Uma simulação social possibilita a visão de questões centrais de um cenário, permitindo traçar e aplicar estratégias de intervenção em sociedades artificiais e verificar seus resultados antes de infligir teste em sociedades humanas [Robison-Cox et al. 2007]. Desta forma, pode ser possível aplicar estratégias para redução dos desequilíbrios de gênero, pois o aumento da participação feminina nas ciências e tecnologias pode trazer grandes transformações sociais e econômicas de impacto favorável para a sociedade.

3. Simulação Social Baseada em Agentes

Uma forma de definir a simulação social baseada em agentes (SSBA) é que ela constitui a interseção de três campos científicos: a computação baseada em agentes; as ciências sociais; e a simulação computacional, que podem ser observados na Figura 1 [Davidsson 2002].

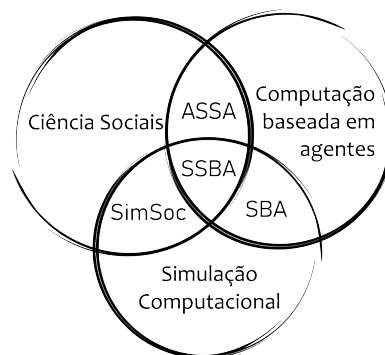


Figura 1. Áreas que definem a SSBA. Adaptado de [Davidsson 2002]

As ciências sociais são o campo de estudos que englobam diferentes ciências que estudam os aspectos sociais do mundo humano, interação entre entidades sociais, a vida social de indivíduos e grupos humanos. A computação baseada em agentes, por sua vez

³Que denomina a onda de discriminação em que a posição de liderança feminina está associada a um maior risco de fracasso.

⁴O labirinto simboliza ao caminho sinuoso, que requer persistência, consciência do próprio progresso e visão dos desafios a serem enfrentados como a discriminação por motivação de gênero; o preconceito com a mulher no papel de liderança; dúvidas quanto à competência feminina e a exigência da conciliação da vida familiar com a profissional.

⁵Que indica que as mulheres tendem a ser empregadas em locais de trabalho onde a jornada de trabalho é em tempo parcial pois precisam dar conta de questões familiares.

⁶Ilustra que barreiras educacionais, como a falta de esclarecimento sobre as possibilidades de escolha da carreira na área tecnológica afastam e desincentivam as meninas.

é uma área de pesquisa dentro da ciência da computação que estuda a modelagem para conceber agentes, sua arquitetura, suas organizações e interações de modo genérico. A simulação computacional é uma modelagem que visa obter uma melhor compreensão de algumas características do contexto simulado, realizando estudos sobre sistemas reais, sua reação ante influências externas e internas, ou sua abrangência no meio ambiente [Strack 1984].

Os estudos ilustrados na Figura 1, onde apenas dois dos campos se cruzam são: os Aspectos Sociais dos Sistemas de Agentes (ASSA), composto pela interseção entre as ciências sociais e a computação baseada em agentes e que diz respeito ao estudo de normas, instituições, organizações, cooperação, competição entre agentes. A Simulação Baseada em Agentes (SBA) compreende as atividades pertencentes à interseção entre computação baseada em agentes e a simulação computacional e estuda o uso de tecnologia de agente para simular qualquer fenômeno em um computador. Por último, a Simulação Social (SimSoc) compreende a interseção entre as ciências sociais e simulação computacional e corresponde à simulação de fenômenos sociais em um computador utilizando qualquer técnica de simulação [Davidsson 2002].

Delineados os campos de interseção, a SSBA é caracterizada como uma extensão natural de todas as áreas abordadas sendo ramo que investiga o uso de tecnologia de agentes para simular fenômenos sociais em um computador [Davidsson 2002]. A SSBA possibilita a visão de questões centrais de um cenário, permitindo traçar e aplicar estratégias de intervenção em sociedades artificiais e verificar seus resultados antes de infligir teste em sociedades humanas.

4. Resultados Preliminares: SSBA Reprograma

A construção de uma simulação pode ser dividida em três grandes etapas: etapa de concepção e modelagem, onde é realizado o levantamento de informações e viabilidade de desenvolvimento do modelo; etapa de experimentos, na qual são realizados experimentos sobre o modelo criado através da alteração de parâmetros e variáveis; e por fim, a etapa de validação, onde são realizadas comparações dos dados experimentais obtidos com a simulação e comparação com a realidade [Frozza 1997].

Na primeira etapa foi estudada a área em que se aplicaria o estudo, posteriormente foram feitas definições do domínio do problema e definidas perguntas de pesquisa. O cenário alvo de modelagem do sistema foi definido como o ambiente profissional e acadêmico das áreas da tecnologia da informação. O sujeito central é a mulher cis⁷ que escolhe a carreira tecnológica. A pergunta de pesquisa que a simulação visa responder é: Quais são os fatores que amplificam os desequilíbrios de gênero na tecnologia da informação? Contudo, pretende-se também buscar resposta a seguinte pergunta secundária: É possível traçar estratégias para reduzir os desequilíbrios de gênero?

Para o embasamento da modelagem foi utilizada a pesquisa bibliográfica descritiva. Optou-se por essa técnica pois procura explicar um problema a partir de referência

⁷Cisgênero é o indivíduo que se identifica com o sexo biológico com o qual nasceu. Um exemplo de cisgênero é uma pessoa que nasceu com genitália feminina e cresceu com características físicas de “mulher”, além disso adotou padrões sociais ligados ao feminino, comumente expressados em roupas, gestos, tom de voz.

teóricas publicadas, observando, registrando e correlacionando fatos e fenômenos, procurando descobrir com maior precisão possível a frequência com que um fenômeno ocorre e sua relação e conexão com outros. Para [Cervo et al. 1980], a pesquisa descritiva busca conhecer diversas situações e relações que ocorrem na vida social, política e os aspectos do comportamento humano tomado como indivíduo ou grupo em comunidades mais complexas.

Para complementar as informações sobre o modelo, será utilizado o questionário de perguntas abertas como instrumento de coleta de dados. O questionário contém um conjunto de questões logicamente relacionados a um problema central e pode ser aplicado de forma simultânea a um número maior de indivíduos sendo um instrumento que possibilita extrair com mais exatidão o que se deseja [Cervo et al. 1980].

Após as informações obtidas, será criado o modelo conceitual da simulação, onde tem-se a definição dos agentes e ambiente, suas definições, relações e características. A construção deste modelo visa traduzir os conceitos do sistema que se pretende modelar, dando uma estrutura que irá orientar e facilitar sua transição para a modelagem computacional. Para isto, serão criados os diagramas de Caso de Uso e os Diagramas de Classe.

O diagrama de Caso de Uso constitui uma técnica baseada em cenários para levantamento de requisitos. Em sua forma mais simples, identifica e especifica o tipo da interação e os agentes envolvidos, dando uma ideia geral de como o sistema irá se comportar [Sommerville 2014]. O Diagrama de classes, por sua vez, é uma representação estática utilizada na área da programação para descrever a estrutura de um sistema, apresentando suas classes, atributos, operações e as relações entre os objetos [Sommerville 2014].

A implementação é dedicada à programação da simulação proposta a partir da modelagem conceitual construída. Para este trabalho, optou-se pela utilização de um software na construção da simulação, visto que oferece benefícios de praticidade e simplicidade no desenvolvimento de uma simulação em relação aos kits de ferramentas. A ferramenta selecionada foi NetLogo⁸. Para encerrar a primeira etapa de desenvolvimento, o processo de verificação da simulação será realizado utilizando o método de teste unitário. O teste unitário é uma metodologia que procura verificar o código em sua menor fração buscando problemas na implementação.

5. Conclusões e Trabalhos Futuros

Nos últimos anos, houveram consideráveis avanços nas pesquisas de gênero, ciência e tecnologia. Este fenômeno evidencia que este é um campo fértil para novas discussões acerca da naturalização da masculinidade que envolve a área e para contribuições que transformem esses espaços para mais democráticos e igualitários.

Entendemos que há muito a ser feito para impulsionar a participação plena das mulheres na área, por isso este projeto teve como objetivo levantar os aspectos teóricos e aplicados para o desenvolvimento de uma simulação social baseada em agentes para análise dos desequilíbrios de gênero na tecnologia, bem como verificar fatores que amplificam as desigualdades e estratégias que possam revertê-las.

De modo conclusivo, reiteramos que muito trabalho ainda deve ser feito. Apesar de termos encontrado ótimas produções bibliográficas para a embasar a modelagem da

⁸<https://ccl.northwestern.edu/netlogo/>

simulação, percebemos que ainda existe uma lacuna, quando tratamos de indicadores de mulheres trans e/ou negras na tecnologia e sugerimos como trabalhos futuros o desenvolvimento de uma simulação com a representação dessas interseccionalidades. Esperamos que com o uso desta SSBA, possamos apresentar novos resultados, de forma a contribuir para a área, propondo novos cenários de análise do problema.

Referências

- Cardoso, A. et al. (2017). Atitudes de mulheres com relação ao teto de vidro: um estudo em empresas do setor de tecnologia de informação e comunicação.
- Cervo, A. L., Bervian, P. A., Da Silva, R., et al. (1980). Metodologia científica.
- Costa, A. A. (2000). Gênero, poder e empoderamento das mulheres. <http://pactoglobalcreapr.files.wordpress.com/2012/02/5-empoderamento-analice.pdf>. Acesso em, 8:08–14.
- Crosby, F. J., Williams, J. C., and Biernat, M. (2004). The maternal wall. *Journal of Social Issues*, 60(4):675–682.
- Davidsson, P. (2002). Agent based social simulation: A computer science view. *Journal of artificial societies and social simulation*, 5(1).
- Eagly, A. H., Eagly, L. L. C. A. H., and Carli, L. L. (2007). *Through the labyrinth: The truth about how women become leaders*. Harvard Business Press.
- Elsevier (2017). *Gender in the Global Research Landscape: Analysis of Research Performance Through a Gender Lens Across 20 Years, 12 Geographies, and 27 Subject Areas*. Elsevier.
- Frozza, R. (1997). Simula: Ambiente para desenvolvimento de sistemas multiagentes reativos.
- INEP (2018). Notas estatísticas 2017. Coordenada por INEP, Disponível em.
- Kekelis, L. S., Ancheta, R. W., and Heber, E. (2005). Hurdles in the pipeline: Girls and technology careers. *Frontiers: A Journal of Women Studies*, pages 99–109.
- Lima, B. S. (2008). Teto de vidro ou labirinto de cristal? as margens femininas das ciências.
- Madalozzo, R., Martins, S. R., and Shiratori, L. (2010). Participação no mercado de trabalho e no trabalho doméstico: homens e mulheres têm condições iguais? *Revista Estudos Feministas*, 18(2):547–566.
- Nunes, D. (2018). Educação superior em computação, estatísticas 2017. *Sociedade Brasileira de Computação-SBC*.
- PNAD (2018). *Estatísticas de gênero: indicadores sociais das mulheres no Brasil*. IBGE e Coordenação de População e Indicadores Sociais.
- Programaria (2015). Mulheres perderam representatividade nos cursos de computação.
- Robison-Cox, J. F., Martell, R. F., and Emrich, C. G. (2007). Simulating gender stratification. *Journal of Artificial Societies and Social Simulation*, 10(3):8.

Ryan, M. K. and Haslam, S. A. (2005). The glass cliff: Evidence that women are over-represented in precarious leadership positions. *British Journal of management*, 16(2):81–90.

Sommerville, I. (2014). *Engenharia de software*.

Strack, J. (1984). *GPSS: modelagem e simulação de sistemas*. LTC.

Modelagem da Organização de um Sistema Multiagente no Contexto de Recursos Hídricos *

Míriam Blank Born¹, Diana F. Adamatti², Marilton Sanchotene de Aguiar¹

¹Programa de Pós-Graduação em Computação
Universidade Federal de Pelotas (UFPEL) – Pelotas – RS – Brasil

²Centro de Ciências Computacionais (C3)
Universidade Federal do Rio Grande (FURG) – Rio Grande – RS – Brasil


{mborn, marilton}@inf.ufpel.edu.br, dianaada@gmail.com

Abstract. *The study of multi-agent systems (MAS) applied to natural resource management presents numerous challenges, even if it is limited only to the distribution and sharing of water resources in this study. Therefore, the proper administration of this resource is essential in the search for solutions for this type of scenario. This paper aims to model and simulate a multi-agent system composed of three types of agents: regulators, producers, and supervisors. For that, it presents a case study of the watershed of Lagoa Mirim and Canal São Gonçalo. In Moise+, we introduced the organization's initial modeling of two roles that make up the system. This definition will assist in the implementation of the MAS because the intention is to use the JaCaMo framework since this platform integrates Jason, CArtaGO, and Moise+ technologies for multi-agent programming.*

Resumo. *O estudo de sistemas multiagente (SMA) aplicado ao gerenciamento de recursos naturais apresenta inúmeros desafios, mesmo limitando-se apenas à distribuição e ao compartilhamento de recursos hídricos neste estudo. Sendo assim, é indispensável a gestão adequada deste recurso na busca de soluções para este tipo de cenário. Este artigo apresenta como objetivo a modelagem e a simulação de um sistema multiagente composto por três tipos de agentes: reguladores, produtores e fiscalizadores. Além disso, apresenta um estudo de caso da bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo. Ainda, apresenta-se a modelagem inicial da organização no Moise+ de dois papéis que compõem o sistema. Esta definição auxiliará na implementação do SMA pois pretende-se utilizar o framework JaCaMo visto que esta plataforma integra as tecnologias Jason, CArtaGO e Moise+ para a programação multiagente.*

1. Introdução

O gerenciamento dos recursos hídricos torna-se a cada ano um dos grandes desafios vividos pela sociedade na atualidade. Assim, o estudo, a análise e a elaboração de novos modelos de gestão tornam-se relevantes para a busca de possíveis soluções neste

* O trabalho Modelagem da Organização de um Sistema Multiagente no Contexto de Recursos Hídricos de Míriam Blank Born, Marilton Sanchotene de Aguiar e Diana Francisca Adamatti está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

contexto. O Brasil dispõe de grandes bacias hidrográficas em seu território, entretanto apresenta diversos tipos de conflitos relacionados à distribuição e compartilhamento do recurso [Brito et al. 2020, Born et al. 2019b].

A modelagem multiagente apresenta-se como um recurso eficaz para a simulação de sistemas ecológicos e sociais complexos. Para a gestão participativa de recursos hídricos e o auxílio na tomada de decisão, as ferramentas baseadas em agentes são amplamente utilizadas em diversos domínios de aplicação [Filatova et al. 2013]. No contexto de uma bacia hidrográfica, por exemplo, o estudo e a gestão de recursos renováveis é um importante caminho na busca de possíveis soluções aos problemas enfrentados pelos interessados em um determinado ecossistema [Adamatti 2007].

Além disso, para a modelagem e o desenvolvimento destes sistemas complexos, torna-se necessária uma plataforma e/ou *framework* que forneça recursos computacionais adequados que facilitem o entendimento e detalhamento do problema como, no âmbito deste trabalho, a especificação da organização de um sistema multiagente (SMA), a partir de uma ferramenta como o Moise+ (*Model of Organization for multi-agent SystEms*) [Hübner 2003], e a modelagem do sistema, com o *framework* JaCaMo [Boissier et al. 2016], amplamente utilizado na programação multiagente.

Neste trabalho, a partir da modelagem de um jogo computacional baseado em Sistemas Multiagente e Jogos de Papéis (RPG) apresentada em [Leitzke et al. 2019], propõe-se a organização de um sistema multiagente composto por três tipos de agentes i) reguladores, ii) produtores e iii) fiscalizadores, com um estudo de caso da bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo. O estudo também define o diagrama de integração [Born et al. 2019b] apresentando a visão geral do sistema como os grupos de agentes, suas ações e recursos do ambiente. Os diagramas de classe e de sequência definem a formalização destas ações e as interações entre estes agentes.

Visto a complexidade do desenvolvimento de um sistema multiagente associado a um jogo computacional busca-se neste trabalho, como primeiro passo, a definição desta organização multiagente. A modelagem está sendo especificada no Moise+, visando contribuir para a implementação futura de todo o sistema. O *framework* JaCaMo será utilizado para desenvolvimento e simulação deste modelo, pois a plataforma integra as tecnologias Jason, CArAgO e Moise+ para a programação multiagente.

O artigo encontra-se organizado da seguinte forma. Na Seção 2 são apresentados conceitos fundamentais sobre Sistemas Multiagente. Na Seção 3 é introduzido um panorama geral sobre o *framework* JaCaMo, com suas características e a integração entre as ferramentas. Na Seção 4 é apresentado o estudo de caso, bem como a modelagem de uma organização multiagente no Moise+. E por fim, a Seção 5 apresenta as considerações finais e trabalhos futuros.

2. Sistemas Multiagente

A aplicação de SMA em modelagem e simulação no contexto do gerenciamento de recursos naturais é estudada há alguns anos, com inúmeras pesquisas publicadas e resultados promissores. Os modelos biológicos, assim como as interações sociais, servem de inspiração para o desenvolvimento de sistemas onde agentes inteligentes podem ser concebidos por meio de dispositivos de *hardware* e/ou *software* [Artero 2009]. Os

agentes representados por estes equipamentos ou programas devem ter a capacidade de perceber seu ambiente por meio de sensores e de agir sobre este por meio de atuadores [Russell and Norvig 2013].

Dentre alguns trabalhos neste âmbito pode-se mencionar [Adamatti 2007], onde jogadores virtuais são inseridos em um RPG com intuito de apoiar decisões em grupo no contexto de recursos naturais. Em [Gaudou et al. 2014], foi desenvolvida uma plataforma de modelagem e simulação baseada em agentes para estudar os impactos ambientais, econômicos e sociais de regulamentações relacionadas à utilização e ao gerenciamento da água. [Théron et al. 2014] utiliza a plataforma MAELIA no estudo de problemas de escassez da água e o estudo [Taillandier and Buard 2009] apresenta uma formalização do comportamento de agentes utilizando uma combinação de regras de produção e um método de tomada de decisão multicritérios no domínio da Ecologia.

[Nwana 1996] menciona que os agentes possuem três características importantes: *cooperar, aprender e agir de maneira autônoma*. [Bordini et al. 2001] ainda acrescentam coordenação, competição e negociação como aspectos relevantes na concepção de agentes. A partir destas características, existem diferentes maneiras de classificar os agentes [Nwana 1996, Artero 2009, Coppin 2010], compreendendo-se basicamente em agentes reativos, agentes colaborativos, agentes de comunicação e agentes de aprendizado.

As propriedades, a arquitetura e a estrutura dos agentes, bem como o ambiente em que estes estão inseridos, são implementadas de acordo com o problema a ser resolvido, a complexidade e o domínio específico de cada aplicação. Em [Russell and Norvig 2013, Rezende 2005, Luger 2013], diversas aplicações e algoritmos são apresentados de forma a exemplificar a busca de soluções para esta demanda crescente de problemas.

Para [Alvares and Sichman 1997] e [Bordini et al. 2001], inúmeros são os benefícios da utilização de um SMA: i) rapidez na resolução de problemas visto a inerência do processamento concorrente; ii) aumento da flexibilidade e escalabilidade através da conexão de vários sistemas; e, iii) aumento da capacidade de resposta à um determinado problema pelo fato de todos os recursos estarem localizados no mesmo ambiente e também, a modularidade obtida mediante esta técnica.

No desenvolvimento de SMA, uma arquitetura comumente utilizada é a BDI (*Beliefs-Desires-Intentions*), baseada em um modelo cognitivo que representam crenças, desejos e intenções [Hübner et al. 2004]. Para [Wooldridge 2002], *crenças* representam o que o agente sabe sobre si mesmo, dos demais agentes e o ambiente ao qual está inserido; *desejos* representam os estados que o agente almeja atingir, geralmente são objetivos; e, *intenções* são representadas pela sequência de ações que determinado agente comprometeu-se em executar para alcançar um objetivo.

3. Framework JaCaMo

O *framework* de programação multiagente JaCaMo [Boissier et al. 2016] tem sido desenvolvido ao longo dos anos e integra as plataformas Jason [Bordini et al. 2007], utilizado no desenvolvimento de agentes autônomos; CArTAgO [Ricci et al. 2011], aplicado no desenvolvimento de ambientes compartilhados; e, o Moise+ [Hübner et al. 2007] desenvolvido para a modelagem de organizações multiagente. A partir desta integração é proporcionado aos desenvolvedores um *framework* completo para aplicações de SMA.

Em JaCaMo consideram-se três: dimensões agente, ambiente e organização, com intuito de facilitar ao desenvolvedor a modelagem e implementação de sistemas multi-agente complexos [Thomasi 2014]. As conexões decorrem entre agente e ambiente (A-E) mediante o mapeamento das ações externas do agente para as operações nos artefatos (CArtAgO). As propriedades observáveis dos artefatos, observadas pelos agentes, são mapeadas para a base de crença do agente.

Entre organização e ambiente (O-E), considerando que a infraestrutura organizacional faz parte do ambiente no qual os agentes estão localizados, os artefatos organizacionais proveem operações utilizadas pelo agente e também para o gerenciamento da organização por agentes organizacionais. Em agente e organização (A-O), ocorre a partir do mapeamento dos objetivos, os quais são definidos na dimensão organizacional para os objetivos individuais de cada agente. Os objetivos da organização são delegados aos agentes a partir das obrigações [Hübner et al. 2010a].

A plataforma JaCaMo, a partir da integração destas ferramentas, composta pelos agentes, ambiente e interações, oferece um recurso para a escalabilidade de aplicativos complexos permitindo assim sua distribuição em diversos nós. Jason e Jade [Bellifemine et al. 2007] são ferramentas utilizadas na programação em nível de agente. No Jason, que é um interpretador para a linguagem *AgentSpeak-L*, é possível programar o comportamento do agente de acordo com uma abordagem declarativa, baseada em lógica e arquitetura BDI. Jade (*Java Agent Development Framework*) é utilizado para gerenciar o ciclo de vida dos agentes e possibilitar a troca de mensagens entre estes.

A partir da integração de Jason ao Jade tem-se a implementação da semântica dos atos de fala, as performativas do FIPA-ACL são integradas ao estado mental do agente. Com a integração de Jason ao CArtAgO permite-se a percepção e ações, sendo que todas as percepções dos artefatos são mapeadas em crenças e ações em planos para as operações do artefato [Boissier et al. 2016]. O CArtAgO possui artefatos especiais possibilitando o gerenciamento da estrutura do ambiente, que os agentes gerenciem o ciclo de vida do ambiente de trabalho e que interajam com o ambiente externo através de serviços disponíveis como: criar, destruir, ingressar e sair de áreas de trabalho locais ou remotas.

No JaCaMo, a dimensão organizacional é gerenciada por artefatos ORA4MAS (*ORganizational Artifacts for Multi-Agent Systems*), o qual consiste em uma abordagem baseada no metamodelo de agentes e artefatos (A&A). A infraestrutura para o gerenciamento organizacional do ORA4MAS é constituída por agentes e artefatos organizacionais. Os artefatos organizacionais são utilizados por estes agentes afim de facilitar suas atividades [Boissier et al. 2011, Thomasi 2014].

Em [Boissier et al. 2016], apresentou-se a utilização do JaCaMo em diversas áreas, dentre as quais na área de computação ambiental, como no gerenciamento inteligente de construções, ambientes virtuais inteligentes, gestão do conhecimento ou gerenciamento de crises.

4. Modelagem da Organização

Para este trabalho foi desenvolvida a modelagem inicial do problema, baseando-se na representação das interações básicas entre os papéis do sistema e a atuação destes no ambiente, de acordo com a especificação proposta em [Born et al. 2019a]. Neste estudo,

os agentes são classificados de acordo com os papéis que assumem e são divididos em três grupos principais (reguladores, fiscalizadores ou produtores), conforme Figura 1.

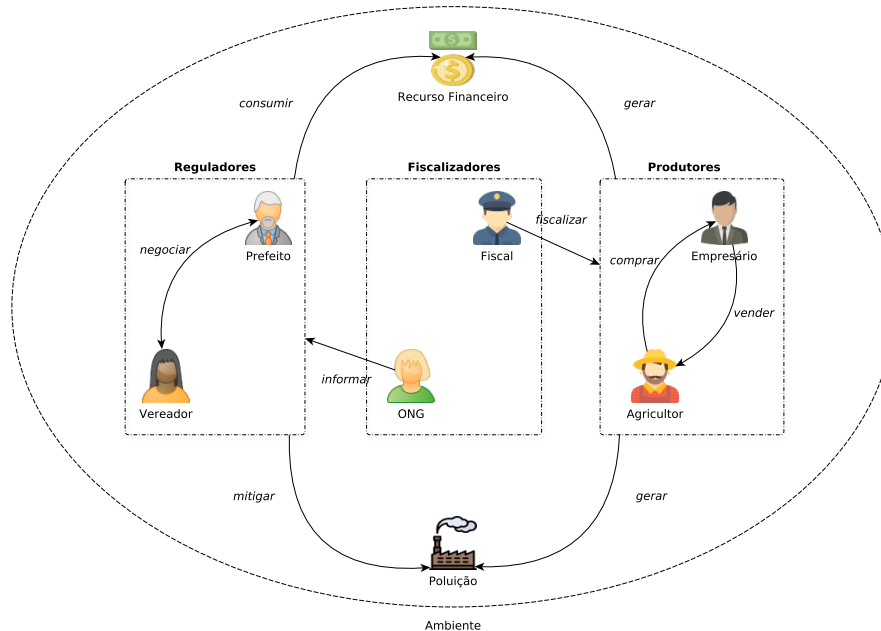


Figura 1. Diagrama de Integração do modelo proposto [Born et al. 2019a].

De acordo com [Born et al. 2019a], os agentes reguladores podem assumir os papéis de prefeito ou vereador, sendo responsáveis pela administração de recursos financeiros, com o objetivo de controlar ou mitigar a poluição do ambiente. Os agentes fiscalizadores possuem como atribuição fiscalizar ou informar irregularidades que impactam no ambiente a partir da produção e assumem os papéis de fiscal ambiental ou ONG (Organização Não-Governamental). Os agentes produtores, nos papéis de empresário ou agricultor, exploram o ambiente para suas produções e seu principal objetivo é obter recursos financeiros, a partir da venda dos seus produtos, de modo equilibrado com a poluição do meio ambiente. É relevante salientar que, não há restrição de interação entre os papéis neste sistema.

O Moise+ caracteriza-se por ser um modelo organizacional para sistemas multiagente composto de três dimensões: i) *estrutural*, constituída pelos grupos, papéis e ligações; ii) *funcional*, englobando planos globais, metas e missões; e, iii) *normativa* ou *deôntica*, a qual define as obrigações e permissões dos agentes. Em [Hübner et al. 2010b] encontra-se um amplo tutorial com a conceitualização e exemplos de utilização do Moise+.

No desenvolvimento deste trabalho, baseando-se no diagrama de interação da Figura 1, elaborou-se a especificação estrutural da organização para este estudo de caso, conforme a Figura 2. Esta especificação estrutural também foi baseada na definição do RPG [Born et al. 2019a] elaborado para a gestão participativa dos recursos naturais no contexto da bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo.

A especificação estrutural possui quatro grupos: `Jogo`, `Regulador`, `Fiscalizador` e `Produtor`, considerando que a interação pode ser intragrupo e externa a

estes. Todos os papéis são herdados do papel Jogador e possuem interação bilateral. Cada grupo e papel possui as cardinalidades correspondentes ao número de papéis definidos no trabalho [Born et al. 2019a], sendo que esta elaboração baseou-se no trabalho de [Hübner et al. 2002], com as devidas adaptações ao contexto deste estudo. No RPG, como o papel de jogador é uma pessoa, todos os demais papéis são herdados deste, por isso nesta especificação todos os papéis herdam características do papel de Jogador.

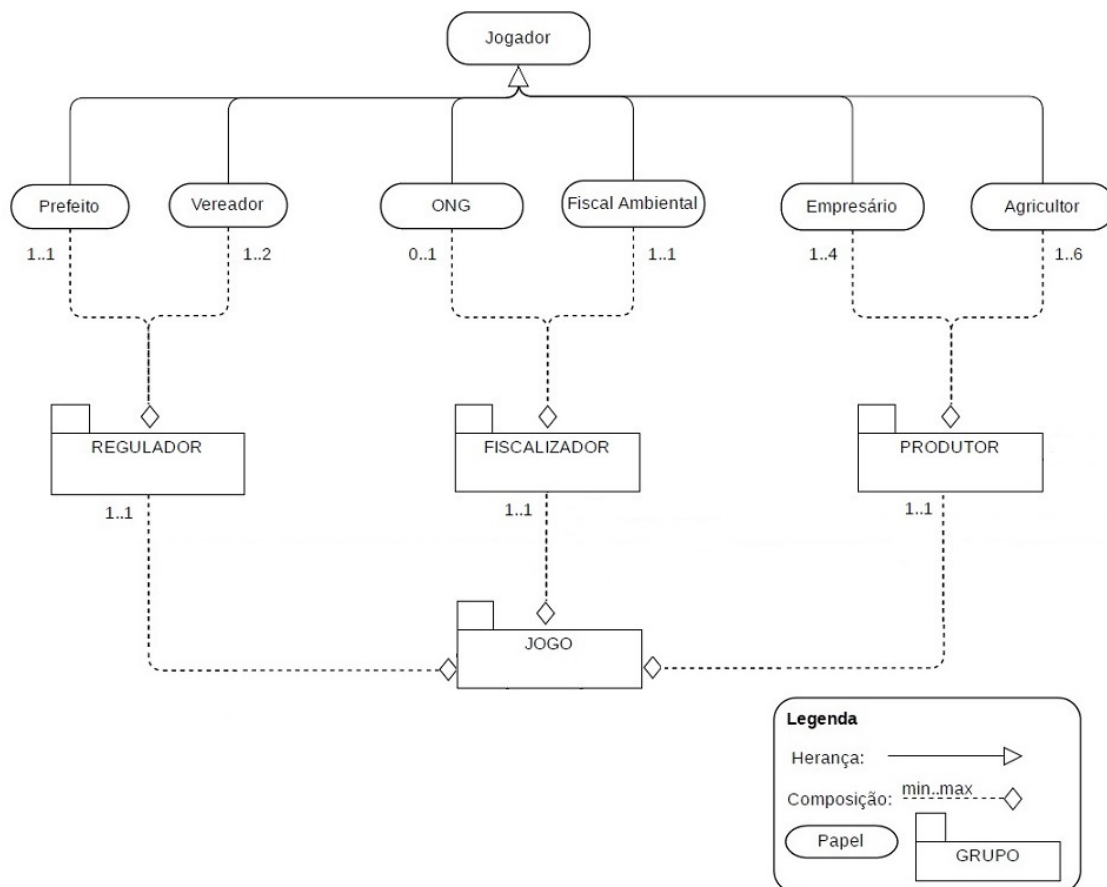


Figura 2. Especificação Estrutural da Organização para o modelo proposto.

Em relação às cardinalidades, no grupo Regulador, o agente pode assumir o papel de prefeito ou vereador, permitindo-se apenas um prefeito e dois vereadores. No grupo Fiscalizador, o agente pode assumir o papel de ONG ou de fiscal ambiental, considerando-se que um agente ou nenhum pode assumir o papel da ONG, visto que este papel é um NPC (do inglês, *Non-Player Character*) na modelagem do estudo de caso e é obrigatório pelo menos um fiscal ambiental. No grupo Produtor, o agente pode assumir o papel de empresário ou agricultor, sendo no mínimo um agente para cada papel e, no caso do empresário, no máximo quatro, e do agricultor, no máximo seis. Por fim, o grupo Jogo deve conter pelo menos um grupo de cada um dos demais.

Este trabalho encontra-se em fase de definição da parte funcional da organização, por isso, nesta concepção foi considerada a parte da especificação estrutural da organização. Para a elaboração do entendimento desta especificação funcional, de quais são as

ações de cada um dos papéis e como as mesmas serão realizadas, considerou-se o grupo Regulador e seus respectivos papéis, Prefeito e Vereador. Os três operadores representados na Figura 3 fazem parte da especificação para apresentar como as ações podem ser realizadas se em sequência, ou seleção ou ainda em paralelo.

A Figura 3 representa parte da especificação funcional da organização considerando, num primeiro momento, quais são as ações de cada papel. O papel de Prefeito possui três missões: (M1) tomar medida, seja de consultar o saldo disponível na prefeitura ou medidas de prevenção (como: tratamento de água, de lixo ou de esgoto); (M2) alterar imposto, podendo diminuir, manter ou aumentar a taxa; e, (M3) negociar, ou seja, fornecer ou receber propina, salientando que esta negociação pode ocorrer entre quaisquer papéis da organização. O papel de Vereador possui duas missões: (M4) propor ao papel de Prefeito a alteração dos impostos, podendo diminuir ou aumentar esta taxa; e, (M5) negociar, seguindo a mesma ideia do papel de Prefeito.

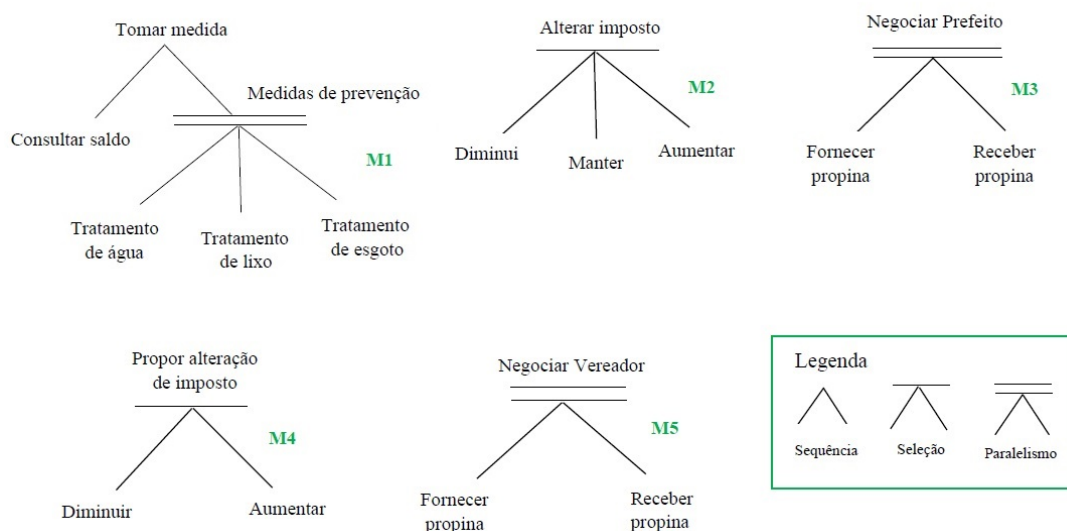


Figura 3. Parte da Especificação Funcional da Organização para o modelo proposto.

Observa-se na Tabela 1 o aspecto deôntico de cada papel especificado na coluna Tipo, os papéis de Prefeito e de Vereador. Os papéis se comprometem a realizar determinada missão, ou seja, têm obrigação de realizá-las ou o papel têm permissão para se comprometer ou não com determinada missão. As missões M1 e M2 do papel de Prefeito devem ser executadas e a missão M3 pode ser executada tanto para fornecimento ou recebimento de propina. As missões M4 e M5, correspondem ao papel de Vereador, onde este papel pode se comprometer ou não em executá-las.

É importante salientar que este artigo apresenta as primeiras percepções de como

Tabela 1. Parte da Especificação Normativa da Organização.

id	Papel	Tipo	Missão
n1	Prefeito	Obrigação	M1
n2	Prefeito	Obrigação	M2
n3	Prefeito	Permissão	M3
n4	Vereador	Permissão	M4
n5	Vereador	Permissão	M5

ocorrerá a modelagem da organização concebida em [Born et al. 2019a]. Por tratar-se de um sistema complexo, devido ao número de agentes, as ações e a sincronização destas no referido ambiente, suas especificações ainda não foram completamente definidas. Então, busca-se neste momento, o entendimento de como todo o sistema poderá ser modelado para posterior implementação.

5. Considerações e Trabalhos Futuros

Este trabalho apresentou a modelagem inicial de um sistema multiagente de uma organização no contexto de estudo dos recursos hídricos. O estudo de caso, no qual baseou-se este artigo, apresenta a definição da modelagem do sistema para a região da bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo, através do diagrama de interação (Figura 1).

O sistema é composto por três grupos de agentes e as sessões de RPG, desenvolvidas até o presente o momento, auxiliam no entendimento do problema relacionado ao recurso hídrico da bacia e como os agentes interagem neste ambiente.

Ao longo dos anos, o gerenciamento de recursos naturais, neste caso especificamente o recurso hídrico, tem se mostrado com grande importância. Desta forma, a aplicação de SMA e de um *framework* que permita a especificação da organização do sistema e sua implementação de forma integrada, pode fornecer resultados promissores. A modelagem da organização deste estudo é complexa, devido aos inúmeros aspectos envolvidos, tais como: as especificações de todos os papéis que os agentes podem assumir devem ser bem definidas, bem como suas metas e relacionamentos, trocas de mensagens e ambiente.

Desta forma, a modelagem completa desta organização auxiliará na implementação do SMA e o *framework* JaCaMo apresenta subsídios necessários para especificação, modelagem, implementação e simulação de cenários, permitindo um maior entendimento do problema em questão e de análise de estratégias tomadas pelos agentes do sistema quando assumem determinado papel.

Como trabalhos futuros, pretende-se estender esta modelagem inicial da organização no Moise+, bem como sua implementação e integração no *framework* JaCaMo.

Agradecimentos

Os autores deste artigo agradecem pelo auxílio financeiro no desenvolvimento desta pesquisa ao Programa de Apoio ao Ensino e à Pesquisa Científica e Tecnológica em Regulação e Gestão de Recursos Hídricos – Pró-Recursos Hídricos Chamada N° 16/2017.

Referências

- Adamatti, D. F. (2007). *Inserção de jogadores virtuais em jogos de papéis para uso em sistemas de apoio à decisão em grupo: um experimento no domínio da gestão de recursos naturais*. PhD thesis, Escola Politécnica – Universidade de São Paulo, São Paulo, Brasil. doi:10.11606/T.3.2007.tde-07012008-154915.
- Alvares, L. O. and Sichman, J. S. (1997). Introdução aos sistemas multiagentes. In *XVII Congresso da SBC-Anais JAI'97*.
- Artero, A. O. (2009). *Inteligência Artificial: Teoria e Prática*. Editora Livraria da Física, São Paulo/SP, 1a edition.
- Bellifemine, F. L., Caire, G., and Greenwood, D. (2007). *Developing multi-agent systems with JADE*, volume 7. John Wiley & Sons.
- Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2011). Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*.
- Boissier, O., Hübner, J. F., and Ricci, A. (2016). The jacamo framework. In *Social coordination frameworks for social technical systems*, pages 125–151. Springer.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons.
- Bordini, R. H., Vieira, R., and Moreira, A. F. (2001). Fundamentos de sistemas multiagentes. In *Anais do XXI Congresso da Sociedade Brasileira de Computação (SBC2001)*, volume 2, pages 3–41.
- Born, M., Leitzke, B., Farias, G., Melo, M., Gonçalves, M., Rodrigues, P., Martins, V., Barbosa, R., Aguiar, M., and Adamatti, D. (2019a). Sistema multiagente para gestão de recursos hídricos: Modelagem da bacia do são gonçalo e da lagoa mirim. In *Anais do X Workshop de Computação Aplicada a Gestão do Meio Ambiente e Recursos Naturais*, pages 87–96, Belém/PA. SBC, sol.sbc.org.br.
- Born, M., Leitzke, B. S., Farias, G., Aguiar, M., and Adamatti, D. F. (2019b). Modelagem baseada em agentes para análise de recursos hídricos. In *Anais do XIII Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações (WESAAC 2019)*, pages 107–118, Florianópolis/SC. wesaac.c3.furg.br.
- Brito, A. D., Lopes, J. C., and dos Anjos Neta, M. M. S. (2020). Tripé da governança: Poder público, setor privado e a sociedade civil em busca de uma gestão integrada dos recursos hídricos. *Revista Gestão & Sustentabilidade Ambiental*, 8(4):506–522.
- Coppin, B. (2010). *Inteligência Artificial*. Rio de Janeiro: LTC, 3a edition.
- Filatova, T., Verburg, P. H., Parker, D. C., and Stannard, C. A. (2013). Spatial agent-based models for socio-ecological systems: Challenges and prospects. *Environmental modelling & software*, 45:1–7.
- Gaudou, B., Sibertin-Blanc, C., Therond, O., Amblard, F., Auda, Y., Arcangeli, J.-P., Balestrat, M., Charron-Moirez, M.-H., Gondet, E., Hong, Y., Lardy, R., Louail, T., Mayor, E., Panzoli, D., Sauvage, S., Sánchez-Pérez, J.-M., Taillandier, P., Van Bai, N., Vavasseur, M., and Mazzega, P. (2014). The MAELIA multi-agent platform for

- integrated analysis of interactions between agricultural land-use and low-water management strategies. In Alam, S. J. and Parunak, H. V. D., editors, *Multi-Agent-Based Simulation XIV*, pages 85–100, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Hübner, J. F. (2003). *Um modelo de reorganização de sistemas multiagentes*. PhD thesis, Universidade de São Paulo.
- Hübner, J. F., Boissier, O., Kitio, R., and Ricci, A. (2010a). Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous agents and multi-agent systems*, 20(3):369–400.
- Hübner, J. F., Bordini, R. H., and Vieira, R. (2004). Introdução ao desenvolvimento de sistemas multiagentes com jason. *XII Escola de Informática da SBC*, 2:51–89.
- Hübner, J. F., Sichman, J. S., and Boissier, O. (2002). Moise+ towards a structural, functional, and deontic model for mas organization. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 501–502.
- Hübner, J. F., Sichman, J. S., and Boissier, O. (2007). Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3/4):370–395.
- Hübner, J. F., Sichman, J. S., and Boissier, O. (2010b). Moise+ tutorial.
- Leitzke, B., Born, M., Farias, G., Melo, M., Gonçalves, M., Rodrigues, P., Martins, V., Barbosa, R., Aguiar, M., Adamatti, D. F., et al. (2019). Sistemas multiagente e jogos de papéis para gestão de recursos naturais. In *Anais do XIII Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações (WESAAC 2019)*, pages 218–223, Florianópolis/SC. wesaac.c3.furg.br.
- Luger, G. F. (2013). *Artificial intelligence : structures and strategies for complex problem solving*. Addison-Wesley, Boston, Massachusetts, EUA, 6a edition.
- Nwana, H. S. (1996). Software agents: An overview. *The knowledge engineering review*, 11(3):205–244.
- Rezende, S. O. (2005). *Sistemas inteligentes: fundamentos e aplicações*. Editora Manole Ltda, Barueri/SP, 1a edition.
- Ricci, A., Piunti, M., and Viroli, M. (2011). Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*, 23(2):158–192.
- Russell, S. and Norvig, P. (2013). *Inteligência Artificial*. Elsevier Ltda, Rio de Janeiro/RJ, 3a edition.
- Taillandier, P. and Buard, E. (2009). Designing agent behaviour in agent-based simulation through participatory method. In Yang, J.-J., Yokoo, M., Ito, T., Jin, Z., and Scerri, P., editors, *Principles of Practice in Multi-Agent Systems*, pages 571–578, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Thérond, O., Sibertin-Blanc, C., Lardy, R., Gaudou, B., Balestrat, M., Hong, Y., Louail, T., Nguyen, V. B., Panzoli, D., Sanchez-Perez, J.-M., Sauvage, S., Taillandier, P., Vavasseur, M., and Mazzega, P. (2014). Integrated modelling of social-ecological systems: The MAELIA high-resolution multi-agent platform to deal with water scarcity

problems. In *7th International Environmental Modelling and Software Society (iEMSs 2014)*, page pp. 1, San Diego, California, United States.

Thomasi, C. D. (2014). Orias: uma infraestrutura de nível micro-organizacional baseada em artefatos para sistemas multiagentes. Master's thesis, Programa de Pós-Graduação em Computação, UFRGS, Porto Alegre/RS.

Wooldridge, M. (2002). An introduction to multi agent systems, department of computer science, university of liverpool, uk.

Agent Programming for Industrial Applications: Some Advantages and Drawbacks^{*†}

Otávio A. Matoso¹, Luis P. A. Lampert¹, Jomi F. Hübner¹,
Mateus Conceição¹, Sérgio P. Bernardes¹, Cleber J. Amaral^{1,2},
Maicon R. Zatelli¹, Marcelo L. de Lima³

¹Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

²Instituto Federal de Santa Catarina (IFSC)
São José – SC – Brazil


³Petrobras-Cenpes
Ilha do Fundão, Rio de Janeiro – RJ – Brazil

otaviomatoso@yahoo.com.br, {lp.lampert,sergiopb1998}@gmail.com,
{maicon.zatelli,jomi.hubner}@ufsc.br, mateusconceicao1@hotmail.com,
cleber.amaral@ifsc.edu.br, marceloll@petrobras.com.br

Abstract. *Autonomous agents are seen as a prominent technology to be applied in industrial scenarios. Classical automation solutions are struggling with challenges related to high dynamism, prompt actuation, heterogeneous entities, including humans, and decentralised decision-making. Besides promoting concepts, languages, and tools to face such challenges, agents must also provide high reliability. To assess how appropriate and mature are agents for industrial applications, we have investigated its application in two scenarios of the gas and oil industry. This paper presents the development of systems and the initial results highlighting the advantages and drawbacks of the agents approach when compared with the existing automation solutions.*

1. Introduction

Agents programming offers useful tools to be applied in complex scenarios, with high dynamism and decentralised decision-making, which are common challenges in industrial applications. The agents may take advantage of virtualised information, bringing reactivity, autonomy and often intelligence in decision-making processes acting over the real devices through their digital representations. In fact, many studies are suggesting the use of Multi-Agent Systems (MAS) in industrial applications [Dias et al. 2017, Kotak et al. 2003, Barbosa et al. 2018, Mualla et al. 2018, Roloff et al. 2017, Wada et al. 1998, Zhou et al. 2007].

 Agent Programming for Industrial Applications: Some Advantages and Drawbacks by Otávio A. Matoso, Luis P. A. Lampert, Jomi F. Hübner, Mateus Conceição, Sérgio P. Bernardes, Cleber J. Amaral, Maicon R. Zatelli and Marcelo L. de Lima is licensed under a Creative Commons Attribution 4.0 International License. <https://creativecommons.org/licenses/by/4.0/>

^{*}Supported by Petrobras project AG-BR, IFSC and UFSC.

Although the use of agents in industrial scenarios is reasonable, it is not clear whether they can bring more benefits in practical scenarios, especially when compared to classical solutions. This paper presents the development of MAS based solutions for two cases in the gas and oil industry to assess if agents can be applied and which advantages and drawbacks they can bring. The first case regards to a scenario in which a controller must actuate to protect a heat exchanger. The second case regards to adjustments in parameters of the processes that control artificial oil lifting, which are done through the integration of several systems. Both cases are real scenarios and refer to critical processes which are currently automated by classical control solutions. We have developed prototypes using the JaCaMo¹ framework for both investigations. In the performed tests the agents use real data usually interacting with real services of the plant.

This paper is organised as follows: Section 2 presents an overview of how MAS are currently being applied in industrial scenarios, Section 3 presents the heat exchangers scenario, the current solution and our agents solution as well as the results obtained from tests. Section 4 presents the artificial lifting scenario, a list of requirements for an enhanced solution, our MAS solution and results. Finally, Section 5 summarises our conclusions about the advantages and drawbacks that agents' application may bring for the industry.

2. MAS in the industry

It is often said that the upcoming fourth industrial revolution would be the result of the application of technologies such as Internet of Things (IoT), Internet of Services (IoS), Big Data, Machine Learning, among others [Rojko 2017, Thoben et al. 2017, Cimini et al. 2017, Lu 2017]. Although these technologies may enhance production, control and relationship with customers in many ways, integrating and managing them all together is a challenge [Leitão et al. 2013, Monostori et al. 2016].

A factory has many sensors, actuators, controllers, software and other entities that can be virtualised in a standard way, which can allow them to interact with each other. Many approaches are based on the concepts of Cyber-Physical Systems (CPS). This is a standard abstraction of entities that aims to manage and integrate information and systems, allowing a broad use of data and high integration among all sort of devices.

The agent technology has been successfully applied in a variety of areas in the industry context, namely production planning, scheduling, and logistics. The agents in such scenarios are bringing reactivity, autonomy and intelligence. They take advantage of virtualised information for their decision-making processes, acting over real devices through their digital representations. Multi-Agent Systems (MAS) can be the core of the CPS virtualised entities, providing interoperability and control [Cruz and Vogel-Heuser 2017, Leitão et al. 2013, Monostori et al. 2006, Pěchouček and Mařík 2008]. It brings several advantages in terms of reconfigurability, scalability, and productivity, which are translated

¹JaCaMo is an open-source Agent-Oriented Programming platform available in <http://jacamo.sourceforge.net>. More details are presented in [Boissier et al. 2013, Boissier et al. 2016, Boissier et al. 2019].

into measurable competitive advantages such as less production time and better use of resources [Leitão et al. 2013]. When applied to realise CPS, agent-based approaches leverage their negotiation and communication mechanisms to present benefits regarding flexibility, adaptability, and pro-activity [Amaral et al. 2019b, Monostori et al. 2016, Cruz and Vogel-Heuser 2017].

On the other hand, many other issues are still open challenges. There is a lack of suitable, mature simulation and engineering tools, security, safety, real-time capabilities, and standardisation [Pěchouček and Mařík 2008, Leitão et al. 2013]. In addition to the technological barrier, there is also a human factor-related barrier, such as the radical control paradigm shift from a centralised to a modularised control view, with distributed decision-making entities, and the absence of interaction mechanisms to support humans, especially using mobile devices [Leitão et al. 2013, Cruz and Vogel-Heuser 2017]. As a consequence, industrial applications of agents are very limited and their possible benefits in practical means are not well experimented and proven.

3. Applying agents to protect heat exchangers

The first application faced in this work refers to a protection control for heat exchangers used in Floating Production Storage and Offloading (FPSO) processes for the Oil and Gas industry. Heat exchangers are responsible for transferring thermal energy between a fluid and a material which can also be a fluid [Shah and Sekulić 2003]. In FPSO processes, they are used to cool the extracted fluid. Without proper control of the refrigeration, the heat exchange can be exposed to extremes of temperature reducing its lifetime.

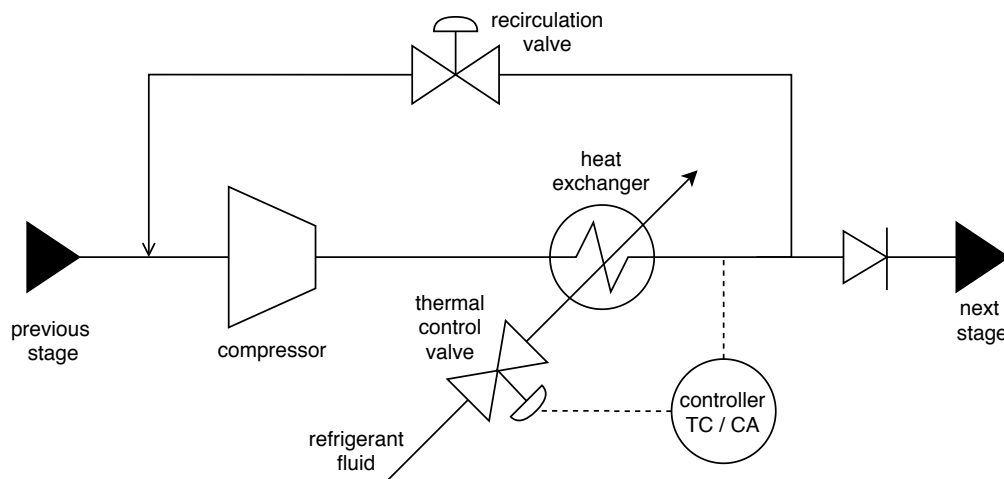


Figure 1. Typical compression stage in FPSO.

In the study case (see Figure 1), the control is performed by two different devices: (i) a Temperature Controller (TC) which is responsible for the processes most of the time controlling the flow of the refrigerant fluid, and (ii) a control application (CA) triggered by special conditions such as compressor starting and stopping processes, unstable and abnormal thermal conditions. The referred situations are usually transients, and the TCs are less effective for them. In this sense, the CA

protects the heat exchange until the system reaches a permanent regime and then TC assumes the control. Agents are being used to implement CAs.

The CA monitors the temperature and assumes the control of the thermal valve, progressively reducing its flows until a previously established set point. The way the valve is controlled varies according to the respective transient nature. The stabilisation process follows a fuzzy logic controller. Orders for changing the valve flow are sent to a system that validates the compensation, logs each order and effectively apply them on the Programmable Logic Controller (PLC) that acts over the physical valve.

The current CA is developed using classical control devices such as PLCs and supervisory control and data acquisition (SCADA) systems. The logic is implemented using Sequential Flow Charts (SFC)² which acts over a SCADA or directly over a PLC. The code is modularised into diagrams that are triggered according to preconditions which drive the system to execute the appropriated sequence function.

Figure 2 shows a part of a logic control that protects a heat exchanger when the compression process is going to stop. In summary, the application must take control of the valve when the compressor stops. However, there is a switch that can deactivate the application and a variable that indicates if the system is under operation. Finally, a sensor that indicates an abnormal temperature condition.

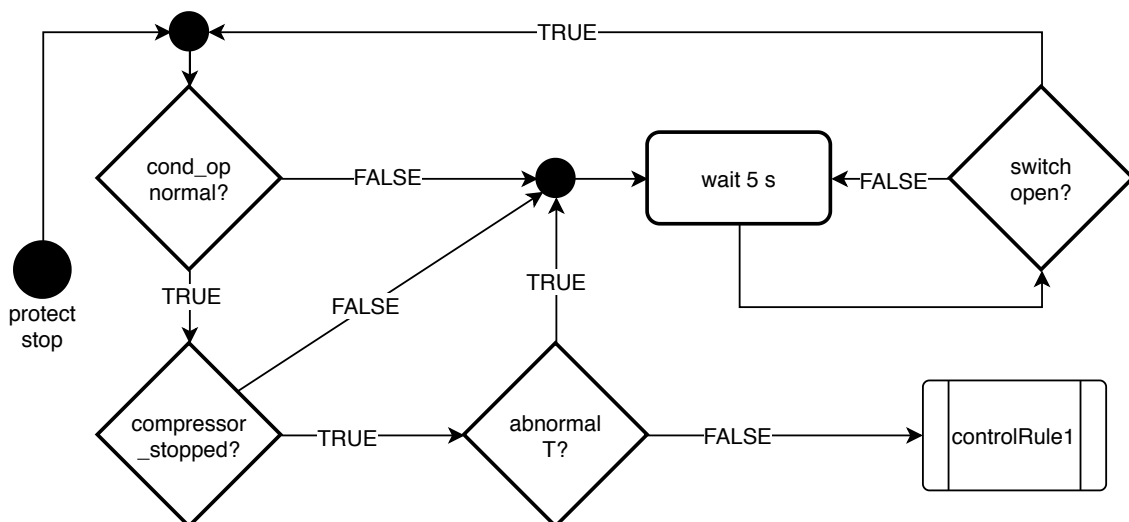


Figure 2. Current implementation of part of stopping compression process.

The classical logic verifies the state of the variables every 5 seconds. The effective control is performed when it is detected that the compressor has stopped, the controller is on, the system is under an operation condition, and the temperature is normal. In this example, we can verify the use of a polling method to check the condition of certain variables. It is also remarkable the presence of the verification `abnormal T?` which regards to a higher priority condition referring to abnormal temperatures. Along with many diagrams of different processes, we can find the same verification since this condition should trigger an overriding process.

²Sequential Flow Charts are standardised in the IEC 61131.

```

+compressor_stopped
  : switch(open) & cond_op(normal)
  <- !!controlRule1.

```

Figure 3. Agents implementation of part of stopping compression process.

In our study, there is no requirement of any enhancement regarding the current control features. The investigation aims to check whether agents can be applied and uncover possible advantages and drawbacks. The cooperation among agents, possibly in a chain of heater exchangers, is out of our scope.

In the solution using agents, we have developed an agent in which most of the CA actuation is programmed. In our case, the agent is coded in Jason, the agent programming framework embedded in JaCaMo. Most of the original implementation have straight-forward counterparts in Jason. However, in some cases, we could take advantage of the Agent-Oriented Programming (AOP) language.

Agents are reactive, i.e., they are sensible to events that can trigger actions when perceiving some change in the environment. This feature is being exploited in order to avoid polling loops. The code in the Figure 3 is an example of how such control logic can be translated to an AOP language. The code defines a plan in which the event `compressor_stopped` may trigger the execution of `controlRule1`. To effectively perform the plan, two conditions must be true: `switch(open)` and `cond_op(normal)`. In this sense, the approaches presented in Figure 2 and Figure 3 produce similar effects. Besides the agent implementation being arguably shorter and simpler, it is often faster since agents reacts promptly to events, while the SCF may delay up to 5 seconds.

In another aspect, the SFC also has a flag to check if the temperature is abnormal. This is a high priority condition that is present in many other parts of the control logic that is being omitted here. Besides redundancy, the need of such check on many parts of the system increases the risk of design mistakes. In agents solution, there is no need to check on each plan this condition since the agent version is taking advantage of agents' goal-oriented behaviour. The agent is coded with the primary goal of protecting the heat exchanger. In this sense, an abnormal temperature condition is triggering another plan that drops any other intention that the agent may have. Doing so, above all, the agent prioritises to perform the reaction to the referred abnormal condition. In this case, an agent-based solution is safer.

Control logic also makes frequent use of timers. Some of them are used to avoid too much stress on checking a condition as presented in Figure 2. Other timers are placed to give some timeout for an operation. Other ones are used for supporting some process that needs to filter outlier conditions, and so on. In the case of agents, the timers were also sensibly simplified since Jason supports scheduled operations such as `.at(t,e)` which executes `e` after `t` period.

With the CA implemented in Jason, we observed a reduction in the code size. An evaluation comparing the CA written on the basis of a literal translation of the classic control logic and the CA written exploiting Jason features has shown that the

latter is about 40% smaller³. Besides facilities like the scheduled operations, some code has been saved because of several verifications become useless when applying the concept of goals priority.

Agents also facilitate to enhance system robustness. We have added on the agent some plans for possible, even unlikely, situations. Indeed, agents can have several plans in order to fulfil a single goal, i.e., in a range of possible plans, the agent chooses to perform the first applicable. It means that for many reasons, plans cannot be executable, or they can fail. Still, in a way, one of them can be successful, resulting in the achievement of the goal.

Although AOP helped in the referred situations, we could not find advantages on the agent's version when acting over control variables and when calculating parameters. The SFC solution is very convenient on procedural tasks, including real-time ones. Industry standard solutions also have lots of tools helping to monitor variables of the industrial plant. The programming interface of the classical solution is more friendly as well.

4. Applying MAS to control artificial lifting

Some oil wells provide enough pressure for the natural lifting of the oil in production flow needs. For wells without enough pressure, especially in deep waters, it is necessary to apply artificial lifting techniques which are usually done by centrifugal pumps or by gas injection. These different artificial approaches are roughly similarly parameterised according to the reservoir pressure condition.

The procedure for changing the actuation of the artificial lifting process follows five main stages: (i) results from new tests on the reservoir are available, (ii) a new mathematical model of the reservoir is drawn, (iii) new control parameters for controlling the flow are defined and optimised, (iv) the control system is set up with new parameters, and (v) an external governmental agency is informed. Due to several uncertainties, the final decision-making process depends on human specialists validations.

Figure 4 illustrates the current process in which the engineer participates on the whole cycle. S/He is responsible for obtaining data, triggering the model design, sending parameters for optimisation and for forwarding setting up commands to the operator. The operator validates the setup with colleagues and then set it up in the control system. The engineer is also responsible for sending to an external agency reports about the production.

An enhanced solution has as main requirement the reduction of the engineer's workload. Although it is mandatory to have a validation of the optimisation from the perspective of an engineer, the coordination tasks can be delegated to an automated system.

Agents' proactivity and goal-oriented behaviour may help with tasks that are not easily automated due to the necessity of interactions and adjustments. Proactive agents chase their goals, in this case, performing the optimisation of the artificial

³For this evaluation, we have compared the code size after compressing the versions.

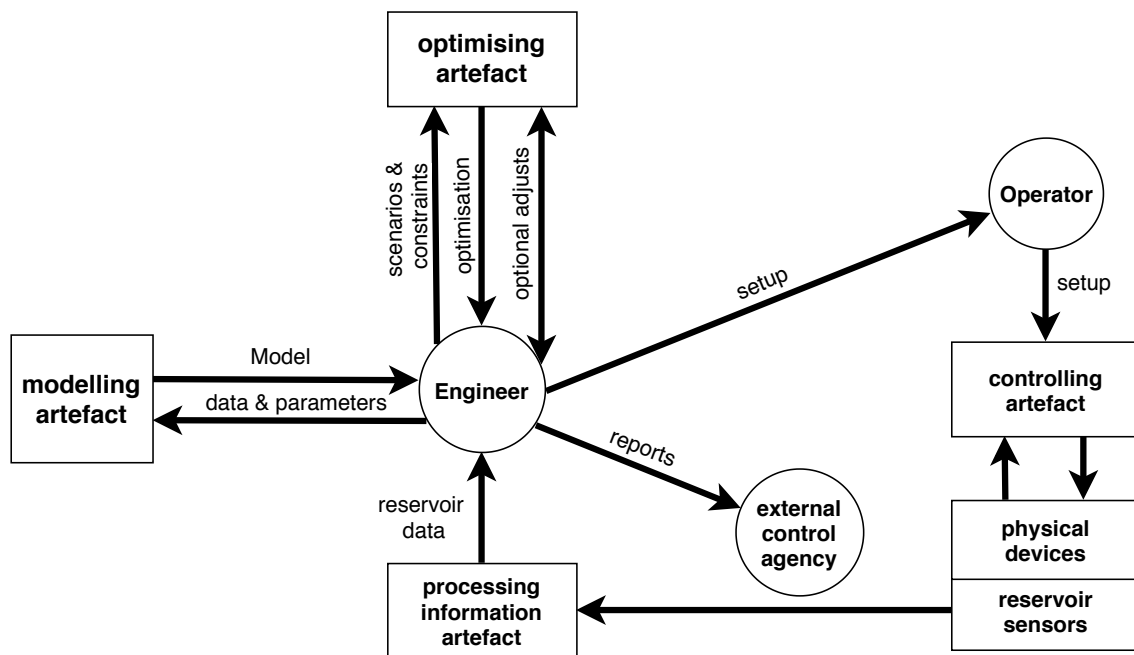


Figure 4. Classic solution for artificial lifting.

lifting process. Properly developed goal-oriented agents have multiple plans to deal with a sort of circumstances, providing necessary adjusts or starting the process from the scratch if needed.

Our MAS approach to this problem is presented in Figure 5. In this solution, the MAS is a new entity in the diagram. It is performing coordination tasks releasing the engineer from this workload. To do so, the MAS is equipped with a mediation tool that helps the agents to communicate with different entities. Some entities are software that perform procedural tasks, i.e., industrial artefacts. Other entities are agents, both human and artificial, that require different communication skills and argumentation capabilities. The system is composed of four agents, as represented in the zoomed area on the top left. These agents play roles in an organisation.

From the communication perspective, the integration between the MAS and external entities is done through Apache Camel mediation tool. The process information system, the modeller, the optimiser, and the control system are integrated through REST and/or CORBA interfaces. The integration among artificial agents is internally resolved by JaCaMo default infrastructure. The interaction among artificial agents and humans is performed through a chat messaging application which is also integrated with the MAS using Apache Camel. To enhance this interaction, the agent *chatbot* has natural language processing skills.

Although integration facilities is provided by an agnostic mediation tool, it is important to mention that the MAS solution has the advantage of providing entities model abstractions for both autonomous agents (including humans) and industrial artefacts. From the agent's perspective, it means that it interacts with these entities in a usual way, i.e., using its messaging system if the entity is modelled as an agent or through action and perception if it is modelled as an accessible tool

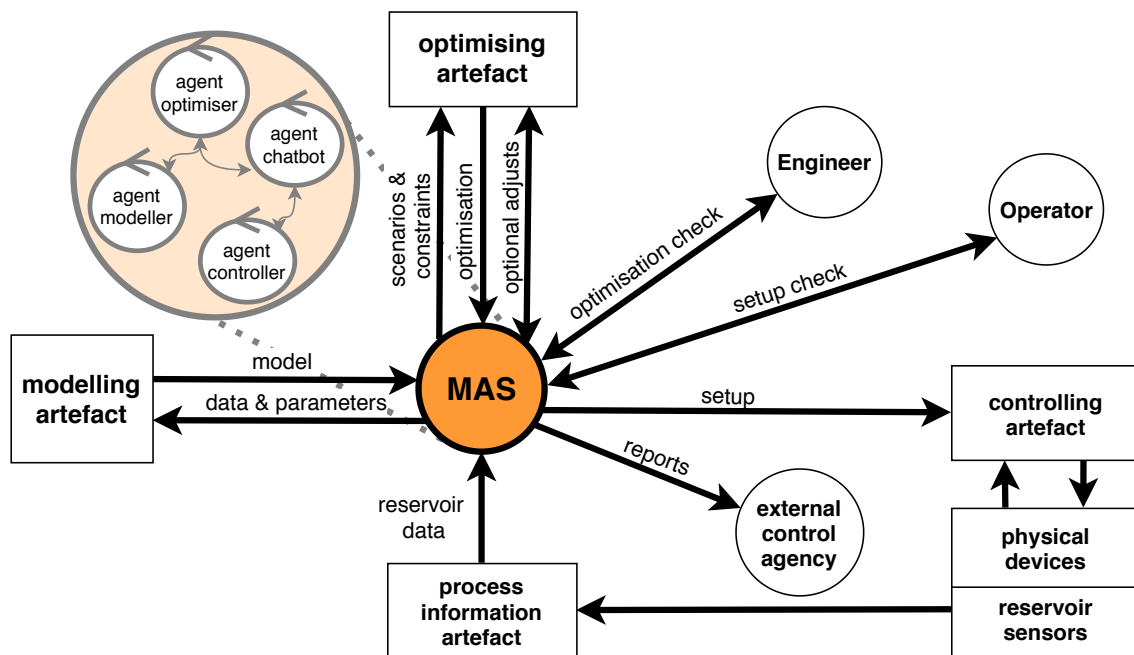


Figure 5. MAS solution for artificial lifting.

of its environment⁴.

In this application, as shown in the Figure 6, agents playing organisational roles are committed to sub-goals of the main goal of optimising the artificial lifting process. The agent *modeller* is in charge of reading reservoir data, sending the data to the modelling artefact for delivering the new model to the agent *optimiser*. This agent sends to the optimisation artefact scenarios obtained from the model and constraints. The artefact answers with an optimisation configuration that is sent to the engineer, through agent *chatbot*. The engineer may contest the parameter which may suffer adjustments. If accepted the agent *chatbot* checks with the operator the proposed setup to be applied. Whether the operator agrees with that, the agent *controller* apply the setup in the controlling artefact.

One advantage of AOP solution regards the clearer division of responsibilities. The division of sub-goal among agents work as a system's modularisation which simplifies significantly individual agent's programming processes. Additionally, the use of an organisational apparatus provides coherence among agents' actions. The coordination of agents by such structure can also simplify rearrangements since sub-goals interactions and dependencies are all defined in a higher abstraction. Indeed, the organisation programming dimension promotes development of macro level features which allows to adjust the behaviour of the system without changing the code of agents and artefacts. For instance, if it is requested to ask the engineer a final validation after the operator's answer, there is no need to change agents' code, but adding to the organisational scheme another sub-goal.

⁴More details of the advantages of such abstractions are available in [Amaral et al. 2019a] publication.

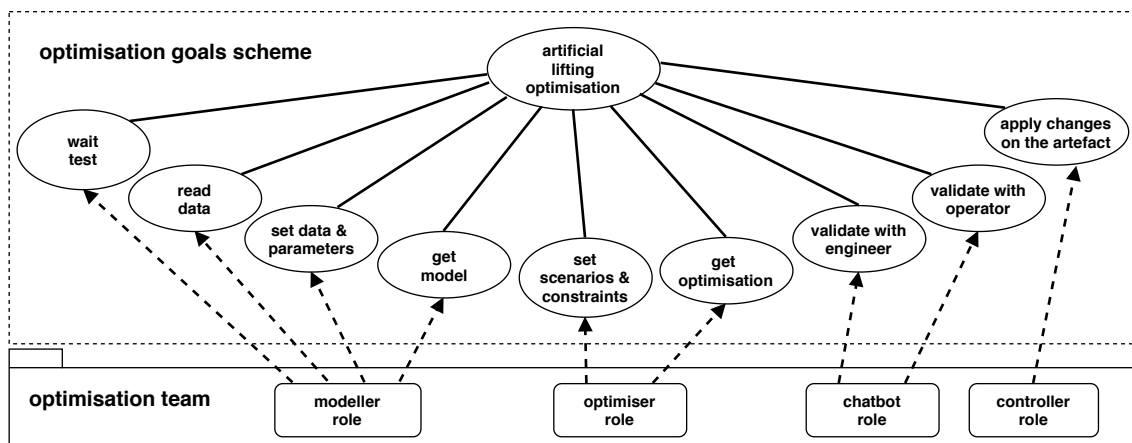


Figure 6. MAS organisation for artificial lifting.

5. Conclusion

In the industrial applications considered in this initial investigation, we can notice some enhancements when agents are applied. In the heat exchangers scenario, the reactive characteristic of agents has shown that the system can respond to environmental changes quickly (no need for polling). However, this is not an exclusive feature of AOP, it is present in any event-driven programming approach. Agent advantages became clearer when it is added some goal-oriented behaviour, which expands the system's capabilities to deal with environment dynamism. Agents can be programmed to be ready for all sorts of circumstances that may occur while they are achieving their goals. Some situations may also change the entire course of the agent's actions as we have illustrated in Section 3 when current intentions were dropped to achieve a higher priority goal. In this sense, AOP version has shown it can be faster and safer.

It is difficult to measure how easy it is coding in a programming language. However, at least taking the comparison in terms of code size, we realise that the code of an agent application can be smaller than an equivalent application coded in a classic control approach. Although, AOP program can be shorter and simpler, when considering legibility, the code reduction is competing against a visual solution that uses SFC diagrams. A visual programming interface can steep the learning curve and it is often preferable for designers. It is also to be mentioned that constructors used in those diagrams are common tools for most engineers, while predicate logic is not so usual.

The second case brought us other situation in which it was expected to enhance the system in order to reduce the workload over the engineer. This case regards to a complex information flow in a chain of processes that requires coordination and decision-making capabilities from a central actor. We have proposed a MAS in this position since, although complex, the requirements follow standards that can be delegated to agents. The first advantage we have noticed regards to the abstractions for autonomous and non-autonomous external entities AOP is providing. The acknowledgement of the nature of the external entity may boosts the integration through proper communication methods. It can also enhance decision-

making processes, for instance, with a better understanding of actions determinism. Another advantage regards to the use of a set of agents as part of an organisation which helps to divide the complexity and allows faster changes in the system, for instance, in the sequence of the actions. In the case of needing a double-check on the setup parameterisation, we have shown that with AOP it is only necessary to introduce this new stage in the goals scheme. No changes in the agents would be needed. It seems to be an appropriate facility since the introduction of a double-check is closer to a coordination functionality, than to an individual agent's business rule. In this sense, besides an arguably better abstraction, this coordination apparatus remarkably improves flexibility.

Besides some advantages, we experienced difficulties in debugging the MAS approach when integrated directly to control devices. Indeed, the MAS has no tools for such supervision and presents no advantages in executing procedural tasks. Although in a fair comparison between paradigms we should not consider the maturity level of their tools, AOP did not bring advantages in procedural tasks. In this case, in a pragmatic point of view, the lack of tools and the cost of change can be considered AOP drawbacks.

An important issue one may be concerned about the MAS adoption is related to its reliability. Classic control systems are highly reliable. In MAS, it is possible to develop agents to monitor other agents in order to replace them in case of fails, for instance. However, in this investigation, unfortunately, we could not test such a possibility and did not stress the MAS or use it broadly, for instance, in several heat exchangers or in long-term plant operation. In this sense, we still have no evidence that MAS is reliable and robust enough.

In this sense, we find the MAS solution a prominent approach for the core of the CPS. Acknowledging the advantages of classical solution and the need of more evidences on reliability of agents, we find a better solution by combining both approaches (agents and classical) together. We opt to give the agents the responsibility for events, timers, and for triggering smaller and simplified versions of SFCs. They are also responsible for coordination processes. The control procedures, including real-time ones, remain on classical devices.

As future work we intend to: (i) exploit advantages on cooperation among agents in a chain of heat exchangers, (ii) adapt the agent platform to be integrated with monitoring tools, (iii) develop fault tolerance facilities such as agents' health monitoring and actuators, (iv) develop a unit testing tool for agents, and (v) expand system tests for long-term running sandboxes and real plants for checking issues such as robustness and reliability.

References

- Amaral, C. J., Bernardes, S. P., Conceição, M., Hübner, J. F., Lampert, L. P. A., Matoso, O. A., and Zатели, M. R. (2019a). Finding new routes for integrating Multi-Agent Systems using Apache Camel. In *Anais do XIII Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações*, WESAAC 2019.
- Amaral, C. J., Cranefield, S., Hübner, J. F., and Roloff, M. L. (2019b). Giving Camel to Artifacts for Industry 4.0 Integration Challenges. *Lecture Notes in Computer*

- Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11523 LNAI:232–236.
- Barbosa, J., Leitao, P., Ferreira, A., Queiroz, J., Angione, G., and Lo Duca, G. (2018). Smart Inspection Tools Combining Multi-Agent Systems and Advanced Quality Control. *Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018*, pages 828–833.
- Boissier, O., Bordini, R. H., Hübner, J. F., and Ricci, A. (2019). Dimensions in programming multi-agent systems. *The Knowledge Engineering Review*, 34.
- Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013). Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*, 78(6):747–761.
- Boissier, O., Hübner, J. F., and Ricci, A. (2016). *The JaCaMo Framework*, pages 125–151. Springer International Publishing.
- Cimini, C., Pinto, R., and Cavalieri, S. (2017). The business transformation towards smart manufacturing: a literature overview about reference models and research agenda. *IFAC-PapersOnLine*, 50(1):14952–14957.
- Cruz, S. L. A. and Vogel-Heuser, B. (2017). Comparison of agent oriented software methodologies to apply in cyber physical production systems. In *Proceedings - 2017 IEEE 15th International Conference on Industrial Informatics, INDIN 2017*.
- Dias, J., Vallhagen, J., Barbosa, J., and Leitao, P. (2017). Agent-based reconfiguration in a micro-flow production cell. *Proceedings - 2017 IEEE 15th International Conference on Industrial Informatics, INDIN 2017*, pages 1123–1128.
- Kotak, D., Wu, S., Fleetwood, M., and Tamoto, H. (2003). Agent-based holonic design and operations environment for distributed manufacturing. *Computers in Industry*, 52(2):95–108.
- Leitão, P., Mařík, V., and Vrba, P. (2013). Past, present, and future of industrial agent applications. *IEEE Transactions on Industrial Informatics*, 9(4):2360–2372.
- Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6:1–10.
- Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., and Ueda, K. (2016). Cyber-physical systems in manufacturing. *CIRP Annals*, 65(2):621 – 641.
- Monostori, L., Váncza, J., and Kumara, S. R. (2006). Agent-based systems for manufacturing. *CIRP Annals - Manufacturing Technology*, 55(2):697–720.
- Mualla, Y., Vanet, R., Najjar, A., Boissier, O., and Galland, S. (2018). Agentoil: A multiagent-based simulation of the drilling process in oilfields. In Demazeau, Y., An, B., Bajo, J., and Fernández-Caballero, A., editors, *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, pages 339–343, Cham. Springer International Publishing.
- Pěchouček, M. and Mařík, V. (2008). Industrial deployment of multi-agent technologies: Review and selected case studies. *Autonomous Agents and Multi-Agent Systems*, 17(3):397–431.

- Rojko, A. (2017). Industry 4.0 Concept: Background and Overview. *International Journal of Interactive Mobile Technologies (iJIM)*, 11(5):77.
- Roloff, M. L., Amaral, C. J., Stivanello, M. E., and Stemmer, M. R. (2017). MAS4SSP: A Multi-Agent Reference Architecture for the configuration and monitoring of Small Series Production lines. *2016 12th IEEE International Conference on Industry Applications, INDUSCON 2016*.
- Shah, R. K. and Sekulić, D. P. (2003). *Fundamentals of Heat Exchanger Design*. John Wiley & Sons, Ltd.
- Thoben, K.-D., Wiesner, S., and Wuest, T. (2017). "Industrie 4.0" and Smart Manufacturing – A Review of Research Issues and Application Examples. *IJAT (International Journal of Automation Technology)*, IJAT 11:4–16.
- Wada, H., Sakuraba, Y., and Negishi, M. (1998). Machinery Control System using Autonomous Agents. *Yokogawa Electric Corporation*, pages 1–6.
- Zhou, B.-h., Li, C.-c., and Zhao, X. (2007). Fipa agent-based control system design for fms. *The International Journal of Advanced Manufacturing Technology*, 31(9):969–977.

Farmer: Sistema Multiagente para a Manutenção Autônoma em uma Agricultura Utilizando IoT *Middleware* *

Alice Trinta, Fabian C. B. Manoel, Carlos Eduardo Pantoja

¹Centro Federal de Educação Tecnológica (CEFET-RJ)
20785-220 – Rio de Janeiro – RJ – Brasil

maria.trinta@aluno.cefet-rj.br, fabiancpbm@gmail.com


pantoja@cefet-rj.br

Abstract. *the agricultural sector, the demand for increased productivity has resulted in problems such as scarcity and waste of natural resources. As a solution, technologies to control and monitor these resources have emerged to make their consumption sustainable. However, these tools still do not offer autonomy at the hardware level, which can delay decision-making about cultivation. The objective of this work is to present a system composed of devices with actuators, sensors, and an embedded SMA that performs autonomous control of resources used in a crop. Besides, this system allows you to connect the devices on an IoT network, and with that, the crop data is made available in real-time. Finally, a WEB page will request data from the IoT network and display it to the user.*

Resumo. *No setor agrícola, a demanda por aumento na produtividade resultou em problemas como escassez e desperdício de recursos naturais. Como solução, surgiram tecnologias de controle e monitoramento destes recursos com o objetivo de tornar seus consumos sustentáveis. Porém, estas ferramentas ainda não oferecem autonomia no nível de hardware, o que pode atrasar tomadas de decisão sobre o cultivo. O objetivo deste trabalho é apresentar um sistema composto de dispositivos com atuadores, sensores e um SMA embarcado que realiza o controle autônomo de um cultivo. Além disso, este sistema permite conectar os dispositivos em um rede IoT, e com isso, os dados do cultivo são disponibilizados em tempo real. Por fim, uma página WEB irá requisitar os dados do vindos da rede IoT e exibir ao usuário.*

1. Introdução

A modernização da agricultura tem um de seus marcos durante os movimentos de urbanização das décadas de 1950, 1960 e 1970, onde o apelo por uma maior demanda de produção, levou à criação de tecnologias destinadas ao aumento de produtividade dos campos. Entretanto, a demanda excessiva de produção trouxe problemas como escassez e desperdício de recursos naturais, e como solução começaram a surgir tecnologias de controle de recursos utilizados pelo meio agrícola [Suzuki 2007].

*  O trabalho Farmer: Sistema Multiagente para a Manutenção Autônoma em uma Agricultura Utilizando IoT Middleware de Maria Alice Trinta, Carlos Eduardo Pantoja, Fabian Brandão está licenciado com uma Licença Creative Commons - Atribuição 4.0 Internacional. <http://creativecommons.org/licenses/by-nc/4.0/>

Um Agente Inteligente pode ser descrito como uma entidade física ou virtual autônoma capaz de perceber o meio e agir sobre ele, buscando através de planos e ações, alcançar seus objetivos. Eles são considerados entidades inteligentes e autônomas pois são capazes de deliberar suas próprias ações e não necessitam de intervenção humana ou de outros sistemas [Chin et al. 2014]. Um Sistema Multiagente (SMA) é um conjunto de agentes inteligentes situados em um ambiente e organizados de modo a atingirem seus objetivos comuns ou competirem entre si [Wooldridge 2009]. Já a Internet das coisas (IoT) é uma rede que interliga objetos, permitindo a interação objeto-objeto e usuário-objeto. Ao estender a internet ao mundo físico é possível realizar o gerenciamento remoto de qualquer objeto, o rastreamento de objetos através da internet, e o compartilhamento de informações [Zhang et al. 2012].

A criação de um sistema hipotético que gerencie os recursos de uma agricultura utilizando a IoT e SMA, pode trazer consigo benefícios para um cultivo. Uma vez que a água e a luz são elementos básicos para a sobrevivência de uma planta, é possível designar a um SMA o gerenciamento autônomo considerando parâmetros essenciais como umidade do solo, incidência de luz e previsões climáticas, evitando assim o afogamento ou ressecamento do cultivo e evitando também a falta de luz solar sob as plantas. Com isso, o usuário, através da IoT, seria capaz também de acessar as informações sobre o cultivo de qualquer lugar. Inicialmente, este sistema pode auxiliar pequenos cultivos residenciais. No entanto, em larga-escala, este tipo de sistema poderia levar à automatização da agricultura em grandes campos, elevando a produção reduzindo a necessidade de interferência humana.

Existem trabalhos que desenvolveram soluções para o gerenciamento de recursos em agriculturas, como por exemplo o [Salazar et al. 2013] que apresenta um controle de irrigação autônomo a partir de um SMA e um Arduino. No entanto, o trabalho apresenta uma dependência de periféricos, uma vez que as informações só serão processadas com auxílio de um computador. Um outro trabalho [Jin et al. 2018] retrata o desenvolvimento de uma rede de comunicação baseada em um protocolo para compartilhamento de informações sobre um cultivo, mas deixa o gerenciamento dos recursos a serviço do usuário, limitando a autonomia. Por último, o [Shi 2014] retrata a construção de um sistema de manutenção que recolhe, transmite e entrega os dados de um cultivo ao usuário, mas não apresenta autonomia para o gerenciamento do cultivo e de seus recursos.

O objetivo deste trabalho é apresentar o desenvolvimento de um dispositivo controlado por um SMA embarcado para fazer o controle autônomo dos recursos utilizados em um cultivo conforme as condições do ambiente, e simultaneamente disponibilizar em tempo real o status do cultivo para o usuário. Para isso, será criado um dispositivo composto por uma *raspberry* e um Arduino. Já para a comunicação entre os controladores e o SMA será utilizada a arquitetura ARGO [Pantoja et al. 2016] que permite a comunicação entre o framework Jason [Bordini et al. 2007] e microcontroladores. O SMA embarcado irá, por meio de crenças capturadas por sensores observar e ponderar se agirá, por meio de atuadores, ou não sobre o ambiente para benefício do cultivo. Para a ligação entre os dispositivos e a camada de gerenciamento dos recursos será utilizado o *middleware* ContextNet, que permitirá aos dispositivos disponibilizarem todas as informações do ambiente em tempo real para o usuário através de uma página *WEB*.

Este trabalho está organizado da seguinte maneira: na segunda seção o referencial

teórico, na terceira seção os trabalhos relacionadas, na quarta seção a metodologia, na quinta seção as avaliações experimentais, na sexta seção as considerações finais, e por fim as referências utilizadas.

2. Referencial Teórico

Nesta seção será apresentado o referencial teórico que contém as definições e as tecnologias utilizadas nas seções posteriores para o detalhamento do desenvolvimento do trabalho, suas funções e seu funcionamento.

Um sistema embarcado é a união entre partes de software e hardware responsável por originar dispositivos independentes para atender a somente uma aplicação. Uma vantagem sobre o uso de sistemas embarcados é a possibilidade de criação de dispositivos que independem de qualquer outro tipo de componente externo, tendo todo o necessário para seu funcionamento disponível em um só aparato [Peckol 2019]. No âmbito deste artigo, há a criação de dispositivos que integram software, pois abrigam um SMA, e hardware, através do uso de um Arduino para captação de dados do ambiente. Além disso, todos os dados recebidos poderão ser enviados ao usuário, permitindo a interação entre SMA e usuário. Para isso, há a necessidade de um agente capaz de se comunicar com uma camada de gerenciamento dos recursos, e um agente capaz de se comunicar com o microcontrolador para recolher as informações dos sensores.

Um agente ARGO é uma arquitetura Jason personalizada que tem por finalidade possibilitar a programação de agentes robóticos que são capazes de acessar hardware. O agente é capaz de decidir quando parar ou iniciar o fluxo de percepções vinda dos sensores e definir um intervalo fixo entre cada percepção, para controlar o comportamento perceptivo de um agente. O ARGO utiliza o *middleware* Javino [Lazarin and Pantoja 2015], para possibilitar a troca de informações entre a linguagem Java (alto nível) e a do Arduino (baixo nível), trabalhando como uma via de mão dupla através de uma comunicação serial segura contra perda de informações.

A Internet das Coisas — *Internet of Things* (IoT) — é uma rede que interliga dispositivos físicos para compartilhamento de informações e controles remotos. Neste trabalho, a rede IoT será disponibilizada através do *middleware* ContextNet, que possibilita interligar dispositivos móveis oferecendo o compartilhamento de informações em larga escala [Endler et al. 2011].

3. Sistema de Gerenciamento e Dispositivos *Farmer*

O trabalho propõe uma arquitetura capaz de gerenciar uma plantação de forma autônoma através da criação de um dispositivo embarcado com um SMA capaz de realizar a manutenção dos recursos e informar em tempo real o status do cultivo ao usuário. Nesta seção será descrita toda a metodologia por trás do desenvolvimento do trabalho, que será dividida por camadas, especificando desde a criação do dispositivo e seu funcionamento, até sua aplicação em conjunto com outras tecnologias responsáveis pela virtualização dos dispositivos.

Na Figura 1, a solução proposta está organizada de maneira a por o dispositivo fixado no chão da horta, de onde serão recolhidos os dados sobre umidade do solo e luminosidade para que sejam processados pelo SMA embarcado no dispositivo e utilizados

como base para qualquer gerenciamento feito na horta, como por exemplo o fornecimento de água ou o alerta para se o cultivo recebe ou não luz o suficiente. Os dados recolhidos serão repassados ao usuário através da rede IoT para uma página *WEB*, que terá acesso ao status do cultivo de qualquer lugar.

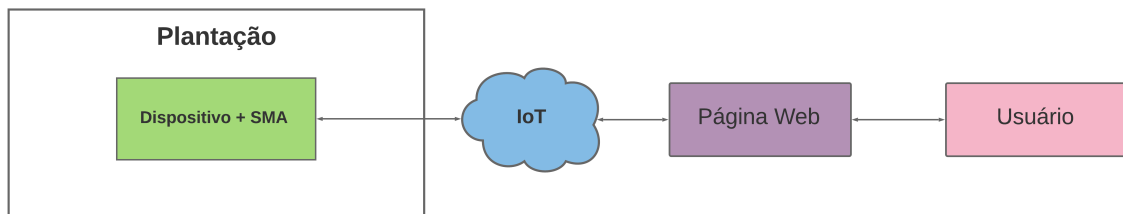


Figura 1. A arquitetura do projeto de gerenciamento autônomo de um cultivo.

Em uma horta pode-se encontrar diversos tipos de plantas, cada uma com necessidades específicas que devem ser levadas em conta, contudo uma planta precisa essencialmente de água e luz para sobreviver, mas para que o cultivo como um todo atinja um alto nível de produtividade é necessário que os recursos essenciais sejam devidamente gerenciados, considerando as especificidades de cada espécie. Logo, o responsável pela horta tem de permanecer atento aos níveis de umidade do solo e de incidência solar, pois caso não haja atenção ao gerenciar estes recursos, problemas como seca ou afogamento podem comprometer o desenvolvimento e levar o cultivo a morte.

No entanto, este responsável pode nem sempre estar disponível para cuidar do cultivo, ou devido a uma falta de experiência o indivíduo pode fracassar e desistir da atividade. Visando uma tecnologia capaz de gerenciar uma plantação sem necessidade de intervenção humana, os dispositivos *Farmer* auxiliarão, de forma autônoma, estes indivíduos a realizar tarefas agrícolas.

Um dispositivo contém sensores capazes de recolher informações sobre a umidade do solo e o índice de luminosidade. A partir destas informações e das necessidades específicas da plantação, os agentes do SMA embarcado serão capazes de gerenciar o acionamento de irrigadores e informar ao usuário caso o cultivo esteja recebendo luz o suficiente ou não. Cada dispositivo *Farmer* possui a função de obtenção de dados externos sobre o clima se conectado a internet para evitar o acionamento de irrigadores caso o local da plantação esteja propício a temporais. Se por acaso uma plantação esteja afogada, muito seca ou não esteja em contato com a luz os agentes do SMA embarcado no dispositivo irão detectar e informar ao usuário através das páginas *WEB*, onde o usuário pode ter acesso ao status do plantio.

Em um cenário onde existem dois *Farmers*, um cuidando de uma plantação de manjeriço e outro de uma plantação de babosas, o *Farmer* do manjeriço irá exigir mais água do que o *Farmer* da babosa, logo cada dispositivo irá funcionar de forma que as necessidades específicas de cada plantio sejam atendidas. Em caso de chuvas no ambiente, o acionamento da irrigação será controlado para benefício do plantio. Caso por exemplo, o solo da plantação de manjeriço esteja com umidade excessiva a qual o *Farmer* não consiga regular, o dispositivo informará o problema ao usuário para que seja resolvido.

3.1. O dispositivo *Farmer*

Para criar um dispositivo que controle os recursos básicos utilizados em um cultivo de forma autônoma, e fazer com que os dados recolhidos sobre o ambiente cheguem até o usuário é preciso atuadores e sensores capazes de coletar dados e atuar sobre o ambiente, um modelo cognitivo para avaliar e agir sobre esses dados e, por fim, um meio que possibilite a transmissão e visualização dos dados recolhidos.

O dispositivo é composto de um microcontrolador Arduino para permitir o controle dos sensores e dos atuadores, um mini-computador *Raspberry* para abrigar os agentes cognitivos e uma rede IoT criada com o *middleware* ContextNet para a comunicação com uma aplicação WEB.

É possível observar a arquitetura dos dispositivos na Figura 2, onde o Arduino controla os sensores de umidade do solo e de luminosidade que são utilizados para verificar se o plantio está recebendo todos os elementos necessários para sua sobrevivência. O Arduino controla também o irrigador, que nesse caso faz papel de atuador, sendo acionado somente quando necessário.

A decisão sobre o acionamento do irrigador vem após o recolhimento e processamento das informações feito pelo SMA embarcado na *raspberry*. Para fazer a interação entre o microcontrolador e o mini-computador, foram utilizados nos SMA agentes ARGO, capazes de fazer a captação dos dados recolhidos pelo Arduino, possibilitando o processamento das informações, o envio para a camada IoT e a devolução de ações de acionamento de atuadores ao Arduino.

Para que as informações sejam enviadas ao usuário, foi necessária a criação de uma rede IoT através do *middleware* ContextNet, originando assim, a camada de gerenciamento de recursos, uma arquitetura voltada para o gerenciamento de objetos IoT que permite armazenar informações recolhidas pelo dispositivo em um banco de dados onde os mesmos são registrados, cada um possuindo uma identificação única, possibilitando também o gerenciamento manual a distância permitindo o acionamento ou desligamento de irrigadores.

3.2. SMA, RML e aplicação WEB

Cada dispositivo *Farmer* contém um SMA composto por dois agentes capazes de fazer o controle de suas funções. O primeiro agente, chamado controlador, é uma arquitetura ARGO que recebe crenças baseadas no que se observa através dos sensores e o segundo agente, chamado mediador, é uma arquitetura jason padrão que recebe as crenças do controlador, retorna ordens e envia as informações através do *middleware* ContextNet para a camada de gerenciamento de recursos, onde o dispositivo é cadastrado e suas informações são salvas em um banco de dados.

A camada de gerenciamento de recursos (Resource Management Layer - RML)[Pantoja et al. 2019] é uma arquitetura que possibilita a organização dos objetos IoT de um ambiente. Este recurso estabelecido pelo *middleware* ContextNet é uma aplicação servidor que permite armazenar informações sobre objetos IoT em um banco de dados para que os mesmos sejam registrados e possam ser gerenciados. A utilização da RML permite que sejam criadas aplicações móveis que possibilitam a visualização dos dados destes objetos a partir de qualquer lugar, a qualquer momento.

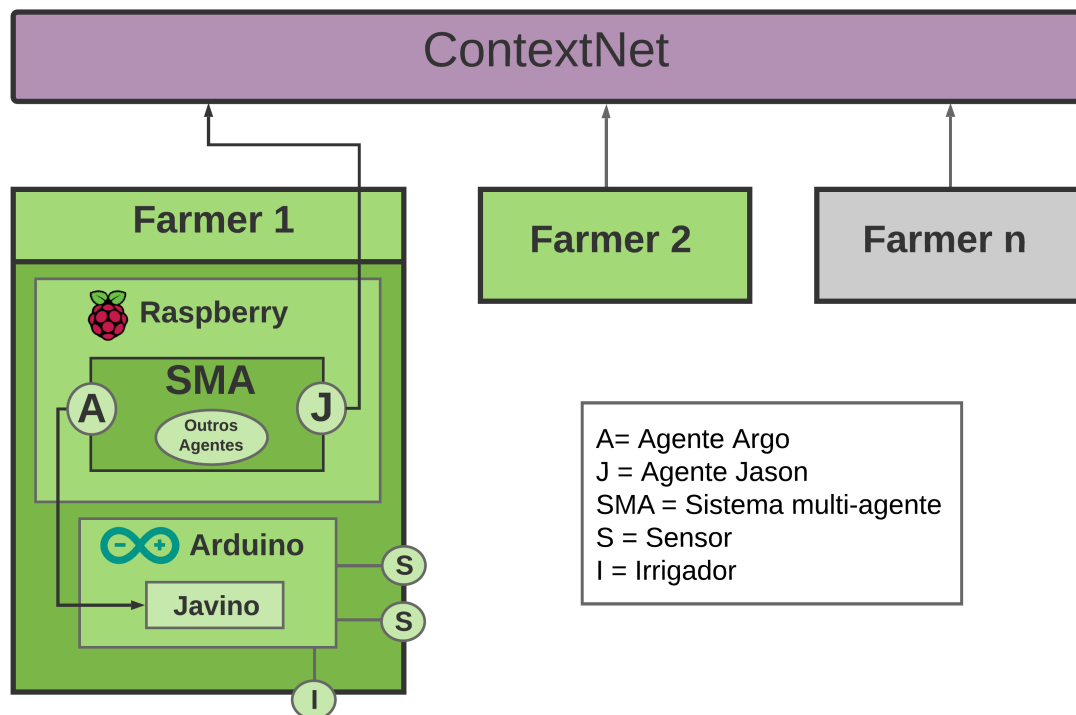


Figura 2. Comunicação entre o dispositivo e a camada ContextNet / gerenciamento de recursos.

O agente controlador por ser uma arquitetura ARGO, possui funções que em conjunto do *middleware* javino possibilitam a extração de dados sobre umidade e incidência luminosa do ambiente do cultivo, estes dados passam a ser crenças e estas crenças são compartilhadas com o agente mediador. As informações são extraídas a partir de um plano onde o comando *.port* sinaliza a porta referente ao arduino e o comando *.percepts(open/block)* permite a passagem das informações pela porta. O agente mediador ao receber estas crenças, devolve ao controlador ordens de acordo com suas crenças e executa duas ações internas, uma referente à uma busca por incidência de chuvas na região e outra responsável pela comunicação com a RML passando todas as informações sobre o cultivo. O agente mediador através de suas crenças e através de sua ação interna que verifica incidência de chuva, vai devolver ao controlador especificamente ordens sobre o acionamento ou não do sistema de irrigação.

Devida a utilização da RML foi possibilitado ao usuário visualizar as informações a cerca da horta a partir de qualquer lugar através de uma aplicação *WEB*. Assim que a RML recebe as informações sobre o dispositivo, suas características e funções são cadastrados em um banco de dados. Logo, foi criada uma página *WEB* (Figura 3) onde o usuário pode visualizar as informações coletadas pelos sensores dos dispositivos *Farmer*.

Pode-se então ver a arquitetura completa do trabalho na Figura 4, que vai desde os dispositivos *Farmer*, seus sensores e seu SMA, até a camada RML onde há um banco de dados (Virtualized Components DataBase - VCDB) para o cadastro dos dispositivos e dos dados recolhidos e por fim a conexão desse banco de dados com a camada *WEB*, onde foi criada uma aplicação em *Java Server Pages* (JSP) para que todas as informações

Tabela 1. Design do caso de estudo

Design	Descrição
Objetivo	Analisar o funcionamento do dispositivo <i>Farmer</i> , bem como o sistema em geral.
Caso	Foram comparados dois vasos de planta, um gerenciado pelo sistema e outro por um voluntário.
Questões	O sistema funciona como planejado? É uma ferramenta que facilita a prática da agricultura? É tão eficiente quanto um ser humano ou mais? É possível designar a função de cuidar de um cultivo a este sistema ao invés de um ser humano?
Método	Método de observação.

coletadas cheguem ao usuário.



Figura 3. Página WEB do Sistema de Gerenciamento.

4. Avaliação Experimental

Após a criação do dispositivo foram feitos testes referentes às funções do sistema no meio prático. Foi constatado através de experimentos que, o solo onde ficou instalado o dispositivo estava sempre hidratado adequadamente, enquanto o solo que não era regido pelo sistema ficava em constante seca. Na Tabela 1, pode-se ver como foi montado e em que se baseou o caso de estudo.

Os testes que duraram duas semanas consistiram em um cenário onde se encontravam dois vasos de plantas, um com o dispositivo e outro sem, colocados em condições onde um voluntário regava o vaso de planta sem o dispositivo sempre que possível. Em Figura 5, pode-se ver a esquerda da imagem o vaso gerenciado pelo sistema, com o dispositivo ao lado recolhendo informações e uma válvula solenoide que permite a passagem de água se necessário para irrigação, e a direita da imagem o vaso gerenciado pelo voluntário. Na Tabela 2, pode-se ver quais foram os resultados obtidos durante o período de observação.

Sistemas multiagentes para a manutenção em uma agricultura domiciliar utilizando IoT

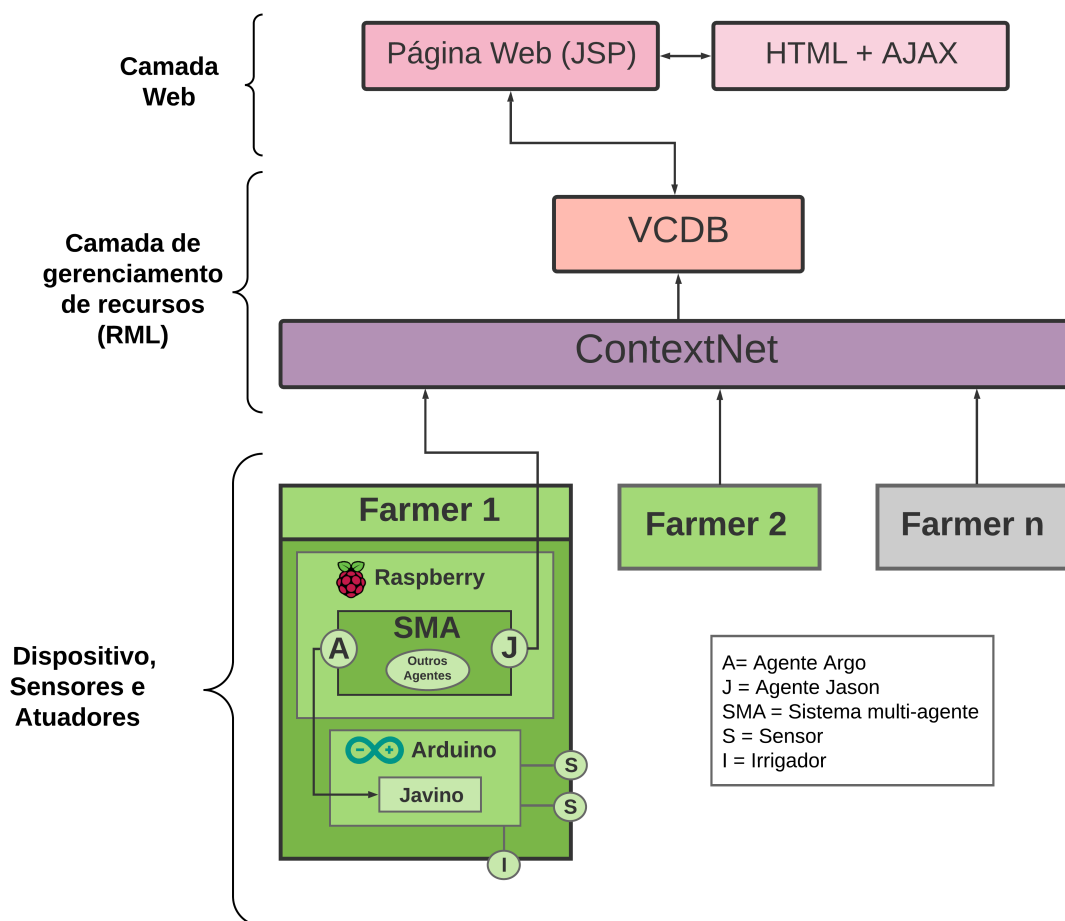


Figura 4. Arquitetura completa do trabalho.

Foram observados ao longo do período de testes alguns eventos que serviram de base para as constatações anteriores, o primeiro foi a falta de irrigação constante causada pelo esquecimento após uma longa jornada de trabalho, ou rotina exaustiva, fazendo com que a irrigação da planta pelo voluntário acontecesse nenhuma, uma ou duas vezes ao dia. Em um outro momento, o voluntário regou a planta e logo em seguida ocorreram chuvas, o que alagou o cultivo. Em comparação, o *Farmer* manteve uma irrigação regular de duas vezes ao dia, em dias de baixa incidência solar, e em dias muito quentes até três se necessário, e evitou também o afogamento do cultivo no dia em que ocorreram as chuvas.

Através das imagens, é possível concluir o melhor desenvolvimento do vaso gerenciado pelo dispositivo, em contrapartida o vaso gerenciado pelo voluntário não conseguiu se desenvolver pois a falta de irrigação constante gradativamente o levou a seca. Durante o período de testes, o sensor de luminosidade se manteve funcional informando constantemente se o cultivo estava recebendo luz ou não.

Tabela 2. Resultados obtidos

Teste	Descrição	Resultado
Comunicação entre o microcontrolador e o microcomputador.	Porcentagem de sucesso (%) quando há troca de dados entre o SMA e o Arduíno.	100%
Comunicação entre o SMA e a RML.	Porcentagem de sucesso (%) quando há troca de dados entre o SMA e a RML através da ContextNET.	100%
Verificar se a planta do FARMER dura mais tempo do que se fosse gerenciada por um humano.	Análise feita a partir da Figura 5, comparando a planta gerenciada pelo sistema e a planta gerenciada pelo voluntário.	Sim

5. Trabalhos Relacionados

Nesta seção serão explorados os trabalhos que partilham do objetivo de criar tecnologias inteligentes para a manutenção sustentável e autônoma de uma agricultura.

O primeiro trabalho [Salazar et al. 2013] apresenta o desenvolvimento de um sistema de controle de irrigação a partir de um SMA e um Arduino, que recolhe a informação dos sensores e as envia a um computador para que essas informações sejam processadas pelo SMA e exibidas ao usuário por meio de uma ferramenta gráfica. O trabalho possui autonomia, pois os agentes conduzem as operações a partir das informações processadas, no entanto acaba sendo dependente de periféricos, uma vez que há a necessidade de um computador para abrigar o SMA.

No segundo trabalho [Jin et al. 2018] é descrito o desenvolvimento de uma rede baseada nos protocolos LoRa (Long Range) [Bor et al. 2016] um transceptor desenvolvido pela Samtech que permite a transmissão e recepção de dados por longas distâncias. O trabalho desenvolve uma ferramenta IoT para dispositivos que coletam informações sobre uma plantação, proporcionando o compartilhamento de dados por longas distâncias. Porém, não há presença de nenhuma inteligência capaz de gerenciar a partir do processamento dessas informações, fazendo com que o trabalho seja mais uma ferramenta de apoio a manutenção do que uma ferramenta para a manutenção propriamente dita.

Por último temos o [Shi 2014], trabalho que apresenta um sistema de manutenção para uma agricultura baseado em uma rede IoT que possibilite o controle de informações e equipamentos da indústria agrária. O trabalho explora ferramentas *WEB* e diversos *frameworks*, dispositivos e softwares como modos de acesso a informação e controle remoto de uma agricultura. Todavia, o trabalho acaba apresentando uma proposta de sistema que não define uma ferramenta específica, fazendo um estudo sobre possíveis ferramentas para cuidar de um plantio que utilizem de uma rede IoT para controle de recursos.

O presente trabalho oferece um dispositivo IoT que independente de periféricos, pois todos os softwares são abrigados em um dispositivo embarcado, que seja autônomo, pois possui um SMA capaz de gerenciar o cultivo sem necessidade de intervenção humana, e que contém uma camada *WEB* estabelecida para que o usuário tenha acesso a



Figura 5. Avaliação experimental.

todas as informações sobre o cultivo em tempo real, trazendo uma implementação real, e não apenas propostas.

6. Considerações finais

O trabalho apresentou uma solução autônoma e sustentável utilizando um SMA embarcado que permite o gerenciamento para evitar tanto o desperdício de recursos hídricos quanto o afogamento do cultivo, sendo a abordagem de agentes central no desenvolvimento deste trabalho. O trabalho se provou funcional através da experimentação proposta onde o dispositivo foi inserido na prática para cuidar e gerenciar os recursos de uma planta, executando todas as funções propostas.

As contribuições que este trabalho pretende trazer são para a área de agentes, expandindo suas funcionalidades e explorando suas utilidades no meio prático, para a área da IoT, expandindo suas possibilidades de uso ao designa-la para auxílio no gerenciamento de uma horta, e também para atividades agrícolas de um modo geral, uma vez que o principal objetivo deste trabalho é desenvolver uma solução para o gerenciamento autônomo que evite o desperdício de recursos.

Para trabalhos futuros pretende-se inserir o sistema em uma horta de ensino, encarregando os cuidados de um ambiente real ao sistema, e também fazer mais testes comparativos entre um cultivo com o sistema e um sem o sistema, fazendo um acompanhamento do nascimento à morte da plantação. Além disso, é preciso permitir com que os SMA presentes em cada dispositivo se comuniquem para cuidar de cultivos cada vez maiores e

mais diversificados, compartilhando informações e atuando para que todo o campo obtenha a melhor produtividade possível de forma autônoma. Também é desejado desenvolver um aplicativo *Android* capaz de exercer as mesmas funções da página *WEB*, expandindo as possibilidades de supervisão a distância para outras plataformas e adicionar um sensor de temperatura, para melhorar a qualidade do diagnóstico ao tentar manter as condições de temperatura estáveis.

Referências

- Bor, M., Vidler, J. E., and Roedig, U. (2016). Lora for the internet of things.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). Programming Multi-Agent Systems in AgentSpeak using Jason. John Wiley & Sons Ltd.
- Chin, K. O., Gan, K. S., Alfred, R., Anthony, P., and Lukose, D. (2014). Agent architecture: An overview. Transactions on science and technology, 1(1):18–35.
- Endler, M., Baptista, G., Silva, L., Vasconcelos, R., Malcher, M., Pantoja, V., Pinheiro, V., and Viterbo, J. (2011). Contextnet: context reasoning and sharing middleware for large-scale pervasive collaboration and social networking. In Proceedings of the Workshop on Posters and Demos Track, page 2. ACM.
- Jin, J., Ma, Y., Zhang, Y., and Huang, Q. (2018). Design and implementation of an agricultural iot based on lora. In MATEC Web of Conferences, volume 189, page 04011. EDP Sciences.
- Lazarin, N. M. and Pantoja, C. E. (2015). A Robotic-Agent Platform for Embedding Software Agents using Raspberry Pi and Arduino Boards. In 9th Software Agents, Environments and Applications School.
- Pantoja, C., Dorneles Soares, H., Viterbo, J., Alexandre, T., Seghrouchni, A., and Casals, A. (2019). Exposing iot objects in the internet using the resource management architecture. International Journal of Software Engineering and Knowledge Engineering, 29:1703–1725.
- Pantoja, C. E., Stabile Jr, M. F., Lazarin, N. M., and Sichman, J. S. (2016). ARGO: A customized jason architecture for programming embedded robotic agents. Fourth International Workshop on Engineering Multi-Agent Systems (EMAS 2016).
- Peckol, J. K. (2019). Embedded systems: a contemporary design tool. John Wiley & Sons.
- Salazar, R., Rangel, J. C., Pinzón, C., and Rodríguez, A. (2013). Irrigation system through intelligent agents implemented with arduino technology.
- Shi, D. L. (2014). Research and design of intelligent agriculture management system based on the internet of things. In Applied Mechanics and Materials, volume 687, pages 1868–1871. Trans Tech Publ.
- Suzuki, J. C. (2007). Modernização, território e relação campo-cidade: uma outra leitura da modernização da agricultura. Agrária (São Paulo. Online), (6):83–95.
- Wooldridge, M. (2009). An Introduction to Multi-Agent Systems. Wiley.

Zhang, D., Ning, H., Xu, K. S., Lin, F., and Yang, L. T. (2012). Internet of things j. ucs special issue. Journal of Universal Computer Science, 18(9):1069–1071.

A Systematic Review of Literature on Recommendation Systems and Machine Learning Applied to Multiagent Systems *

Karine Pestana Ramos¹ and Marilton Sanchotene de Aguiar¹

¹ Programa de Pós-Graduação em Computação
Centro de Desenvolvimento Tecnológico
Universidade Federal de Pelotas
Pelotas, RS – Brazil

{kpramos,marilton}@inf.ufpel.edu.br

Abstract. *This study developed a systematic review of the literature (SRL), a formal study that is used to map a specific area of knowledge. The main question is defined that will guide the entire search during SRL including other formal steps. This SLR has been defined to synthesize and integrate the areas of multi-agent systems, machine learning, and recommendation systems. At the end of the SRL, six studies with different characteristics were found that were adequate to the main question and that satisfy the selection criteria.*

1. Introduction


A Systematic Review of Literature (SRL) is a methodology study that aims to synthesize formally through well-defined steps, the main studies available in a specific area [Okoli and Schabram 2010].

According to [Galvão and Pereira 2014] the systematic reviews must be comprehensive and non-biased in its preparation. So that the researcher can repeat the steps taken and achieve the same results.

To do a good SRL is necessary a well-defined question beyond criteria for identifying, selecting, evaluating, and synthesizing the study results. This work aims to, from an SRL according to the steps mentioned above, raise the possibilities of integrating the areas of Multi-Agent Systems (MAS), Machine Learning (ML), and Recommendation Systems (RS).

This SRL is developed as a part of a major study. It aims to be used as a base for an application in ML for a MAS. To provide a good basis for future works this research is very important to know some information such as, the state of art in the area, how authors have been creating this kind of application, and the system and tools that are used.

The paper is structured as following: in Section 2 are presented the main concepts in the theoretical framework of MAS (Sec. 2.1), ML (Sec. 2.2) and, RS (Sec. 2.3). Section 3 presents the methodology (Sec. 3.1) applied to SRL in this paper (such as

 O trabalho A Systematic Review of Literature on Recommendation Systems and Machine Learning Applied to Multiagent Systems de Karine Pestana Ramos e Marilton Sanchotene de Aguiar está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial 4.0 Internacional. <http://creativecommons.org/licenses/by-nc/4.0/>

databases and keywords), the evaluation steps (Sec. 3.2), the search results (Sec. 3.3), and the discussion of selected articles. Finally, in Section 4 the conclusions are presented.

2. Theoretical Framework

This Section presents briefly the main concepts about Multi-Agents Systems, Machine Learning, and Recommendation Systems, necessary for understanding the SRL.

2.1. Multi-Agent Systems

According to [Singh 1994], a Multi-Agent System (MAS) is an example of distributed computing, composed of the interactions of various computational entities. They are systems capable of modeling theories and human societies.

The term agent is widely used in computing and is usually abstracted from their knowledge, intentions, and desires [Singh 1994]. An agent is considered a basic unit of system intelligence in the scope of this study.

This area aims to solve problems collectively, in a way that a single agent would not be able to perform such a task alone [Hübner et al. 2004]. Thus, it has its focus on the collectivity of the environment, defining the integration between agents and their forms of organization among themselves.

Some characteristics are common to MAS [Weiss 1999]: agents have only incomplete information; there are restrictions on their actions; the control of the system decentralized or distributed; the data is also distributed; and, there is asynchronicity.

As part of an intelligent system, an agent can perceive and act on the environment and, at least partially, his actions are defined by his experience in the environment [Weiss 1999]. In MAS, an autonomous agent is an agent that is independent of other agents. It is important to emphasize that an agent will always be limited by his knowledge.

2.2. Machine Learning

Machine Learning (ML) can be seen as a branch of Artificial Intelligence (AI) as it is a statistical model that has had its progressive growth in the field of computer science.

It has become a common means for any task that requires significant pattern recognition in large databases [Shalev-Shwartz and Ben-David 2014]. Learning applications must be able to learn and adapt. The goal is to program software able to learn given a certain input. Effectively performing specific tasks without the use of explicit instructions, with the use of patterns and inferences [Harrington 2012].

According to [Shalev-Shwartz and Ben-David 2014] learning is the process of incorporating experiences so it becomes specialties or knowledge. The entries in the algorithm are expected to serve as experience to be possible to generate inferences, generating knowledge in its output.

Knowledge of the algorithm should not be limited to its initial training, so like humans, a good algorithm can during its execution be constantly learning as new entries are inserted and learn about their errors [El Naqa and Murphy 2015].

There are two main machine learning classifications: supervised learning and unsupervised learning. In supervised learning, it is necessary a previous step named data training, more specifically, a set of data that must be used as a learning base for the eventual classification of new entries.

In unsupervised learning, however, learning occurs without previous knowledge. Thus, the algorithm does not necessarily attempt to associate each entry with a label (as in supervised learning). The algorithm learns about the input data entered.

Finally, there is learning classification, semi-supervised learning that blends the two previous types of learning, where one part of the data has one classification and the other part doesn't. Thus the already classified part can be used to assist in the classification of the unclassified part.

2.3. Recommendation Systems

Recommendation Systems (RS) are algorithms to establish filtering for the user. It is intended to assist individuals in their choices among a usually large set of options that could cause difficulty for the user to decide. These are systems that seek to facilitate the laborious activity of searching for interesting content.

Word of mouth is a common form of indication where the individual relies on someone else's recommendation (usually of a similar taste or profile). The purpose of recommendation systems is precisely to make this connection even that no one directly indicates something to the individual.

For the recommendation to succeed the group that should receive the recommendation must include individuals with common interests (explicit) or common behavior (implicit) [Cazella et al. 2010]. However, it is a big challenge to do this mapping correctly, creating a correct match for the expectations of users with the products or services that may be recommended to them. To define and discover this relationship of interests can be difficult.

Recommendation systems are widely used in e-commerce sites, using various techniques to provide the most appropriate product recommendations to customers, thereby increasing their profitability.

3. Systematic Review

This Section contains the specification of each step of the methodology defined for the execution of the SRL and a presentation of the main studies found by this review.

3.1. Methodology

The first step to be defined is the question that will guide the entire literature search, which was established as "What are the main articles involved in the context of multi-agent systems using machine learning and recommendation systems?".

The Table 1 is intended to summarize how the main elements were defined for the review, including the main issue, among other information such as objective search criteria, inclusion, and exclusion criteria. The next step is to define the databases for literature research. The selected databases for this study are listed in Table 2 as suggested by [Kofod-Petersen 2012].

Table 1. Systematic literature review protocol

Main Question	“What are the main articles involved in the context of multi-agent systems using machine learning and recommendation systems?”
Goal	Search the literature for studies that are in the context of multi-agent systems using machine learning and recommendation systems only from the last 10 years (2010–2019).
Inclusion Criteria	Articles that contain the following key terms and synonyms: i) Multi-agent Systems ; ii) Machine Learning ; and, iii) Recommendation Systems
Exclusion Criteria	i) the study was not in English; ii) the study was not found in the databases searched; and, iii) the study does not have the searched keywords.

Table 2. Databases used in the study

Name	URL
ACM	https://dl.acm.org/
IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp
Science Direct	https://www.sciencedirect.com/
Springer Link	https://link.springer.com/

Since the protocols of the research and the databases have already been established, the next step is necessary to define the key terms that will be used in the searches. According to [Kofod-Petersen 2012] the main point is to form a string that represents the key terms searched. For instance, if we wanted to search for the term multiagent, the string should not contain only this variation of the term but should also include other variations that appear in the literature, such as multi-agent, multiagent system, among others.

The same method was used for the other searched terms, so the terms are included with their synonyms. In the search result, we desire to find articles that contain any of the terms of that synonym group and equally for the two other term groups. Boolean logical operators like OR and AND are used to achieve this. Finally, it is possible to observe the compilation of these terms in a string generated as a result in Table 3.

After collecting articles found in the four databases cited in Table3, we use two software for reference management to help us organize the results: JabRef and Mendeley. JabRef is used to remove duplicate articles, considering that sometimes an article is found in more than one database [Team et al. 2011]. Mendeley is used for reading and annotating in the articles, working synchronously for several devices [Singh et al. 2010].

3.2. Evaluation Steps

The evaluation steps of the articles found were applied as indicated in the study of [Kofod-Petersen 2012]. The method consists of three steps: title and abstract evaluation, introduction evaluation, and full-text evaluation, applied in exactly that order.

In title and abstract evaluation, the authors read the title and abstract to evaluate if each study is in the context of SRL’s main question, considering the inclusion and exclusion criteria defined in Table 1. So, studies outside the theme are eliminated from the research.

In the introduction evaluation, the authors read introductions from every article approved in the stage before. The evaluation occurs by analyzing if the introduction is

Table 3. Key terms used to define the search string

“multiagent system”	OR	“multiagent systems”	OR
“multiagents system”	OR	“multiagents systems”	OR
“multi-agent system”	OR	“multi-agent systems”	OR
“multi-agents system”	OR	“multi-agents systems”	OR
“agent-based system”	OR	“agent-based systems”	OR
“agent system”	OR	“agent systems”	OR
“agents system”	OR	“agents systems”	OR
“agent based model”	OR	“agent based model”	OR
“agent-based model”	OR	“agent-based models”	
AND			
“machine learning”	OR	“automatic learning”	OR
“machine learning techniques”			
AND			
“decision support system”	OR	“decision support systems”	OR
“recommendation system”	OR	“recommendation systems”	

related to SRL’s main question, besides considering the inclusion and exclusion criteria. The studies approved in this stage still need to pass by a final evaluation.

The last evaluation stage is the reading of the full text of the articles that remain under evaluation. In full-text evaluation, the methodology used is very specific of computing area, where for each article read we analyzed and answered ten questions, whose answers can be yes (1 point), partially (1/2 point) or no (0 points).

Composing a final grade for each article evaluated. The cutoff note is defined by the researchers. In this study, the cutoff note is 6 points. Below are listed the 10 specified questions defined by [Kofod-Petersen 2012]:

1. “Is there is a clear statement of the aim of the research?”
2. “Is the study put into context of other studies and research?”
3. “Are system or algorithmic design decisions justified?”
4. “Is the test data set reproducible?”
5. “Is the study algorithm reproducible?”
6. “Is the experimental procedure thoroughly explained and reproducible?”
7. “Is it clearly stated in the study which other algorithms the study’s algorithm(s) have been compared with?”
8. “Are the performance metrics used in the study explained and justified?”
9. “Are the test results thoroughly analyzed?”
10. “Does the test evidence support the findings presented?”

3.3. Search Results

When performing the searches we collected 206 articles referring to the review in question, which is distributed according to Table 4. Of the 206 articles extracted from the database after removing the duplicates, it remained 190 which were finally analyzed by the study.

In the first evaluation step, title and summary led to the removal of 163 of the articles, leaving only 27 to be analyzed in the next step. The evaluation of the introduction

Table 4. Database search results

Database	Quantity of Articles
ACM	21
IEEE Xplore	22
Science Direct	110
Springer Link	53

restricted the study to only 11 articles to be evaluated for their complete text. The analysis of the last step is summarized in Table 5, where the articles are evaluated as defined in Section 3.2. Numbers 1 through 10 correspond to the questions set out in the same Section (3.2), respectively.

Table 5. Results of last stage

Article Title	1	2	3	4	5	6	7	8	9	10	Points
A Multiagent Approach to Ambulance Allocation Based on Social Welfare and Local Search [Shaft and Cohen 2013]	1	1	0.5	0	0.5	0.5	0.5	0.5	1	1	6.5
A network aware approach for the scheduling of virtual machine migration during peak loads [Duggan et al. 2017]	1	1	0.5	0	0.5	0.5	1	1	1	1	7.5
A novel meta learning system and its application to optimization of computing agents' results [Kazík et al. 2012]	1	1	1	0	0	0.5	1	1	1	1	7.5
Agent-based Decision Support System for Technology Recommendation [Legien et al. 2017]	1	1	1	0.5	0.5	1	1	0.5	0.5	0.5	7.5
ALBidS: A Decision Support System for Strategic Bidding in Electricity Markets [Pinto and Vale 2019]	1	0.5	0.5	0	0	0.5	0	0	0	0	2.5
Decision Support System for Opponents Selection in Electricity Markets Bilateral Negotiations [Silva et al. 2019]	1	0	1	0	0	0	0	0	0	0	2
Decision support system for predicting stock prices based on sentiments in social media [Chornous and Iarmolenko 2017]	1	1	1	0.5	0.5	0.5	1	0	0	0	5.5
Intelligent Digital Learning : Agent-Based Recommender System [Brigui-Chtioui et al. 2017]	1	1	1	0.5	0.5	0.5	1	0.5	0	0	6
Key Technologies of Confrontational Intelligent Decision Support for Multi-Agent Systems [Zhang 2018]	0.5	0.5	1	0	0.5	0.5	0.5	0	0	0	3.5
Real-time Machine Learning Prediction of an Agent-Based Model for Urban Decision-making [Zhang et al. 2018]	1	0	0	0	0.5	0.5	0	0.5	0	0.5	3
Strategic advice provision in repeated human-agent interactions [Azaria et al. 2016]	1	1	0.5	0.5	0.5	0.5	1	1	1	1	6

For the selection of the final articles, they were ranked according to their final score from Table 5. In case of the same score, the rank order presented in alphabetical order. As a cutoff note, it was decided to use at least 6 points, so the study needs to address at least 60% of punctuated criteria.

3.4. Selected Papers

In [Duggan et al. 2017] the study aims at mastering available resource utilization when there are high cloud network saturation peaks. This analysis and prediction of volume traffic used by the real-time network have been studied in ways to efficiently utilize the resources available at times of high access to cloud data centers. Enabling the migration of virtual machines (VMs) without congesting network traffic.

The difficulty of transmission is to balance network resources with network traffic without loss of resources. A single agent is implemented that through pro-reinforcement learning can detect the best moment for live migrations. Thus, the approach is adaptive and able to learn the best moments for the migration of VMs to occur, reducing the network resource saturation at peak times, and the migration time itself.

This study produces a testing environment to simulate a band and its traffic demands. Simulating how the agent can handle the migration situation VMs allowing the agent to decide to schedule the migration or just wait. The reinforcement learning system acts as decision support in the live migration process for testing the developed environment for simulation. The environment allows the agent to observe and maximize rewards based on actions taken. The agent is allowed to migrate or wait.

The study shows through simulated scenarios as results that the agent can learn to migrate when network traffic utilization is low, reducing congestion at peak times and improving the use of network resources during off-peak hours and decreasing the costs. Thus is concluded that the reinforcement learning approach is satisfactory as it provides a live migration framework that has no impact on a cloud computing system, increasing quality of service for cloud clients and providers.

In [Kazík et al. 2012], authors develop a multi-agent system that incorporates different data mining methods. The formulated environment is capable of testing different mining methods and finds the best settings. The AGR model is employed (agent, group, and role), where roles allow the flexible composition of multi-agent systems into group structures. The agent may join a group only by performing a function defined in the structure of the group in question. Two agents interact through communication protocol defined between two roles they assumed. For interaction between two agents, they must be in the same group.

The system learns from the results of previous tasks, gathering experience, becoming able to find the best method possible even for the new datasets. This learning works through basic training.

The developed system tests are divided into two sets. The first set used a simple training error as a comparison criterion, while the second group performed cross-validation five times on the data set. Three search algorithms were compared: search by genetic algorithm (with and without elitism), simulated annealing, and random searching. Study experiments have shown that learning can lead to better models. Considering simpler sets local searches are sufficient and for more sophisticated sets genetic algorithms give better results.

The work of [Legien et al. 2017] is a multi-agent implementation of a labeled deduction system in which agents are used to decomposing the system in knowledge distri-

bution and processing. Agents share a common API that enables their communication. Agents are required to register in the system, so the list of registrations is always updated as agents register or unregister.

Machine learning algorithms can be performed during an inference process. In the developed implementation the concept of breadth-first search (BFS) is used. This math is performed by a specific agent and for distribution math, the reasoning process can be performed by a group of agents working in parallel.

The structural part of the project is divided into two parts, one being the API of the agents and another part are the components of the environment. The first involves the interface and the agent request classes for cooperation with other agents. The second, registration in the environment, sending requests, and sending responses.

The environment has an agent that acts as an administrator, able to register agents working in the environment, and statistics. The study experiments presented in [Legien et al. 2017] simulate a situation in which an environment based on agents is a back end of the metallurgical field recommendation system. Looking for better foundry recommendations. In the beginning, all agents need to register with the system. They submit a request for reasoning, make inferences, calculate a cost of proof, and display the results of the process, cooperating in a distributed environment. The developed model has its advantages. It is flexible and adding modules is easy.

The scenario involved in the study of [Shaft and Cohen 2013] involves a local search in a health care environment with multiple victims and hospitals throughout an accident. The goal is that given an accident with many victims with different severity of injuries they need to be allocated to an ambulance and eventually to a hospital. Generically the problem can be seen as a situation of multi-agent resource allocation.

For the correct allocation of ambulances, it is important to know the state of patients' health to be included in the decision process. The simulation assumes that everyone involved in the accident needs medical attention and needs to go to a hospital, so is created a one-to-one relationship with the number of injuries and ambulances available. Besides, the resources available at the hospital do not necessarily meet the needs of patients.

The problem addressed by the study [Shaft and Cohen 2013] is an optimization problem and therefore the goal is to find a solution that maximizes the total cost function. Three points influence the cost function. The first point is related to hospital capacity. If there are more patients assigned to a hospital than their capacity, the cost function should return to an unfavorable result. The second point involves the resources, i.e. if the patient is assigned for a hospital with fewer resources than necessary for its care it should also return unfavorable values. This situation implies that the patient will have to wait at the hospital for resources to be available that they need. The last point is a balance involving the experience of the ambulance driver, patient severity, and distance to the hospital. Experiments were developed to test the above approach and the results were satisfactory. Scenarios were tested with distributions of different accidents. The study also indicates the application of the work in real scenarios, despite citing several possible enhancements and expansions.

The study presented in [Brigui-Chtioui et al. 2017] is an approach about digital learning, so an agent-based recommendation system is developed which aims to help

students supply knowledge in areas that they are deficient by suggesting relevant learning. The main objective is that with the right support you can make learning more efficient.

The system is cooperative and agent-based, agents are autonomous and able to update recommendations, enabling improvements in the recommendation's outcome based on previous learning platform experiences. The recommendation process occurs from a recommender agent who has unrestricted access to platform data and can be viewed as estimate scores for items not discovered by the user. The recommending agent can judge the relevance of each event. In general, each agent has a communication module that allows message exchange according to the communication protocol.

In the system in question the recommendation deals with learning resources and the users are students. The item rating is represented by a score. The system is based on item decomposition into singular values for analysis of score vs. item. A collaborative filtering process occurs, where similarity is calculated between the references of different users. This assumption works because it assumes that those who liked a particular item on the market tend to continue to like that particular item (or very similar items).

On the developed platform new users have initial data through assessments, but on other systems, one may have no knowledge about the user and then the most popular activities (set as default) are recommended. Initial assessments allow the recommendation system to learn student preferences. Despite highlighting points that could be further explored, the study achieves its objective in developing the proposed system, creating a system with a set of autonomous cooperative agents able to recommend useful resources to the students, to fill their gaps or deficiencies detected.

The study by [Azaria et al. 2016] aims to develop strategies that allow the system to advise on scenarios involving computers agents and people. Scenarios have been modeled in which humans or computing agents can share certain goals, but are essentially interested in themselves.

The authors proposed a social agent to provide counseling. This agent is responsible for modeling the selection of human choice. For the agent to achieve success, their advice must have a high probability of acceptance by people without disregarding the fulfillment of the agent's individual goals. The agent is tested in two different environments, using hundreds of human subjects, scenarios varying in complexity, human behavior modeling, and information that is available to people to make them decide whether or not to accept the agent's advice.

The study demonstrates two possible agent designs, one using a Markov Decision Process (MDP) and one using a social preference model. The proposed agent is a social agent for providing counseling (SAP), which generates counseling according to the following social model. The approach balances the costs for the two participants in the selection process based on social weight. Experiments for the evaluation of SAP and MDP agents were tested in a route selection environment. Four possible commute routes for the driver to commute to work. The system advises a route but the driver does not necessarily have to choose it. The driver's goal is to make the journey in less time. The system considers other factors such as minimum fuel consumption and route safety. Only the agent knows the cost of each route. The tests were applied to men and women who had only as information that their objective was to optimize the route time.

It is concluded that SAP achieves goals and outperforms other implementations, and is simple to implement because its advisory strategy does not depend on the history of interaction with your current user, making it possible to use it in many common situations where there is no knowledge of the number of times that the users have used the system.

The studies discussed so far present different approaches and each paper focus on a very singular problem. The Table 6 summarizes and highlights the use of multi-agent systems and recommendation systems in the final articles.

Table 6. Comparative summary of the selected papers

Papers	Features (Multi-Agent Systems and Recommender Systems)
[Duggan et al. 2017]	MAS: A single agent that learns through reinforcement learning. RS: It allows the system to schedule optimal times for live migration.
[Kazík et al. 2012]	MAS: Multiple agents are implemented as computational intelligence methods. RS: Capable of recommending the appropriate method according to the data.
[Legien et al. 2017]	MAS: Multiple agents are responsible for distributing and processing knowledge. RS: Depends on the application, in experiments of the article the environment based on agents is a back end of the metallurgical field recommendation system.
[Shaft and Cohen 2013]	MAS: Agents are treated as resource allocation problems. RS: The system is capable of deciding how to allocate and schedule ambulances during a mass casualty incident through a relative to a chosen social welfare function.
[Brigui-Chtioui et al. 2017]	MAS: Use of autonomous agents able to recommend and update recommendations for the learning in the platform. RS: Agents act as recommender agents.
[Azaria et al. 2016]	MAS: Propose a social agent provision (SAP) able to generate advice. RS: Uses the proposed agent.

4. Conclusions

SRL aims to synthesize the best studies of specific subarea in the literature. This study has detailed stage by stage of the procedures for elaboration of an SRL for the integration of multi-agent systems, machine learning, and recommendation systems areas, obtaining satisfactory results in this research and demonstrating it is possible to find this intersection point. Methodology and steps performed in this study were well described here making this SRL, as it should be, reproducible.

Although the six finalist studies are about multi-agent systems, recommendation systems, and machine learning with their topics and purposes differ in many ways. Studies diverge in the broadest aspects some are more detailed and others do not describe so deeply their implementation or testing or even the experiments of the work in question.

The studies have different characteristics, as the article by [Duggan et al. 2017] is only implemented with a single agent, and in the study by [Azaria et al. 2016] the implemented agents are for comparison purposes. Unlike other studies that also implement more than one agent, however, they work cooperatively or just perform different functions in the system.

In the study described by [Brigui-Chtioui et al. 2017] there is no information about experiments and there is no discussion about some system tests developed, other studies all included some system tests. Some use searches to support the decision process, which may be local searches, as in the study by [Shaft and Cohen 2013] or as in the study by [Kazík et al. 2012] which makes use of genetic algorithm searching, simulated tempering and random searching. The study covered by [Duggan et al. 2017] is the only that explicitly cites the use of reinforcement learning.

At the end of the SRL, six studies were found that were adequate to the main question and that satisfy the selection criteria. Although the studies have several different points as discussed in Section 3.4, essentially these studies represent a set of the intersection of MAS, machine learning, and recommendation systems as the main point of this research.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES/Brasil) and Agência Nacional de Águas (ANA/Brasil) – Edital N° 16/2017.

References

- Azaria, A., Gal, Y., Kraus, S., and Goldman, C. V. (2016). Strategic advice provision in repeated human-agent interactions. *Autonomous Agents and Multi-Agent Systems*, 30(1):4–29.
- Brigui-Chtioui, I., Caillou, P., and Negre, E. (2017). Intelligent digital learning: agent-based recommender system. In *Proceedings of the 9th International Conference on Machine Learning and Computing*, pages 71–76.
- Cazella, S. C., Nunes, M., and Reategui, E. (2010). A ciência da opinião: Estado da arte em sistemas de recomendação. *André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski.(Org.). Jornada de Atualização de Informática-JAI*, pages 161–216.
- Chornous, G. and Iarmolenko, I. (2017). Decision support system for predicting stock prices based on sentiments in social media. In *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing*, pages 1–4.
- Duggan, M., Duggan, J., Howley, E., and Barrett, E. (2017). A network aware approach for the scheduling of virtual machine migration during peak loads. *Cluster Computing*, 20(3):2083–2094.
- El Naqa, I. and Murphy, M. J. (2015). What is machine learning? In *machine learning in radiation oncology*, pages 3–11. Springer.
- Galvão, T. F. and Pereira, M. G. (2014). Revisões sistemáticas da literatura: passos para sua elaboração. *Epidemiologia e Serviços de Saúde*, 23:183–184.
- Harrington, P. (2012). *Machine learning in action*. Manning Publications Co.
- Hübner, J. F., Bordini, R. H., and Vieira, R. (2004). Introdução ao desenvolvimento de sistemas multiagentes com jason. *XII Escola de Informática da SBC*, 2:51–89.

- Kazík, O., Peková, K., Pilát, M., and Neruda, R. (2012). A novel meta learning system and its application to optimization of computing agents' results. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 170–174. IEEE.
- Kofod-Petersen, A. (2012). How to do a structured literature review in computer science. *Ver. 0.1. October*, 1.
- Legien, G., Sniezynski, B., Wilk-Kołodziejczyk, D., Kluska-Nawarecka, S., Nawarecki, E., and Jaśkowicz, K. (2017). Agent-based decision support system for technology recommendation. *Procedia Computer Science*, 108:897–906.
- Okoli, C. and Schabram, K. (2010). A guide to conducting a systematic literature review of information systems research. *SSRN Electronic Journal*.
- Pinto, T. and Vale, Z. (2019). Albids: A decision support system for strategic bidding in electricity markets. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2375–2377. International Foundation for Autonomous Agents and Multiagent Systems.
- Shaft, D. and Cohen, R. (2013). A multiagent approach to ambulance allocation based on social welfare and local search. In *2013 12th International Conference on Machine Learning and Applications*, volume 2, pages 384–389. IEEE.
- Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.
- Silva, F., Pinto, T., and Vale, Z. (2019). Decision support system for opponents selection in electricity markets bilateral negotiations. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2384–2386. International Foundation for Autonomous Agents and Multiagent Systems.
- Singh, J. et al. (2010). Mendeley: A free research management tool for desktop and web. *Journal of Pharmacology and Pharmacotherapeutics*, 1(1):62.
- Singh, M. P. (1994). *Multiagent systems*. Springer.
- Team, J. D. et al. (2011). Jabref [software].
- Weiss, G. (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press.
- Zhang, Y. (2018). Key technologies of confrontational intelligent decision support for multi-agent systems. *Automatic Control and Computer Sciences*, 52(4):283–290.
- Zhang, Y., Grignard, A., Lyons, K., Aubuchon, A., and Larson, K. (2018). Real-time machine learning prediction of an agent-based model for urban decision-making. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2171–2173. International Foundation for Autonomous Agents and Multiagent Systems.

Towards Jacamo-rest: A Resource-Oriented Abstraction for Managing Multi-Agent Systems^{*†}

Cleber J. Amaral^{1,2}, Jomi F. Hübner¹ and Timotheus Kampik³

¹Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

²Instituto Federal de Santa Catarina (IFSC)
São José – SC – Brazil


³Umeå University
Umeå – Sweden

cleber.amaral@ifsc.edu.br, jomi.hubner@ufsc.br and tkampik@cs.umu.se

Abstract. *The Multi-Agent Oriented Programming (MAOP) paradigm provides abstractions to model and implements entities of agents, as well as of their organisations and environments. In recent years, researchers have started to explore the integration of MAOP and the resource-oriented web architecture (REST). This paper further advances this line of research by presenting an ongoing work on jacamo-rest, a resource-oriented web-based abstraction for the multi-agent programming platform JaCaMo. Jacamo-rest takes Multi-Agent System (MAS) interoperability to a new level, enabling MAS to not only interact with services or applications of the World Wide Web but also to be managed and updated in their specifications by other applications. To add a developer interface to JaCaMo that is suitable for the Web, we provide a novel conceptual perspective on the management of MAOP specification entities as web resources. We tested jacamo-rest using it as a middleware of a programming interface application that provides modern software engineering facilities such as continuous deployments and iterative software development for MAS.*

1. Introduction

Multi-Agent Oriented Programming (MAOP) is an approach that uses first-class abstractions to design and develop distinct dimensions that characterise an MAS, among them agents, environments, and organisations [Boissier et al. 2013]. The *agent* dimension provides abstractions for the development of autonomous entities responsible for decision-making processes, called agents. The *environment* dimension uses abstractions for the definition and programming of environmental artefacts,

 Towards Jacamo-rest: A Resource-Oriented Abstraction for Managing Multi-Agent Systems by Cleber J. Amaral, Jomi F. Hübner and Timotheus Kampik is licensed under a Creative Commons Attribution 4.0 International License. <https://creativecommons.org/licenses/by/4.0/>

^{*}This study was partially funded by Project AG-BR of Petrobras and by the program PrInt CAPES-UFSC “Automação 4.0” and partially funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP).

i.e., distributed non-autonomous resources that can be used and shared by agents to achieve their goals. Finally, the *organisation* dimension allows structuring, coordinating, and regulating the interactions in the system, among agents, as well as between agents and the shared environment.

The Web, in contrast, has its abstractions specifically designed to sustainably support the Internet at scale. On the Web, interactions among applications and services frequently use REpresentational State Transfer (REST) Application Programming Interfaces (APIs) [Costa et al. 2014]. REST refers to an architecture that models the elements of the Web as resources [Fielding and Taylor 2002].

Recent works have already explored the intersection of the Web, REST, and MAOP [Ciortea et al. 2018], focusing on the integration of MAS and Web abstractions to facilitate run-time interaction between MAS and traditional web applications. In particular, web service abstractions that allow applications to interact with Jason agents [Rafalimanana et al. 2020], as well as with agents written in ASTRA [W. Collier et al. 2019], have been developed. While the MAS integration with other applications is broadly covered in the literature, the support for the development and run-time modification of MAS specifications and code has so far not been considered. In this paper, we call this type of integration *management integration*. In particular, the paper presents an early version of the *jacamo-rest* implementation that places MAS on the Web as a resource-oriented interface that supports management integration.

This paper is organised as follows: Section 2 presents MAS entity abstractions, whereas Section 3 presents the Web abstractions counterpart. Then, Section 4 introduces *jacamo-rest*, an interface implementation for MAS on the Web. In particular, the section highlights the concepts of *management* and *interaction* integration and presents the results of an early evaluation. Subsequently, Section 5 discusses the presented work, before Section 6 presents future research directions and Section 7 concludes the paper.

2. The Multi-Agent Oriented Model

The multi-agent oriented programming approach proposes a distinction of entities in dimensions of programming. The division in dimensions facilitates the development of entities of each class, making use of specific abstractions, languages and paradigms. In their work, [Boissier et al. 2013, Boissier et al. 2019] propose three dimensions: (i) the environment dimension, (ii) the organisation dimension, and (iii) the agent dimension. Figure 1 illustrates these dimensions and relationships between agents and environmental and organisational artefacts.

The dimension of the environment is composed of *artefacts* and *workspaces*. Artefacts are virtual elements with observable properties and operations that can be triggered by agents [Ricci et al. 2011]. They represent non-autonomous entities that agents can use and share. Some examples of entities that can be modelled as artefacts are physical devices, virtual and physical resources such as non-autonomous software, tools and all sort of objects. An example of a physical device is a machine that has *displays* for showing its state (observable properties) and buttons (operations). When an artefact represents an external entity, it usually has a mean to

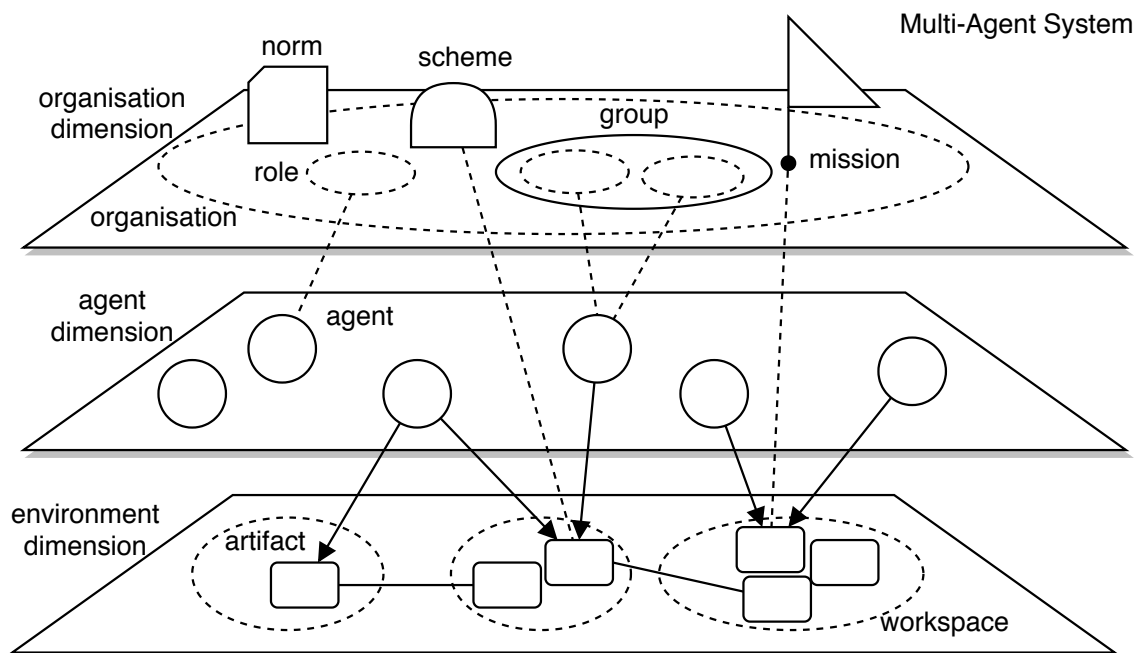


Figure 1. The JaCaMo multi-agent oriented programming model, adapted from [Boissier et al. 2013].

communicate with this concrete entity and mirrors this object, which allows agents to observe and control it just as if they could control the corresponding real-world object. *Workspaces* are virtual boundaries for artefacts, working similarly to physical boundaries for real-world objects. An agent has to enter a workspace to observe and control artefacts within its boundaries.

The *organisation* dimension is composed of organisational artefacts that describe abstract entities of organisations. Organisations can be an MAS's internal or external entities. Some organisational artefacts describe how agents are structured in groups, others define sub-goals schemes that must be achieved to fulfil a global goal, and yet others represent missions associated with sub-goals and organisational roles [Hübner et al. 2010]. These artefacts help maintain coherence among agents and usually provide means to coordinate them.

Finally, the *agent* dimension is composed of autonomous entities that are situated in an environment, perceiving and acting in it. An artificial agent is a computer program with distinguishing features, of which we highlight: (i) the agent is autonomous, *i.e.*, takes its own decisions including rejecting orders from other agents, even from humans; (ii) the agent is proactive, *i.e.*, it is goal-oriented, which in practice means it has initiative and for chasing a goal it may perform several different plans; (iii) an agent is social, which means it is aware of the existence of other agents and can act in a cooperative way towards mutual goals [Wooldridge 2002]. Among possible actions that agents may perform we have: entering workspaces, reading observable properties of artefacts, operating artefacts, adopting organisational roles, fulfilling individual and collective goals, and communicating with other agents.

3. Resource-Oriented Architecture and Representational State Transfer

With the emergence of the Web as the world's most important application ecosystem, a new generation of software architecture paradigms have risen to prominence. One of those paradigms is *resource-oriented architecture* and in particular REpresentational State Transfer (REST) [Fielding 2000], which is—at least so some extent—reflected in most web applications. In REST, any information object that can be named, for example, an image, a document, or the virtual presentation of a physical object, is considered a resource [Fielding 2000]. REST is a technology-agnostic protocol that specifies how such resources should be exchanged between a server and a client.

According to REST, each resource on a server has a Unique Resource Identifier (URI). Clients can create, retrieve, update, and delete resources on a server by sending requests against this server, whereby existing resources are accessed and manipulated via their URI. When retrieving data from a resource type's endpoint, all resources of the corresponding type will be listed, or a subset of these resources, if a filter is applied by specifying query parameters. REST aims to be stateless in that all state about a client's session is maintained by the client and not by the server. Typically, REST is used in conjunction with the Hypertext Transfer Protocol (HTTP). Then, REST can make use of HTTP verbs (**GET**, **POST**, **PUT**, **DELETE**, **OPTIONS**, *et cetera*) to access and manipulate resources. REST does not specify a data encoding for resources; the format can be, for example, HTML, XML, or JSON. *RESTful services* are modelled using resource-oriented architecture (ROA), make proper use of HTTP verbs, and allow their Application Interfaces (APIs) to be discovered by other systems [Richardson and Ruby 2008].

4. The *jacamo-rest* middleware

The JaCaMo platform¹ joins different *frameworks* for developing distinct dimensions of MAS. The agents that run on this platform inherit facilities from the Jason *framework*, allowing the interpretation of the *AgentSpeak* language, which includes among other functionalities sending messages among agents. Although this feature is a powerful tool, this integration is restricted to agents that run on the same platform instance. Among extensions of the JaCaMo ecosystem, we have the possibility to use JADE [Bellifemine et al. 2003] as an infrastructure. In this case, the JADE framework serves as an integration middleware for agents. However, this integration is also limited to artificial agents, which in this case can be Jason agents or JADE agents.

Another way to address integration in JaCaMo is through artefacts. An artefact can be used for integrating an agent and a non-autonomous external entity, and for integrating agents themselves. In the former case, an artefact can be built to communicate with an external entity embedding a specific protocol and virtualising the external entity, which allows agents to manipulate this representation. In the latter case, an artefact can be used as a means for communication among agents, for instance, a telephone or an email account artefact. Such artefacts may support

¹<http://jacamo.sourceforge.net/>.

communication between internal agents and external autonomous entities, for example, with humans. The limitation of this approach is that each kind of external entity may require the development of a specific integration.

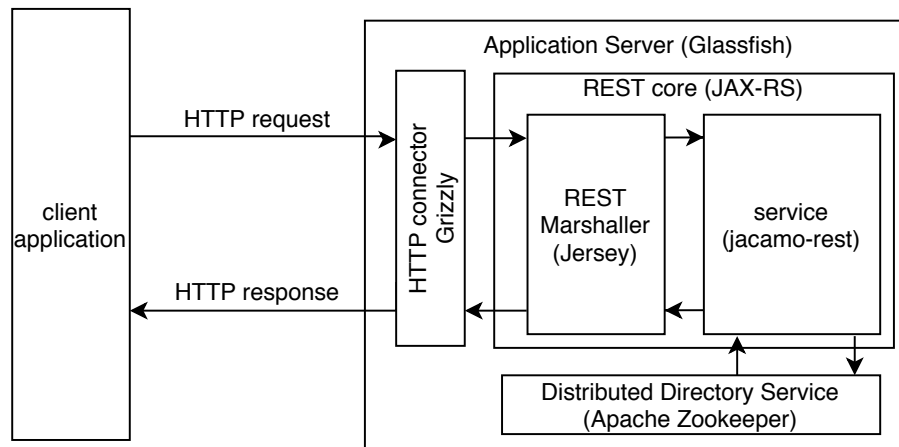


Figure 2. Data flow among libraries and frameworks used by *jacamo-rest*

To expose JaCaMo entities as resources on the Web, we introduce *jacamo-rest*. The *jacamo-rest* extension² is developed in Java, the same technology that the JaCaMo platform is based on. To support REST functionalities, *jacamo-rest* adds libraries for defining endpoints and web infrastructure for providing web-server facilities. Figure 2 illustrates the main libraries used in the extension and how data flows between them.

Jacamo-rest is instantiated by the *Glassfish* application server. The client, when communicating with *jacamo-rest*, makes requests over the HTTP protocol. The *Grizzly* HTTP connector receives and sends data in this protocol, executing low-level functionalities for mediating and routing HTTP messages. The connector delivers the content of the messages, *i.e.*, the payload, to the REST *Jersey* marshaller. *Jersey* forwards the data, which results in method invocations defined in *jacamo-rest* classes. Each method follows the specification of the *JAX-RS* Java library, which defines how data enters and leaves the REST implementation. In other words, each REST method answers to an *endpoint* provided by *jacamo-rest*. Additionally, *jacamo-rest* provides a directory service facility that publishes the list of agents and their functionalities similar to a distributed yellow-pages service.

Jacamo-rest makes available mainly three collections of *endpoints*: (i) related to agents (*/agents*), (ii) related to the environment (*/workspaces*), and (iii) related to organisations (*/organisations*). Each one of these collections is defined in a specific class that implements REST methods. For instance, the class *RestImplAg* implements the *endpoint* *GET /agents* that returns the list of agents and the *endpoint* *GET /agents/{agentuid}* returns data about the agent identified by *agentuid*.

Jacamo-rest has implementations for mediating data that flows between

²Jacamo-rest is an open-source project available at <https://github.com/jacamo-lang/jacamo-rest>.

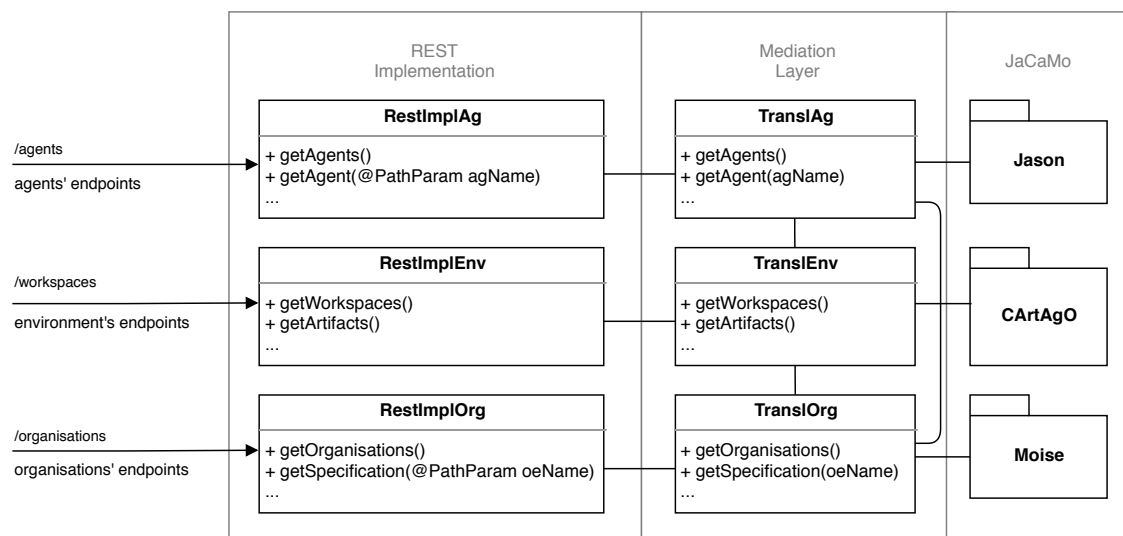


Figure 3. Jacamo-rest's collections of endpoints and software layers view

REST implementation classes and the JaCaMo platform. These mediation classes are responsible for delivering data from JaCaMo in the format that will be sent to the client through a REST interface. The mediation classes communicate among each other, which allows for the reuse of some methods. For example, the mediator of the environmental dimension has methods to format the data of artefacts. When an agent's endpoint needs to show artefact information, for instance, the artefacts an agent is observing, the mediator of agents uses data of the mediator of the environment to produce the final content. Figure 3 illustrates how these classes are interconnected.

4.1. Interaction Integration and Management Integration

Existing works primarily shed light on the *interaction integration* of agents and multi-agent systems on the Web [Ciortea et al. 2018, Ciortea et al. 2019, Rafalimanana et al. 2020, W. Collier et al. 2019]; *i.e.*, the works propose ways for agents and multi-agent systems to interact at run-time via the Web. The purpose of RESTful APIs is, in this context, the provision and update of *application state*. Interaction integration is a requirement from the execution process view, *i.e.*, it is concerned with enabling communications between an MAS and third-party systems according to the requirements of the systems' application domain.

In our approach, the integration among applications is provided by message exchange, which is the standard way agents interact with each other. Indeed, the interaction integration view merely requires access to the services the MAS can provide, which translates to write access to send messages to agents' mailboxes (POST). It is important to mention that the `/inbox` endpoint is only suitable for inbound requests, *i.e.*, the MAS working as a server. In Section 4.2, we point to our previous work that allows exposing MAS as clients as well. It is also worth noting that messages to an agent's inbox can be considered a Remote Procedure Call (RPC) style functionality; it does not retrieve or manipulate particular resources; instead, it triggers agent actions. A concern with this approach is its vulnerability on overloading

the system with a possible flood of requisitions [Richardson and Ruby 2008]. In the context of REST, RPCs are considered anti-patterns, which triggers the question of how these JaCaMo API endpoints can be exposed in a more idiomatic manner. We touch upon this issue in the Section 5. Still about interactions, one may point out that *jacamo-rest* is not exposing environmental and organisational artefacts. Indeed, currently, we are considering them as internal MAS representations. In this sense, if needed, agents should mediate the sharing of such resources.

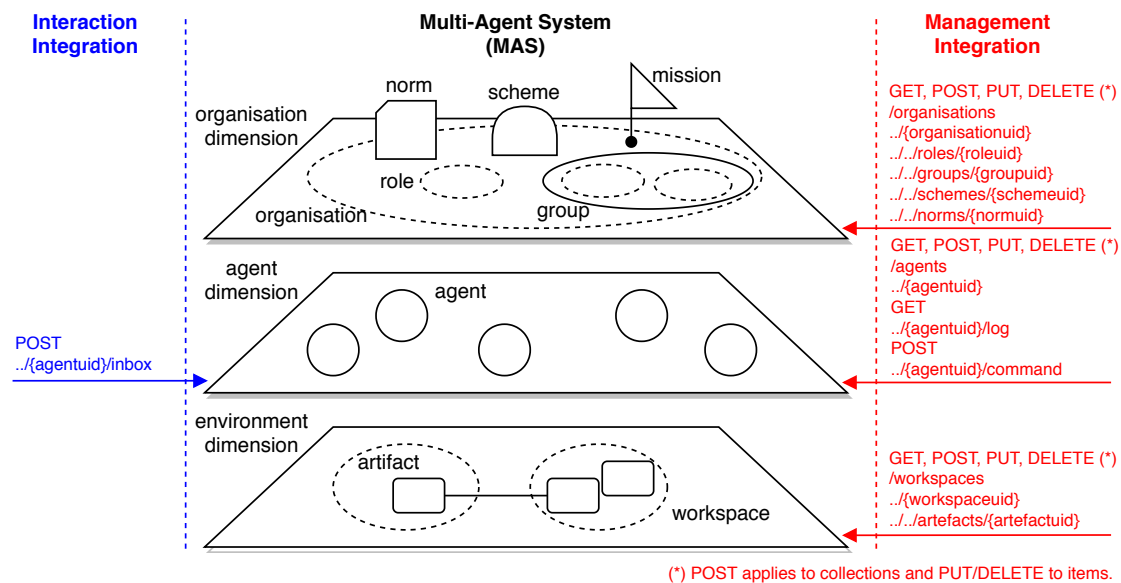


Figure 4. Endpoints and methods (HTTP verbs) shared between external applications and the MAS (left, blue), and for the MAS development (right, red).

The notably facilities of *jacamo-rest* is on *management integration*, *i.e.*, the update and retrieval of *models* of agents, artefacts and their environment. Management integration is concerned with the development and operations view, and can be seen in the light of the *DevOps* best practices in modern software development [Ebert et al. 2016], which attempt to facilitate the creation of autonomous development teams that are in charge of the whole process from development via testing and deployment to operations, and make use of a sophisticated and highly automated toolchain. Management integration requires a superset of the API endpoints that the interaction integration requires, because read and write access to both application state and models is important to deploy, update, debug and hot-fix the MAS and its models.³ Consequently, the approach is well-aligned with modern software engineering best practices. This new perspective further facilitates interoperability, distribution, and agility of MAS, as it allows developers to update MAS on the fly, and to automate these updates via continuous integration. Figure 4 provides an overview of the resources and methods that are shared between application and management view (on the left-hand side), as well as the of ones that are specific to the management view (on the right-hand side).

Most *management integration* endpoints are compliant with RESTful design best practices. MAS entities are properly modelled as resources, allowing operations

³Our study does not take into account security issues.

such as retrieve, create, update, and delete resources. HTTP verbs are used as suggested, *i.e.*, GET only retrieves data retrieve (and is safe and idempotent), PUT and DELETE are both idempotent, and POST creates, in most scenarios, new resources. The exception is the POST `/command` which is, just like the `/inbox` endpoint, an RPC-like method.

Finally, our implementation allows navigating application state via its API, in accordance with the *Hypermedia As the Engine of Application State* (HATEOAS) concept. To do so, we use the Jersey facility that implements the OPTIONS verb for each exposed endpoint. This operation provides to the client all endpoints and possible methods that can be triggered. Developers and artificial agents can take advantage of this information and use it as a manual that helps them find resources and facilities provided by the API.

4.2. Application Example

We tested the *jacamo-rest* concept of how MAS abstractions can be modelled as resources by implementing the *jacamo-web* application⁴ [Amaral and Hübner 2019]. The *Jacamo-web* Integrated Development Environment (IDE) allows programming MAS interactively by direct manipulation of running instances with collaborative and integrative tools [Amaral et al. 2020]. The IDE uses most of the *endpoints* provided by *jacamo-rest*.

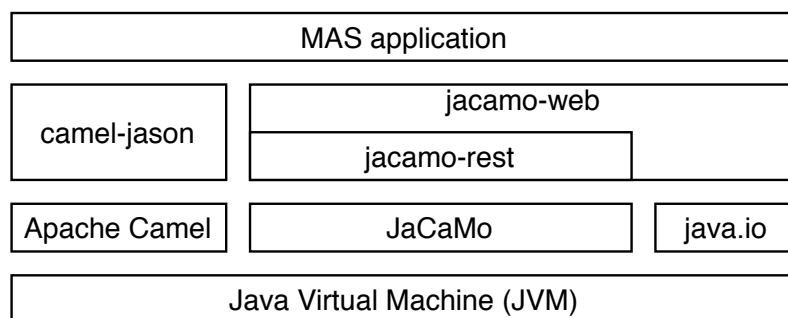


Figure 5. Architecture of an example of application using *jacamo-rest*

Figure 5 illustrates an integration architecture, which in this case extends the project *jacamo-rest* to provide the web-based user interface *jacamo-web*. Besides inheriting *jacamo-rest* functionalities, the extension adds support for file access through the *java.io* library. This is necessary because *jacamo-web* provides static files, like HTML pages and JavaScript files, to the client that then renders the interface. *Jacamo-web* also accesses files of the JaCaMo platform, such as agent source code files, which can be manipulated through the *jacamo-web* user interface.

In addition, we illustrate in Figure 5 an implementation of an MAS application on the top of the architecture. In this example, the agents need to request data from external entities, as discussed in Section 4.1, something that *jacamo-rest* does not provide native support for. To address this issue, we have used the approach proposed by [Amaral et al. 2019], adding an Apache Camel component for agents that

⁴Jacamo-web is available at <https://github.com/jacamo-lang/jacamo-web>.

allows specifying communication routes using an extensive range of communication components for different protocols.

In this sense, this application is taking advantage of management integration for providing interfaces for changing agent, artefact and organisation implementations. Engineers that are using this IDE may collaborate when developing the MAS interactively, introducing pieces of code to the MAS through *jacamo-rest* facilities.

5. Discussion

The specific instantiation of the more generic resource-oriented management and interaction integration design, the implementation of the *jacamo-rest* API, as well as the embedding of *jacamo-rest* into *jacamo-web*, raised some intriguing questions. An interesting aspect when providing a RESTful design interface to an MAS is the implementation of an interface that enables the navigation of historical states (*time travel*⁵). Clients that request models from an MAS server need to be able to check whether a model resource they retrieve reflects the same model version of the previously retrieved resource. This can be achieved by controlling resource revisions and providing them using specific endpoints such as `/agents/{agentuid}/revisions/{revisionuid}`.

A further question, as mentioned in Subsection 4.1, is how to expose the RPC-like JaCaMo interfaces to an agent's mailbox as well as to its command interface in a more RESTful manner. This can potentially be achieved by treating messages and commands as resources, which are created by a POST request of the client, who then receives a UID of the corresponding message or command resource. Retrieving the resource with a GET request can provide the status of the message or command, *i.e.*, a description of how the agent has processed the message so far. However, from a pragmatic perspective, one could alternatively conclude that REST is not a suitable protocol for the message and command interface. Instead, one could implement these parts of the API using a protocol that has real-time support as a first-class concern, for example, a publish/subscribe or socket-based interface.

We have to mention that our approach can be seen as limited since it has no native facility for the MAS working as a *client* on the Web. One may suggest that besides a `/inbox` endpoint we can provide an `/outbox` facility. In this sense, an agent may add a message to a queue that should be managed by a process which later sends it to some external entity. However, it would also be necessary to provide proper representations for external entities which is out of the scope of this paper. In this sense, since the *camel-jason* component [Amaral et al. 2019] provides such representations and since it is compatible with REST, we propose it for outbound communication needs.

6. Future Work

This paper presents a snapshot of a work in progress. The following research directions can be considered as promising to further advance the line of work.

⁵Time-travelling functionality is a feature that is, in particular, proposed in the context of mainstream programming languages like JavaScript [Barr et al. 2016].

Implement Revisioning and Time-Travel Functionality.

In Section 5, we have proposed integration of RESTful resource provision by the server and version control of specification object state. While the integration of a version control system and *jacamo-rest* is a feature of the *jacamo-rest*-based *jacamo-web* system, the implementation is not yet adhering to the proposed approach. Also, a more thorough evaluation of useful revisioning and versioning functionality will likely yield additional features, such as the explicit versioning of agents (*minor*, *major*, *patch*) to enable version-based compatibility checking.

Generalise the jacamo-rest interface for other MAOP systems.

While the *jacamo-rest* implementation is MAOP platform-specific, other AOP platforms can be exposed through a similar interface. To enable true interoperability, framework-agnostic interfaces can be designed and implemented, building a heterogeneous ecosystem of agents and resources that are integrated at both run-time and design-time.

Implement outbound facilities, as well as additional inbound facilities.

We have proposed an `/outbox` facility for agents that serve as clients. Besides this, environmental and organisational artefacts can also be shared from both a local system's perspective and an external system's perspective. For these functionalities, additional representations need to be modelled, and suitable endpoints need to be implemented.

Add better support of HATEOAS.

Besides providing the `OPTIONS` method for each endpoint, it is recommended to provide links for each relation the retrieved resource has. For instance, when retrieving data about an agent, the agent resource should link to each artefact the agent is observing, which helps to find the right location of the resource.

7. Conclusion

In this paper, we have presented *jacamo-rest*, a system that enables a resource-oriented approach to provide abstractions on multi-agent systems and their agents, artefacts, and organisations. In contrast to related works, the focus of *jacamo-rest* is on the provision of *management* abstractions on MAS, and not primarily on enabling resource-oriented *interactions* between agents and other Web resources. Consequently, the presented work can be considered a stepping stone towards the integration of multi-agent oriented programming and modern software engineering approaches, in which the automated and (somewhat) autonomous management of software artefacts is a key concern.

References

- Amaral, C. J., Bernardes, S. P., Conceição, M., Hübner, J. F., Lampert, L. P. A., Matoso, O. A., and Zatelli, M. R. (2019). Finding new routes for integrating Multi-Agent Systems using Apache Camel. *Workshop-School on Agents, Environments, and Applications (WESAAC) 2019 proceedings*.
- Amaral, C. J. and Hübner, J. F. (2019). Jacamo-web is on the fly: an interactive multi-agent system ide. In *7th International Workshop on Engineering Multi-Agent Systems (EMAS 2019)*.

- Amaral, C. J., Kampik, T., and Cranefield, S. (2020). A framework for collaborative and interactive agent-oriented developer operations. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '20*, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.
- Barr, E. T., Marron, M., Maurer, E., Moseley, D., and Seth, G. (2016). Time-travel debugging for javascript/node.js. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE 2016*, page 1003–1007, New York, NY, USA. Association for Computing Machinery.
- Bellifemine, F., Caire, G., Poggi, A., and Rimassa, G. (2003). JADE A White Paper. *exp*, 3(September):6–19.
- Boissier, O., Bordini, R. H., Hübner, J. F., and Ricci, A. (2019). Dimensions in programming multi-agent systems. *The Knowledge Engineering Review*, 34.
- Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013). Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747 – 761.
- Ciortea, A., Mayer, S., Gandon, F., Boissier, O., Ricci, A., and Zimmermann, A. (2019). A decade in hindsight: The missing bridge between multi-agent systems and the world wide web. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19*, page 1659–1663, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.
- Ciortea, A., Zimmermann, A., and Florea, A. M. (2018). *Give Agents Some REST: Hypermedia-driven Agent Environments*, pages 125–141. springer.
- Costa, B., Pires, P. F., Delicato, F. C., and Merson, P. (2014). Evaluating a representational state transfer (rest) architecture: What is the impact of rest in my architecture? In *2014 IEEE/IFIP Conference on Software Architecture*, pages 105–114.
- Ebert, C., Gallardo, G., Hernantes, J., and Serrano, N. (2016). Devops. *IEEE Software*, 33(3):94–100.
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. Doctoral dissertation, University of California, Irvine.
- Fielding, R. T. and Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Trans. Internet Technol.*, 2(2):115–150.
- Hübner, J. F., Boissier, O., Kitio, R., and Ricci, A. (2010). Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, 20(3):369–400.
- Rafalimanana, H. F., Razafindramintsa, J. L., Cherrier, S., Mahatody, T., George, L., and Manantsoa, V. (2020). Jason-rs, a collaboration between agents and an iot platform. In Boumerdassi, S., Renault, É., and Mühlethaler, P., editors, *Machine Learning for Networking*, pages 403–413, Cham. Springer International Publishing.

- Ricci, A., Piunti, M., and Viroli, M. (2011). Environment programming in multi-agent systems: an artifact-based perspective. *Auton Agent Multi-Agent Syst*, 23(2):158–192.
- Richardson, L. and Ruby, S. (2008). *RESTful web services*. ” O’Reilly Media, Inc.”.
- W. Collier, R., O’Neill, E., Lillis, D., and O’Hare, G. (2019). Mams: Multi-agent microservices. In *Companion Proceedings of The 2019 World Wide Web Conference, WWW ’19*, page 655–662, New York, NY, USA. Association for Computing Machinery.
- Wooldridge, M. (2002). Intelligent agents: The key concepts. In Mařík, V., Štěpánková, O., Krautwurmová, H., and Luck, M., editors, *Multi-Agent Systems and Applications II*, pages 3–43, Berlin, Heidelberg. Springer Berlin Heidelberg.

Implantação de Precificação Dinâmica em um Estacionamento Inteligente utilizando Agentes*

Alexandre L. L. Mellado¹, Gleifer Vaz Alves¹, Paulo Leitão², André Pinz Borges¹

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

²Research Centre in Digitalization and Intelligent Robotics (CeDRI)
Instituto Politécnico de Bragança (IPB)
Bragança – Portugal

mellado@alunos.utfpr.edu.br, gleifer@utfpr.edu.br

pleitao@ipb.pt, apborges@utfpr.edu.br


Abstract. *In large cities, parking is a limited resource and drivers spend a lot of time looking for a spot, which causes wasted fuel, congestion and pollution. Here, we develop a method to handle the spot allocation using a dynamic pricing engine. This engine, programmed in Java and implemented in a Multi-Agent System developed in the JaCaMO framework, dynamically changes prices in parking lots. The simulations performed show that a specialized pricing module is better for parking than not using any module.*

Resumo. *Em grandes cidades, estacionamento é um recurso limitado e motoristas gastam muito tempo à procura de uma vaga, o que causa desperdício de combustível, congestionamento e poluição. Neste trabalho é desenvolvido um método para tentar lidar com a alocação de vagas através do uso de um módulo de precificação dinâmica. Esse módulo, programado em Java e implantado em um Sistema Multi-Agente (SMA) desenvolvido no framework JaCaMO, realiza mudanças de preços de estacionamentos dinamicamente. As simulações executadas evidenciam que um módulo de precificação especializado é melhor para o estacionamento do que não usar nenhum módulo.*

1. Introdução

Estacionamentos são recursos essenciais para mobilidade sustentável em áreas urbanas. Esse recurso é limitado em grandes metrópoles, o que leva os motoristas a transitarem por um longo período de tempo para encontrar uma vaga disponível [Shoup 2006]. [Polycarpou et al. 2013] mostra que, durante a procura por vagas de estacionamento, motoristas desperdiçam tempo e combustível, aumentando o congestionamento e a poluição.

Devido a essa limitação de vagas, o preço das vagas pode ser utilizado para orientar as opções de estacionamento dos motoristas de forma dinâmica, buscando melhorar

* O trabalho Implantação de Precificação Dinâmica em um Estacionamento Inteligente utilizando Agentes de Alexandre L. L. Mellado, Gleifer Vaz Alves, Paulo Leitão, André Pinz Borges está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

a taxa de ocupação. Este valor pode ser controlado de forma eficiente em um Estacionamento Inteligente (EI) que é diferenciado por utilizar tecnologias para automatizar e melhorar suas tarefas diárias, como alocação e pagamento de vagas [Di Nocera et al. 2014].

A precificação dinâmica é uma abordagem de alteração do preço de serviços ou mercadorias com base em mudanças no mercado. A precificação pode maximizar a receita de um estacionamento, cobrando um maior valor pelas vagas quando a demanda é mais alta. Isso compensa as potenciais perdas durante um período de baixa demanda. Além disso, o valor reduzido em períodos de baixa demanda pode incentivar os motoristas a utilizarem os recursos dos estacionamentos [Tian et al. 2018].

Além de ser vantajoso para o estacionamento, a precificação dinâmica também é benéfica para os motoristas. Isso é devido às taxas do estacionamento condizerem com o valor real das vagas, em vez de uma cobrança estática arbitrária. Os motoristas que usualmente estacionam durante períodos de baixa demanda ou em áreas com menor demanda perceberão que acabam pagando valores mais baixos pelas vagas [Tian et al. 2018].

O trabalho apresentado neste artigo é uma extensão do projeto MAPS (UTFPR) para gerenciamento de EIs. Alguns resultados já obtidos neste projeto foram: **i.** [Castro et al. 2017] implementam um SMA para alocação de vagas baseado no grau de confiança dos agentes motoristas; **ii.** em [Ducheiko et al. 2018], é apresentado um SMA, onde os agentes negociam por vagas seguindo um modelo aberto e descentralizado; **iii.** [Botelho et al. 2019] propõem a implantação de um EI empregando agentes embarcados.

Neste artigo é proposto um módulo de precificação que pode ser acoplado em um SMA [Wooldridge 2009] de forma que sejam consideradas as variáveis do estacionamento que podem ser modificadas dinamicamente: demanda de vagas e o fator temporal. Esse módulo foi desenvolvido usando como base o SMA, desenvolvido em [Castro et al. 2017]. O SMA foi implementado usando o *framework* [JACAMO 2011], composto por três linguagens: Jason, responsável pela programação dos agentes; Cartago, utilizado para implementar os artefatos do ambiente e Moise, usado na organização dos indivíduos do sistema. JaCaMo foi escolhido por facilitar no desenvolvimento de um SMA ao possuir as linguagens interligadas e porque foi utilizado no SMA usado como base neste trabalho.

O SMA desenvolvido previamente [Castro et al. 2017] apresenta um mecanismo básico de negociação e alocação de vagas para os agentes motoristas, facilitando o acoplamento de um módulo de precificação dinâmica. Esse SMA compõe a parte de negociação de um Smart Parking. Neste trabalho é desenvolvido um módulo de precificação com viés de identificar o seu impacto sobre a dinâmica de negociações. Almeja-se que utilizando tal módulo, seja possível identificar se a utilização de um módulo de precificação resulta em ganho monetário para um estacionamento.

O restante do artigo está organizado da seguinte maneira. Na seção 2 são descritos os trabalhos relacionados ao paradigma de precificação dinâmica e projetos MAPS. Na seção 3 tem-se o funcionamento do módulo de precificação. Na seção 4 apresentam-se os resultados das simulações executadas. Na seção 5 tem-se as considerações finais.

2. Trabalhos Relacionados

Nesta seção destacam-se trabalhos que precificação dinâmica e implementação de SMA para EIs.

Em [Chen and Sheldon 2015] é realizada uma medição de como preços dinâmicos afetam a oferta de trabalho de motoristas do Uber, utilizando medições das elasticidades de motoristas e de ofertas positivas. Observa-se que o aumento no preço das viagens diminuem a taxa de pausa, o que leva os motoristas a ajustarem suas programações para dirigir por mais tempo e fornecer mais viagens nos momentos com maior demanda.

Em [Williams 2018] foi criado um modelo de cálculo dinâmico de preços de companhias aéreas, considerando as principais interações entre o padrão de chegada de categorias de consumidores e a capacidade restante sob demanda estocástica. O modelo aumentou os lucros e consumidores do sistema.

Um modelo de precificação dinâmica para reserva de vagas foi apresentado em [Tian et al. 2018]. Apesar do modelo considerar apenas a demanda e número de vagas, este resultou em melhor utilização de vagas e aumento de lucros.

Em [Pierce and Shoup 2013] é revisado o aplicativo *SFpark* que ajusta preços para a aumentar a disponibilidade de vagas. Foram realizados testes com 5000 mudanças de preço, mostrando que o preço médio do medidor caiu 1% durante o primeiro ano. Logo, o *SFpark* ajustou os preços sem aumentá-los, em geral.

Os artigos anteriores mostram exemplos de soluções utilizando precificação dinâmica e sua universalidade na era da informação. Observa-se que a precificação dinâmica pode auxiliar na gestão de alocação de recursos e gerar lucro ao proprietário utilizando estes métodos.

Em relação à trabalhos usando SMA para EIs, [Castro et al. 2017] utilizou o *framework* JaCaMo para alocação de vagas. Este sistema é composto por um agente *manager* e agentes *drivers*. O *manager* é responsável por administrar as vagas que serão alocadas aos *drivers*, de acordo com os respectivos graus de confiança destes agentes. Esse grau de confiança é um valor que mostra o comprometimento do *driver* em relação ao estacionamento. Foram realizadas diversas simulações em cenários empíricos que, em linhas gerais, indicam que um grau de confiança maior resulta em um menor tempo de espera para o agente *driver*.

O trabalho de [Ducheiko et al. 2018] expande o projeto MAPS ao criar um modelo de raciocínio para desenvolver o mecanismo de negociação de vagas de forma descentralizada. Este descreve os protocolos e fórmulas elaboradas para a negociação entre os agentes, e é implementado um SMA utilizando o *framework* JaCaMo, destacando a organização social do sistema através de regras do Moise.

Neste artigo tem-se como objetivo complementar os trabalhos de [Castro et al. 2017] e [Ducheiko et al. 2018] por meio do desenvolvimento de um módulo de precificação genérico, capaz de ser implantado em diferentes tipos de estacionamentos.

3. Desenvolvimento

Este trabalho visa desenvolver um módulo de precificação dinâmico que possa ser usado em qualquer EI. Para implantar esse módulo foi necessário a utilização de um SMA já existente. Esse sistema em questão [Castro et al. 2017] foi modificado para realizar negociações simples entre os agentes *driver* e *manager*.

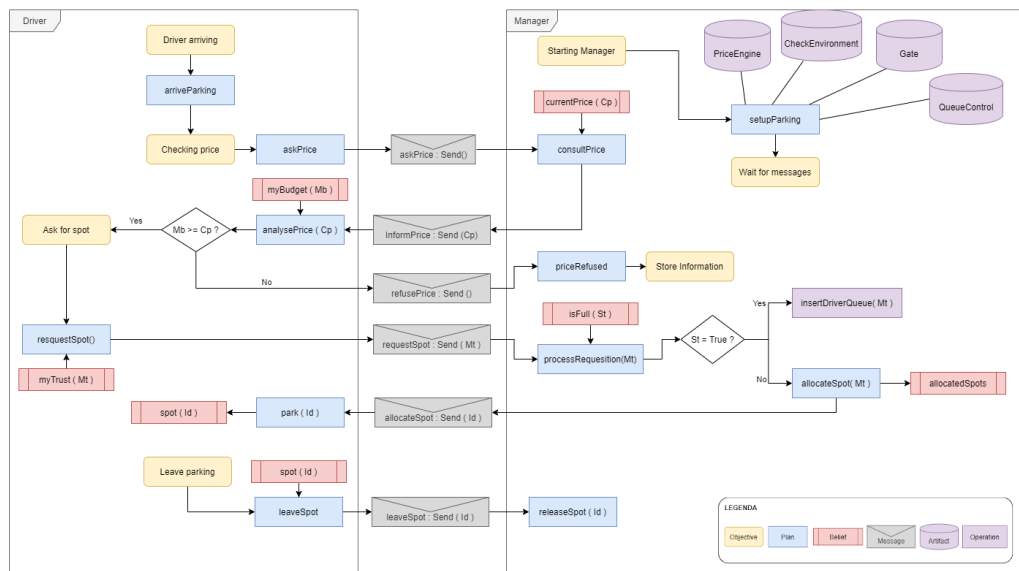


Figura 1. Modelagem do SMA de um EI

A Figura 1 ilustra a modelagem de um EI utilizado como base da implantação do módulo de precificação. Neste SMA centralizado o agente *manager* controla a entrada, alocação e pagamento dos agentes *drivers* interessados em estacionar. Os agentes *driver* encarregados apenas de decidir se aceitam ou não o preço das vagas. A decisão é tomada a partir da comparação entre os créditos disponíveis ao *driver* e o valor das vagas indicado pelo *manager*.

O agente *manager* é um ponto central de gerência de todo o estacionamento. Ele está encarregado de receber, informar e alocar agentes *drivers*. O *manager* também é responsável por decidir quando alterar o preço das vagas de forma dinâmica (após observar o ambiente) ou de forma estática (utilizando planos internos).

O módulo de precificação é encarregado de retornar o novo preço das vagas quando requisitado pelo agente *manager*. O valor é definido a partir das variáveis do ambiente atual e também das regras definidas na configuração do estacionamento (ver Subseção 3.1).

A seguir são descritos os componentes do módulo de precificação: o arquivo de configuração que define as regras seguidas pelo módulo e as operações do módulo que observam a situação do ambiente a fim de atualizar o valor da vaga.

3.1. Regras do Sistema

Parte integral deste módulo de precificação dinâmica encontra-se no arquivo de configuração. Neste sistema foi utilizado um arquivo CSV, contudo o método de processamento do arquivo possibilita utilizar outras extensões de arquivos. No arquivo de configuração são definidas as regras utilizadas para a mudança dos preços das vagas.

A construção deste arquivo deve ser feita da seguinte forma: primeiramente, é criada uma legenda que contém o ID, a *String* e os intervalos das regras. Em seguida, uma tabela contendo as categorias, condições e a porcentagem de mudança do preço.

Na tabela de legenda, a coluna *ID* é usada para identificar o tipo de regras, onde 1 indica uma mudança por demanda; 2 mudança por Clima e 3 mudança por Horário do dia. A coluna *String* representa a palavra usada na tabela de regras para cada *ID*; E a coluna intervalos estabelece os valores usados para escolher a condição de uma categoria.

Na tabela de regras, a primeira coluna contém as categorias que definem o tipo de regra. A segunda coluna contém as condições que especificam uma categoria. A terceira coluna contém os valores de cada combinação categoria vs. condição.

Tabela 1. Conf 1

1	ID	String	Intervalos
2	1	Demanda	0-0.3-0.6-0.8-1
3	3	Horario	0-6-12-15-18-24
4			
5	Categoria	Condição	Valor
6	Horario	Madrugada	-0.30
7	Horario	Manha	0.10
8	Horario	MeioDia	0.30
9	Horario	Tarde	0.10
10	Horario	Noite	-0.15
11	Demanda	Baixa	0.0
12	Demanda	Normal	0.15
13	Demanda	Alta	0.30
14	Demanda	Enorme	0.60

Tabela 2. Conf 3

1	ID	String	Intervalos
2	1	Dmd	0-0.4-0.7-1
3	2	Tmp	0-1-2
4	3	Hr	0-12-18-24
5			
6	Categoria	Condição	Valor
7	Tmp	Ensolarado	-0.1
8	Tmp	Nublado	0
9	Tmp	Chuvoso	0.05
10	Dmd	Baixa	0.0
11	Dmd	Media	0.20
12	Dmd	Alta	0.50
13	Hr	Manha	0.0
14	Hr	Tarde	0.05
15	Hr	Noite	-0.05

A tabela de legendas na Tabela 1 e Tabela 2 são utilizadas como meio de identificar quais tipos de regras serão usadas no estacionamento e como elas são definidas. Na Tabela 1 por exemplo, serão utilizadas regras de demanda, clima e horário, enquanto na Tabela 2 serão usados apenas demanda e horário. Os intervalos são usados como meio de conectar a situação do ambiente com as condições disponíveis na tabela de regras. Os valores desses intervalos depende do contexto das regras. Os intervalos de demanda representam porcentagens e precisam conter os valores de 0 a 1 (0 a 100%). Os de clima são identificadores de cada clima observado pelo estacionamento. Os de horário equivalem as horas do dia e devem conter de 00:00 a 24:00 que representam o começo e fim do dia respectivamente.

Cada linha na segunda tabela contém uma regra em que o estacionamento deseja realizar uma mudança do preço. Usando a coluna de intervalos da legenda, no exemplo 1 uma demanda de 65% se encontra entre 0.6 e 1 e logo se encaixa com a condição de *Alta*. A combinação *Demanda Alta*, encontrada na décima quinta linha do exemplo 1, resulta em um aumento de 20% no valor das vagas. Porém, a mesma demanda de 65% encontra-se entre 0.4 e 0.7 na legenda do exemplo 2 e se encaixa na condição *Media*. A combinação *Dmd Media* mantém o valor do preço.

Como o valor base das vagas não muda durante a simulação, o maior e o menor valor possível são determinados pelo arquivo de configuração. Por exemplo, ao usar a Tabela 1 como configuração, o maior preço é encontrado ao ocorrer a combinação de regras *Horario:MeioDia/Demanda:Alta/Temporal:Tempestate*. Usando a mesma tabela, o menor preço ocorre com o uso das regras *Horario:Madrugada/Demanda:Baixa/Temporal:Sol*.

3.2. Variáveis do sistema

Durante a simulação do EI, os valores de algumas variáveis modificam relativamente o funcionamento do sistema. O agente *driver*, quando criado, possui quatro variáveis que determinam suas ações. Essas variáveis recebem valores aleatórios dentro de um intervalo pré-definido. Segue as descrições de suas funções e valores possíveis:

- *timeToArrive*: Determina o tempo em que o agente *driver* irá chegar ao estacionamento após ser criado. Definido entre 0 e 24h (tempo de simulação);
- *timeToSpend*: Determina o tempo em que o agente *driver* irá passar estacionado em sua vaga. Definido entre 30 minutos e 2 dias (tempo de simulação);
- *myTrust*: Variável que determina o valor de confiança usada como meio de priorização de motoristas ao negociar com o agente *manager*. Definido entre 0 e 1000;
- *myBudget*: Valor dos créditos disponíveis ao agente *driver* para requerir uma vaga no estacionamento. Definido entre 5 e 20 créditos.

O agente *manager* possui quatro variáveis. Destas, duas são usadas no gerenciamento do estacionamento e duas são utilizadas para o controle do preço das vagas:

- *nSpotsMax*: Determina o número total de vagas no estacionamento. Este valor é definido estaticamente e altera o espaço total disponível do estacionamento;
- *nSpotsUsed*: Determina a quantidade de vagas ocupadas. A partir desse valor, que é iniciado em zero, e o total de vagas é possível encontrar a ocupação do estacionamento;
- *basePrice*: Valor base do preço das vagas. Esta é a variável usada durante os cálculos de alteração de preço. O seu valor é estático durante a simulação;
- *currentPrice*: Determina o preço atual das vagas. Este valor começa igual ao valor base e é alterado após execuções do módulo de precificação.

3.3. Módulo de Precificação

O módulo de precificação utiliza as regras estabelecidas pelo estacionamento como base para realizar as mudanças de preço. Essas regras determinam quais são as características dinâmicas do ambiente que o estacionamento deseja observar e quanto cada uma delas afeta o preço. Esse módulo é encarregado de receber a circunstância presente do ambiente e modificar o preço das vagas de acordo.

3.3.1. Planos do Agente Driver

O agente *driver* representa ações de um motorista procurando por uma vaga. Neste agente são contidos planos de execução para chegar a um estacionamento, confirmar preço em relação a um orçamento pessoal, estacionar e deixar o estacionamento. Os planos a seguir são parcialmente modificados a partir do sistema desenvolvido em [Castro et al. 2017].

- *setupDriver*: Criação das crenças do agente *driver* e definição de quanto tempo o agente *driver* irá demorar até chegar ao estacionamento. Nesta crenças são contidas: o tempo que o agente deseja ficar no estacionamento, a confiança do agente e o seus créditos disponíveis;

- *arriveParking*: Plano que indica ao sistema que o agente *driver* chegou no estacionamento. É definido como objetivo inicial durante a criação do agente. Ao final do plano é definido como objetivo requisitar o preço das vagas;
- *askPrice*: Agente *driver* envia uma mensagem para o agente *manager* solicitando o preço atual das vagas do estacionamento;
- *confirmPrice*: Plano é iniciado quando o agente *manager* retorna uma mensagem contendo o preço das vagas. Neste plano é realizada a consideração desse valor em relação aos créditos disponíveis do agente. Caso esse valor seja aceito, o agente define como próximo objetivo pedir uma vaga. Caso contrário, o agente envia uma mensagem para o agente *manager* recusando o preço;
- *requestSpot*: Plano é iniciado caso o agente tenha aceitado o preço de uma vaga do estacionamento. Neste plano é enviada uma mensagem ao agente *manager* pedindo por uma vaga disponível;
- *park*: Plano iniciado quando o agente *manager* retorna uma mensagem contendo a vaga alocada. Neste plano o agente *driver* cria uma crença com os dados desta vaga indicada pelo agente *manager*;
- *leaveSpot*: Plano iniciado quando o tempo de estadia no estacionamento termina. Neste plano o agente *driver* remove a crença contendo os dados da sua vaga e envia uma mensagem para o agente *manager* que esta se retirando do estacionamento.

3.3.2. Planos do Agente Manager

O *manager* controla todos os aspectos do EI. Nele é implementada a recepção, alocação e remoção de agentes *drivers*. Ainda são utilizados os artefatos que realizam a verificação do ambiente e alteração do preço das vagas. Os planos de gerenciamento das vagas foram parcialmente modificados a partir do sistema em [Castro et al. 2017]. Os novos planos são aqueles que lidam com a verificação do ambiente e alteração do preço.

- *setupParking*: Inicialização dos artefatos utilizados pelo agente *manager* e criação de suas crenças. Os artefatos em questão são a cancela de entrada do estacionamento, o controle da fila de espera, o sistema de checagem do ambiente, o módulo de precificação e o criador de *log* de eventos.
- *consultPrice*: Plano iniciado após receber uma mensagem do agente *driver* pedindo pelo preço atual das vagas. O preço atual está contido nas crenças do agente *manager* e uma mensagem com esse valor é retornado ao agente *driver*.
- *processRequisition*: Plano iniciado após receber uma requisição de vaga através de uma mensagem vinda do agente *driver*. Neste plano é verificado se existe vaga disponível no estacionamento e desta forma determinar qual é o próximo objetivo do agente *manager*.
- *priceRefused*: Plano iniciado quando o agente *manager* receber uma mensagem de rejeição do preço de um agente *driver*. Como sistema desenvolvido não possui contra-proposta, o *driver* vai embora do sistema após a rejeição do preço.
- *allocateSpot*: Neste plano é feita uma busca por uma vaga disponível e os dados dessa são alocados ao agente *driver* em questão.
- *insertDriverQueue*: Caso o estacionamento esteja lotado este plano insere o agente *drive* em uma fila de espera utilizando o artefato de controle dessa fila.

Posteriormente esse *driver* irá ser alocado em uma vaga quando uma estiver disponível.

- *releaseSpot*: Plano iniciado quando um agente *driver* estacionado decide sair do estacionamento. Esse agente é retirado da vaga e removido do sistema.
- *checkEnvironment*: Neste plano é acionado o artefato de verificação do ambiente e o módulo de precificação. Caso tenha ocorrido alguma mudança no ambiente, prevista no arquivo de configuração (Seção 3.1) o preço das vagas será atualizado de acordo com a nova situação.
- *addPromotion*: Plano que atualiza o preço sem verificar o ambiente. Neste plano o valor das vagas é alterado de acordo com o evento de promoção previamente definido no arquivo de configuração.

3.3.3. Processo de Precificação

Na Simulação implementada o SMA possui artefatos capazes de observar o ambiente e ativar o módulo de precificação, quando ocorrem mudanças.

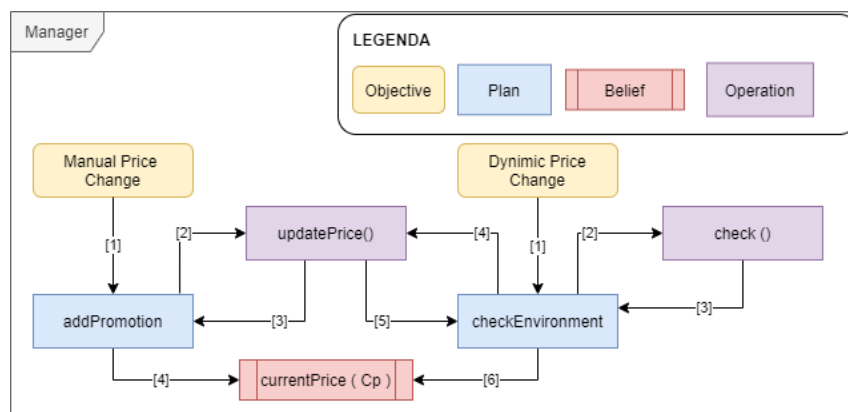


Figura 2. Utilização do módulo de precificação pelo agente *manager*

A Figura 2 ilustra o processo de utilização do módulo de precificação. Em um momento pré-programado no sistema, o agente *manager* inicia o processo de alterar o preço. Dependendo do que causou o início do processo, o *manager* decide entre uma mudança manual ou dinâmica do preço.

Uma mudança manual ocorre quando o agente *manager* deseja mudar o preço a partir de uma crença própria. Na Figura 2, por exemplo, o agente decide acionar o plano *addPromotion* que inclui uma promoção no valor das vagas. Neste caminho é chamada a operação *updatePrice* usando como regra *Evento:Promocao* (ver Tabela 4). Após consultar o arquivo de configuração, essa operação retorna o novo preço e o agente *manager* atualiza o valor das vagas.

Uma mudança dinâmica ocorre quando o agente *manager* deseja verificar se a situação atual do ambiente teve alguma alteração, desde a última checagem. Para isso é iniciado o plano *checkEnvironment*, chamando o artefato que observa o ambiente. A operação *check* do artefato retorna a condição do ambiente e o procedimento utiliza esses dados na operação *updatePrice*. Uma possível condição do ambiente resulta na regra

Demanda:Normal/Temporal:Sol (ver Tabela 4). Essa operação verifica o arquivo de configuração, retorna o novo valor e o agente *manager* atualiza o preço das vagas.

4. Resultados

Aqui são descritos os resultados das simulações do EI. Foram realizadas testes de cenários com e sem a utilização do módulo de precificação.

4.1. Cenários de Testes

Os cenários de teste foram definidos por combinações de números máximos de vagas disponíveis com arquivos de configurações distintos. A partir da combinação entre 50, 100 e 200 vagas, o uso de cinco configurações e ainda desconsiderando o módulo de precificação, no total foram realizadas 18 simulações. Cada cenário foi testado durante 30 dias de simulações. Para cada dia foi criado uma média de 30 *drivers* como meio de simular um número aleatório de clientes do estacionamento. Os cinco cenários desenvolvidos podem ser vistos nas tabelas 1, 2, 3, 4 e 5.

Tabela 4. Conf. 4

ID	String	Intervalos
1	Demanda	0-0.3-0.6-1
2	Temporal	0-1-2-3
3	Horario	0-6-12-15-19-24
Categoria	Condição	Valor
Horario	Madrugada	-0.20
Horario	Manha	0.0
Horario	MeioDia	0.10
Horario	Tarde	0.0
Horario	Noite	-0.10
Evento	Promocao	-0.50
Demanda	Baixa	-0.20
Demanda	Normal	0.0
Demanda	Alta	0.30
Temporal	Sol	-0.1
Temporal	Nublado	0
Temporal	Chuva	0.05
Temporal	Tempestade	0.1

Tabela 3. Conf. 2

ID	String	Intervalos
1	Demanda	0-0.3-0.8-1
3	Horario	0-6-12-18-24
Categoria	Condição	Valor
Horario	Madrugada	-0.30
Horario	Manha	0.10
Horario	Tarde	0.10
Horario	Noite	-0.15
Demanda	Baixa	0.0
Demanda	Normal	0.15
Demanda	Alta	0.30

Tabela 5. Conf. 5

ID	String	Intervalos
1	Dmd	0-0.4-0.7-1
Categoria	Condição	Valor
Dmd	Baixa	0.0
Dmd	Media	0.2
Dmd	Alta	0.50

4.2. Resultados obtidos

Tendo obtido os resultados das simulações foi feita a avaliação do impacto da inclusão de preço dinâmico em relação a ocupação, considerando uma comparação com o sistema sem o módulo. Também foi verificado se o total de créditos recebidos permaneceram equivalentes ou pioraram entre cenários. Cada simulação resulta em um *log* de eventos que contém todas as operações do sistema durante sua execução.

A Figura 3 mostra, como exemplo, dois pedaços de um *log* de eventos. Os resultados na tabela 6 foram obtidos a partir desses *logs* de eventos da simulação. Em cada ação dos agentes, foi salvo em um arquivo *csv* contendo as informações e valores dessa ação. Posteriormente foram encontrados os dados mostrados na tabela 6 usando os valores guardados no *log* e calculando as porcentagem de ocupação e valor das vagas ao decorrer do tempo.

As figuras 4, 5 e 6 mostram os gráficos contendo a taxa de ocupação e preço das vagas do estacionamento em cada dia da simulação. Os valores no eixo y são porcentagens para os dados de taxa de ocupação e créditos para os dados de preço atual.

Sector	SpotId	Agent	Description	Date(D/M)	Time(H:M:S)	Rule	CurrentPrice(R\$)	OccupationRate(%)
?	?	driver_18	AskedPrice	1/1	1:45:36	?	5	0
?	?	driver_18	RequestedSpot	1/1	1:45:36	?	5	0
B	49	driver_18	SpotAllocated	1/1	1:45:36	?	5	1
?	?	driver_2	AskedPrice	1/1	2:28:52	?	5	1
?	?	driver_2	RequestedSpot	1/1	2:28:52	?	5	1
A	49	driver_2	SpotAllocated	1/1	2:28:52	?	5	2
?	?	driver_1	AskedPrice	1/1	4:49:24	?	5	2
?	?	driver_1	RequestedSpot	1/1	4:49:24	?	5	2
B	48	driver_1	SpotAllocated	1/1	4:49:24	?	5	3
B	39	driver_44	LeavingSpot	2/1	13:1:51	?	3.5	28
?	?	driver_42	AskedPrice	2/1	14:9:25	?	3.5	28
?	?	driver_42	RequestedSpot	2/1	14:9:25	?	3.5	28
B	39	driver_42	SpotAllocated	2/1	14:9:25	?	3.5	29
B	47	driver_15	LeavingSpot	2/1	14:9:25	?	3.5	28
?	?	?	PriceChanged	2/1	15:1:6	Demanda:Normal/ Temporal:Sol	4.5	28
?	?	driver_31	AskedPrice	2/1	15:1:6	?	4.5	28
?	?	driver_31	RequestedSpot	2/1	15:1:6	?	4.5	28
B	47	driver_31	SpotAllocated	2/1	15:1:6	?	4.5	29
B	39	driver_42	LeavingSpot	2/1	15:1:6	?	4.5	28

Figura 3. Exemplo de log de eventos

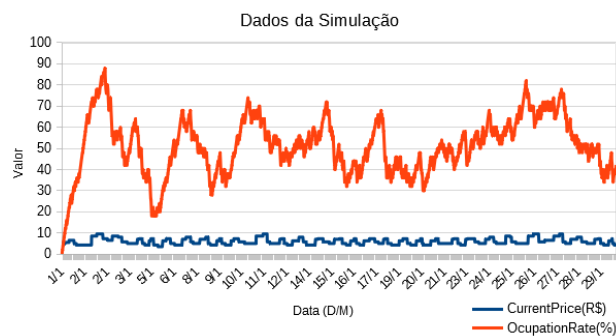


Figura 4. Gráficos de resultados para 50 vagas usando a configuração 1

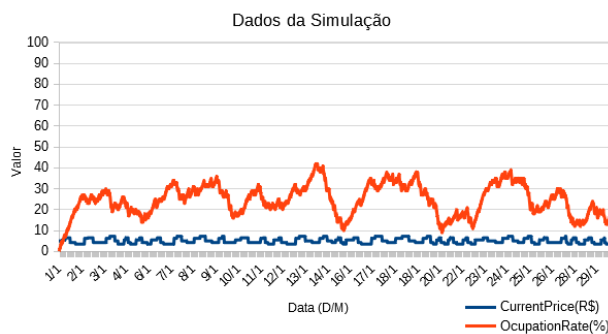


Figura 5. Gráficos de resultados para 100 vagas usando a configuração 1

Os gráficos mostram que os estacionamentos com um maior número de vagas disponíveis tiveram menores taxas de ocupação. Isso ocorrer devido apenas 30 veículos em média entrarem nestes estacionamentos em cada dia. Essa conclusão não é encontrada somente com o uso da configuração 1, como é visto na Tabela 6.

4.3. Discussão dos resultados

A Tabela 6 mostra os resultados obtidos a partir dos logs (ver Figura 3) criados durante as simulações realizadas. Os dados de interesse nestes testes foram o crédito total recebido

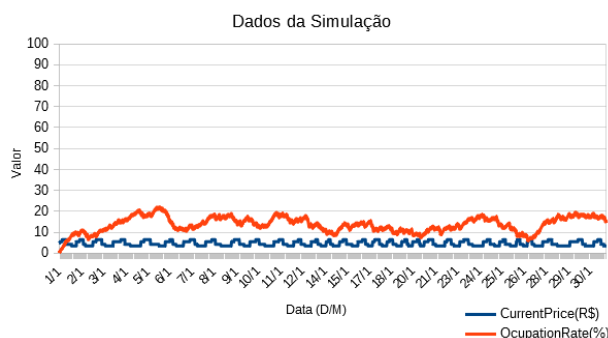


Figura 6. Gráficos de resultados para 200 vagas usando a configuração 1

e a ocupação média durante um mês de simulação.

Tabela 6. Resultados

Módulo	# de Vagas	Crédito Recebido	Ocupação Média
Sem módulo	50 vagas	14220	53,6878%
	100 vagas	14285	23,6682%
	200 vagas	14960	13,4934%
Configuração 1	50 vagas	17747	50,7936%
	100 vagas	14193,75	24,5152%
	200 vagas	13967	13,6244%
Configuração 2	50 vagas	15360,25	49,1860%
	100 vagas	15212,75	26,6203%
	200 vagas	14494,75	13,0428%
Configuração 3	50 vagas	18396,50	47,3752%
	100 vagas	14484,50	24,4134%
	200 vagas	14159,25	14,2786%
Configuração 4	50 vagas	14652,50	44,5488%
	100 vagas	11619,75	24,8263%
	200 vagas	10811,75	13,2768%
Configuração 5	50 vagas	16402,50	45,0777%
	100 vagas	16220	26,0010%
	200 vagas	16085	13,8306%

Em todos os cenários que usaram módulo de precificação, aumentar o número de vagas resultou em uma redução no total de créditos recebidos. Isso é devido às regras de demanda e que o número de veículos não aumenta com o tamanho do estacionamento. Já que os cenários sem módulo não consideram demanda, mais vagas implicam apenas maior número de possíveis clientes e consequentemente mais créditos recebidos.

Apesar da redução dos créditos recebidos em função do aumento do número de vagas, a maioria dos cenários que utilizaram módulos tiveram um valor total maior de créditos recebidos. A clara exceção ocorre na configuração 4 que possui um desconto muito maior em situação de demanda baixa (ver Tabela 4).

Excepcionalmente, quanto maiores as porcentagens de alteração do preço nas regras, maior a diferença nos créditos totais recebidos. Nas configurações dos testes, as regras de demanda possuíram as maiores porcentagens. Especialmente as configurações

1 e 3 (ver Tabela 1 e 2), que tiveram um aumento de 60% e 50% em situações de demanda alta, tiveram os maiores créditos recebidos no cenário de 50 vagas.

É adequado dizer que, se as regras de horário tivessem maiores porcentagens, o número de vagas influenciaria menos o total de créditos recebidos em cada cenário.

5. Conclusão

Em grandes cidades, estacionamentos são fundamentais para a circulação de veículos e redução de congestionamento. A pesquisa de novas tecnologia na área vem a indicar que o uso de sistemas inteligentes é uma alternativa para tentar resolver problemas de trânsito. Neste artigo é apresentada a implementação de um módulo de precificação com o intuito de viabilizar tarifas dinâmicas em um EI. Os resultados da simulação mostram que, a utilização de um módulo de precificação pode acarretar em maiores créditos recebidos como também menores. A ganho total depende de como o módulo foi criado e em onde foram inseridas as maiores porcentagens.

Este trabalho é uma primeira tentativa de desenvolvimento de um sistema de ajustes de preços dinâmicos. Futuramente é necessário a realização de testes explorando uma diversidade maior de cenários, com variáveis mais complexas em relação ao ambiente. Ainda é preciso igualmente aprimorar a simulação para obter resultados mais próximos à realidade. Como trabalho futuro é possível citar uma pesquisa de previsão de ocupação. A partir de mineração de dados armazenados em simulações passadas é possível extrair padrões e realizar recomendações sobre preços futuros das vagas. Esse processo de previsão (na taxa de ocupação) pode ajudar na implementação de um sistema de reserva de vagas com o módulo de precificação dinâmico.

Referências

- Botelho, P. W., Borges, A. P., and Alves, G. V. (2019). Proposta de implantação de um sistema ciber-físico para um smart parking baseado em agentes inteligentes.
- Castro, L. F. S. D., Alves, G. V., and Borges, A. P. (2017). Using trust degree for agents in order to assign spots in a Smart Parking. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 6(2):45–55.
- Chen, M. K. and Sheldon, M. (2015). Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform. *UCLA Anderson*. URL: <https://www.anderson.ucla.edu>.
- Di Nocera, D., Di Napoli, C., and Rossi, S. (2014). A social-aware smart parking application. In *WOA*.
- Ducheiko, F. F., André, P. B., and Gleifer, V. A. (2018). Implementação de Modelo de Raciocínio e Protocolo de Negociação para um Estacionamento Inteligente com Mecanismo de Negociação Descentralizado. *Revista Junior de Iniciação Científica em Ciências Exatas e Engenharia*, 1(19):25–32.
- JACAMO (2011). The jacamo approach.
- Pierce, G. and Shoup, D. (2013). Getting the prices right: an evaluation of pricing parking by demand in san francisco. *Journal of the American Planning Association*, 79(1):67–81.

- Polycarpou, E., Lambrinos, L., and Protopapadakis, E. (2013). Smart parking solutions for urban areas. In *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–6. IEEE.
- Shoup, D. C. (2006). Cruising for parking. *Transport Policy*, 13(6):479–486.
- Tian, Q., Yang, L., Wang, C., and Huang, H.-J. (2018). Dynamic pricing for reservation-based parking system: A revenue management method. *Transport Policy*, 71:36–44.
- Williams, K. (2018). Dynamic airline pricing and seat availability.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

Integrando Sistemas Multi-Agentes Embarcados, Simulação Urbana e Aplicações de IoT *

Lucas Fernando Souza de Castro¹, Fabian Cesar P. B. Manoel²
Vinícius Souza de Jesus², Carlos Eduardo Pantoja²
André Pinz Borges³, Gleifer Vaz Alves³

¹Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brasil

²Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)
Rio de Janeiro – RJ – Brasil


³Programa de Pós-Graduação em Ciência da Computação
Universidade Tecnológica Federal do Paraná (UTFPR)
Ponta Grossa, PR – Brasil

lucas.castro@ic.unicamp.br, {souza.vdj, fabiancpbm}@gmail.com

pantoja@cefet-rj.br, {apborges, gleifer}@utfpr.edu.br

Abstract. *The development of smart city systems connected to the Internet of Things (IoT) has been the goal of several multi-agent system research works. However, few works show how to deploy and make the connection among the employed systems. This paper proposes an approach towards the integration of a MAS using JaCaMo framework along with the Urban Simulation Tool (SUMO), and some IoT applications. The integration presented here is applied in a Smart Parking scenario with real features, where is shown the integration and the connection throughout all layers, from agent level to artifacts, including real environment and simulation, as well as IoT applications. In future works, we intend to establish a methodology which shows how to properly integrate these different applications no matter the applied scenario nor the used tools.*

Resumo. *O desenvolvimento de sistemas para cidades inteligentes conectados com Internet of Things (IoT) tem sido o foco de muitas pesquisas no âmbito de Sistemas Multi-Agentes (SMA). Contudo, poucos trabalhos mostram como realizar a implantação e conexão entre os softwares utilizados nestas diferentes áreas. Assim, este trabalho propõe uma abordagem de integração de um SMA, usando o framework JaCaMo com a ferramenta de Simulação Urbana (SUMO) e aplicações de IoT. A integração apresentada aqui é aplicada em um cenário de Estacionamento Inteligente com característica realista, onde a integração e conexão de todos componentes envolvidos é descrita desde o nível dos agentes e artefatos, passando pelo nível do ambiente real (físico), pelo nível da simulação*

*  O trabalho Integrando Sistemas Multi-Agentes Embarcados, Simulação Urbana e Aplicações de IoT de Lucas Fernando Souza de Castro, Fabian Cesar P. B. Manoel, Vinícius Souza de Jesus, Carlos Eduardo Pantoja, André Pinz Borges, Gleifer Vaz Alves está licenciado com uma Licença Creative Commons - Atribuição-CompartilhaIgual 4.0 Internacional. <http://creativecommons.org/licenses/by-sa/4.0/>

urbana, chegando as aplicações de IoT. Em trabalhos futuros, pretende-se elaborar uma metodologia que mostre como integrar essas diferentes aplicações independentemente dos cenários e das ferramentas utilizadas.

1. Introdução

A Cisco em um de seus relatórios aponta uma estimativa de aproximadamente 50 bilhões de dispositivos para a *Internet of Things* (IoT) em funcionamento a partir de 2022 [Evans 2011]. Além disso, há uma crescente demanda de sistemas computacionais que proporcionem funcionalidades e conectividade para esses dispositivos. O desenvolvimento de tais sistemas deve atender a dinamicidade dos cenários onde os dispositivos para IoT estão localizados, como por exemplo o caso de uma cidade inteligente.

Uma cidade inteligente pode ser definida como uma cidade que provê tecnologias integradas ao seus cidadãos para que tenham qualidade de vida [Albino et al. 2015]. Essa qualidade fornecida à população pode ser diversificada em várias categorias, sendo: governabilidade, segurança, economia, meio ambiente e mobilidade. Como demonstrado em [Neirotti et al. 2014], a questão da mobilidade tem sido uma área de atenção de pesquisa e investimento industrial em cidades inteligentes. Assim, é necessário diante de tais demandas o desenvolvimento de sistemas capazes de fornecer suporte à mobilidade dos cidadãos.

Os Sistemas Multi-Agentes (SMA) são candidatos a fornecer o suporte necessário à mobilidade dos cidadãos, pois são sistemas compostos por múltiplos agentes autônomos, independentes e pro-ativos com raciocínio cognitivo e capacidade de tomada de decisão com habilidade de comunicação entre si, a fim de atingir um objetivo comum ou conflitante [Wooldridge 2000]. Tais características permitem que problemas relacionados à mobilidade em cidades inteligentes possam ser tratados de forma pro-ativa sem interferência ou até conhecimento dos cidadãos, porque agentes inteligentes podem assumir esse papel em benefício do usuário e negociar com outros agentes para tomar as decisões mais pertinentes em dado momento para qualquer serviço. Como em uma cidade inteligente podem existir inúmeros dispositivos capazes de trocar informações, a IoT surge como uma tecnologia essencial na implantação de tais soluções, já que, por definição, IoT é um conjunto de dispositivos interconectados pela Internet. Visto que SMA podem ser explorados tanto de forma simulada (virtual), quanto de forma física, é possível prever dispositivos embarcados com SMA aplicados em aplicações de IoT.

Ao observar a demanda tecnológica requerida pela mobilidade urbana, observa-se a necessidade de um sistema adequado para lidar com situações de inteligência e organização em um ambiente, e sobretudo com suporte da IoT para escalar a resolução do problema. Sendo assim, o objetivo deste trabalho é descrever uma abordagem de integração de diferentes aplicações, envolvendo SMA, Simulação Urbana, Sistemas Embarcados e a Camada IoT de forma que seja possível criar soluções que utilizem tais tecnologias em conjunto para resolver questões de mobilidade, heterogeneidade e baixo acoplamento entre as tecnologias. Consequentemente, essa integração visa prover tanto uma solução capaz de ser aplicada em diferentes cenários e aplicações (heterogeneidade), quanto uma solução que proporcione a possibilidade de utilizar somente uma parte - subconjunta - da integração como uma solução (baixo acoplamento entre as tecnologias).

Para implantação do trabalho utiliza-se o *framework* JaCaMo

[Boissier et al. 2013], uma composição de outros três *frameworks*, Jason, CArTAgO e Moise. Jason [Bordini et al. 2007] interpreta uma linguagem orientada a agentes denominada *AgentSpeak* em Java para a programação de agentes BDI (*Belief-Desire-Intention*). Já o CArTAgO é baseado no modelo *Agents & Artifacts (A&A)* o qual permite o desenvolvimento da camada ambientes do SMA e provê a integração entre os agentes e artefatos de um SMA [Ricci et al. 2006]. Por fim, o Moise implementa um modelo organizacional para SMA que tem fundamentação nos conceitos de agrupamento, comportamento e objetivos [Boissier et al. 2013].

O restante deste artigo está organizado da seguinte maneira. Seção 2 apresenta a abordagem para integração dos sistemas. Na Seção 3 é descrita a integração entre os agentes e artefatos do SMA com a simulação. Após, a Seção 4 detalha a integração de agentes e artefatos com o meio físico. Enquanto, na Seção 5, a integração de agentes com a IoT é descrita. Por fim, a Seção 6 discute as considerações finais.

2. Abordagem para Integração de Sistemas

Nesta seção é apresentada a abordagem para integração dos diferentes Sistemas: SMA, Simulação Urbana, Sistemas Embarcados e a camada de IoT. A abordagem utilizada descreve como desenvolver um SMA embarcado para aplicações de IoT utilizando o framework JaCaMo e integrando diversas ferramentas de forma a facilitar a aplicação e visualização de tais sistemas, inclusive em diferentes domínios de aplicação. A abordagem empregada aqui permite a separação da estrutura de um sistema em níveis onde as responsabilidades do software, hardware e IoT são programadas separadamente para facilitar a integração dependendo da solução que está sendo criada e de quais níveis serão utilizados. A proposta de integração de sistema é composta por quatro níveis bem definidos: Agentes, Ambiente, IoT e nível da Simulação, conforme ilustrado na Fig. 1.

A camada de nível do *agente* é responsável pela criação dos agentes do SMA com uso da linguagem Jason. Os agentes possuem duas extensões específicas: uma para o interfaceamento de hardware usando o ARGO [Pantoja et al. 2016], e uma outra para comunicação e transporte de agentes usando a IoT. Esta camada de nível relaciona-se com o nível de ambiente através dos artefatos disponibilizados pelo CArTAgO [Ricci et al. 2006], com o nível de simulação para que seja possível simular os agentes utilizando a ferramenta SUMO [Krajzewicz et al. 2002], com o nível de ambiente real diretamente através do agente estendido ARGO, e por fim, com o nível IoT para escalar as aplicações com agentes.

O nível do *ambiente* é dividido entre os artefatos que podem ser criados pelo CArTAgO e pelo meio físico, onde as plataformas de hardware com sensores e atuadores podem interagir com o mundo físico. Assim é possível criar os *Artefatos Simulados*, aqueles que mantêm informações apenas em nível de software, e os *Artefatos Físicos*, que realizam a interface de Hardware por meio da utilização de uma Interface Serial. Ambos artefatos podem co-existir em uma solução desenvolvida usando a abordagem proposta, uma vez que as tecnologias utilizadas em cada nível são independentes. Os *Artefatos Físicos* podem também se relacionar com o nível de IoT via de conexões MQTT com o *KonkerLabs* e o *Node-Red Application*.

No nível da IoT, é utilizado o *ContextNet*, um *middleware* para IoT que tem capacidade para lidar com diversas características de sistemas distribuídos como conecti-

vidade, escalabilidade e comunicabilidade, para comunicação entre agentes. Os agentes comunicadores da abordagem proposta são capazes de conectar-se ao *ContextNet* para trocar mensagens ou realizar o transporte de agentes de um SMA embarcado em um dispositivo para um outro SMA de um dispositivo diferente.

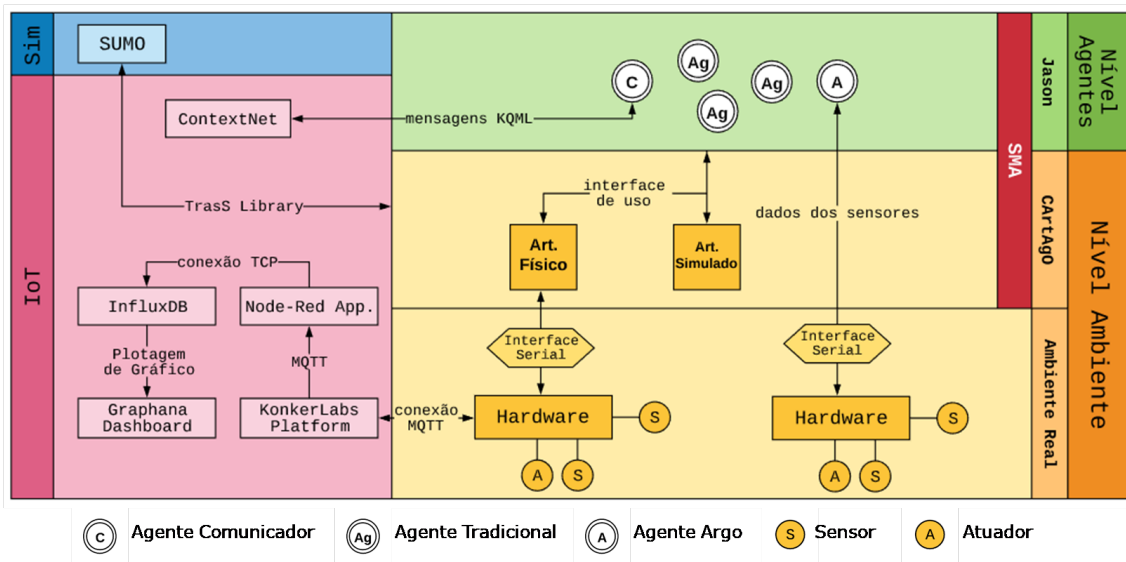


Figura 1. Abordagem para Integração de Sistemas: visão geral

Dessa forma, é possível criar dispositivos utilizando a abordagem de agentes para atuar fisicamente em um Sistema Ciber-Físico e aplicações de IoT. Pode-se definir um Dispositivo como um componente composto de um SMA embarcado utilizando o Jason e o CArTAgO, capaz de interfacear sensores e controladores conectados a hardwares micro-controlados (ATMEGA, PIC, Arduino, etc.) e de se conectar a uma rede IoT para troca de informações.

3. Integração: Agentes, Ambiente (Artefatos) e Simulação

A utilização do *framework* JaCaMo, além de proporcionar três camadas de desenvolvimento para os SMAs: agentes, artefatos e social, possui também a capacidade de interligação com diferentes ferramentas, por exemplo: simulação, sistemas embarcados e IoT. Tal capacidade é derivada da versatilidade do *framework* em prover por meio da utilização de artefatos de ambiente do CArTAgO. Essa seção descreve como é realizada a conexão do JaCaMo e SUMO.

Para ilustrar essa integração é utilizado um SMA desenvolvido no JaCaMo para alocação e negociação de vagas em Estacionamento Inteligentes. Na simulação com o SUMO, utiliza-se uma rede¹ da UTFPR - Campus Ponta Grossa com enfoque no estacionamento de visitantes conforme ilustrado na Figura 2.

A integração do SMA com o SUMO foi dividida em duas camadas: agentes e artefatos. A camada dos agentes compreende a programação dos agentes na linguagem Jason e suas interações sociais. Enquanto a segunda camada apresenta os artefatos desenvolvidos em CArTAgO e suas interações com os agentes. Ambas as camadas são descritas em detalhes na seção seguinte.

¹O termo rede no SUMO (ou *network*) é utilizado para denotar um mapa a ser empregado na simulação.

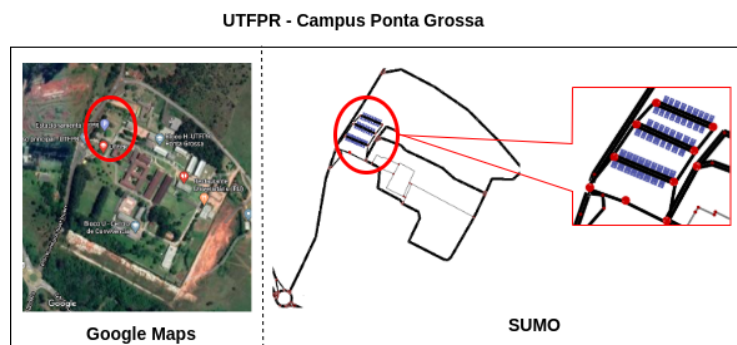


Figura 2. UTFPR (Ponta Grossa) - Mapa e Representação no SUMO

3.1. Camada dos Agentes

Ao todo são três grupos de agentes que compõem o SMA: *builder*, *pspace*, *driver*. Na Figura 3 são ilustrados os agentes, crenças e interações (mensagens).

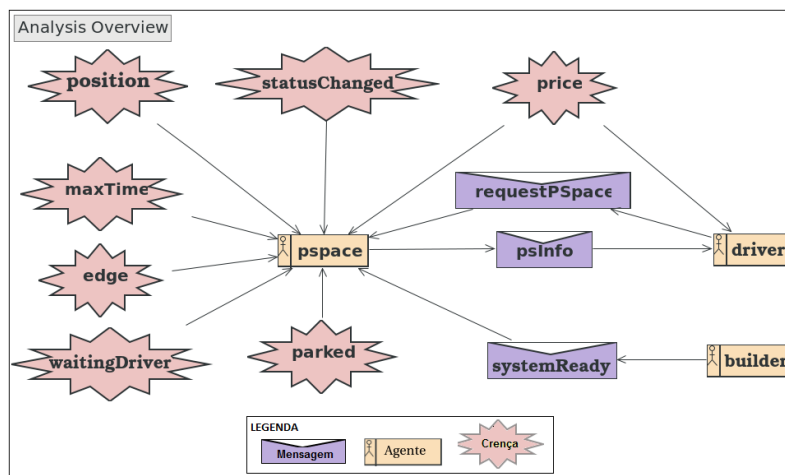


Figura 3. Diagrama Prometheus - Agentes: crenças e mensagens

O grupo de **agentes** possui os seguintes membros:

- *builder*: responsável pela instanciação de todos os agentes *pspace*, *workspaces* e artefatos do Cartago;
- *pspace*: responsável por representar a vaga do estacionamento, tanto de forma física, como virtual (no SUMO, conforme Fig. 2).
- *driver*: representante do motorista que deseja uma vaga de estacionamento.

O agente *pspace* possui as seguintes crenças:

- *price*: preço da vaga por hora utilizada;
- *statusChanged*: valor que corresponde a variação do sensor da vaga (em uso ou livre);
- *position*: posição da via da rede no SUMO (0 - ida / 1 - volta);
- *maxTime*: tempo máximo de permanência que um motorista pode utilizar a vaga. Caso esse tempo seja extrapolado, haverá alterações no valor de preço;
- *edge*: valor que identifica a aresta (rua) da rede do SUMO em que a vaga está posicionada. Por padrão o SUMO fornece os valores pré-estabelecidos como (edgez.ID), sendo esses editáveis;

- *waitingDriver*: valor que informa se a vaga está no status *aguardando motorista*. crença é utilizada no caso de alguma vaga ser ocupada sem ser previamente alocada a um determinado motorista. Assim, caso isso ocorra, a vaga foi ocupada de maneira indevida por algum objeto ou motorista não desejado;
- *parked*: valor que indica a presença de um motorista estacionado na vaga que foi alocada ao motorista.

Entre o grupo de agentes as seguintes mensagens podem ser trocadas:

- *requestPSpace*: mensagem enviada pelo agente *driver* ao agente *pspace* informando a requisição da vaga por um preço pré-determinado pela crença *price*.
- *systemReady*: mensagem enviada ao instanciar os agentes *pspace*, *workspaces* e artefatos do CArtaGO. Neste processo o agente *builder* envia uma mensagem aos agentes *pspace* informando que o sistema está em funcionamento. Assim, os agentes *pspace* estão abertos às requisições dos agentes *drivers*.
- *psInfo*: Quando o agente *pspace* aceita uma requisição pela vaga, esse agente deve informar ao agente respectivo *driver* as informações a respeito da vaga alocada.

Observa-se que está fora do escopo deste artigo detalhar o processo de requisição e negociação de uma vaga no estacionamento. Mas, existem trabalhos relacionados que apresentam soluções de SMA específicos para a negociação de vagas em estacionamentos inteligentes, como: [Castro et al. 2017], [Ducheiko et al. 2018] e [Alves et al. 2019].

3.2. Camada dos Artefatos - Conexão com SUMO

Os artefatos descritos nesse cenário compreendem a interligação do SMA com o SUMO, ilustrado na Fig. 4.

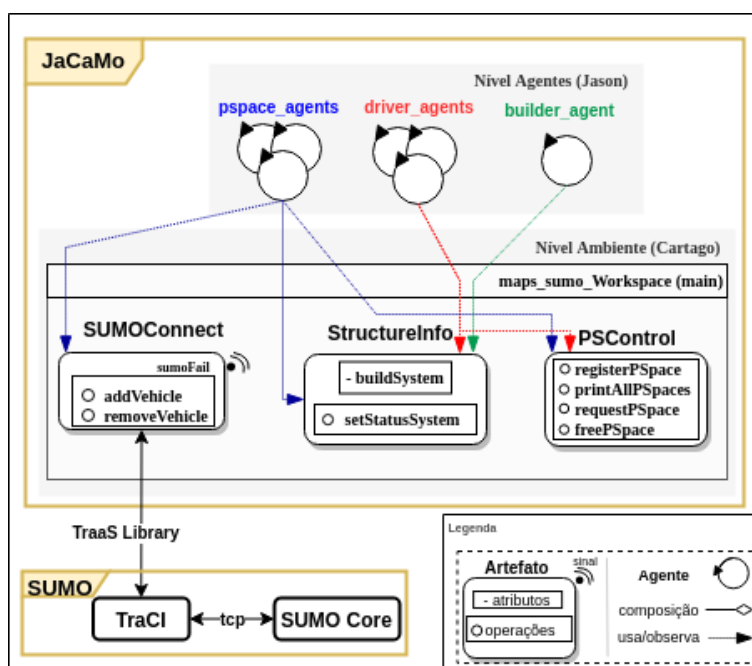


Figura 4. Interação entre JaCaMo e SUMO

Os seguintes artefatos são utilizados no sistema:

- ***SUMOConnect***: Artefato utilizado para a conexão com o SUMO. A conexão com o SUMO é realizada por meio da biblioteca TraaS² [Krumnow 2013]. Há duas operações que os agentes *pspace* podem realizar no artefato *SUMOConnect*.
 - `addVehicle`: método que adiciona veículos na rede do SUMO. Os agentes *driver* do SMA em JaCaMo são considerados como veículos no SUMO, ao passo que os agentes *pspace* são os recursos *parkingArea* do SUMO. Listing 1 apresenta a função `addVehicle`. Os parâmetros da função são o nome do agente (id do veículo do SUMO), o id da estrada do SUMO (*edge*) e a posição da estrada (0 - ida, 1 - volta). A função inicialmente (linha 7) adiciona o veículo na rede do SUMO. Nessa linha ainda tem-se o `routeToParking`, que é previamente definido como uma sequência de estradas usadas pelo veículo para encontrar a vaga destinada a ele. Por fim, a linha 8 define qual vaga o veículo irá ocupar, sendo esta previamente negociada na camada dos agentes.

Listing 1. Função: `addVehicle` - CArtaGO

```

1 public class A_SUMOConnect extends Artifact {
2     SumoTraciConnection conn;
3     @OPERATION
4     public void addVehicle(String agentName, String edge, double position) {
5         conn.do_job_set (Vehicle.add(agentName, "DEFAULT_VEHTYPE", routeToParking,
6             ↪ lane, position, speed, Byte.valueOf("0")));
7         conn.do_job_set (Vehicle.setParkingAreaStop(agentName, parkingAreaName,
8             ↪ duration, timeOut, flag));}

```

- `removeVehicle`: O Código Listing 2 apresenta a função de remoção do veículo da rede no SUMO. Após um agente *driver* finalizar a utilização de uma *pspace* é solicitado que o veículo correspondente no SUMO deixe a *parkingArea* (Linha 4) e inicie a rota para a saída da rede (`routeToLeave`) (Linha 5).

Listing 2. Função: `removeVehicle` - CArtaGO

```

1 @OPERATION
2 public void removeVehicle(String agentName) {
3     conn.do_job_set (Vehicle.resume(agentName));
4     conn.do_job_set (Vehicle.setRoute(agentName, routeToLeave));}

```

Por fim, no *SUMOConnect* há a utilização do sinal `sumoFail` em caso de falha de conexão com o TraCI e problemas com a rede do SUMO.

- ***StructureInfo* e *PSControl***: O artefato *PSControl* é utilizado para registro, alocação, liberação e controle das *pspaces*. Por sua vez, o artefato *StructureInfo* é utilizado pelo agente *builder* para informar aos demais agentes o status do SMA.

4. Integração: Agentes, Artefatos e o Meio Físico

Esta seção apresenta o JaCaMo como um sistema embarcado, o que proporciona a integração de um SMA no Meio Físico. Logo são apresentadas duas formas de conexão:

²Biblioteca baseada em Python que faz a comunicação com o TraCI do SUMO (*Socket* de conexão).

i. integração de um SMA a um meio físico utilizando diretamente a camada de agentes; e ii. integração do meio físico com a camada de ambiente (artefatos) de um SMA. Em ambas as abordagens, uma interface serial será utilizada como ferramenta de conexão com o meio físico. Esta interface é o Javino [Lazarin and Pantoja 2015], um protocolo de comunicação implementado em duas bibliotecas, sendo uma voltada para o *hardware* e a outra para o *software*. A principal característica desta interface serial é a confiabilidade, pois possui um processo de verificação da integridade da mensagem para que não ocorra perda ou qualquer tipo de alteração entre a mensagem enviada e a recebida.

4.1. ARGO: Integração de Agente Jason e Hardware

A integração de Agentes Jason com *hardware* permite desenvolver SMA em ambientes físicos reais. A noção de autonomia, proatividade e colaboração por parte dos agentes de um SMA justificam a necessidade de integrar este sistema a um ambiente real a fim de tratar as imprevisibilidade deste meio físico.

O ARGO [Pantoja et al. 2016] é uma arquitetura customizada de agentes para o *framework* Jason que acrescenta aos agentes a capacidade de controlar microcontroladores, por exemplo, Arduino. Os agentes com essa arquitetura são chamados de agentes ARGO e permitem desenvolver um SMA em um meio físico por meio de protótipos e qualquer dispositivos que utilizam microcontroladores. Os agentes ARGO possuem esta capacidade de controlar microcontroladores, pois na arquitetura deste agente foi incluída a interface de comunicação serial Javino.

Sendo assim, por meio da interface serial Javino, um agente ARGO pode enviar e receber informações de microcontroladores. As informações recebidas são tratadas como percepções que o agente ARGO recebe do ambiente via sensores do *hardware* que o agente está controlando e, portanto, são automaticamente adicionadas como crenças na base de crenças deste agente. Já as informações enviadas para o microcontrolador são tratadas como ações que devem ser realizadas no meio físico via atuadores do *hardware*.

Para utilizar essas novas funcionalidades dos agentes ARGO, foram desenvolvidas quatro novas ações internas (*i.e.*, ações ou comportamentos pré-programados inerente ao agente [Bordini et al. 2007]) exclusivas para fazer o controle de microcontroladores, sendo elas:

- **.port("Porta serial")**: Define qual porta serial o agente irá controlar e, consequentemente, qual microcontrolador o agente irá controlar. Esta ação interna possui um argumento que representa qual a porta serial o agente irá controlar.
- **.perceive(open/block)**: Define se o agente irá perceber ou não o ambiente. Esta ação interna possui um argumento e tem somente duas opções de valores possíveis: *open*, que é utilizado para fazer o agente abrir as percepções e receber informações do *hardware*; e *block*, usada para fechar as percepções do agente e não mais receber informações.
- **.limit(Tempo em milissegundos)**: Define um intervalo de tempo para o agente perceber o ambiente. Esta ação interna possui um argumento que representa o tempo em milissegundos que o agente irá alternar a percepção de aberta para fechada.
- **.act("Ação")**: Define uma ação que o microcontrolador deve realizar. Esta ação interna possui um argumento que representa a ação que deverá ser enviada para o microcontrolador executar.

Além dos agentes ARGO, que permitem desenvolver SMA em ambientes físicos reais, existem também os Artefatos Físicos que permitem integrar artefatos convencionais do *framework* CARtAgO com dispositivos físicos e seus microcontroladores.

4.2. Artefatos Físicos: Integração CARtAgO e Hardware

A arquitetura proposta para o JaCaMo [Boissier et al. 2013] apresenta a diferença entre dois tipos de ambientes: o ambiente interno, aonde estão localizados os artefatos de um SMA, e o ambiente externo, que podem ser oferecidos por cenários simulados ou reais. Ao integrar a camada de ambiente do SMA com o meio físico, é possível desenvolver Sistemas Cyber-Físicos (SCF) usando artefatos.

O uso de artefatos para produzir SCF permite conectar o SMA e o meio físico. Portanto, os agentes do SMA não precisam incluir as responsabilidades de integração em seu ciclo de raciocínio. Conseqüentemente, esta abordagem é necessária para permitir que um agente trabalhe em um meio físico sem necessariamente ocupar todo o seu ciclo de raciocínio com os dados fornecidos por este meio. A vantagem da abordagem proposta é permitir que os agentes possam continuar interagindo com a camada de ambiente localizada dentro do SMA sem que eles precisem tomar conhecimento de que os artefatos estão ligados ao meio físico. O artefato responsável por esta integração com o meio físico é chamado Artefato Físico.

Um Artefato Físico é uma extensão de artefatos convencionais desenvolvida dentro do *framework* CARtAgO e portanto, também possui (1) um conjunto de operações que podem ser realizadas pelos agentes, (2) instruções que descrevem como estes artefatos devem ter suas funcionalidades acessadas, (3) propósito de existência, e (4) as estruturas internas que definem as implementações de suas funcionalidades [Ricci et al. 2006]. Com o CARtAgO, um artefato pode ser programado em linguagem Java, onde podem ser implementados métodos chamados de Operações, que determinam o comportamento do artefato, e também podem ser implementadas instâncias conhecidos como Propriedades Observáveis, que permitem ao artefato notificar o agente de algum evento do ambiente. Um artefato convencional do CARtAgO é considerado como um Artefato Físico quando ele troca mensagens com os dispositivos do meio físico, controlando os atuadores e monitorando os sensores deste dispositivo. Assim, um Artefato Físico pode fornecer ao agente os dados do meio físico através das Propriedades Observáveis, e os agentes podem realizar comandos que interfiram neste meio através das Operações.

A ponte de conexão entre a camada de ambiente e o meio físico é criada utilizando o Javino entre o Artefato Físico e o microcontrolador do dispositivo contido neste meio. Sendo assim, esta interface serial é incorporada como uma ferramenta de suporte aos Artefatos Físicos, possibilitando a comunicação artefato/dispositivo. Com isso, para implementar Artefatos Físicos em um projeto específico, três configurações devem ser feitas por três métodos abstratos:

- ***String definePort()***: retorna o valor da porta de comunicação a ser utilizada para trocar dados com o microcontrolador;
- ***int defineAttemptsAfterFailure()***: retorna o número de vezes que o artefato tentará enviar uma mesma mensagem quando algum erro tiver ocorrido durante a troca de mensagens;

- ***int defineWaitTimeout()***: retorna o tempo de espera em milissegundos entre dois pedidos feitos para o dispositivo;

Quando é necessário implementar as Operações e as Propriedades Observáveis dos Artefatos Físicos, dois métodos podem ser utilizados para realizar a comunicação com o dispositivo físico:

- ***String read()***: retorna uma mensagem enviada pelo microcontrolador do dispositivo, normalmente apresentando as medidas coletadas pelos seus recursos.
- ***void send(String message)***: envia a mensagem passada por parâmetro ao dispositivo, normalmente apresentando um comando para operar os recursos deste dispositivo.

A abordagem de Artefatos Físicos permite que a conexão entre o SMA e o meio físico não ultrapasse o Nível de Ambiente proposto, e com isso, o acoplamento entre as tecnologias é diminuído.

5. Integração: Agentes e Aplicações IoT

Uma vez que SMAs, dispositivos, plataforma e ferramentas IoT possuem diferentes escopos e implementações, é necessário então a utilização de diferentes tecnologias que realizem a interface de comunicação entre eles.

A integração desenvolvida nesse trabalho utiliza diferentes módulos e conexões para essa interface de comunicação seja realizada. A Figura 5 apresenta a integração, onde os módulos, bem como suas conexões e interações sequenciais são descritas abaixo:

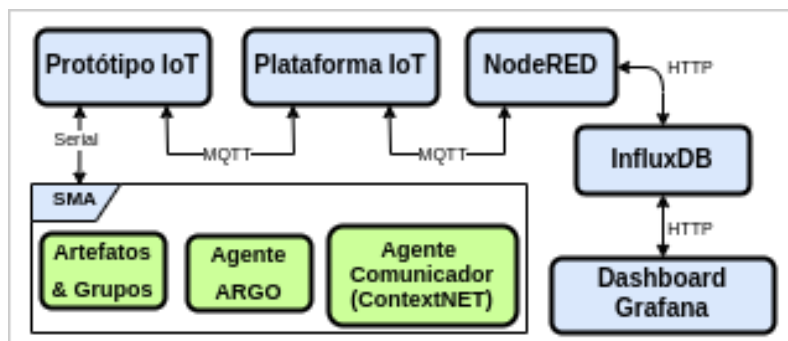


Figura 5. Integração: Protótipo, SMA, plataforma e ferramentas IoT

- **Protótipo IoT:** O protótipo é composto por três itens: NodeMCU 1.0 (ESP 8266), Arduino UNO e um Raspberry Pi 3. Inicialmente o Arduino realiza a leitura e envia os dados dos sensores (ultrasônico de distância) via serial para o NodeMCU. Assim, o NodeMCU gera um pacote JSON e envia via MQTT para a Plataforma IoT. O Raspberry Pi 3 é utilizado na hospedagem dos módulos do NodeRED, InfluxDB e Grafana;
- **SMA:** O SMA é composto por artefatos (desenvolvidos no Cartago), grupos sociais (providos pelo Moise) e agentes, sendo eles:
 - **Agente ARGO:** Estabelece a conexão com o protótipo IoT (Arduino UNO) para a percepção do ambiente físico por meio dos sensores ultrasônicos de distância. Ao obter as percepções, o agente ARGO notifica o protótipo IoT (NodeMCU) e o agente Comunicador caso ocorram mudanças no ambiente (alteração significativa nos valores dos sensores).

- Agente Comunicador (ContextNET): Agente responsável pela comunicação com outros SMAs via rede ContextNET. Com a utilização desse agente, o SMA é capaz de se comunicar com outros SMAs com a finalidade de uma múltipla cooperação entre os sistemas. Por exemplo, uma rede de estacionamentos inteligentes disponibilizados em vários locais de uma mesma cidade. Assim, um único estacionamento poderá prover vagas de n-estacionamentos.
- **Plataforma IoT:** A plataforma IoT visa a integração entre diferentes dispositivos IoT por meio de rotas MQTT (publish/subscribe). Ao receber as mensagens via MQTT do protótipo, a plataforma redireciona as mensagens via MQTT para o NodeRED. A plataforma utilizada é a fornecida pela KonkerLabs, contudo há outras soluções fornecidas pela Eclipse, Google, entre outros;
- **NodeRED:** Utilizado para a decomposição e extração das mensagens JSON provenientes da plataforma IoT e armazenagem no InfluxDB via HTTP. A decomposição e extração é utilizada para a obtenção do valor dos sensores do protótipo;
- **InfluxDB:** Banco de dados de séries de tempo (*time series databases - TSDB*). A utilização desse tipo de banco de dados é devido a sua característica de ter como chave primária um *timestamp*. Assim, a armazenagem dos valores dos sensores é efetuada de acordo com o tempo de medição;
- **Dashboard Grafana:** O *dashboard* do Grafana provê diferentes componentes para a visualização de dados em tempo real. O dashboard acessa os dados dos sensores via HTTP no banco de dados do InfluxDB.

6. Conclusão

Este trabalho apresenta uma abordagem para integração de um SMA desenvolvido em JaCaMo com ferramentas de Simulação Urbana, Sistemas Embarcados e a camada IoT. O objetivo principal não é apenas descrever tal integração, mas também não estabelecer um forte acoplamento entre os níveis de ambiente, agentes, IoT e simulação.

A vantagem desta abordagem está na sua generalização. Ou seja, os níveis propostos são heterogêneos, e portanto permitem que diferentes aplicações utilizem esta abordagem como suporte para integração de sistemas. Além disso, outra vantagem está no baixo acoplamento entre as camadas, o que dá flexibilidade para utilizar tanto a arquitetura completamente integrada, como apenas partes dela. Através deste trabalho problemas que demandem escalabilidade e um sistema inteligente podem ser resolvidos com auxílio da IoT integrada a um SMA. Ainda é possível definir se o tipo de ambiente a ser utilizado será simulado ou físico.

Como trabalhos futuros, busca-se desenvolver uma metodologia de integração como um *framework* para soluções de mobilidade urbana integrado SMA, IoT, Simulação e Sistemas Embarcados. Além disso, é necessário implantar uma aplicação (mais robusta e complexa) no domínio de Cidades Inteligentes, a qual utilize todos os níveis propostos. Com tal aplicação igualmente será possível executar experimentos para avaliar características da metodologia de integração, como: escalabilidade, baixo acoplamento, confiabilidade na troca de mensagens e informações, entre outras. Por fim, pretende-se contemplar a integração da camada de organização social com o Moise (JaCaMo). Assim, será possível estabelecer regras e comportamentos em sociedades de agentes, promovendo a organização social e a criação de grupos sociais de agentes, esses elementos podem representar de forma adequada características da mobilidade urbana.

Referências

- Albino, V., Berardi, U., and Dangelico, R. M. (2015). Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of urban technology*, 22(1):3–21.
- Alves, B. R., Alves, G. V., Borges, A. P., and Leitão, P. (2019). Experimentation of Negotiation Protocols for Consensus Problems in Smart Parking Systems. In Marik, V., Kadera, P., Rzevski, G., Zoitl, A., Anderst-Kotsis, G., Tjoa, A. M., and Khalil, I., editors, *Industrial Applications of Holonic and Multi-Agent Systems*, Lecture Notes in Computer Science, pages 189–202, Cham. Springer International Publishing.
- Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013). Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747–761.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons Ltd.
- Castro, L. F. S. D., Alves, G. V., and Borges, A. P. (2017). Using trust degree for agents in order to assign spots in a Smart Parking. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 6(2):45–55.
- Ducheiko, F. F., André, P. B., and Gleifer, V. A. (2018). Implementação de Modelo de Raciocínio e Protocolo de Negociação para um Estacionamento Inteligente com Mecanismo de Negociação Descentralizado. *Revista Junior de Iniciação Científica em Ciências Exatas e Engenharia*, 1(19):25–32.
- Evans, D. (2011). How the Next Evolution of the Internet of Things Is Changing Everything. *Cisco Internet Business Solutions Group*, page 11.
- Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P. (2002). Sumo (simulation of urban mobility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, pages 183–187.
- Krumnow, M. (2013). Sumo as a service—building up a web service to interact with sumo. In *Simulation of Urban MObility User Conference*, pages 62–70. Springer.
- Lazarin, N. M. and Pantoja, C. E. (2015). A Robotic-Agent Platform for Embedding Software Agents using Raspberry Pi and Arduino Boards. In *9th Software Agents, Environments and Applications School*.
- Neirotti, P., Marco, A. D., Cagliano, A. C., Mangano, G., and Scorrano, F. (2014). Current trends in smart city initiatives: Some stylised facts. *Cities*, 38:25 – 36.
- Pantoja, C. E., Stabile Jr, M. F., Lazarin, N. M., and Sichman, J. S. (2016). Argo: A customized jason architecture for programming embedded robotic agents. *Fourth International Workshop on Engineering Multi-Agent Systems (EMAS 2016)*.
- Ricci, A., Viroli, M., and Omicini, A. (2006). Programming MAS with artifacts. 3862 LNAI:206–221.
- Wooldridge, M. J. (2000). *Reasoning about rational agents*. MIT press.

Simulação do Comportamento de um Agente Inteligente Guiado por Regras de Trânsito *

Vithor Tozetto Ferreira¹, Gleifer Vaz Alves¹

¹Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná (UTFPR)
Rua Doutor Washington Subtil Chueire, 330 - 84017-220 – Ponta Grossa – PR – Brasil

vithorferreira@alunos.utfpr.edu.br, gleifer@utfpr.edu.br


Abstract. *An Autonomous Vehicle (AV), situated in an ordinary urban traffic environment, must act according to local traffic rules, and its controller needs to be able to consider these rules when planning actions. Such controller could be modelled as an intelligent agent, which needs a comprehensible representation of the urban traffic rules. In this paper we present a model of an environment for simulating an agent, which is responsible to control the AV according to a subset of traffic rules related to road junctions.*

Resumo. *Um Veículo Autônomo (VA), presente em um ambiente de tráfego urbano comum, deverá agir de acordo com as regras de trânsito locais, e seu controlador precisa ser capaz de considerar tais regras durante o planejamento de suas ações. Tal controlador, pode ser modelado como um agente inteligente que por sua vez, precisa de uma representação compreensível das regras de trânsito urbano. Neste trabalho é apresentado um modelo para um ambiente para simulação de um agente, que controla o VA de acordo com um fragmento das regras destinadas aos cruzamentos urbanos.*

1. Introdução

Muitas tecnologias desenvolvidas nos últimos anos visam facilitar o cotidiano através da automatização de tarefas. Dentre tais tarefas, o desenvolvimento de veículos autônomos (VA's) será um grande avanço tecnológico para a indústria automotiva [Blankesteijn et al. 2019]. Em grandes centros urbanos motoristas gastam muito tempo no trânsito, e a utilização desta tecnologia cria uma lacuna de tempo para a realização de outras tarefas durante o período de locomoção [Silberg et al. 2012].

Antes da utilização cotidiana de um VA, é necessário garantir a segurança do funcionamento destes veículos, e para o contexto deste artigo, considera-se especificamente a segurança em relação ao comportamento do VA de acordo com as regras de trânsito. Esses veículos já são capazes de, através de várias tecnologias, interagir com segurança e autonomia em diversas situações [Gomes 2014]. Entretanto, esta capacidade não engloba a utilização de regras de trânsito, que são um aspecto essencial do ambiente de

 O trabalho Simulação do Comportamento de um Agente Inteligente Guiado por Regras de Trânsito de Vithor Tozetto Ferreira, Gleifer Vaz Alves está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-Compartilha Igual 4.0 Internacional. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

atuação do veículo e necessárias para garantir a segurança do funcionamento de um VA [Vellinga 2017].

Em [Prakken 2017], é mencionado que a interação com regras de trânsito em geral não é considerada durante a fase de desenvolvimento de um VA. Idealmente, o veículo deve conseguir perceber seu ambiente, considerar suas tarefas e planejar suas ações de forma segura e condizente com as leis de trânsito em vigor no local de utilização do VA, e assim atingir os objetivos de forma autônoma conforme o ambiente de trânsito no qual está inserido. Dessa forma, um VA precisa ser controlado por um sistema computacional com a capacidade de realizar tal tarefa, e tal sistema pode ser representado através de um agente, situado em um ambiente e capaz de realizar ações autônomas para atingir seus objetivos [Wooldridge and Jennings 1995].

Como visto em [Ferreira and Alves 2019a, Ferreira and Alves 2019b], existe uma dificuldade em fazer com que um sistema computacional autônomo utilize as regras de trânsito em seu planejamento de ações. Nestes artigos, é proposta a utilização de uma ontologia para representação de objetos e ações do trânsito, tentando encurtar a distância entre a linguagem natural utilizada em códigos de trânsito [BRASIL 1997, Department for Transport 2017] e uma linguagem compreensível para um agente, levando também em consideração a possibilidade de mobilidade urbana do veículo ao representar mais de um conjunto de leis de trânsito. Esta representação pode ser traduzida para o contexto de um agente inteligente, e pode também ser utilizada no desenvolvimento de um ambiente para tal agente. Neste trabalho é utilizado o Código de Trânsito Britânico como referência das regras devido ao seguinte: **i.** já ser utilizado em trabalhos anteriores que igualmente exploram a relação entre este conjunto de leis e VA's; e **ii.** a existência de uma consulta oficial por parte do governo Britânico para o desenvolvimento de um código de leis de trânsito que considere os VA's [Law Commission - UK 2020].

O agente controlador de um VA é desenvolvido por meio da linguagem de programação de agentes GWENDOLEN. Esta linguagem permite a criação de agentes BDI (*Belief-Desire-Intention*), delimitando suas percepções iniciais, planos e objetivos [Dennis 2017]. Este agente está inserido em um ambiente, desenvolvido em Java, que representa uma abstração de um cruzamento de trânsito, disposto de objetos presentes nas ontologias mencionadas anteriormente. Neste ambiente também são expressas as possíveis ações que o agente pode tomar e quais percepções os objetos do ambiente representam ao agente.

O objetivo específico deste artigo é apresentar um modelo para um ambiente que representa um cenário de cruzamento urbano com alguns objetos presentes em cenários reais, e também a proposta de um agente representando o controlador de um VA que utilize regras de trânsito em suas ações. Este artigo está diretamente relacionado com os seguintes trabalhos [Alves et al. 2018] e [Alves et al. 2019], onde é apresentada uma arquitetura baseada em agentes para representação de um conjunto de regras de trânsito do Reino Unido usando a lógica temporal.

2. Veículos Autônomos e Código de Trânsito

Na literatura da área, geralmente é utilizado o termo direção automatizada, e raramente o termo direção autônoma. O primeiro termo envolve um conjunto de ferramentas que auxiliam na direção, enquanto o segundo refere-se ao estado final da automação,

onde um sistema teria controle total sobre todas as funções de controle do veículo [Herrmann et al. 2018]. Em resumo, um VA é um veículo controlado totalmente por um sistema, sem o auxílio de um motorista. Visando garantir a segurança de suas ações no quesito de tráfego urbano, é necessário que o VA leve em consideração as Regras de Trânsito do local onde está situado.

Conforme mencionado por Prakken [Prakken 2017] e reforçado por Alves et. al [Alves et al. 2018], existe uma lacuna no desenvolvimento de VA's no que diz respeito aos seguintes aspectos: i.) A implantação de um VA considera adequadamente as regras de trânsito? ii.) O comportamento de um VA no tráfego urbano dá-se em acordo com as regras de trânsito? iii.) É necessário em alguma instância alterar e adaptar as regras de trânsito para o adequado comportamento de um VA? Esses aspectos reforçam a necessidade de representar o conhecimento das regras de trânsito para que sejam utilizadas pelo sistema controlador do VA.

Essa lacuna é perceptível ao se aprofundar na idealização de um comportamento adequado em relação as regras de trânsito. Os VA's que são testados atualmente conseguem, por exemplo, perceber a informação que um semáforo transmite. O veículo identifica que o semáforo está no sinal vermelho e logo irá parar, assim como avançará caso o sinal esteja verde. Mas, quando se considera um sinal amarelo, o comportamento em relação a essa informação difere entre motoristas humanos. Alguns aceleram e avançam, enquanto outros diminuem a velocidade e param. No código britânico [Department for Transport 2017], o comportamento em relação ao sinal amarelo é delimitado pela Regra **175**, onde um motorista só pode avançar um sinal amarelo caso já tenha ultrapassado a linha de parada ou acredite que possa causar uma colisão ao parar bruscamente. Nesse caso, a interação do VA não seria influenciada apenas pela informação do semáforo, sendo necessário que o controlador avalie a situação e decida sobre como agir. Para garantir a segurança dessa ação o VA, além de conhecer o ambiente em que está situado, precisa igualmente considerar a Regra de Trânsito **175**.

De acordo com [BRASIL 1997], se configura como trânsito a movimentação e imobilização de veículos, pessoas e animais nas vias terrestres. Cada país, dentro de seus territórios, delimitam um conjunto de regras destinadas a controlar e garantir a segurança do tráfego. É esperado que todo usuário presente no trânsito esteja ciente destas regras, e que as cumpra visando manter a harmonia do ambiente de tráfego urbano. Portanto, para os VA's estarem totalmente inseridos no trânsito, é necessária sua adequação as regras de circulação de seus ambientes de atuação.

Para o desenvolvimento deste trabalho são consideradas um conjunto de regras de trânsito referentes a cruzamentos urbanos (*Road Junctions*). Esse conjunto é formado pelas regras **170** à **183** do Código de Trânsito Britânico [Department for Transport 2017]. Essas regras descrevem o comportamento esperado dos motoristas, quais objetos ou usuários devem ser observados para a realização de manobras, e também como proceder em diferentes cenários referentes aos cruzamentos urbanos.

3. Ambiente de Simulação

Como descrito por [Wooldridge and Jennings 1995], um agente é uma entidade situada em certo ambiente. Dessa forma, para desenvolver um agente também é necessária a existência de um ambiente, onde o agente irá receber percepções e interagir com objetos

para realizar suas ações. Neste artigo, o ambiente funciona como uma abstração de um cenário real de trânsito urbano: um cruzamento entre duas vias, contendo artefatos presentes em cenários reais, como semáforos e pedestres. Este ambiente permite a execução de um agente controlador de um VA, capaz de interagir com objetos de trânsito de acordo com as leis de trânsito referentes a cruzamentos urbanos.

Para o desenvolvimento do ambiente está sendo utilizado o *framework* MCAPL (*Model Checking Agent Programming Language*) [Dennis et al. 2012], que possui ferramentas para a criação de interpretadores de linguagens de agentes. O MCAPL também pode ser utilizado para realizar a verificação formal do funcionamento de um agente, permitindo verificar se o agente se comporta corretamente. Porém, esta etapa não foi contemplada neste trabalho. O *framework* é composto pelo AIL (*Agent Infrastructure Layer*), utilizado para a criação dos interpretadores, e o *Model Checker* AJPF (*Agent Java Pathfinder*) [Dennis 2018]. Na distribuição do *framework* são inclusas algumas linguagens de programação implementadas no AIL, sendo utilizada neste trabalho a linguagem GWENDOLEN.

O ambiente é executado em conjunto com o agente GWENDOLEN, e nele estão contidos os objetos com que o agente pode interagir. Este ambiente é totalmente abstrato, sendo utilizado somente como um espaço virtual para permitir a execução do agente. Na próxima etapa deste trabalho será criada uma representação visual, que permitirá observar o funcionamento do agente. A Fig. 1 ilustra o modelo deste ambiente de simulação.

Este ambiente consiste de uma classe Java, estendida do ambiente padrão da linguagem GWENDOLEN. O espaço físico é representado como uma grade quadriculada, e os objetos são posicionados em coordenadas desta grade. O agente move o veículo uma posição de cada vez. No ambiente também existe uma variável incrementada a cada ação do veículo, utilizada como uma representação simples da passagem de tempo no ambiente, permitindo que semáforos mudem de estado ou pedestres atravessem a via. Cada incremento nesta variável é considerado como um turno, e em cada turno o VA pode realizar uma ação de movimento ou esperar durante o período do turno. Após o agente realizar a sua ação o turno acaba, a variável é incrementada e os objetos do ambiente são atualizados de acordo. O ambiente não possui um limite de turnos, já que o VA pode esperar por um período indeterminado enquanto não acreditar que é seguro avançar em seu percurso.

Em [Ferreira and Alves 2019a, Ferreira and Alves 2019b] são demonstradas ontologias representando objetos e ações presentes em um ambiente de trânsito. Nestes trabalhos, as informações foram extraídas a partir da análise de fragmentos referentes a cruzamentos urbanos presentes nos Códigos de Trânsito Brasileiro [BRASIL 1997] e do Reino Unido [Department for Transport 2017], já que foi considerada durante o desenvolvimento das ontologias a possibilidade de um VA transitar entre dois territórios que possuam legislações de trânsito distintas, e que o controlador do veículo precise adaptar-se ao novo conjunto de regras durante seu percurso. A partir dessas ontologias, foi possível delimitar quais artefatos devem estar presentes no ambiente para permitir o desenvolvimento de um agente que considere e obedeça as leis de trânsito em suas ações.

Os objetos presentes no ambiente atualmente e representados na Fig. 1, considerando o ambiente como uma matriz de células, são os seguintes: as vias por onde o VA

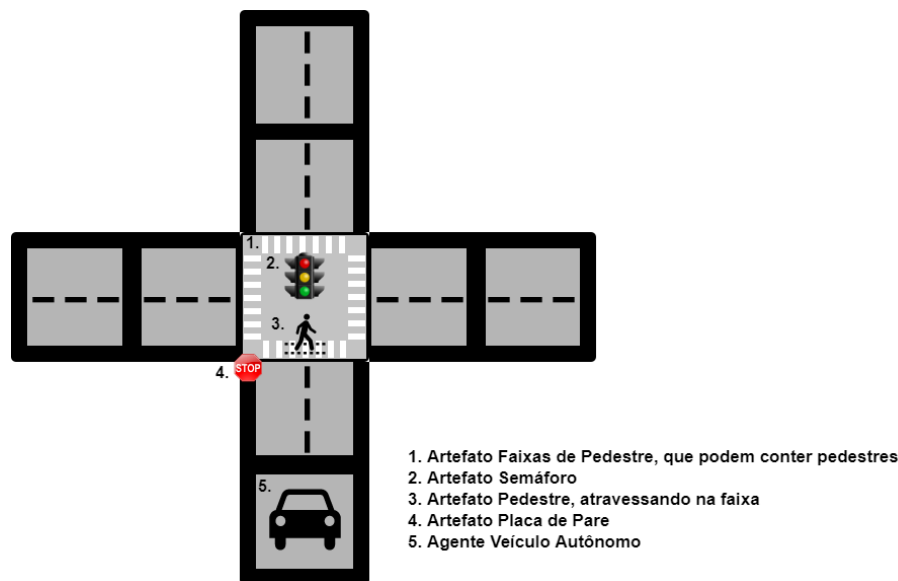


Figura 1. Modelo do ambiente de simulação

pode trafegar, cada uma sendo uma célula com uma coordenada (X, Y) ; o VA, que se move pelas posições das vias; uma placa de Pare; um semáforo, que muda de estado com a passagem de tempo do ambiente; uma faixa de pedestre, que simula um pedestre atravessando a via; o cruzamento, centralizado na interseção das vias. O agente pode executar as seguintes ações: mover, deslocando o veículo; esperar, onde o agente espera uma unidade de tempo devido a alguma condição do ambiente, como a existência de uma placa de Pare; verificar caminho, onde o agente deve averiguar se existem e quais objetos estão na direção que está seguindo; parar, onde o agente imobiliza o veículo.

4. Agente Controlador do VA

A palavra agente é utilizada na computação como um conceito abrangente, com diferentes perspectivas sobre o que é exatamente um agente de acordo com cada autor. Um agente pode significar simplesmente um *script*, executado após uma certa condição, mas também pode se referir a um programa sofisticado capaz de raciocinar sobre diferentes informações para atingir objetivos complexos [Rao and Wooldridge 1999]. Porém, entre diversas definições de agentes, existem conceitos comuns, principalmente a ideia de que um agente deve ser capaz de realizar tarefas de forma autônoma. Dessa forma, foi desenvolvido através da linguagem GWENDOLEN um agente, visando representar a entidade computacional no controle de um VA capaz de comportar-se conforme as regras de trânsito. Este agente está inserido exclusivamente em situações referentes a cruzamentos urbanos, e deve lidar com percepções e ações nestas situações em específico.

Inicialmente, o agente possui as crenças da sua posição atual e de uma posição final, criando um caminho que utiliza o cruzamento. O objetivo inicial do agente é atravessar o cruzamento e chegar na posição final. O agente deve percorrer o trajeto, identificando quais regras devem ser seguidas de acordo com as percepções que recebe dos objetos posicionados no ambiente. Essas percepções indicam ao agente a presença de um certo objeto em determinada posição (X, Y) na grade, e o agente deve então determinar quais planos devem ser utilizados para interagir corretamente com esses objetos.

O conhecimento das regras de trânsito pelo agente torna-se viável através dos planos que o mesmo possui, criados a partir de regras presentes no Código de Trânsito do Reino Unido [Department for Transport 2017]. Esses planos delimitam as condições para o agente mover-se ou esperar de acordo com percepções do ambiente. Por exemplo, considerando a regra 171 do Código de Trânsito do Reino Unido:

“You MUST stop behind the line at a junction with a ‘Stop’ sign and a solid white line across the road. Wait for a safe gap in the traffic before you move off.”

Essa regra indica que um veículo deve parar antes de entrar em um cruzamento, caso exista uma placa de Pare (ver Fig. 2). Os objetos *solid white line across the road* e *safe gap* não serão considerados neste exemplo. Mas, igualmente devem ser representado na ontologia das regras de trânsito. Essa regra é (parcialmente) descrita para o agente através do seguinte plano:

```
+!rule171 [perform] :
{ B stop_sign_at(X,Y), B at(X,Y), ~B stopped(X,Y) } <-
stop, wait(stopsign), +stopped(X,Y);
```

O plano `rule171` é utilizado para que o agente cumpra a regra 171 do Código de Trânsito Britânico. Esse plano é utilizado quando as seguintes condições são satisfeitas:

- o agente acredita que existe uma placa de pare na posição (X, Y) (essa percepção é originada quando o agente percebe a placa em seu caminho, como mostrado na Fig. 2 a);
- o agente acredita que está em (X, Y) ;
- o agente ainda não parou em (X, Y) .

Quando estas condições são satisfeitas, o agente executa o corpo do plano:

- parar no local que está, já que acredita que está na posição indicada pela placa de Pare;
- esperar um turno, com o veículo parado;
- adicionar a crença `stopped(X, Y)`, informação que o agente acredita ser verdadeira e indica que já parou no local necessário.

Assim, o agente seguiu um plano para agir de acordo com a regra de trânsito. Nesse cenário, o agente voltará a verificar o ambiente, e quando acreditar que não existe nenhum objeto ou regra impedindo seu deslocamento (*e.g.* um pedestre atravessando a via), o agente continuará o percurso (ver Fig. 2 c).

5. Considerações Finais

Em trabalhos futuros, as propostas apresentadas neste artigo serão estendidas, visando obter conclusões referentes a questões presentes na literatura de VA's, como aquelas presentes em [Prakken 2017, Vellinga 2017]: quais meios podem ser utilizados para a representação das regras de trânsito para o contexto de um VA; quais são as limitações de um VA no que se refere a compreensão e execução das regras de trânsito; e quais mudanças podem ser necessárias na legislação de trânsito para a adequação do uso de VA's.

A proposta deste artigo é demonstrar o estado inicial do desenvolvimento de um agente e um ambiente de simulação. O agente, modelado como o sistema controlador de

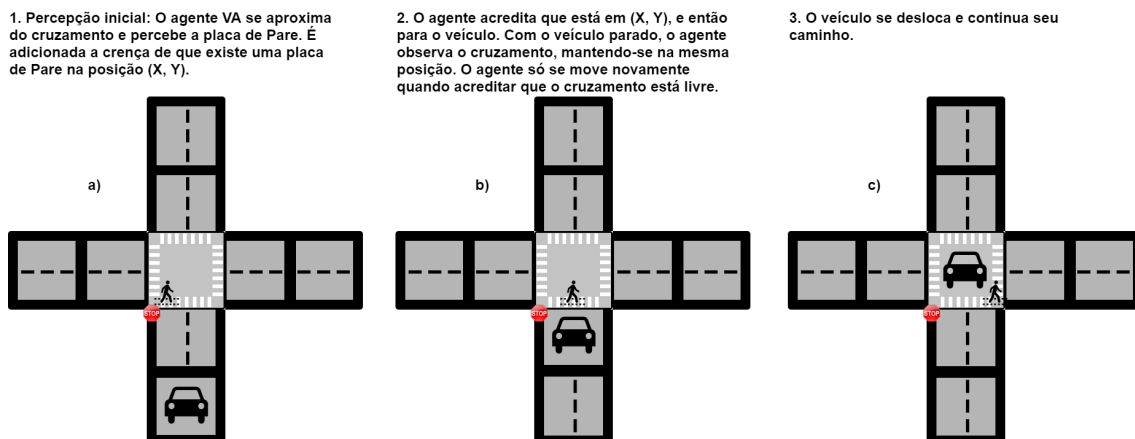


Figura 2. Representação da sequência de ações do agente no ambiente

um VA, possui uma representação das regras de trânsito, e deve utilizá-las em suas ações. O ambiente, é um cenário representando um cruzamento urbano, contendo em si objetos percebidos pelo agente. O ambiente foi criado para suprir a necessidade que um agente possui em existir um ambiente para sua atuação. A partir do ambiente mostrado neste artigo, podem ser criados novos cenários, com a adição de novos objetos, permitindo o teste de novos fragmentos de regras de trânsito.

Como trabalho futuro, pretende-se expandir o ambiente, com a adição de novos objetos para a simulação de outros cenários, permitindo o teste de diferentes regras de trânsito, incluindo a possibilidade de outros Códigos de Trânsito também serem representados. O agente também receberá novos planos, de forma a cumprir as regras de trânsito ao interagir com mais objetos em diferentes cenários. Também será possível obter informações conclusivas sobre o funcionamento do agente ao realizar a verificação formal de seu comportamento, através da ferramenta de *Model Checking* AJPF (*Agent Java Pathfinder*) [Dennis et al. 2012].

Referências

- Alves, G., Dennis, L., and Fisher, M. (2019). An Agent-Based Architecture supported by Temporal Logic for representing the Rules of the Road on Autonomous Vehicles. In Luckcuck, M., Farrell, M., and Fisher, M., editors, *FMAS Workshop Pre-Proceedings 2019*, volume 1, pages 41–48, Porto, Portugal. FMAS - A satellite workshop of Formal Methods 2019, Porto, Portugal.
- Alves, G. V., Dennis, L., and Fisher, M. (2018). Formalisation of the Rules of the Road for embedding into an Autonomous Vehicle Agent. In *International Workshop on Verification and Validation of Autonomous Systems*, pages 1–2, Oxford, UK.
- Blankesteijn, M., Jong, F. D., and Bossink, B. (2019). Closed-open innovation strategy for autonomous vehicle development. *International Journal of Automotive Technology and Management*, 19(1-2):74–103.
- BRASIL, D. (1997). Código brasileiro de trânsito. http://www.planalto.gov.br/ccivil_03/LEIS/L9503.htm.

- Dennis, L. A. (2017). Gwendolen Semantics: 2017. Technical Report ULCS-17-001, University of Liverpool, Department of Computer Science.
- Dennis, L. A. (2018). The mcapl framework including the agent infrastructure layer and agent java pathfinder. *The Journal of Open Source Software*.
- Dennis, L. A., Fisher, M., Webster, M. P., and Bordini, R. H. (2012). Model checking agent programming languages. *Automated Software Engineering*, 19(1):5–63.
- Department for Transport, U. (2017). Using the road (159 to 203) - The Highway Code - Guidance - GOV.UK.
- Ferreira, V. T. and Alves, G. (2019a). Desenvolvimento de uma ontologia para representação de objetos e ações do código de trânsito em um veículo autônomo. In *Anais do III Workshop de Pesquisa em Computação dos Campos Gerais*, volume 3, Ponta Grossa, Paraná. UTFPR Campus Ponta Grossa.
- Ferreira, V. T. and Alves, G. (2019b). Representação de Objetos do Código de Trânsito Através de Uma Ontologia Para Aplicação em um Veículo Autônomo. In *13th Workshop-School on Agents, Environments, and Applications (WESAAC 2019)*, Florianópolis. Editora UFSC.
- Gomes, L. (2014). Hidden obstacles for google’s self-driving cars. *MIT Technology Review*.
- Herrmann, A., Brenner, W., and Stadler, R. (2018). *Autonomous driving: how the driverless revolution will change the world*. Emerald Publishing, Bingley North America Japan India Malaysia China, first edition. OCLC: 1031123857.
- Law Commission - UK (2020). Automated Vehicles: Summary of the Analysis of Responses to the Preliminary Consultation Paper. <https://www.lawcom.gov.uk/project/automated-vehicles/>.
- Prakken, H. (2017). On the problem of making autonomous vehicles conform to traffic law. *Artificial Intelligence and Law*, 25(3):341–363.
- Rao, A. S. and Wooldridge, M. (1999). Foundations of rational agency. In *Foundations of rational agency*, pages 1–10. Springer.
- Silberg, G., Wallace, R., Matuszak, G., Plessers, J., Brower, C., and Subramanian, D. (2012). Self-driving cars: The next revolution. *White paper, KPMG LLP & Center of Automotive Research*, page 36.
- Vellinga, N. E. (2017). From the testing to the deployment of self-driving cars: legal challenges to policymakers on the road ahead. *Computer Law & Security Review*, 33(6):847–863.
- Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2):115–152.

Desenvolvimento de Aplicação para um Estacionamento Inteligente via Computação em Nuvem baseada em Agentes e Sistema Ciber-Físico *

Pedro W. Botelho¹, Gleifer Vaz Alves¹, Paulo Leitão², André Pinz Borges¹

¹Departamento Acadêmico de Informática –
Universidade Tecnológica Federal do Paraná (UTFPR)
Rua Doutor Washington Subtil Chueire, 330 - 84017-220 – Ponta Grossa – PR – Brasil


²Research Centre in Digitalization and Intelligent Robotics (CeDRI)
Instituto Politécnico de Bragança
Campus de Santa Apolónia, 5300-253 – Bragança – Portugal

pbotelho@alunos.utfpr.edu.br, pleitao@ipb.pt

{gleifer, apborges}@utfpr.edu.br

Abstract. *Systems for managing and allocating parking spaces aim to reduce urban traffic and improve the occupancy of parking spaces. Some applications used, in isolated forms, Multi-Agent Systems (MAS), Internet of Things (IoT) and Cloud Computing (CC). Integrating these technologies is critical to build smart parking applications. This article presents an implementation of an architecture that integrates MAS technologies for the allocation of vacancies, IoT for detecting cars and a cloud based database for storing information. The integration took place through a cloud API communicating a mobile application to the MAS, the physical device and the database which was proved to be efficient to communicate the devices.*

Resumo. *Sistemas para gerenciamento e alocação de vagas de estacionamento são desenvolvidos para reduzir tráfego urbano e melhorar a ocupação de estacionamentos. Aplicações desenvolvidas utilizam, de maneira isolada, Sistemas Multi-Agentes (SMA), Internet das Coisas (IoT), e Computação em Nuvem (CN). Integrar tais tecnologias é fundamental para a criação de aplicativos inteligentes para estacionamentos. Aqui é apresentada a implementação de uma arquitetura que integra as tecnologias de SMA para alocação de vagas, IoT para detecção de carros e um banco de dados na nuvem para o armazenamento de informações. A integração deu-se por meio de uma API comunicando um aplicativo móvel ao SMA, ao dispositivo físico e ao banco de dados, a qual se mostrou eficiente para comunicar os dispositivos.*

*  O trabalho Desenvolvimento de Aplicação para um Estacionamento Inteligente via Computação em Nuvem baseada em Agentes e Sistema Ciber-Físico de Pedro Warmling Botelho, Gleifer Vaz Alves, Paulo Leitão, André Pinz Borges está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

1. Introdução

Problemas de trânsito são encontrados em cidades dos mais diversos portes, desde pequenas até grandes cidades. Estudos mostram que cerca de um terço dos veículos circulando em uma cidade são motoristas em busca de estacionamento [Jazdi 2014]. O método mais comum de busca por vagas é manual, sem auxílio de sistemas computacionais. A solução menos eficiente para esse problema é aumentar o número de vagas, assim os motoristas terão maiores possibilidades de encontrar um estacionamento. Contudo, há métodos inteligentes e, conseqüentemente, mais eficientes para atenuar ou até resolver o problema, como os estacionamentos inteligentes [Geng and Cassandras 2013].

Estacionamento Inteligente (EI) é um sistema de estacionamento que auxilia os motoristas a encontrar vagas utilizando sensores que detectam se há ou não um veículo e então o direcionam para a vaga [Hassoune et al. 2016]. O objetivo desses sistemas é garantir que o motorista encontre uma vaga para estacionar no local desejado rapidamente. Assim, o motorista pode reservar uma vaga antes mesmo de entrar no veículo [Castro et al. 2017].

No projeto *Smart Parking* (SP) desenvolvido em parceria pelo Instituto Politécnico de Bragança (IPB) com a Universidade Tecnológica Federal do Paraná campus Ponta Grossa (UTFPR-PG), diferentes abordagens foram implementadas neste domínio, como a negociação de vagas em um estacionamento inteligente [Ducheiko et al. 2018] [Alves et al. 2019]. O projeto, além da negociação, aborda também a implementação de um estacionamento inteligente como um todo, contendo sistemas físicos, também já implementados [Botelho et al. 2019] [Sakurada et al. 2019]. Porém, o projeto ainda não conta com uma aplicação que o usuário possa utilizar para fazer a reserva de vagas e também não conta com um meio de integrar todos estes componentes, sendo estes itens propostos aqui.

Para solucionar o problema dos estacionamentos inteligentes, foi adotado, pelo projeto, uma abordagem baseada em agentes devido a capacidade de inteligência distribuída dos agentes e pela escalabilidade oferecida [Wooldridge 2009]. A arquitetura estabelece dois tipos de agentes: o agente motorista, o qual representa a pessoa que dirige o veículo e o agente vaga que representa uma vaga de estacionamento. Neste contexto, os agentes negociam uma vaga até chegar em um consenso e só então o motorista pode se deslocar ao estacionamento.

Desse modo, com o número da vaga já de posse do motorista, é necessário um sistema físico para realizar o controle da vaga e dos setores do estacionamento, como detectar quando um carro é estacionado. Porém, atualmente no projeto, ainda não é possível fazer com que uma requisição do usuário seja transformada em uma negociação pelo Sistema Multi-Agente (SMA) e, conseqüentemente, no ato de detectar o carro estacionado na vaga.

Para alcançar tal objetivo, é necessário implementar uma interface que permita ao motorista requisitar uma vaga e integrar esta interface ao SMA e a parte ciber-física. Para isso, neste trabalho é implementado um aplicativo móvel e uma solução em nuvem para permitir que os componentes de hardware troquem mensagens e realizem o controle das vagas do estacionamento com apoio de tecnologia em nuvem. Cabe destacar que o foco do artigo não está no processo de negociação, uma vez que foram utilizados mecanismos

de negociação simples, com valores iniciais aleatório e sem a preocupação com tempo de negociação.

O restante deste artigo está organizado da seguinte forma. Na seção 2 são descritos alguns trabalhos relacionados ao tema de Sistemas Multi-Agentes, plataformas de hardware e computação em nuvem. Na seção 3, tem-se a descrição da arquitetura implementada no artigo e como cada componente se comporta dentro dela. A seção 4 apresenta os resultados obtidos e a seção 5 as considerações finais a respeito do desenvolvimento do trabalho.

2. Trabalhos Relacionados

Nesta seção descreve-se brevemente alguns trabalhos desenvolvidos pelo grupo de pesquisa e alguns que relacionam estacionamentos inteligentes com componentes físicos e com computação em nuvem.

Em [Ducheiko et al. 2018], é proposto um modelo de raciocínio e protocolo de negociação para alocação de vagas em um estacionamento utilizando um mecanismo de negociação descentralizado. Neste mecanismo, existe apenas um tipo de agente no sistema, o agente motorista, o qual pode assumir o papel de vendedor (quando deixa a vaga) ou comprador (quando requisita a vaga).

O trabalho de [Pham et al. 2015] propõe um sistema de estacionamento que auxilia o usuário a encontrar automaticamente, sem o uso de agentes, uma vaga livre calculando o preço com base na distância do usuário e o número total de vagas gratuitas em cada estacionamento. Este artigo utiliza o conceito da nuvem de forma implícita para realizar a integração entre vários estacionamentos espalhados pela cidade, assim como um aplicativo móvel que permite ao usuário requisitar uma vaga.

Em [Alves et al. 2019] é apresentado um estudo a respeito de protocolos de negociação para problemas de consenso em estacionamentos inteligentes. Algumas estratégias de negociação, como o protocolo *Contract Net*, o leilão inglês e o leilão holandês foram implementados utilizando o *framework* JADE para evidenciar a melhor estratégia para aplicar em um SMA aplicado à estacionamentos inteligentes.

Em [Khanna and R. Anand] um sistema de estacionamento é proposto também a computação em nuvem para integrar os componentes de IoT que detecta a presença de um veículo e um dispositivo móvel para realizar as requisições do usuário. O sistema fornece informações em tempo real do estado da vaga e permite ao motorista escolher uma vaga livre apenas, sem passar por um processo de negociação.

Em [Sakurada et al. 2019] uma arquitetura ciber-física para estacionamentos inteligentes é apresentada, porém, esta arquitetura considera um estacionamento para bicicletas. Cada vaga possui um Raspberry Pi com um agente embarcado, o qual controla um dispositivo físico capaz de liberar o acesso a vaga da bicicleta.

Como foi observado, os trabalhos citados utilizam de maneira isolada diferentes componentes que fazem parte de um estacionamento inteligente como um todo. A partir disso, uma integração entre todos estes componentes é necessária para a construção do aplicativo móvel para estacionamentos inteligentes.

Um trabalho externo ao grupo de pesquisa, desenvolvido por [Bar 2019], apre-

senta uma revisão da literatura sobre tecnologias (e.g. componentes, sensores e softwares) empregadas em Estacionamentos Inteligentes. Os autores identificam os tipos mais utilizados de cada componente e destacam as tendências de uso, além de analisar vários trabalhos cujo foco está no desenvolvimento de softwares para Estacionamentos Inteligente. Este trabalho é utilizado como referência quanto às tecnologias e componentes que podem ser utilizados neste artigo.

3. Metodologia

A computação em nuvem está em uma crescente nos últimos anos e vem auxiliando empresas a se tornarem mais responsivas e ágeis, podendo ser acessada de qualquer lugar e em qualquer dispositivo [Senyo et al. 2018]. Nesse contexto, a arquitetura desenvolvida neste artigo é baseada na possibilidade de conectar componentes de modo independente. Estes componentes vão desde sensores capazes de informar se há a presença de um veículo em uma vaga (e.g. infra-vermelho), dispositivo de comunicação destes sensores com Raspberry Pi (e.g. ESP), Arduino, aplicativo móvel até serviços em nuvem (e.g. *Amazon Web Services - AWS*).

Uma visão geral da arquitetura e componentes utilizados é apresentada na Figura 1. A arquitetura contém: (i) um aplicativo móvel para a conexão do motorista com o estacionamento; (ii) um módulo de computação em nuvem (AWS) usado para replicar mensagens trocadas entre os componentes e armazenar todas as informações do estacionamento; (iii) um SMA responsável pela negociação da vaga requisitada pelo motorista e (iv) a plataforma de hardware é responsável por detectar a presença de um veículo na vaga.

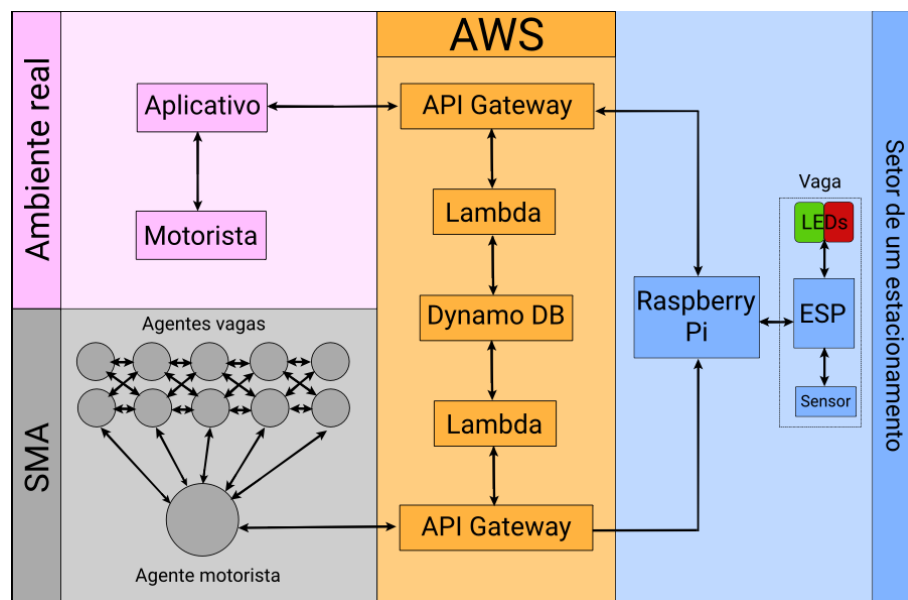


Figura 1. Visão geral da arquitetura

De maneira geral, o sistema inicia quando um motorista requer uma vaga de estacionamento pelo aplicativo móvel (Aplicativo). Este se comunica com a nuvem (AWS) a qual repassa esta informação ao SMA. O SMA realiza uma negociação com base nos critérios informados pelo motorista e devolve para a nuvem o setor e o número da vaga em

que o motorista deve estacionar. Neste momento, o motorista é registrado como locatário da vaga no banco de dados e o setor e o número da vaga são repassados ao aplicativo, o qual informa ao motorista que ele deve se dirigir a vaga. O motorista, por sua vez, estaciona na vaga e então, a plataforma de hardware vinculada a vaga reconhece o carro do motorista e armazena na nuvem os dados necessários.

3.1. Aplicativo Smart Parking

Para o desenvolvimento do aplicativo, o framework Flutter [Flutter 2017] foi utilizado por possuir uma metodologia de desenvolvimento híbrido capaz de compilar o aplicativo para Android e iOS mantendo apenas uma base de código. Assim, o aplicativo une o usuário e o restante do sistema com uma interface de fácil uso, uma vez que consiste em poucas funcionalidades, como cadastrar um novo veículo, estacionar e consultar seu histórico de estacionamento.

Para requisitar uma vaga, o motorista deve, primeiro, cadastrar a placa do carro em e então requisitar uma vaga, onde ele irá visualizar as vagas disponíveis e selecionar uma, informando o preço que deseja pagar na vaga. Como dito anteriormente, neste trabalho os valores a pagar pelas vagas, tempo de estacionamento, quantidade de vagas, tempo de permanência e valores relativos à negociação são gerados apenas para simulação. Ao término da requisição (processo descrito na seção anterior) o aplicativo notifica o motorista se a vaga requerida por ele foi, ou não, alocada para ele. Nota-se que uma outra vaga, que seja parecida com a escolhida por ele pode ser alocada, essa informação dar-se-á pela implementação do SMA.

3.2. Sistema Multi-Agente

Uma parte importante do sistema é a negociação entre o motorista e a vaga. No entanto, a implementação deste componente foi realizado por outra parte do projeto e integrado à esta arquitetura (ver [Alves et al. 2019]). Portanto, considera-se que, quando uma vaga for requisitada, um agente assumirá o papel de motorista e irá conter os requisitos da vaga informadas pelo usuário no momento da escolha da vaga, outro agente assumirá o papel de vaga e ambos irão negociar uma vaga até que cheguem em um consenso. Este conjunto de requisitos leva em consideração a vaga escolhida pelo motorista.

A implementação da negociação pode ocorrer seguindo três abordagens, o Protocolo Contract Net (CNP), o leilão holandês e o leilão inglês. O CNP é basicamente uma licitação, onde o agente motorista envia uma mensagem para todos os agentes contendo seus requisitos e todos os agentes vagas que possuem algum dos requisitos enviam ao agente motorista suas características, assim, o agente motorista decide qual das vagas possui mais características baseado nos seus requisitos.

O leilão inglês tenta obter um consenso entre o agente motorista e o agente vaga mudando o preço a cada rodada, começando com um preço baixo até que nenhum dos agentes vagas aceite mais mudar o preço. Já o leilão holandês começa com um preço alto e vai baixando até que um agente vaga aceite.

3.3. Plataforma de Hardware

A Plataforma de Hardware é um componente fundamental para o correto funcionamento da arquitetura, pois tem a funcionalidade de detectar se um carro foi estacionado e alertar

os outros componentes (da arquitetura) para alteração do status da vaga. A implantação desta plataforma foi realizada em [Botelho et al. 2019], a qual possui um agente desenvolvido em JADE embarcado em um Raspberry Pi¹ gerenciando um setor de um estacionamento com n vagas e cada vaga possui um ESP-12e conectado a sensores de presença e a dois LEDs de cores vermelho (*indicando vaga ocupada*) e verde (*representando vaga livre*).

O sensor de presença localizado em cada vaga permite que o ESP-12e receba um sinal quando um carro é estacionado na vaga, assim, é possível enviar uma mensagem ao Raspberry Pi, informando-o que a vaga está ocupada. Esta comunicação entre os dispositivos é realizada através do protocolo MQTT [MQTT 1999] que utilizando o princípio *Publish-Subscribe*, permite a troca de mensagens por meio de tópicos entre dispositivos conectados à Internet.

Quando é enviada a mensagem: “existe um carro estacionado” ao Raspberry Pi, o agente realiza uma comunicação com o serviço em nuvem, informando que o status da vaga estacionada deve mudar para ocupado. Após obter a resposta positiva do serviço em nuvem, o agente envia ao ESP-12e uma mensagem indicando que o LED vermelho deve ser ligado. Caso receba uma resposta negativa, ou seja, o status da vaga estacionada não é um status de negociação, o LED vermelho começa a piscar.

O processo relacionado a quando o carro deixa a vaga é semelhante. O agente recebe a mensagem que o carro saiu da vaga e repassa a mesma para o serviço em nuvem, o qual irá atualizar o status da vaga para livre, habilitando novamente a vaga para negociação. Após receber uma mensagem positiva da nuvem, o agente informa o ESP-12e que o LED verde pode ser ligado e o vermelho desligado.

3.4. Computação em Nuvem

Para realizar a ligação entre todos os componentes citados nas sub-seções anteriores, tem-se um conjunto de micro-serviços hospedados na *Amazon Web Services (AWS)*². O primeiro deles é o *DynamoDB*³, um banco de dados não relacional que armazena as informações do motorista e das vagas. Todos os setores e as vagas do estacionamento estão mapeados no banco de dados, contendo informações sobre o motorista que ocupa a vaga, como a placa do carro estacionado, e sobre a própria vaga, se ela está livre, ocupada ou em negociação. No banco de dados, também são armazenados os dados do motorista no momento do cadastro no sistema e um histórico de todas as vezes que o motorista estacionou.

O segundo micro-serviço utilizado é a *API Gateway*⁴ com comunicação via HTTP e via *WebSocket*, criando uma conexão entre o aplicativo, o SMA e o controlador físico, enviando os dados compartilhados por eles dentro do sistema. Quando uma mensagem é enviada por um dos componentes para a API, um terceiro micro-serviço é acionado, chamado de *função Lambda*. Esta função é uma rotina, escrita utilizando a linguagem GO [Golang 2009], que executa diferentes tipos de ações dependendo de quem a chama. Basicamente, existe uma Lambda para cada processo executado no sistema, como criar

¹www.raspberrypi.org

²<https://aws.amazon.com/>

³<https://aws.amazon.com/pt/dynamodb/>

⁴<https://aws.amazon.com/pt/api-gateway/>

usuário no banco, atualizar o status da vaga, enviar a requisição do usuário para o SMA, entre outros.

A seguir é descrito o processo de como cada função é executada de modo a informar o motorista o número do setor e da vaga a ser estacionada.

1. O motorista clica no botão de estacionar;
2. Uma requisição HTTP é feita para o *API Gateway*, o qual executa uma função *Lambda* que busca no banco de dados *DynamoDB* as vagas livres e ocupadas do sistema e retorna para o aplicativo;
3. O usuário visualiza as vagas e seleciona a que melhor se encaixa nos seus requisitos;
4. Uma requisição via *WebSocket* é feita para o *API Gateway*, o qual executa uma função *Lambda* que replica a mensagem para o SMA. A mesma função atualiza o status da vaga para em negociação. O controlador físico também recebe esta mensagem e o LED de cor verde fica piscando;
5. O SMA realiza a negociação da vaga e realiza uma requisição, via *WebSocket*, para o *API Gateway*, que executa uma função que envia para o aplicativo o número da vaga negociada;
6. O motorista recebe uma notificação de que a vaga foi negociada, neste momento ele pode aceitar ou não a vaga. Caso aceite, uma requisição HTTP é feita para o *API Gateway*, vinculando os dados do motorista àquela vaga. Caso não aceite, uma requisição via *Web Socket* é feita, informando o controlador físico que o LED de cor verde pode parar de piscar;
7. O motorista agora pode se dirigir a vaga.

A parte física se comunica com a nuvem de modo semelhante: após o processo descrito na sub-seção anterior, o Raspberry Pi responsável pelo setor é quem realiza as chamadas HTTP para o *API Gateway*, o qual executa uma função *Lambda* para atualizar o status da vaga no banco de dados.

4. Resultados

Para analisar os resultados, uma vaga foi requisitada no aplicativo, então, esperou-se até obter a resposta da negociação da vaga, uma vez que o sucesso deste processo exemplifica todo os passos descritos anteriormente.

A Figura 2 ilustra os setores e as vagas do estacionamento, cada círculo representa uma vaga e é necessário clicar em uma bola verde para requisitar uma vaga, nota-se que bolas vermelhas são ocupadas e azuis são em negociação. A Figura 3 ilustra a mensagem que uma vaga foi requisitada. Neste momento a comunicação do aplicativo com a nuvem é realizada, repassando a informação ao SMA que inicia a negociação.

O processo de negociação descrito neste artigo consiste da verificação da disponibilidade da vaga e se há saldo para efetuar o pagamento. Um dos próximos passos da pesquisa é incorporar, neste trabalho, um método de negociação desenvolvido neste projeto, por exemplo [Alves et al. 2019, Ducheiko et al. 2018].

A Figura 4 ilustra a mensagem que a vaga requisitada foi negociada e atribuída ao motorista com êxito e a Figura 5 ilustra o histórico de estacionamento daquele motorista no sistema.

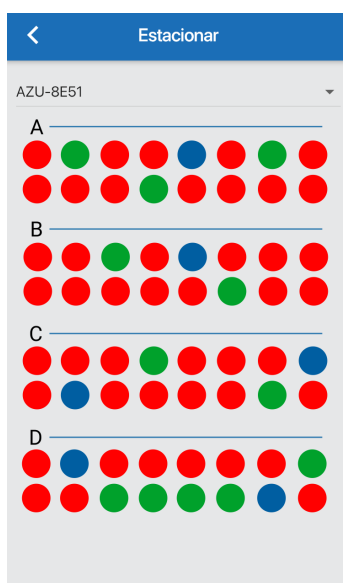


Figura 2. Escolha de vaga



Figura 3. Vaga requisitada

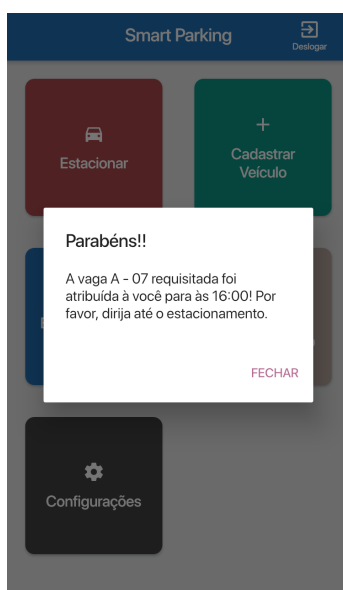


Figura 4. Resultado da requisição

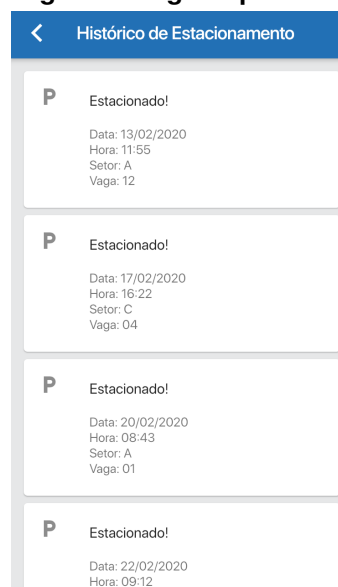


Figura 5. Histórico de estacionamento

5. Conclusões

Este artigo apresenta a implementação de um aplicativo móvel e uma API na nuvem como uma solução para integrar os componentes presentes em um estacionamento inteligente. A arquitetura possui como uma de suas características a versatilidade, sendo possível utilizar qualquer dispositivo móvel para realizar uma requisição. Outra característica diz respeito a facilidade da comunicação entre os dispositivos, sendo necessário apenas de acesso a internet para que os componentes possam se comunicar.

Tendo como base os experimentos apresentados na seção anterior, é possível evidenciar que, para o correto funcionamento desta arquitetura, não é necessário implementar exatamente os componentes aqui citados, como o mesmo SMA, ou o mesmo aplicativo.

A computação em nuvem consegue tornar esta arquitetura flexível, o que significa que um outro SMA, utilizando outro protocolo de negociação pode ser utilizado, desde que receba como entrada uma vaga para negociar e retorne como saída uma vaga negociada.

Como continuidade imediata deste trabalho, deseja-se aplicar testes de desempenho nesta aplicação a fim de evidenciar possíveis gargalos de requisições simultâneas e testar a aplicabilidade desta arquitetura como um todo em um cenário com componentes de um ambiente físico.

Referências

- (2019). Smart parking: A literature review from the technological perspective. *Applied Sciences (Switzerland)*, 9(21).
- Alves, B. R., Alves, G. V., Borges, A. P., and Leitão, P. (2019). Experimentation of Negotiation Protocols for Consensus Problems in Smart Parking Systems. In Mařík, V., Kadera, P., Rzevski, G., Zoitl, A., Anderst-Kotsis, G., Tjoa, A. M., and Khalil, I., editors, *Industrial Applications of Holonic and Multi-Agent Systems*, Lecture Notes in Computer Science, pages 189–202, Cham. Springer International Publishing.
- Botelho, P. W., Alves, G. V., and Borges, A. P. (2019). Proposta de implantação de um sistema ciber-físico para um smart parking baseado em agentes inteligentes. In *Workshop-School on Agents, Environments, and Applications*, pages 259–264.
- Castro, L. F. S. D., Alves, G. V., and Borges, A. P. (2017). Using trust degree for agents in order to assign spots in a Smart Parking. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 6(2):45–55.
- Ducheiko, F. F., Alves, G. V., and Borges, A. P. (2018). Implantação de um modelo de raciocínio e protocolo de negociação para um estacionamento inteligente com mecanismo de negociação descentralizado. In *Revista Junior – ICCEE*, pages 25–32.
- Flutter (2017). Flutter documentation.
- Geng, Y. and Cassandras, C. G. (2013). New “smart parking” system based on resource allocation and reservations. *Proceedings of the IEEE Transactions on Intelligent Transportation Systems*, 14(3):1129–1139.
- Golang (2009). Go documentation.
- Hassoune, K., Dachry, W., Moutaouakkil, F., and Medromi, H. (2016). Smart parking systems: A survey. In *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pages 1–6. IEEE.
- Jazdi, N. (2014). Cyber physical systems in the context of industry 4.0. *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR'14)*, pages 1–4.
- Khanna, A. and R. Anand, journal=International Conference on Internet of Things and Applications (IOTA), t. y. p. m.
- MQTT (1999). Mqtt documentation.
- Pham, T. N., Tsai, M. F., Nguyen, D. B., Dow, C. R., and Deng, D. J. (2015). A cloud-based smart-parking system based on internet-of-things technologies. *IEEE Access*, 3(1):1581–1591.

- Sakurada, L., Barbosa, J., Leitão, P., Botelho, P. W., Alves, G. V., and Borges, A. P. (2019). Development of agent-based cps for smartparking systems. In *IECON - Annual Conference of the IEEE Industrial Electronics Society*.
- Senyo, P. K., Boateng, B., and Addae, E. (2018). Cloud computing research: A review of research themes, frameworks, methods and future research directions. *International Journal of Information Management*, pages 128–137.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

Gerenciamento de Tráfego Urbano em uma Cidade Inteligente baseado em Agentes Autônomos e Descentralizados *

Gleifer V. Alves¹, André P. Borges¹, Paulo Leitão²

¹Programa de Pós-Graduação em Ciência da Computação
Universidade Tecnológica Federal do Paraná (UTFPR)
Ponta Grossa – PR – Brasil

²Research Centre in Digitalization and Intelligent Robotics (CeDRI)
Instituto Politécnico de Bragança (IPB)
Bragança – Portugal


{gleifer, apborges}@utfpr.edu.br, pleitao@ipb.pt

Abstract. *Here we present the proposal of a model based on autonomous agents to represent the urban traffic management in a smart city. In our model we consider decentralised agents which should perceive the environment (e.g. at road junctions), from these perceptions the agents should be able to communicate, cooperate and coordinate each other actions to autonomously decide how to handle urban traffic issues. The issues considered in our approach may range from a simple parking space (in on-street parking) that is blocked, to a complete disruption of an urban road because of a car crash. With our proposed model, we aim to run urban simulations within the agents using tools (e.g. SUMO) to assess the impact of our model in the urban traffic flow and management.*

Resumo. *Este artigo apresenta uma proposta de um modelo baseado em agentes autônomos para representar o gerenciamento de tráfego urbano em uma cidade inteligente. É considerado um modelo de agentes descentralizados que devem perceber alterações no ambiente em que estão situados (e.g. em cruzamentos em vias urbanas), a partir das percepções obtidas, os agentes devem procurar comunicar, cooperar e coordenar entre si ações para atingir uma decisão (autônoma) e solucionar respectivos problemas. Os problemas considerados em nossa abordagem podem variar deste uma simples interdição de uma vaga de estacionamento (externo), até o bloqueio (total) de uma via urbana devido a um acidente de trânsito. Almeja-se por meio deste modelo executar simulações urbanas dos agentes usando ferramentas (e.g. SUMO) para avaliar o impacto do modelo no fluxo e gerenciamento de tráfego urbano.*

1. Introdução

A utilização de técnicas computacionais para explorar soluções para Cidades Inteligentes tem sido amplamente aplicada em problemas envolvendo tráfego urbano. As soluções

* O trabalho Gerenciamento de Tráfego Urbano em uma Cidade Inteligente baseado em Agentes Autônomos e Descentralizados de Gleifer Vaz Alves, André Pinz Borges, Paulo Leitão está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

correspondem ao uso de mecanismos de inteligência, de processamento de imagens, sistemas embarcados, sensores, agentes inteligentes, entre outros [Batty et al. 2012]. O foco das soluções está nas diferentes ramificações de uma Cidade Inteligente, como Mobilidade, Transportes, Economia, Meio-Ambiente, Saúde, entre outras [Koster et al. 2014]. Observa-se que o uso de técnicas computacionais em cenários de Cidades Inteligentes almeja utilizar de forma eficiente os recursos disponíveis, tentando solucionar problemas existentes nos ambientes urbanos.

No quesito Mobilidade e Transportes, as alternativas buscam gerenciar e resolver problemas relacionados a alocação de recursos (e.g. vagas em estacionamentos), fluxo de trânsito em vias, pontos de recarga para veículos elétricos, etc. Especificamente, destaca-se o problema de gerenciamento de tráfego urbano, o qual está associado a um recente convênio entre a UTFPR e o DETRAN-PR. Neste convênio buscam-se soluções diversas para temas no contexto de Cidades Inteligentes.

Uma das questões a serem desenvolvidas diz respeito ao gerenciamento de tráfego em vias urbanas visando a soluções de problemas, como vias bloqueadas, otimizar fluxo de vias, entre outros; de forma a lidar com situações inesperadas que podem ocorrer a qualquer momento em um cenário tão dinâmico. Cenário esse que pode ser adequadamente representado por meio de uma modelagem baseada em agentes. Agentes inteligentes permitem a representação dos componentes existentes em cenários de tráfego urbano, considerando suas respectivas características dinâmicas.

Dito isso, este artigo apresenta uma proposta (inicial) para usar soluções baseadas em agentes autônomos e descentralizados no problema de gerenciamento de tráfego urbano. Aqui entende-se como gerenciamento de tráfego, a verificação e solução de problemas existentes no fluxo de trânsito em vias urbanas. Por exemplo, caso uma determinada via esteja bloqueada: como fazer para desviar o fluxo de veículos (de forma autônoma e eficiente), e assim evitar problemas no tráfego? Como fazer para buscar a otimização do fluxo das vias?

Por meio deste trabalho (inicial) será desenvolvido um projeto de pesquisa, o qual almeja as seguintes contribuições: **i.** conceber uma modelagem (de cenários de tráfego urbano) baseada em agentes descentralizados e autônomos; **ii.** desenvolver técnicas de cooperação e coordenação de agentes aplicadas aos cenários do problema; **iii.** simular as técnicas desenvolvidas em ferramentas de simulação urbana, fazendo uso de dados reais, para poder avaliar a possibilidade de aplicação das soluções desenvolvidas.

O restante deste artigo apresenta uma breve revisão de alguns trabalhos relacionados ao gerenciamento de tráfego. Após, um modelo (simples) é definido, em conjunto com a ilustração de um possível cenário de aplicação. Ao final, algumas considerações e possibilidades de implantação do trabalho são discutidas.

2. Gerenciamento de Tráfego Urbano

Existem diferentes abordagens para lidar com o gerenciamento de tráfego urbano. Alguns trabalhos objetivam soluções para desenvolver semáforos inteligentes e assim otimizar o fluxo em vias; outros buscam soluções para intersecções de trânsito de forma geral, procurando lidar com conflitos ou ainda evitar colisões de veículos. Algumas abordagens consideram somente veículos autônomos ou veículos com motoristas humanos, ou

ainda ambos os casos. Ainda existem trabalhos com o foco nas simulações urbanas para evidenciar a aplicação de técnicas em cenários reais de trânsito.

Nesta seção são brevemente descritos alguns desses trabalhos, sendo que ao final da seção igualmente são mencionados trabalhos com o foco em Estacionamentos Inteligentes, que de certa forma estão relacionados com a proposta deste artigo, devido ao uso de agentes na solução de disputa por recursos em um cenário de Cidade Inteligente.

Em [Dresner and Stone 2008], os autores apresentam uma solução (que até hoje é referência na comunidade científica de agentes) para gerenciamento de intersecções baseada em multi-agentes. A solução apresenta um mecanismo para coordenação de Veículos Autônomos nas intersecções urbanas. Mas, também considera veículos com motoristas humanos e ainda veículos de emergência. Este trabalho procura lidar com incidentes, evitar colisões entre veículos e simular o comportamento dos veículos. Na modelagem do sistema, tanto os motoristas como as intersecções são considerados agentes no SMA. Os autores apresentam um protocolo e um algoritmo para gerenciamento das intersecções. Ainda existe o papel de um gerente da intersecção que deve estabelecer uma comunicação com os agentes motoristas para aplicar a respectiva política de gerenciamento das intersecções e lidar com as requisições dos diversos agentes motoristas.

Em [Chen and Englund 2016], os autores apresentam uma revisão da literatura a respeito do gerenciamento de intersecções de trânsito, onde são analisadas as principais técnicas e soluções pesquisadas envolvendo intersecções sinalizadas e não-sinalizadas. Métodos cooperativos, incluindo horários e reservas de espaço, planejamento de trajetória e semáforos virtuais, são discutidos pelos autores. Métodos de prevenção de colisão entre veículos e de veículos com pedestres, também são abordados. Além disso, é apresentada uma introdução aos principais projetos relacionados ao gerenciamento de intersecções.

Por sua vez, [Carlino et al. 2013] explora a possibilidade de utilizar leilões em cada intersecção para determinar a ordem em que os motoristas executam movimentos conflitantes. Embora este esquema seja inviável para motoristas humanos, veículos autônomos são capazes de fazer manobras de forma rápida e sem problemas em nome de passageiros humanos. Especificamente, o artigo investiga a aplicação de leilões de veículos autônomos em cruzamentos tradicionais usando sinais de parada e sinais de trânsito, bem como protocolos de reserva autônomos. O artigo também aborda a questão da equidade, com uma proposta benevolente do agente do sistema para manter um tempo de viagem razoável para motoristas com orçamentos baixos.

O trabalho de [Guériaux 2019] por sua vez visa organizar o fluxo de veículos usando um Sistema Multiagente baseado na física newtoniana. Neste trabalho, o veículo é considerado um agente físico com comportamento pessoal dividido em três camadas: percepção baseada em sensor, decisão baseada em simulação física e aplicação auxiliada por robótica embarcada.

O artigo [Wang et al. 2019] apresenta um sistema de simulação on-line de controle de tráfego urbano rodoviário, chamado SD-UTCS. O sistema permite realizar a construção e simulação de vários tipos de cenários de tráfego, porém sem mencionar o uso de agentes inteligentes.

Analisando os trabalhos destacados anteriormente a respeito do gerenciamento de intersecções de trânsito é possível observar que boa parte das abordagens não considera o

gerenciamento dos recursos do tráfego urbano, como cruzamentos, veículos, sinalizações, entre outros, de maneira integrada. Além disso, os autores não deixam claro a relação existente entre os agentes e os recursos e ambiente de atuação dos agentes, se há mecanismos de coordenação e cooperação do uso de recursos e auto-organização dos agentes. Possivelmente a exceção está no trabalho de [Dresner and Stone 2008], onde os autores descrevem um protocolo para gerenciamento da intersecção. Este trabalho poderá servir como uma referência para desenvolver aspectos do projeto a ser desenvolvido a partir da proposta deste artigo.

Em termos de abordagens que apresentam uso de ferramentas de simulação e agentes é possível citar os trabalhos de [Blank Born 2015] e [Junior et al. 2015] Ambos trabalhos fazem a utilização do simulador SUMO [Krajzewicz et al. 2012] com modelos baseados em agentes para problemas de tráfego urbano. O trabalho de [Blank Born 2015] utilizou como estudo de caso a cidade de Rio Grande-RS, ao passo que o trabalho de [Junior et al. 2015] utilizou a cidade Porto Alegre-RS como cenário de aplicação. Isso evidencia a aplicação de agentes e SUMO em cenários reais, o que ilustra a viabilidade de aplicação da ferramenta SUMO na sequência do projeto apresentado neste artigo.

Cabe destacar que estes dois trabalhos mencionados têm o objetivo de melhorar a programação dos semáforos nas vias urbanas. Na proposta descrita neste artigo, objetiva-se utilizar técnicas de cooperação e coordenação entre agentes para atingir uma solução descentralizada e autônoma para situações emergenciais que podem ocorrer em vias urbanas, o que necessariamente não está associado a programação de semáforos, mas pode sim englobar outros componentes de cenários de tráfego urbano, como vias bloqueadas, alterações em sentidos de vias, etc.

Em relação à gerência de estacionamentos inteligentes é possível citar alguns trabalhos realizados pelos autores deste artigo, onde agentes e Sistemas Multi-Agentes (SMA) são empregados para implementar e simular negociação de vagas em estacionamentos inteligentes, visando obter as melhores taxas de ocupação dos recursos, i.e. as vagas de estacionamento. Em [Castro et al. 2017], os autores apresentam o uso de SMA e negociação entre agentes, baseada no grau de confiança dos agentes. Em [Ducheiko et al. 2018], os autores descrevem a implementação de um mecanismo de negociação descentralizada entre agentes. Ainda, em [Alves et al. 2019], tem-se a descrição de um mecanismo de negociação entre agentes, baseado em leilões. Por fim, em [Sakurada et al. 2019] os autores propõem uma arquitetura ciber-física para implantar agentes embarcados em estacionamentos inteligentes.

Além destes trabalhos, ainda cita-se o artigo desenvolvido por [Barriga et al. 2019] apresenta uma revisão da literatura sobre tecnologias empregadas em Estacionamentos Inteligentes. Os autores identificam os tipos mais utilizados de cada componente das arquiteturas de Estacionamentos Inteligente (sensores, protocolos de comunicação e soluções de software) e destacam as tendências de uso, além de analisar vários trabalhos cujo foco está no desenvolvimento de softwares para Estacionamentos Inteligente.

Os resultados mencionados pelos trabalhos relacionados bem como as pesquisas realizadas com Estacionamentos inteligentes possuem como ênfase soluções que empregam agentes e SMAs. Com isso é possível vislumbrar a extensão do projeto dentro

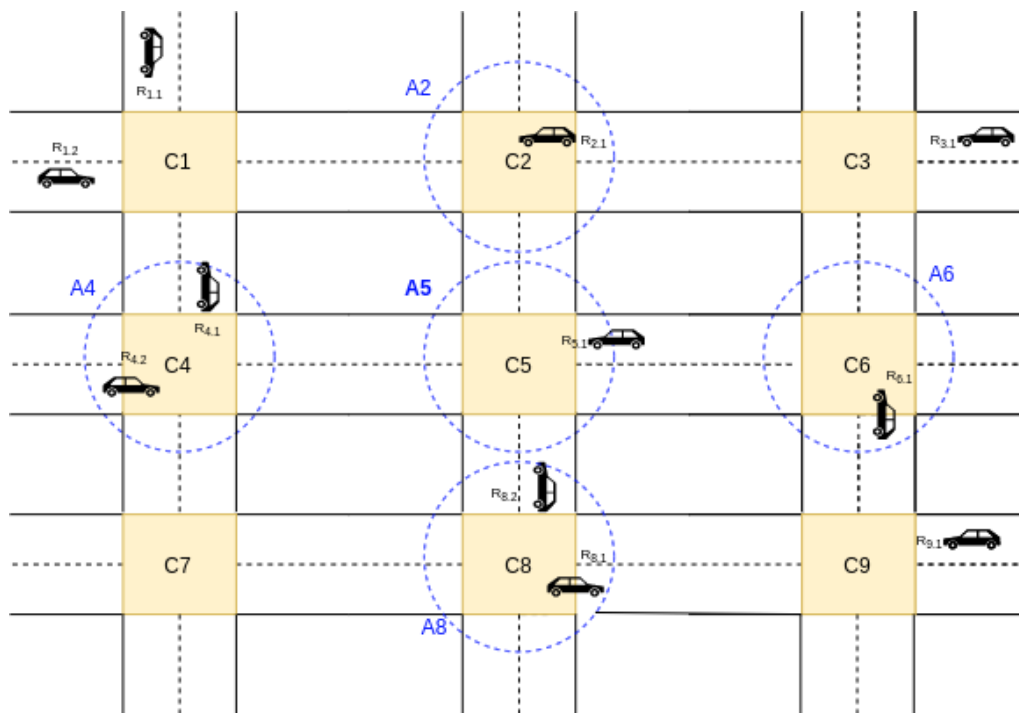


Figure 1. Modelo Geral e Relação de Vizinhança

da ramificação de Mobilidade e Transportes, de forma que agora as soluções de estacionamentos possam ser estendidas para lidar com a problemática existente no tráfego urbano de Cidades Inteligentes.

3. Modelo Proposto

O modelo apresentado neste artigo tem como base um cenário simples de trânsito urbano. Tem-se como ambiente uma Cidade Inteligente (CI) composta por i cruzamentos urbanos (C_i), onde cada C_i deve ter um agente (A_i); cada A_i tem um **raio de ação**, definido aqui pela região de atuação do respectivo agente, sendo composto por vias urbanas, nas quais estão dispostos N artefatos, ou recursos (R_n), conforme ilustra a Figura 1, onde, por exemplo, em C_8 há o agente A_8 e dois recursos (veículos que trafegam nas vias): $R_{8,1}$ e $R_{8,2}$.

Cada agente A_i é situado em um respectivo cruzamento C_i e tem um raio de ação único e de sua responsabilidade. Ainda, na Figura 1, é possível observar a noção de **vizinhança** usada neste trabalho. O agente A_5 possui quatro vizinhos, um em cada direção: norte (A_2), sul (A_8), leste (A_6) e oeste (A_4).

Os agentes existentes na CI seguirão a arquitetura para implementação de agentes racionais BDI (*Belief, Desire, Intention*) e devem ter as seguintes capacidades: percepção, comunicação (descentralizada), negociação e decisão (autônoma). Em linhas gerais, um agente tem suas percepções obtidas dos artefatos (ou recursos) que estão situados em seu raio de ação. A partir de uma dada percepção, o agente A_4 pode iniciar uma troca de mensagens com o agente A_5 , por exemplo, desde que eles sejam agentes vizinhos. A_4 e A_5 podem negociar para gerenciar algum problema de tráfego em seus respectivos cruzamentos (C_4 , C_5). Ao final da comunicação e negociação, A_4 deve tomar uma decisão

e solucionar um dado problema. Imagine que A_5 identificou um mal funcionamento no semáforo localizado no raio de ação de A_5 , agora A_5 precisa comunicar para A_4 que o fluxo de veículos não pode ser direcionado para o raio de ação de A_5 . A_4 deve tomar uma decisão (e.g. desviar o fluxo para via secundária) e comunicar ao A_5 o que foi alterado no gerenciamento da via urbana.

No ambiente considerado neste trabalho serão considerados como artefatos os seguintes elementos de uma *CI*: veículos, via urbana, vagas externas de estacionamento, semáforo, faixa de pedestres, placas de sinalização.

Nas subseções que seguem o protocolo de comunicação entre os agentes é apresentado, seguido de um exemplo de aplicação.

3.1. Protocolo de Comunicação

Os agentes devem comunicar-se por meio de um protocolo, no qual a especificação a seguir é utilizada, apenas quando existe algum problema a ser reportado por um dado agente (dentro do raio de ação dele).

`<ag-dest;tipo-problema;localização;lista-recursos;"msg">`

onde,

- `ag-dest` identifica o agente que deve receber a mensagem;
- `tipo-problema` informa qual das seguintes três classes pertence o problema reportado:

verde: representa situações simples de tráfego, por exemplo uma vaga externa de estacionamento que esteja interdita, que não exigem uma alteração na via urbana, é apenas uma notificação.

amarela: indica problemas intermediários de tráfego urbano que podem exigir alguma forma de alteração na via urbana. Exemplo: semáforo não está funcionando.

vermelha: representa situação grave em uma via urbana, onde certamente é necessário uma alteração na via urbana. Exemplo: ocorreu um acidente e faz-se necessário interditar por completo a via urbana.

- `localização` determina em qual cruzamento o problema está localizado;
- `lista-recursos` define a lista de artefatos (ou recursos) envolvidos no problema;
- `"msg"` o agente que está enviando a mensagem pode completar a informação, relatando ao agente destino mensagens adicionais, como uma nova crença, um plano que deve ser utilizado, entre outras mensagens.

3.2. Exemplo de Aplicação

Para ilustrar o modelo proposto é apresentado um simples exemplo considerando o cenário visto na Figura 2.

O cenário mostra que uma das vias urbana no raio de ação do agente A_3 está bloqueada e portanto o seguinte fluxo de ações e mensagens é executado:

1. A_3 recebe a percepção a partir recurso $R_{3.1}$ (i.e., um veículo) que a via está bloqueada. Considere que $R_{3.1}$ é um veículo que teve algum dano, precisou parar e agora a via está parcialmente bloqueada;

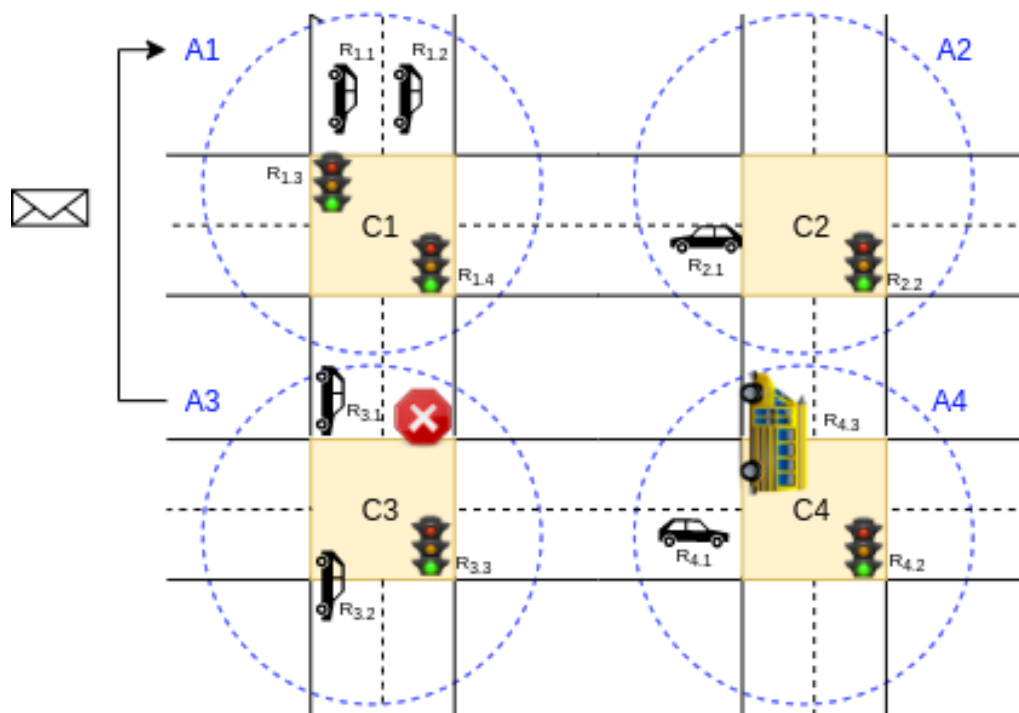


Figure 2. Exemplo de Aplicação

2. A_3 altera as sinalizações no seu respectivo raio de ação, no caso o recurso $R_{3.3}$ sendo um semáforo. Como o fluxo de trânsito será (possivelmente) intenso, devido ao bloqueio de uma via, o semáforo poderá ter que ajustar a frequência da sua sinalização para otimizar o fluxo do trânsito.
3. A_3 envia a seguinte mensagem para A_1 :


```
< A1; vermelha; C3; R3.1; "via parcialmente bloqueada;
      alterar sinalização" >
```
4. A_1 recebe a mensagem de A_3 , onde A_1 deve considerar uma nova crença "via parcialmente bloqueada" e acionar um novo plano "alterar sinalização";
5. A_1 deve também alterar as sinalizações no seu respectivo raio de ação, no caso os recursos $R_{1.3}$ e $R_{1.4}$, respectivamente dois semáforos (um para cada via que compõe o cruzamento). Agora o semáforo $R_{1.4}$ só deverá permitir que os veículos oriundo de C_2 sigam reto em sua direção, evitando dobrar à esquerda (na via parcialmente bloqueada). Enquanto, o semáforo $R_{1.3}$ deverá recomendar que os veículos dobrem à direita (e assim evitar a via bloqueada) e possivelmente ajustar a frequência de sua sinalização para tentar otimizar o fluxo de trânsito nesta via. Essas ações devem ser realizadas para tentar otimizar o fluxo de trânsito em direção a via (parcialmente) bloqueada. Note que aqui é apenas ilustrado um exemplo simples e hipotético para evidenciar a modelagem do cenário em termos de agentes, mas sem um devido estudo da área de engenharia de tráfego urbano.
6. Assim, (idealmente) o recurso $R_{1.1}$, um veículo, quando atingir o cruzamento C_1 deverá virar à direita, evitando assim trafegar na via parcialmente bloqueada;
7. Note que a comunicação é prioritariamente iniciada entre A_3 e A_1 , pois são os agentes que compartilham a via urbana que está parcialmente bloqueada;

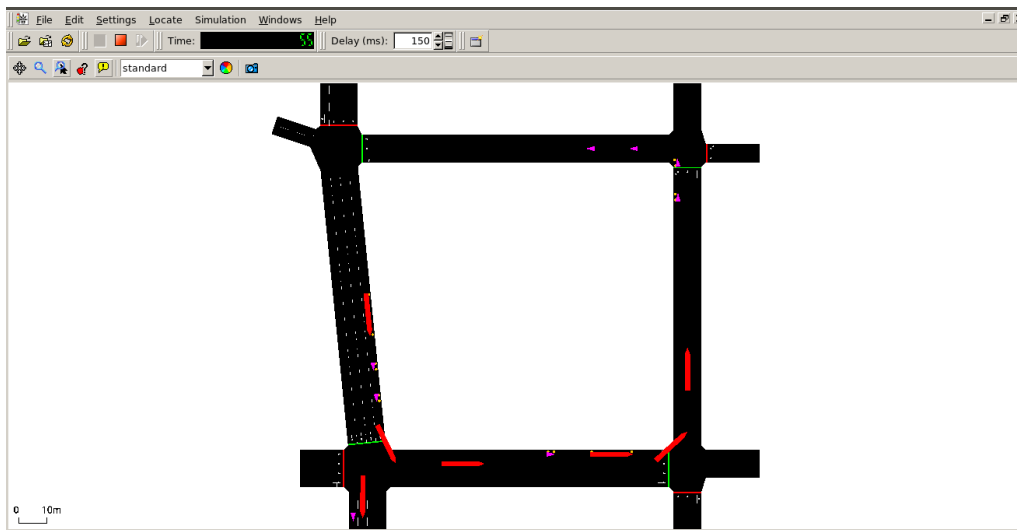


Figure 3. Exemplo de Aplicação - Simulador Sumo 1

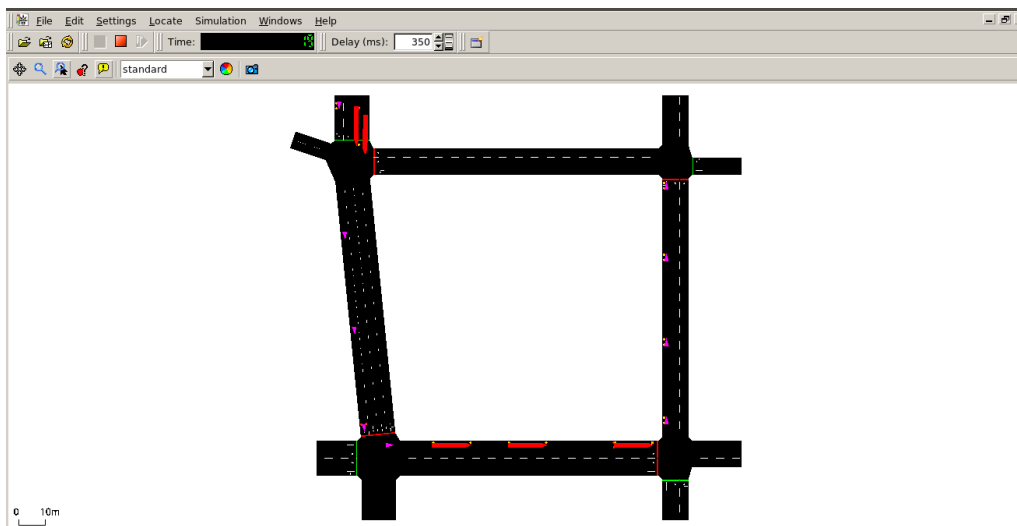


Figure 4. Exemplo de Aplicação - Simulador Sumo 2

8. Ainda destaca-se que esse exemplo apresenta os recursos $R_{2,2}$ e $R_{4,2}$, que são semáforos nos cruzamentos C_2 e C_4 , respectivamente. Todos demais recursos ilustrados neste exemplo são veículos (carros e ônibus - $R_{4,3}$).

Ainda destaca-se brevemente um primeiro exemplo de utilização da ferramenta SUMO, ilustrado nas Figuras 3 e 4, as quais representam uma simples simulação de tráfego urbano. Essa simulação considera o exemplo de aplicação visto anteriormente na Figura 2. Essas simulações utilizando o SUMO foram criadas a partir de uma extração do mapa de um cruzamento na região central da cidade de Ponta Grossa, PR. Nesta simulação estão representados alguns elementos presentes na Figura 2, e.g., carro, ônibus e semáforo.

4. Considerações Finais

Este artigo apresenta uma proposta inicial de um gerenciamento de tráfego urbano baseado em agentes. Espera-se por meio deste modelo explorar algumas características

fundamentais de agentes e SMA, como percepções, decisões autônomas, comunicação, coordenação e cooperação descentralizadas, entre outras. Destacando que a modelagem baseada em agentes é indicada justamente por conseguir capturar os aspectos dinâmicos inerentes ao cenário de trânsito urbano, onde a quantidade de agentes é bastante variável, bem como as suas respectivas ações e decisões.

As características podem representar adequadamente problemas reais existentes no gerenciamento de tráfego urbano, buscando assim uma solução autônoma e descentralizada. Autônoma, pois considera-se que os agentes podem ter planos capazes de ativar procedimentos e ações, sem necessidade de intervenção externa para tal. Descentralizada, pois no cenário apresentado aqui, acredita-se que é possível ter diferentes agentes espalhados na malha de vias urbanas, comunicando, coordenado e cooperando entre si, para resolver eventuais problemas, tudo isso sem a necessidade de um ponto central de gerenciamento urbano.

O trabalho de [Dresner and Stone 2008] (previamente descrito na Seção 2) poderá ser utilizado como referência para desenvolvimento de um mecanismo de coordenação das intersecções de trânsito urbano, no contexto apresentado neste artigo.

Almeja-se que este estudo inicial possa ser estendido em etapas subsequentes, para assim servir de base para trabalhos que estejam associados ao convênio da UTFPR e DETRAN-PR. Um destes trabalhos será em direção ao uso da ferramenta de simulação urbana SUMO [Krajzewicz et al. 2012], onde será possível implementar e simular o modelo proposto neste artigo. Para tal, será também utilizado um *middleware* desenvolvido para a implementação de agentes com o SUMO [Heijmeijer and Alves 2018].

Apesar de ser um trabalho em estágio inicial é possível elencar alguns aspectos que devem ser analisados para a adequada implantação do modelo:

- (i.) escalabilidade: simular e avaliar a possibilidade de representação de uma grande malha de vias urbanas, verificando se realmente é viável ter um agente para cada cruzamento;
- (ii.) excesso de trocas de mensagens: na implementação de comunicação em um SMA, é conhecido que a troca de mensagens pode ser um gargalo, assim a simulação servirá para avaliar se haverá excesso de troca de mensagens entre os agentes do SMA, caso ocorra será necessário rever o protocolo proposto;
- (iii.) necessidade de filtros de percepção: no modelo apresentado existem n recursos em cada cruzamento, imagine um dado agente obtendo percepções oriundas de diversos recursos, isso pode ocasionar excesso de percepções, que poderiam ser resolvidas pelo uso de filtros de percepção;
- (iv.) decisões conflitantes: avaliar as possibilidades de decisões conflitantes num SMA descentralizado, por exemplo, no cenário ilustrado na Figura 2 a decisão de alterar o recurso $R_{1,3}$ para virar somente à direita, poderia conflitar com uma decisão (paralela) ocorrida em outro cruzamento, C_2 , onde alguma via tenha sido igualmente bloqueada, de forma que o fluxo de veículos está também direcionado à direita;
- (v.) recursos que estejam em áreas sombreadas: identificar a possibilidade de existir recursos que estejam fora do raio de ação de qualquer agente do modelo.

References

- Alves, B. R., Alves, G. V., Borges, A. P., and Leitão, P. (2019). Experimentation of Negotiation Protocols for Consensus Problems in Smart Parking Systems. In Marik, V., Kadera, P., Rzevski, G., Zoitl, A., Anderst-Kotsis, G., Tjoa, A. M., and Khalil, I., editors, *Industrial Applications of Holonic and Multi-Agent Systems*, Lecture Notes in Computer Science, pages 189–202, Cham. Springer International Publishing.
- Barriga, J., Sulca, J., León, J., Ulloa, A., Portero, D., Andrade, R., and Yoo, S. G. (2019). Smart parking: A literature review from the technological perspective. *Applied Sciences*, 9:4569.
- Batty, M., Axhausen, K., Fosca, G., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G., and Portugali, Y. (2012). *Smart Cities of the Future*.
- Blank Born, M. (2015). *Uma proposta híbrida baseada em agentes e algoritmos genéticos para a determinação dos tempos de semáforo visando a redução da poluição: estudo de caso do centro de Rio Grande/RS*. Dissertação (Mestrado em Modelagem Computacional), Universidade Federal do Rio Grande, Rio Grande - RS. <https://argo.furg.br/?BDTD10687>.
- Carlino, D., Boyles, S. D., and Stone, P. (2013). Auction-based autonomous intersection management. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 529–534.
- Castro, L. F. S. D., Alves, G. V., and Borges, A. P. (2017). Using trust degree for agents in order to assign spots in a Smart Parking. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 6(2):45–55.
- Chen, L. and Englund, C. (2016). Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586.
- Dresner, K. and Stone, P. (2008). A multiagent approach to autonomous intersection management. *J. Artif. Int. Res.*, 31(1):591–656.
- Ducheiko, F. F., André, P. B., and Gleifer, V. A. (2018). Implementação de Modelo de Raciocínio e Protocolo de Negociação para um Estacionamento Inteligente com Mecanismo de Negociação Descentralizado. *Revista Junior de Iniciação Científica em Ciências Exatas e Engenharia*, 1(19):25–32.
- Guériaux, M. (2019). Self-organizing traffic based on dynamic platoon configuration. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1618–1622.
- Heijmeijer, A. v. H. and Alves, G. V. (2018). Development of a Middleware between SUMO simulation tool and JaCaMo framework. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 7(2):5–15–15.
- Junior, G., Frozza, R., and Molz, R. (2015). Simulação de controle adaptativo de tráfego urbano por meio de sistema multiagentes e com base em dados reais. *Revista Brasileira de Computação Aplicada*, 7(3):65–81.
- Koster, A., Koch, F., and Bazzan, A. L. C. (2014). Incentivising Crowdsourced Parking Solutions. In Nin, J. and Villatoro, D., editors, *Citizen in Sensor Networks*, number

8313 in *Lecture Notes in Computer Science*, pages 36–43. Springer International Publishing.

Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138.

Sakurada, L., Barbosa, J., Leitão, P., Alves, G., Borges, A. P., and Botelho, P. (2019). Development of Agent-Based CPS for Smart Parking Systems. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 2964–2969. ISSN: 1553-572X.

Wang, L., Zhang, L., Zhang, L., Li, M., Zhang, H., Li, K., and Xiu, W. (2019). On-line simulation system of urban road traffic signal control based on scene driven. In *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*, pages 1213–1218.

Agile O-MaSE: Framework Ágil Orientado a Agentes*

Fabiana Ribeiro Ferraz Gominho¹, Rosa Maria Esteves Moreira da Costa¹, Vera Maria B. Werneck¹

¹Programa de Mestrado em Ciências Computacionais, Universidade do Estado do Rio de Janeiro (UERJ), Brasil

fgominho@gmail.com, {rcosta, [vera](mailto:vera@ime.uerj.br)}@ime.uerj.br

Abstract. *Agile methods have been proposed to support the development of multi-agent system and currently, some methodologies already incorporate this concept. However, they consider only one specific agile approach. The objective of this paper is to present an agile extension incorporated into the O-MaSE method of developing multiagent systems. The framework Agile O-MaSE was built to be used in modern iterative and incremental projects, replacing the traditional cascade approach. Agile O-MaSE maintained characteristics of the standard O-MaSE methodology, adding activities that characterize and control agile iterations, with a set of activities, artifacts, and agile practices. After defining Agile O-MaSE, a pilot study was carried out with the development of an educational medical game geared towards agents. Preliminary results show positive aspects in the composition of Agile O-MaSE's activities.*

Resumo. *Métodos ágeis têm sido propostos para apoiar o desenvolvimento de sistema multiagentes e atualmente, algumas metodologias já incorporam esse conceito. Entretanto, elas consideram apenas uma única abordagem ágil específica. O objetivo deste artigo é apresentar uma extensão ágil, incorporada ao método O-MaSE de desenvolvimento de sistemas multiagentes. O framework Agile O-MaSE foi construído para ser utilizado em projetos iterativos e incrementais modernos, substituindo a tradicional abordagem cascata. O Agile O-MaSE manteve as características da metodologia padrão O-MaSE, adicionando atividades que caracterizam e controlam as iterações ágeis, com um conjunto de atividades, artefatos e práticas ágeis. Após a definição da Agile O-MaSE, foi realizado um estudo piloto com o desenvolvimento de um jogo médico educacional orientado a agentes. Resultados preliminares apontam aspectos positivos na composição das atividades da Agile O-MaSE.*

1. Introdução

A disseminação de métodos ágeis no desenvolvimento de *software* tem sido bastante aplicada, provocando uma revolução na literatura, na área profissional, e assim, aumentando a busca dos profissionais por conhecimento e atualização sobre *frameworks* ágeis.

Porém, os *frameworks* ágeis já existentes, simplificam os artefatos de modelagem, mas não são tão eficientes em projetos de sistemas multiagentes (MAS - *MultiAgent System*), devido ao nível de complexidade exigido na modelagem de atividades inerentes a esses sistemas (Ferreira et al, 2015).



* O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilha Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

No âmbito dos sistemas multiagentes, ao longo dos anos, foram adaptadas metodologias orientadas a agentes de processos tradicionais (González-Moreno et al, 2014), (Chella et al, 2006), esses métodos adaptados introduzem um único método ágil e conservam vários artefatos, que no futuro podem ainda ser condensados.

Motivada pela disseminação das metodologias ágeis e pela proposta de adaptação em metodologia orientadas a agentes, esse trabalho colabora com a transformação de um método tradicional orientado a agentes, O-MaSE (*Organization - based Multiagent Systems Engineering*), para uma abordagem ágil, denominada: *Agile O-MaSE*. A metodologia O-MaSE foi escolhida, por ter uma proposta de uso interativo e incremental.

Além da introdução, este artigo é composto por mais 5 seções. A seção 2 aborda uma visão geral do desenvolvimento ágil e das abordagens ágeis encontradas em trabalhos correlatos sobre métodos orientados a agentes. A seção 3 introduz o método O-MaSE. A seção 4 apresenta a proposta do *framework Agile O-MaSE*. A seção 5 apresenta a contextualização e o projeto do estudo piloto. Por fim, a seção 6 discute as considerações finais.

2. Desenvolvimento Ágil de Software Orientado a Agentes


O processo de desenvolvimento de software precisava evoluir rapidamente para reduzir as despesas e atender às necessidades de requisitos de negócios em constante mudanças. Em geral, baseados em uma abordagem iterativa, a especificação, a implementação e os testes são intercalados e os resultados do processo de desenvolvimento de software são decididos por meio de um processo de negociação.

Segundo Beck et al (2001), fevereiro de 2001 é o marco inicial do Manifesto para o Desenvolvimento Ágil (MDA), criado por um grupo de 17 pensadores denominados de “a aliança ágil”. O documento contém termos e conceitos gerais, e passou a guiar o modo ágil de gerenciar projetos. O trabalho realizado chegou aos valores descritos na figura 1.



Figura 1 – Valores do Manifesto de Desenvolvimento Ágil de Software (Beck et al, 2001)

Além dos valores do MDA, também foram criados 12 princípios que guiam o desenvolvimento ágil de software, tais como: satisfação do usuário com entregas adiantadas e contínuas, aceitação de mudanças nos requisitos, pessoas do negócio envolvidas no desenvolvimento, equipe motivada, conversas “cara a cara”, intervalos regulares de revisão.

*  O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilha Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Vários métodos e processos foram propostos para desenvolvimento ágil se destacando o Extreme Programming (XP) e Scrum. O método XP propõe: a criação de histórias de usuário, uso de cartões representativos de classes do sistema (CRC - Class-Responsability-Collaborator), programação em pares e desenvolvimento orientado a testes (TDD - *Test-Driven Development*).

O *framework* Scrum compartimenta o trabalho em *Sprints* (*time-box* de duração de 1 a 4 semanas), identifica o *Product Backlog* (uma lista de funcionalidades desejadas para o sistema) que é definido pelo papel do *Product Owner*. O papel do *Scrum Master* é garantir que o processo Scrum seja seguido e orientar a equipe no uso efetivo do Scrum. Durante a *Sprint* um incremento de produto é liberado e uma nova *Sprint* é iniciada imediatamente após a conclusão da *Sprint* anterior. As *Sprints* contêm eventos de planejamento, reuniões diárias, o trabalho de desenvolvimento, uma revisão da *Sprint* e uma retrospectiva da *Sprint*.

Um mapeamento sistemático sobre as propostas de desenvolvimento ágil orientado a agentes, identificou 11 trabalhos: (Novo *et al*, 2018), (Wang, 2018) (Tenso *et al*, 2017), (Babar *et al*, 2017), (Seghöfer *et al*, 2015), (Ferreira *et al*, 2015), (Noori e Kazemifard, 2015) (Azawi *et al*, 2014), (Yu *et al*, 2014). Foram identificadas as metodologias AgilePassi (Chella, *et al*, 2006) e INGENIAS-Scrum (GONZÁLEZ-MORENO. *et al*, 2014). A Agile-Passi incorpora o método XP e INGENIAS-Scrum adapta o processo Scrum na metodologia INGENIAS.

Steghöfer *et al* (2015) apresentam o método PosoMAS, que é um processo para sistemas multiagentes abertos e auto-organizados, com o ciclo de vida ágil iterativo-incremental do Scrum. Mostra como a nova metodologia foi aplicada em um projeto e as lições aprendidas e por fim, exibe um quadro comparando o PosoMas com outras metodologias de engenharia de software orientada a agentes ágeis.

O trabalho apresentado por Tenso *et al* (2017) afirma que o artefato história de usuário, utilizado em métodos ágeis, não é o suficiente para entender o panorama geral dos requisitos do sistema, mesmo com a existência de métodos que tentam resolver esse problema, e propõem um novo método de modelagem orientado a agentes ágeis (AAOM).

O trabalho apresentado por Azawi *et al* (2014) realiza uma investigação nos métodos de desenvolvimento de jogos e fornece um novo método de desenvolvimento de jogos baseado em modelos de desenvolvimento preditivo e adaptativo criando uma cooperação com engenharia de software orientada a agente (AOSE), que introduz uma metodologia híbrida denominada Metodologia de Desenvolvimento de Jogos Ágil para Agentes (AAGDM).

3. Método O-MaSE (Organization - based Multiagent Systems Engineering)

Segundo DeLoach e Garcia-Ojeda (2014) o O-MaSE foi projetado do zero, ou seja, não partiu de outra metodologia, com um conjunto de fragmentos que poderiam ser montados pelos desenvolvedores para atender aos requisitos específicos de seu projeto. O-MaSE não se compromete com um conjunto pré-definido de Fases, em vez disso, define explicitamente Atividades e Tarefas conforme a figura 2 e permite que engenheiros de métodos organizem as atividades em diferentes caminhos baseados na necessidade do projeto.

A metodologia tem sido utilizada para suportar abordagens iterativas e incrementais modernas, bem como abordagens muito mais simples baseadas em cascata.



Entity	Task	Work Product	Role
Requirements Gathering	Requirements Specification	Requirements Spec	Requirements Engineer
Problem Analysis	Model Goals	Goal Model	Goal Modeler
	Refine Goals	Domain Model	Domain Modeler
Solution Analysis	Model Organization	Organization Model	Organization Modeler
	Interfaces	Role Model	Role Modeler
	Model Roles	Role Description Document	
	Define Roles	Role Goal Model	
Architecture Design	Model Agent Classes	Agent Class Model	Agent Class Modeler
	Model Protocols	Protocol Model	Protocol Modeler
	Model Policies	Policy Model	Policy Modeler
Low Level Design	Model Plans	Agent Plan Model	Plan Modeler
	Model Capabilities	Capabilities Model	Capabilities Modeler
	Model Actions	Action Model	Action Modeler
Code Generation	Generate Code	Source code	Programmer

Figura 2 – Visão geral do O-MaSE (DeLoach e Garcia-Ojeda, 2014)

Ainda segundo DeLoach e Garcia-Ojeda (2014) o fato do O-MaSE não se comprometer com nenhum conjunto específico de Fases, causava um problema por não ter um modelo documentado para descrição de seus processos de design (DPDT - Design Process Documentation Template). Para minimizar este problema, na literatura foi assumido uma abordagem tradicional em cascata, definidas três fases principais: Análise de Requisitos, Projeto e Implementação, conforme a figura 3.

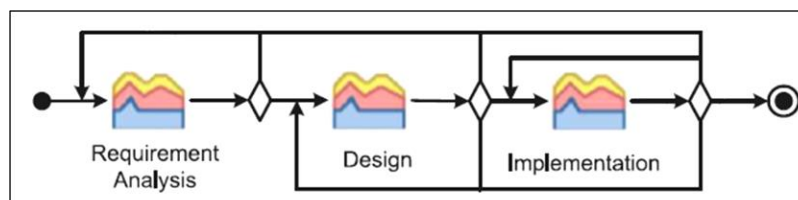


Figura 3 – O-MaSE usando fases em cascata (DeLoach e Garcia-Ojeda, 2014)

Ao utilizar o O-MaSE em um projeto real, o *designer* de processo é livre para definir seu próprio conjunto de Fases e Iterações e atribuir Atividades e Tarefas, conforme apropriado.

4. Proposta *Framework Agile O-MaSE*

Na análise das metodologias ágeis orientadas a agentes encontradas, *Agile-Passi* e *INGENIAS-Scrum*, observou-se que elas adaptaram apenas uma proposta ágil ao método utilizado, *Ingenias* adaptou do *Scrum* e *Passi* adaptou do *XP*, não combinando diversos aspectos, que são propostos em ambos *frameworks* de desenvolvimento ágeis. Contudo, na proposta ágil: *framework Agile O-MaSE*, foram mantidas as Fases da metodologia tradicional



* O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilha Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

da literatura do O-MaSE, com suas Atividades e Tarefas, e foram adicionados fragmentos dos *frameworks* ágeis: Scrum e XP.

Fazendo referência ao MDA, também se destacam o seguinte valor e princípios para construção do *framework Agile* O-MaSE:

- (i) “*Software* em funcionamento, mais que documentação abrangente” (2º Valor do MDA);
- (ii) “Entregar *Software* funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos” (3º Princípio do MDA);
- (iii) “A entrega de *Software* funcional é a medida primária de progresso” (7º Princípio do MDA).

Na **Fase de Análise e Planejamento dos Requisitos**, foram inseridos os produtos: backlog do produto, backlog da iteração e o plano de release originários do Scrum, onde:

Na **Etapa de Levantamento dos Requisitos**, são construídos os Backlogs do Produto e da Iteração; na Etapa de Análise do Problema, são realizadas a Modelagem e Refinamento das Metas e o Plano de Release; na etapa Análise da Solução, são modeladas a interface com a organização e os papéis.

A definição dos requisitos constrói o *Backlog* do Produto com todos os itens que devem ser desenvolvidos para construção do Software. Deve conter requisitos funcionais e não funcionais, e pode ser atualizado a qualquer momento inclusive com os bugs que forem encontrados com o cliente, para acompanhamento das correções. O *Backlog* do Produto deve conter as seguintes informações: item, descrição, tema, prioridade, tipo, estimativa de complexidade, iteração e situação.

O *Backlog* da Iteração, é construído de acordo com a meta/objetivo da iteração, onde o time de desenvolvimento escolhe os requisitos do *backlog* do produto que serão desenvolvidos na iteração corrente. O *Backlog* da Iteração deve conter as informações: meta da iteração, item, história do usuário, tarefas da estória, duração das tarefas, distribuição das tarefas e situação da tarefa.

O Plano de *Release*, é um plano gerencial e deve conter as informações: definição do critério de sucesso da *release*, estimativa dos itens do *backlog* do produto, definição do tamanho e quantidade de iterações, estimativa da velocidade, e associação dos itens do *backlog* do produto as iterações.

Além dos produtos, faltava uma fase importante ligada a qualidade do software, que fizesse referência as características do MDA já destacadas. Dentre os *frameworks* ágeis escolhidos, o XP, se destaca por apresentar uma fase de testes consistente com TDD.

No TDD, o código é desenvolvido de forma incremental, em conjunto com um teste para esse incremento, onde o próximo incremento não é construído até que o desenvolvimento do incremento atual passe no teste. O desenvolvimento de código e os testes são intercalados a cada iteração, conforme a figura 3 (Sommerville, 2019), (Jeffries e Melnik, 2007).



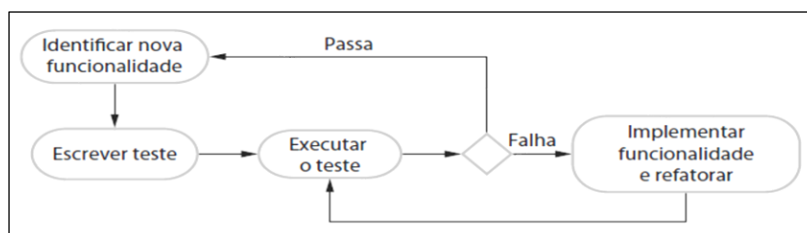


Figura 3 – Fluxo do processo fundamental de TDD (Sommerville, 2019)

Na nova fase, **Fase de Qualidade**, são realizadas duas Atividades: Casos de Testes e Testes.

Na Atividade de Casos de Testes, os testes são escritos a cada iteração, seguindo as histórias selecionadas no *backlog* da iteração.

Na Atividade de Testes, é realizada a cada funcionalidade desenvolvida e só as funcionalidades que estiverem prontas serão liberadas na entrega da iteração, incluindo os testes. As funcionalidades que não passarem nos testes e que não tiveram tempo de ser corrigidas dentro do *time-box* da iteração, devem ser atualizadas na situação da história no *backlog* da iteração, como erro, para que ela seja corrigida e entregue na próxima iteração.

Visando as boas práticas ágeis, foram adicionados ao *framework Agile O-MaSE* os seguintes eventos do *framework Scrum*: revisão da Sprint (reunião de entrega) e retrospectiva da sprint (reunião de revisão da iteração) e para finalizar foi adicionado o Modelo Adaptativo que é clássico nos métodos ágeis, para formalizar as iterações e incrementos, como mostra a figura 4.

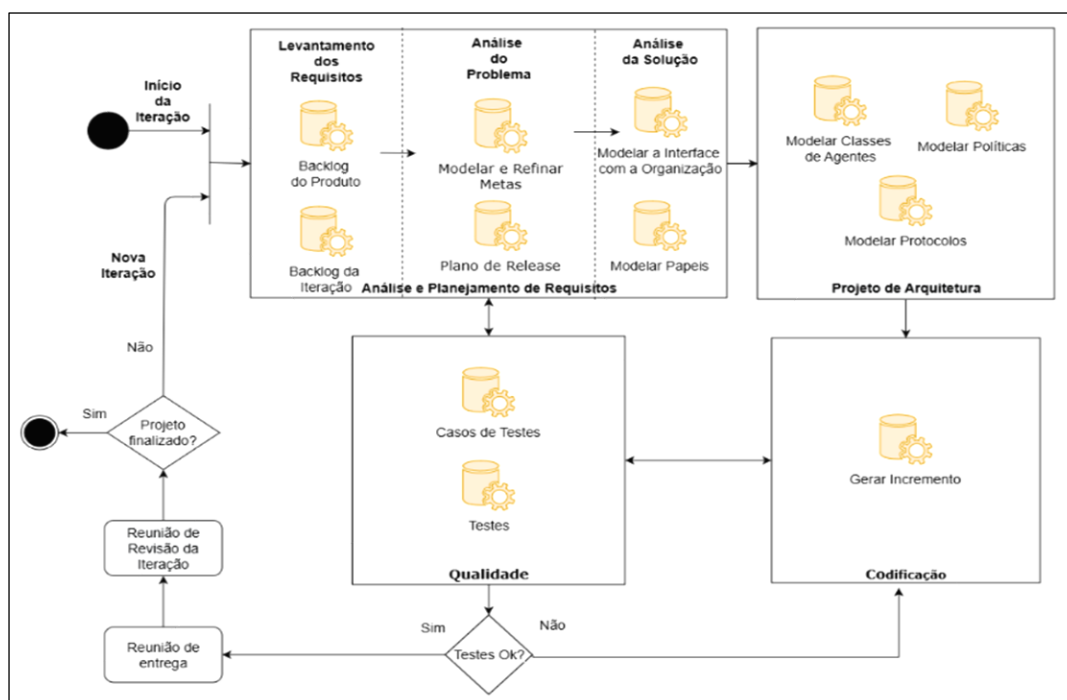


Figura 4 – Framework Agile O-MaSE

Para realizar a construção da proposta piloto do *Agile O-MaSE*, foram considerados: (i) Valores e princípios do Manifesto de Desenvolvimento Ágil (MDA); (ii) Fragmentos dos frameworks XP e Scrum, como parte do processo (descrevendo o que deve ser feito no fragmento específico); (iii) Produtos a serem entregues (artefatos como plano de release, backlogs e casos de testes); (iv) Algumas pré-condições (como os dados de entrada necessários ou a condição de guarda que deve ser verificada antes de iniciar o trabalho especificado no fragmento); (v) Conceitos relacionados a modelos ágeis definidos para serem executados no fragmento de método especificado; (vi) Diretriz (es) que ilustra como aplicar o fragmento e as melhores práticas relacionadas a isso.

5. Estudo Piloto

Esta seção faz uma abordagem de um estudo piloto, com desenvolvimento de um jogo médico educacional orientado a agentes, realizado com uma turma de inteligência artificial (IA) do curso de Ciência da Computação numa Universidade pública do Estado do Rio de Janeiro, com destaque da proposta: *framework Agile O-MaSE*.

O projeto foi dividido em três iterações num *time-box* de 2 semanas, no período de 10 de outubro a 05 de dezembro de 2019. O grupo, composto por quatro alunos, recebeu o cronograma, os artefatos ágeis (*backlog* do produto, *backlog* da iteração e o plano de *release*) e a documentação da metodologia O-MaSE. A mestrandia em ciências computacionais fez a apresentação da metodologia ágil. A Figura 5 apresenta o cronograma.

Id	EDT	Nome da tarefa	Início	Término	Nomes dos recursos
1	1	Projeto de Desenvolvimento do Jogo MedEduc - Metodologia Agile O-MaSE	Qui 17/10/19	Qui 28/11/19	Grupo E; Grupo F
2	1.1	Iteração 1	Qui 17/10/19	Qui 31/10/19	
3	1.1.1	Fase de Análise de Requisitos			
4	1.1.1.1	Atividade Planejar Release: Plano da Release			
5	1.1.1.2	Atividade Levantamento dos Requisitos: Backlog do Produto			
6	1.1.1.3	Atividade Análise do Problema: Modelar e Refinar Metas (Diagrama) e Backlog da Iteração			
7	1.1.1.4	Atividade Análise da Solução: Modelo de Interface com a Organização (Diagrama) e Modelo de Papéis (Diagrama)			
8	1.1.2	Fase de Qualidade			
9	1.1.2.1	Casos de Testes			
10	1.1.3	Fase Projeto de Arquitetura			
11	1.1.3.1	Modelar Classes de Agentes, Modelar Políticas e Modelar Protocolos			
12	1.1.4	Fase de Codificação	Qui 24/10/19	Qui 24/10/19	
13	1.1.4.1	Gerar Incremento			
14	1.1.4.2	Testar a codificação			
15	1.1.5	Fase de Qualidade			
16	1.1.5.1	Realizar Testes			
17	1.1.6	Reunião de Revisão da Iteração - Apresentação dos modelos e Protótipo	Qui 31/10/19	Qui 31/10/19	Todos
18	1.1.7	Reunião de Restropectiva da Iteração	Qui 31/10/19	Qui 31/10/19	Grupo E; Grupo F
19	1.2	Iteração 2	Qui 31/10/19	Qui 31/10/19	
20	1.2.1	Fase de Análise de Requisitos	Qui 31/10/19	Qui 31/10/19	
21	1.2.2	Fase de Qualidade			
22	1.2.3	Fase Projeto de Arquitetura			
23	1.2.4	Fase de Codificação	Qui 07/11/19	Qui 07/11/19	
24	1.2.5	Fase de Qualidade			
25	1.2.6	Reunião de Revisão da Iteração - Apresentação dos modelos e Protótipo	Qui 14/11/19	Qui 14/11/19	Todos
26	1.2.7	Reunião de Restropectiva da Iteração	Qui 14/11/19	Qui 14/11/19	Grupo E; Grupo F
27	1.3	Iteração 3	Qui 14/11/19	Qui 14/11/19	
28	1.3.1	Fase de Análise de Requisitos	Qui 14/11/19	Qui 14/11/19	
29	1.3.2	Fase de Qualidade			
30	1.3.3	Fase Projeto de Arquitetura			
31	1.3.4	Fase de Codificação	Qui 21/11/19	Qui 21/11/19	
32	1.3.5	Reunião de Revisão da Iteração - Apresentação dos modelos e Protótipo	Qui 28/11/19	Qui 28/11/19	Todos
33	1.3.6	Reunião de Restropectiva da Iteração	Qui 28/11/19	Qui 28/11/19	Grupo E; Grupo F

Figura 5 – Cronograma do *framework Agile O-MaSE*

O *backlog* do produto foi entregue com 22 requisitos, já priorizados pelo dono do produto (*product owner*), representado pela mestrandia. O grupo deveria estimar a complexidade de cada requisito utilizando a técnica de Fibonacci. A cada iteração, uma meta de entrega era passada e os alunos selecionavam os requisitos que seriam desenvolvidos, além de apresentar os artefatos ao final de cada iteração. A figura 6, apresenta o *backlog* do produto, a figura 7, apresenta o *backlog* da iteração e a figura 8, apresenta o plano de *release*.

*



O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilha Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>


BACKLOG DO PRODUTO							
ITEM	DESCRIÇÃO	TEMA	PRIORIDADE	TIPO	ESTIMATIVA DE COMPLEXIDADE (Fibonacci - 1 a 13)	ITERAÇÃO	SITUAÇÃO
1	Ao entrar no jogo pela primeira vez, usuário é submetido a questões para definir seu nível e as demais funcionalidades estão indisponíveis.	Nivelamento	10	Requisito	13	1	Pronto
2	A qualquer momento o usuário pode verificar módulos disponíveis para acesso.	Nivelamento, Reforço e Níveis	130	Requisito	5	1	Pronto
3	O usuário visualiza como habilitado apenas módulos disponíveis conforme sua evolução.	Nivelamento, Reforço e Níveis	20	Requisito	1	1	Pronto
4	Ao selecionar o nível disponível na página principal, usuário é direcionado para a primeira questão aleatória do Nível em questão.	Níveis	160	Requisito	1		Pronto
5	Os níveis mais altos têm questões com maior complexidade. As questões disponibilizadas têm níveis de conhecimento de 1 a 5, sendo as questões de nível 1 são mais fáceis do que as questões de nível 5.	Níveis	180	Requisito	2		Pronto

Figura 6 – Backlog do Produto

Na primeira iteração foram desenvolvidos: modelo de metas, modelo de interface com a organização, modelo de papéis, modelo de classes de agentes, de protocolos, além do incremento do software.

BACKLOG DA ITERAÇÃO						
META DA ITERAÇÃO 1	ITEM	HISTÓRIA DE USUÁRIO	TAREFAS DA HISTÓRIA	DURAÇÃO DAS TAREFAS (h)	DISTRIBUIÇÃO DAS TAREFAS	SITUAÇÃO DA TAREFA
Poder realizar as 10 questões do Nivelamento, responder e ter resposta de questões variadas utilizando texto, imagem, áudio ou vídeo, podendo rever as questões já respondidas, saber das respostas corretas quando estiverem erradas, poder parar e reiniciar o nivelamento e no final receber a orientação de qual nível o usuário deve seguir no jogo.	1	Ao entrar no jogo pela primeira vez, usuário é submetido a questões para definir seu nível e as demais funcionalidades estão indisponíveis.	Criar questões	2	█	Pronto
			Criar Login	3	█	Pronto
	2	A qualquer momento o usuário pode verificar módulos disponíveis para acesso.	Criar Nivelamento	5	█	Pronto
			Implementar	3	█	Pronto
	3	O usuário visualiza como habilitado apenas módulos disponíveis conforme sua evolução.	Criar módulos	3	█	Pronto
			Implementar	1	█	Pronto
	8	Cada questão do nivelamento, de nível e reforço é composta por 3 opções de respostas (A, B e C), onde haverá uma opção correta e duas incorretas.	Criar alternativas	1	█	Pronto
			Implementar	1	█	Pronto
	9	Ao selecionar uma opção incorreta em uma questão, usuário é informado da opção correta.	Criar informar opção correta	2	█	Pronto
Implementar			2	█	Pronto	

Figura 7 – Backlog da Iteração

*  O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilha Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

**Plano de Release da metodologia Agile O-MaSE – Grupos E e F
Projeto de Desenvolvimento do Jogo Médico Educacional**

Histórico de Revisões

Data	Versão	Descrição	Autor
17/10/2019	1.0	Criação do documento e preenchimento das informações: 1, 2, 3 e 4.	Fabiana Gominho

1. Definição do Critério de Sucesso da Release

Este projeto é orientado por data, e deve ser finalizado no dia 05/12/19. Consequentemente, as entregas das iterações são do tipo *End-Date Driven*, ou seja, orientada para data de término, cujas as entregas devem estar disponíveis antes da data limite.

Os critérios para considerar uma release terminado com sucesso são:

- A entrega de todas as histórias do *backlog da iteração*;
- A qualidade da entrega: requisitos corretos e sem bugs e
- A apresentação dos modelos e artefatos que a metodologia requer.

O projeto possui um Comitê de Qualidade, composto por: Fabiana Gominho, Vera Werneck e Rosa Costa, que fará a aceitação das entregas de cada iteração.

2. Estimativa dos Itens de Backlog do Produto

A estimativa da complexidade de desenvolvimento deve ser analisada pelo Time de desenvolvimento, item por item do *Backlog* e deve ser iniciado pelo item de maior prioridade. A técnica utilizada na estimativa dos itens do *backlog*, deve ser a sequência de número *Fibonacci*: 1, 2, 3, 5, 8 e 13. Quanto maior for o número, maior será a complexidade de desenvolvimento.

3. Definição do Tamanho e Quantidade de Iterações

O tamanho da iteração está definido limitado a data final do projeto, num *time-box* de 2 semanas, dando um total de 3 iterações.

4. Estimativa da Velocidade

A estimativa da velocidade será realizada de forma manual, observando a primeira iteração e ajustando as demais iterações.

5. Associação dos Itens de Backlog do Produto as Iterações

Iteração	Backlog do Produto
1	1, 2, 3, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
2	4, 5, 6, 18, 19, 20, 21, 22

Figura 8 – Plano de Release

O modelo de metas (figura 9) tem como objetivo transformar os requisitos iniciais do sistema em um conjunto de metas estruturadas para o sistema, refinando cada vez mais cada uma das metas à medida que as iterações avançam.



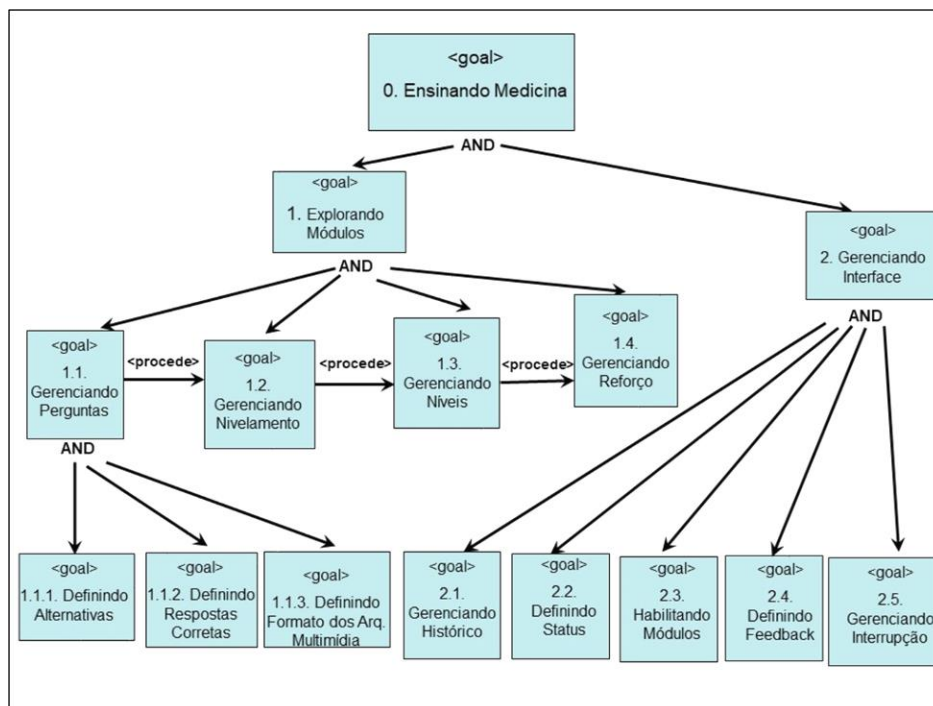


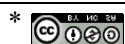
Figura 9 – Modelo de Metas Simplificado

O modelo de papéis (figura 10) identifica os papéis da organização, bem como suas interações entre si e com os atores externos. Foram identificados 4 papéis: avaliador, gerenciador de interface, gerenciador de módulo e gerenciador de perguntas.

O modelo de classe de agentes define o comportamento de um papel em termos de um plano e a capacidade necessária para o agente desempenhar o papel. A figura 11 apresenta o jogador e os agentes, Perguntador, Nivelador e Controlador.

Na segunda iteração, foram atualizados os modelos já existentes, e criado o modelo de políticas, que não havia sido criado na primeira iteração.

Ao final das 3 iterações, foram apresentados além dos modelos finais da metodologia O-MaSE, os *Backlogs* do produto e da Iteração e o Jogo Médico (figuras 12).



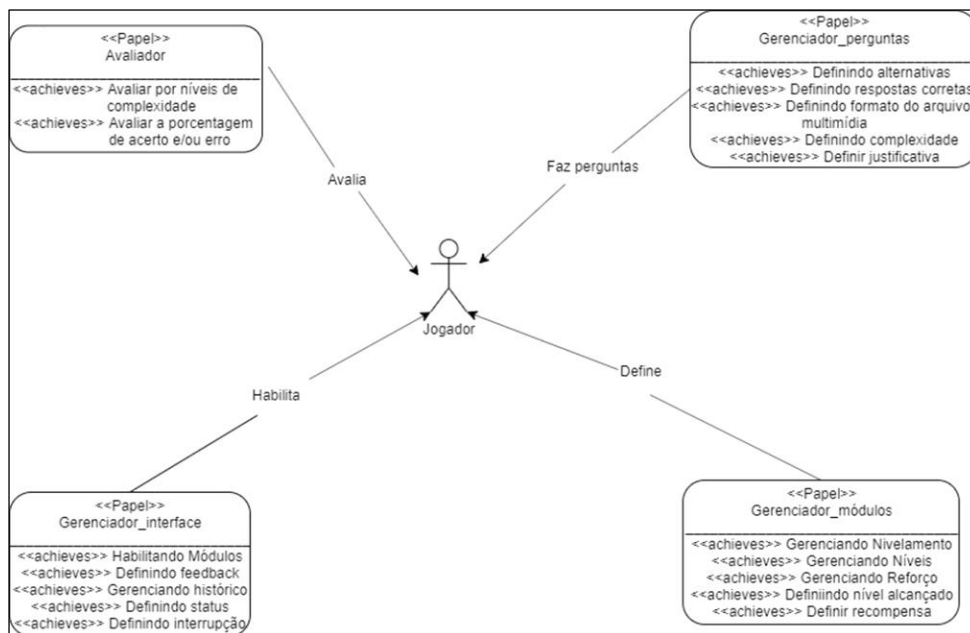


Figura 10 – Modelo de Papéis

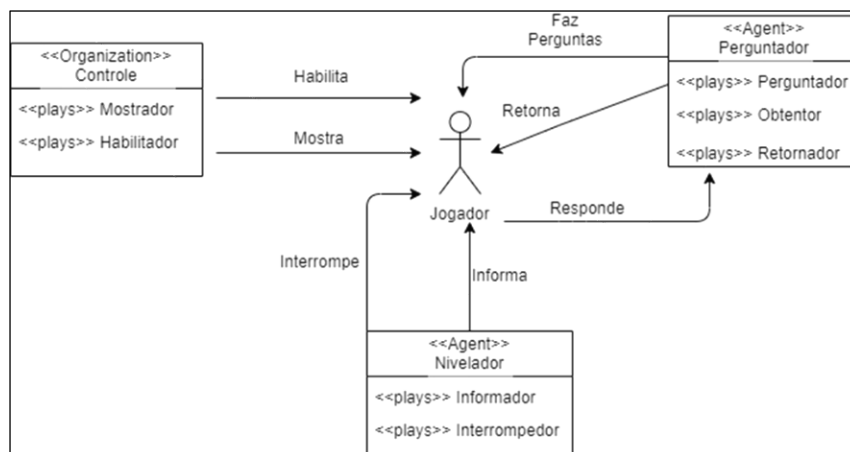


Figura 11 – Modelo de Classe de Agentes

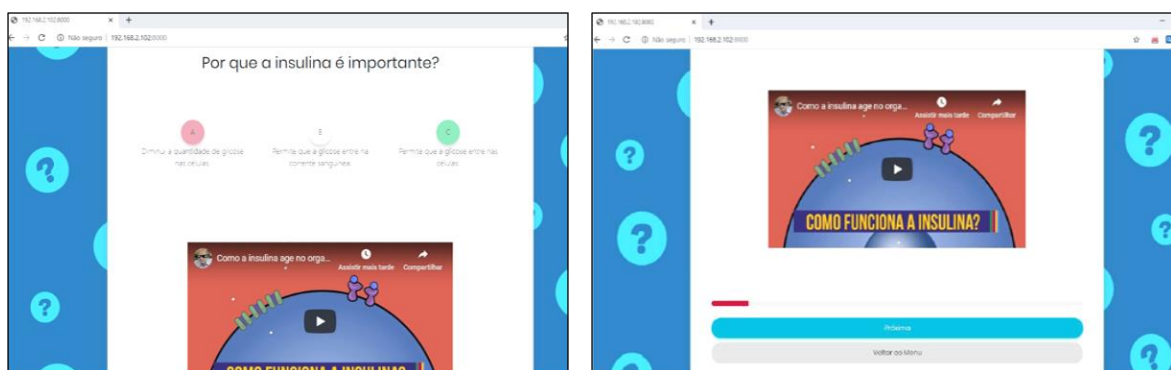


Figura 12 – Telas do Jogo

* O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilha Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

6. Conclusões e Considerações Finais

Este trabalho teve como principal objetivo descrever uma proposta de adaptação de uma metodologia orientada a agente em um *framework* ágil, fundamentada no Manifesto Ágil e nas boas práticas de *frameworks* ágeis, Scrum e XP.

Realizada a experiência no estudo piloto, junto a outras metodologias ágeis e orientadas a agentes, verificou-se que o processo ágil adaptado foi bem utilizado nas entregas e no entendimento geral da agilidade, porém por estarmos num ambiente acadêmico com o horário restrito da aula e devido à ausência dos alunos em dias que não tinha apresentação da iteração, não foi possível executar o *framework* na íntegra, faltando a realização da prática da reunião de revisão da iteração, que seria realizada após as reuniões de apresentação.

Como todo processo ágil, o empenho da equipe faz a diferença no projeto e a tendência é que o trabalho seja maior no início, com a criação de todos os modelos envolvidos e com o passar das iterações esses produtos precisem apenas de atualizações, até a entrega final dos requisitos.

As características: de controle das iterações ágeis, o modelo adaptativo com eventos de revisão da iteração e retrospectiva da iteração, e a inclusão de uma Fase de qualidade para dar suporte às entregas do projeto, possibilitam a apresentação de um *framework* ágil que possa ser integrado a qualquer método orientado a agentes.

Embora as características: eventos de revisão e retrospectiva da iteração, e a Fase de Qualidade, não estejam contempladas em todos os *frameworks* ágeis existentes, estas são importantes para manter o entrosamento da equipe de projeto com o cliente, a aproximação dos componentes da equipe de projeto, a disciplina para obter um bom incremento do produto, para inspecionar e adaptar o processo e a qualidade das entregas.


O grupo que realizou o desenvolvimento utilizando a proposta *Agile O-MaSE*, utilizou duas iterações para realizar os 22 requisitos funcionais, porém seria necessário utilizar a terceira iteração para dar mais qualidade ao software, para tratar erros e requisitos não funcionais como: usabilidade e interface externa do jogo.

Como trabalhos futuros, consideramos que a proposta *Agile O-MaSE* poderá ser aplicada num ambiente onde a frequência dos participantes seja mais homogênea, como alunos bolsistas ou estagiários. Além disso, essa proposta será comparada a outros estudos de caso que foram realizados com as metodologias *Agile-Passi* e *INGENIAS-Scrum* com construção do mesmo jogo.

O estudo piloto permitiu identificar os benefícios da integração desses elementos, que definem a proposta do *framework Agile O-MaSE*. Outros estudos poderão ser realizados para aprimorar a proposta, assim como contribuir com mais evidências do funcionamento da proposta genérica do *framework* ágil que possa ser integrado a métodos orientados a agentes.

Referências

AL-AZAWI, Rula; AYESH, Aladdin; OBAIDY, Mohaned Al. Towards agent-based agile approach for game development methodology. In: 2014 World Congress on Computer Applications and Information Systems (WCCAIS). IEEE, 2014. p. 1-6.

*  O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilha Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

- BABAR, Muhammad et al. The evaluation of agile demand response: An applied methodology. *IEEE Transactions on Smart Grid*, v. 9, n. 6, p. 6118-6127, 2017.
- Beck, Kent; et al. Manifesto for agile software development, 2001. Disponível em: <<http://agilemanifesto.org/>>. Acessado em 27/11/2019.
- CHELLA, Antonio et al. Agile passi: An agile process for designing agents. *International Journal of Computer Systems Science & Engineering*, v. 21, n. 2, p. 133-144, 2006.
- Cossentino, M.; In: IEEE-FIPA. Design Process Documentation Template, jan. 2012. Disponível em: <<http://fipa.org/specs/fipa00097/SC00097B.pdf>>. Acessado em 01/12/2019.
- Deloach, S. A.; Garcia-Ojeda, J. C.; The O-MASE Methodology. In: Cossentino, M. et al. *Handbook on Agent-Oriented Design Processes*. Springer, 2014. p. 253-258.
- FERREIRA, Vitor Manuel Fragoso et al. Developing an educational medical game using AgilePASSI multi-agent methodology. In: 2015 IEEE 28th International Symposium on Computer-Based Medical Systems. IEEE, 2015. p. 298-303.
- GONZÁLEZ-MORENO, Juan C. et al. Ingenias-scrum. In: *Handbook on Agent-Oriented Design Processes*. Springer, Berlin, Heidelberg, 2014. p. 219-251.
- JEFFRIES, Ron; MELNIK, Grigori. Guest Editors' Introduction: TDD--The Art of Fearless Programming. *IEEE Software*, v. 24, n. 3, p. 24-30, 2007.
- NOORI, Fariba; KAZEMIFARD, Mohammad. Simulation of pair programming using multi-agent and MBTI personality model. In: 2015 Sixth International Conference of Cognitive Science (ICCS). IEEE, 2015. p. 29-36.
- NOVO, Fernanda T. et al. Modelando Requisitos de um Jogo Educacional Médico usando a Metodologia INGENIAS SCRUM. In: WER-Workshop of Requirements Engineering. 2018.
- STEGHÖFER, Jan-Philipp et al. PosoMAS: an extensible, modular SE process for open self-organising systems. In: *International Conference on Principles and Practice of Multi-Agent Systems*. Springer, Cham, 2014. p. 1-17.
- Sommerville, I. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019.
- TENSO, Tanel et al. Enhancing requirements engineering in agile methodologies by agent-oriented goal models: Two empirical case studies. In: 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). IEEE, 2017. p. 268-275.
- WANG, Zhe. Teamworking strategies of scrum team: a multi-agent based simulation. In: *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*. 2018. p. 404-408.
- YU, Han et al. A multi-agent game for studying human decision-making. In: *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. 2014. p. 1661-1662.



Ambiente Virtual Personalizado baseado em Estilos de Aprendizagem: uma abordagem avaliativa com as ferramentas Face Reader e Eye Tracker *

Danielli Cossul¹, Brunno José Fagundes², Guilherme Ferreira², Rejane Frozza^{2,3}, William Andrei Nascimento da Silveira²

Universidade de Santa Cruz do Sul

¹Departamento de Ciências da Saúde

²Departamento de Engenharias, Arquitetura e Computação

³Programa de Pós-Graduação em Sistemas e Processos Industriais

daniellicossul@mx2.unisc.br, guilherme27set@gmail.com,
bfagundes@unisc.br, frozza@unisc.br, williamas@mx2unisc.br

Resumo: Esta pesquisa refere-se ao desenvolvimento de um Ambiente Virtual de Aprendizagem personalizado, capacitado para identificar os estilos de aprendizagem dos usuários, com a atuação do Agente Pedagógico Dóris. Objetiva-se identificar a relação estabelecida entre os Estilos de Aprendizagem, apresentação de conteúdos e a interação com a Dóris. O dispositivo Eye Tracker e o software Face Reader foram adotados como ferramentas metodológicas, a fim de avaliar os resultados da interação dos usuários com o Ambiente Virtual. Essa abordagem permitiu observar que a preferência acerca do modelo da apresentação do conteúdo está estreitamente relacionada à orientação do assunto abordado, podendo variar de acordo com a área do conhecimento trabalhada.

Abstract: This research is related to the development of a customized Virtual Learning Environment that is able to identify learning styles of users, with the use of the Pedagogical Agent Doris. The objective is to identify the relation established among the Learning Styles, content presentation and interaction with Doris. The device Eye Tracker and the software Face Reader were used as methodological tools, so that the results of user's interaction with the Virtual Environment were evaluated. This approach made it possible to observe that the preference related to the content presentation model is closely related to the orientation on the subject, being possible a variation according to the worked knowledge area.

1. Introdução

O momento histórico atual pós-moderno (Bauman, 2007), é responsável por inúmeras mudanças nos padrões comportamentais e de interação entre sujeitos, computadores e conhecimento. Fundamentalmente, novos conceitos e definições são produzidos, evidenciando que o modelo de educação se reinvente (Meirelles et al, 2017). Neste caso, especificamente,



* O trabalho Ambiente Virtual Personalizado baseado em Estilos de Aprendizagem: uma abordagem avaliativa com as ferramentas Face Reader e EyeTracker de Danielli Cossul, Brunno José Fagundes, Guilherme Ferreira, Rejane Frozza, William Andrei Nascimento da Silveira está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

refletir acerca dos atores sociais, estudantes e professores, significa pensar em sujeitos digitais que têm possibilidades de acesso, desenvolvimento e compartilhamento de informações em tempo real (Monteiro e Pereira, 2018).

Ferramentas tecnológicas, tais como Ambientes Virtuais de Aprendizagem Personalizados (AVA), são uma possibilidade de suporte e mediação entre prática pedagógica, conhecimento e estudante (Monteiro e Pereira, 2018). Contudo, destaca-se que para o efetivo processo de ensino-aprendizagem, a interdependência entre aprendizado, tecnologia e a organização mental do usuário, relacionada à forma como os conteúdos são recebidos e interpretados, devem ser consideradas. Desta forma, este estudo parte do desenvolvimento de um AVA personalizado, no qual está inserido o Assistente Pedagógico chamado Dóris (Cossul et al., 2018). Este ambiente virtual personalizado é um sistema sendo aperfeiçoado constantemente, desde seus primeiros trabalhos realizados até o presente momento da pesquisa (Santos *et al.*, 2001), (Mainieri *et al.*, 2005), (Frozza *et al.*, 2011), (Borin *et al.*, 2012), (Flores *et al.*, 2013), (Griesang e Frozza, 2013), (Schaefer *et al.*, 2015), (Wegner *et al.*, 2015), (Cortes *et al.*, 2016), (Frozza *et al.*, 2016), (Fagundes e Frozza, 2017) e (Cossul *et al.*, 2018).

A mediação tecnológica personalizada permite que o processo de ensino-aprendizagem aconteça de forma mais eficiente, uma vez que são consideradas as diferentes estratégias de curso adotadas pelos estudantes (Hawasly e Mozzaquatro, 2016). Isto é, proporcionar alternativas facilitadoras para atribuição de sentidos e significados às informações pedagógicas recebidas. Logo, o objetivo principal deste artigo é identificar como ocorre a relação entre os estilos de aprendizagem, apresentação de conteúdos e utilização de um Assistente Virtual chamado Dóris (Cossul et al., 2018). Para o seu alcance, o dispositivo Eye Tracker, o software Face Reader e a aplicação de um questionário foram adotadas como ferramentas metodológicas. Além disso, como objetivo complementar, buscou-se identificar o estado da arte voltada para a utilização de ambientes de ensino personalizados, para isto, foi realizada uma busca bibliométrica na base de dados Scopus.

Essencialmente, esta pesquisa considera as características particulares de cada sujeito, compreendidas como Estilos de Aprendizagem, nas quais as experiências construídas internamente por cada sujeito, durante o processo retenção do conhecimento, ficam em destaque. Tais estilos são abordados por diferentes autores, os quais definem que o Estilo de Aprendizagem de um indivíduo consiste na forma ou preferências com que ele identifica, manuseia e lida com as informações durante o seu processo de construção do conhecimento (Kolb *et al.*, 1999), (Gardner, 1994), (Honey & Mumford, 1992), (Fleming, 1995) e (Felder & Silverman, 1988).

As práticas pedagógicas desempenhadas pelo ambiente desenvolvido são fundamentadas pela teoria de Felder e Silverman (1988). Para estes autores, os indivíduos podem ser distintos entre si, considerando quatro dimensões de acordo com a forma que retêm, percebem, processam e organizam a informação. Respectivamente, Felder e Silverman (1988) pontuam que indivíduos, em relação às formas de retenção da informação podem ter preferências visuais ou verbais; quanto à percepção das informações, podem ser sensoriais ou intuitivos; referente à forma como processam as informações, podem ser diferenciados como ativo ou reflexivo; e quanto à organização das informações, podem ser sequencial ou global.

2. Metodologia

O presente trabalho é caracterizado como de natureza exploratória e de caráter descritivo, uma vez que está baseado em uma abordagem que concilia abordagens quanti-qualitativas, sistematizando e identificando abordagens adotadas pelos pesquisadores (Cardoso, Alarcão e

Antunes, 2010) ao longo dos anos, e além disso, propõe uma análise empírica da interação estudante-ambiente virtual. Para o alcance do objetivo proposto, inicialmente uma busca bibliométrica foi realizada. O método bibliométrico consiste no reconhecimento de pesquisadores, através da literatura publicada dentro de um período estipulado, contribuindo para a produção e evolução do conhecimento científico (Costas, 2017). Desta forma, o volume de investigação produzida ao longo dos anos refere-se, respectivamente, aos trabalhos publicados (Pimenta *et al.*, 2017), os quais são rastreados mediante aplicação de critérios. A seguir, estão contemplados, os critérios de inclusão correspondentes à construção da base de artigos para a pesquisa:

a) Critérios de escolha da base de dados: a base de dados utilizada foi a *Scopus*, pois é uma base multidisciplinar e suas publicações são identificadas com maior nível de qualidade na comunidade científica (Mongeon e Paul-Hus, 2016).

b) Critérios de escolha do período: o período aplicado para o rastreamento de documentos foi 2014-2020, tendo em vista que a literatura coletada seria atual e emergente.

c) Critérios de escolha dos descritores: *learning style*, *personalized learning style*, *virtual learning style*, *FaceReader*, e *Eye Tracker* foram utilizados para o rastreamento de documentos. Os operadores booleanos AND e OR foram utilizados, limitando e ampliando as buscas.

d) Critérios de escolha dos tipos de documentos: foi utilizado como localizar documentos que apresentam algum dos termos de pesquisa no título, resumo e palavras-chave. O tipo de documentos estipulados foi artigos e revisões, considerando esta uma forma de preservar a literatura produzida, na medida em que seriam incluídos nesta bibliometria.

A segunda etapa deste estudo consistiu em elucidar aspectos acerca da interação entre estilos de aprendizagem, utilização de um Assistente Virtual chamado Dóris (Cossul *et al.*, 2018) e apresentação de conteúdos durante o uso de um ambiente virtual. Destaca-se que deste processo participaram 41 usuários com faixa etária entre 20 e 48 anos, com média de 26 anos, composto por 21 homens e 20 mulheres, de diferentes áreas do conhecimento dos cursos de graduação e pós-graduação de uma universidade do sul do Brasil. Os participantes concordaram em participar da pesquisa, com aceitabilidade do Termo de Consentimento livre e esclarecido. O comitê de ética da Instituição aprovou a realização da pesquisa.

Os atributos utilizados para identificar aspectos acerca da interação usuário-ambiente foram, respectivamente: a) dispositivo Eye Tracker, definido como ferramenta complementar pois permite registrar a sequência de observações feitas pelo usuário através do acompanhamento do seu foco visual ou ainda os pontos que o usuário fixa por maior tempo sua atenção (Lin *et al.*, 2017); b) software Face Reader: programa que permite de forma instantânea, a análise quantitativa de expressões faciais (Yu & Ko, 2017), que são, especificamente: felicidade, tristeza, raiva, surpresa, medo, nojo e neutro, identificadas por Ekman (2011). A taxa de exatidão da decodificação realizada pelo software, de acordo com o estudo de Loijens (2013), está em 90%.

Para realizar estas observações, foram desenvolvidos três tipos de apresentação dos conteúdos didáticos. Foram considerados os estilos visual e verbal, e as aulas contemplam: aula i) conteúdo fortemente alinhado à preferência dos usuários, com predominância de imagens para estilos de aprendizagem visual, e uma quantidade menor de texto para o estilo de aprendizagem verbal; aula ii) conteúdo misto, com uma composição equilibrada de textos e imagens; e aula iii) com conteúdo oposto à preferência dos usuários, apresentando quantidade

maior de texto, no caso de um estilo de aprendizagem visual, ou imagens, no caso de um estilo de aprendizagem verbal.

Além das coletas de dados acerca da interação estudante-ambiente, realizadas através de software Face Reader e dispositivo Eye Tracker, um questionário pós-teste composto por 5 questões também foi utilizado. As respectivas perguntas referem-se à identificação da aula preferida pelo aluno; identificação da preferência acerca da discussão sobre novos assuntos; preferência acerca do modelo de apresentação de novos assuntos; e por fim, identificar se o estudante costuma iniciar e terminar uma aula de forma gradual com início, meio e fim, ou prefere entender o todo e somando suas partes.

4. Resultados

Esta seção contempla inicialmente, uma discussão de caráter descritivo acerca dos resultados alcançados a partir da realização da busca bibliométrica. Isto é, os conceitos *Ambiente Virtual Personalizado*, *Agente Pedagógico* são contextualizados, e as dimensões sobre a aplicação da ferramenta Face Reader e do dispositivo Eye Tracker são apresentadas a partir da identificação de suas aplicações em pesquisas acadêmicas contemporâneas. Na segunda parte deste campo, estão evidenciados os aspectos passíveis de identificação interpretação, a partir da interação ambiente-usuário.

4.1 Abordagem quanti-qualitativa

O resultado do rastreamento de documentos na base de dados Scopus, conforme pode ser visualizado no Quadro 1, permitiu identificar que a utilização do software Face Reader para avaliação comportamental de usuários ainda é baixa. Via de regra, destaca-se que através dele, diversos testes e resultados de análises são possíveis. Em relação à utilização combinada dos termos “learning style”, “personalized learning styl”, “virtual learning” com Face Reader nenhum documento foi localizado, já com o dispositivo de Eye Tracker, um documento foi localizado. Desta, a partir da tabela 1 apresentada abaixo, ressalta-se a ausência de pesquisas significativas que apresentam a combinação dos termos utilizados, o que indica uma lacuna de pesquisa.

Tabela 1: Quadro bibliométrico realizado em março de 2020.

SCOPUS	Learning Style	Personalized Learning Style	Virtual Learning	Face Reader	Eye Tracker
Learning Style	2038	3	0	0	1
Personalized Learning Style		3	0	0	0
Virtual Learning			0	0	0
Face Reader				46	2
Eye Tracker					918

SCOPUS	Face Reader	Eye Tracker
“Virtual Learning Style” OR “Personalized Learning Style” OR “Virtual Learning Style”	0	1

Ambientes Virtuais de Aprendizagem são sistemas computacionais que dispõem de mecanismos de interação direcionados ao processo ensino-pedagógico (Frezza *et al.*, 2011). A realidade identificada na interação estudante-ambiente é *duo*. Isto porque aspectos afetivos, cognitivos e simbólicos (Behar *et al.*, 2007), predominantemente humanos, são somados a uma estrutura tecnológica. Além disso, a interação homem, máquina e conteúdo didático desencadeiam diferentes sensações no usuário, as quais são compreendidas como comportamentos verbais e não verbais (Yu e Ko, 2017). Conforme apontado no estudo da universalidade das emoções (Ekman, 2011), estes comportamentos desencadeiam sensações que são manifestadas por gestos e expressões faciais.

Neste contexto, Ambientes Virtuais Personalizados surgem como uma estratégia para potencializar o processo de ensino-aprendizagem. Este trabalho serve como exemplo prático desta perspectiva, uma vez que contempla o Agente Pedagógico Dóris (Cossul, *et al.*, 2018), capaz de interagir por meio de diálogos em textos e áudios, sendo, portanto, uma personagem dotada de características e ações inteligentes, tais como expressões faciais, corporais e emoções, que interagem por linguagem natural, assemelhando-se à interação entre humanos (Figuras 1 e 2).

Figura 1: Agente Dóris



Figura 2: Interação com a Agente Dóris



Além disso, o ambiente em questão aborda os Estilos de Aprendizagem dos usuários. Conforme apontado por Koc-Januchta (2017), técnicas que possibilitem a identificação de Estilos de Aprendizagem são compreendidas como mecanismos para o ajuste de conteúdo didático apresentado em um ambiente virtual. Esta personalização deve favorecer o desenvolvimento de ações cognitivas do sujeito, tais como, a observação, a confrontação, a elaboração de hipóteses, a análise e a sintetização, realizadas pelo estudante ao aprender.

Assim, para cada Estilo de Aprendizagem é pertinente considerar a utilização de diferentes estratégias de ensino que favorecem a formação de conhecimento dos estudantes (Felder e Silverman, 1988). Estas consistem na elaboração direcionada de conteúdos didáticos ao perfil do usuário. A apresentação de conteúdo pode ocorrer através da utilização de elementos que favoreçam a formação do conhecimento, variando entre elementos do tipo texto, imagens, som, vídeo, entre outros (Homrich, Frezza e Schreiber, 2013).

Uma abordagem possível de avaliação dos estudantes durante a utilização do Ambiente Virtual Personalizado adaptado ao Estilo de Aprendizagem é apresentada por Harley (2016), que descreve um levantamento sobre as metodologias utilizadas na pesquisa de

ambiente de aprendizagem baseada em computador. No trabalho, são abordados diferentes pontos de análise possíveis, como, por exemplo, as observações de diferentes canais que podem fornecer informações sobre a emoção momentânea do estudante, dentre elas, as expressões faciais, respostas fisiológicas e postura do corpo. Também são relacionadas as formas de aquisição de informações para avaliação, diferenciando a avaliação *online*, isto é, enquanto o processo está ocorrendo, da avaliação *offline*, através do registro de informações durante o processo para posterior análise.

Neste contexto, a identificação do comportamento do usuário frente ao uso de um ambiente virtual pode ocorrer através do dispositivo Eye Tracker. Este dispositivo permite registrar a sequência de observações feitas pelo estudante durante o acompanhamento do seu foco visual, ou ainda os pontos que o usuário fixa por maior tempo sua atenção (Lin *et al.*, 2017). A sequência de locais observados pelo usuário possibilita identificar pontos de dispersão da atenção na execução de tarefas (Fitzsimmons *et al.*, 2013). No trabalho desenvolvido por Koc-Januchta *et al.* (2017) os autores utilizam-no para validar as dimensões propostas pelo modelo *Object-Spatial Imagery and Verbal Questionnaire* (OSIVQ), o qual considera apenas três dimensões: imagens de objetos, imagens espaciais e dimensão verbal. Com o suporte do *eye tracker* foi possível aos autores, validarem as três dimensões através da observação de usuários que interagiram com um ambiente virtual. Apesar de validar essas três dimensões, o trabalho proposto não aborda os modelos de Felder e Silverman (1988) e também não apresenta uma validação de usabilidade do ambiente utilizado, também não utiliza o *software* FaceReader como ferramenta complementar para avaliar as emoções dos usuários durante a sua interação.

O Face Reader (Vicar Vision, 2007) é um software que realiza análise facial por meio de imagem, detectando as emoções básicas descritas por Ekman (Ekman e Keltner, 1970). Esta ferramenta é baseada no conceito de *Automated Facial Coding* (AFC), uma forma computadorizada e automatizada de classificação que se utiliza do conceito do *Facial Action Coding System* (FACS) (Friesen e Ekmanm 1978), um sistema que categoriza as micro expressões de diferentes músculos da face e os relaciona com determinadas emoções. Destaca-se que o FaceReader permite que estas análises, antes feitas por classificadores humanos, possam ser feitas em tempo real e com grau de confiança semelhante, reduzindo tempo e esforço (Bijlstra e Dotsch, 2011).

O estudo desenvolvido por Yu e Ko (2017), retrata a utilização do software Face Reader para a avaliação de produtos digitais. O estudo relacionou emoções e imagens gráficas apresentadas a consumidores, buscando identificar quais as suas emoções ao visualizarem imagens em diferentes formas de apresentações. Apresenta ainda o software Face Reader como ferramenta de avaliação e validação destas emoções, uma vez que os usuários participantes dos testes foram entrevistados e suas respostas comparadas com os resultados obtidos através da avaliação do software. Os autores consideram os resultados como eficazes.

4.2 Relação usuário-ambiente

Inicialmente, destaca-se que após o desenvolvimento prático da composição da apresentação de conteúdo, ferramentas e métodos utilizados para capturar e analisar os dados, observou-se que é viável: (i) adaptar o Ambiente Virtual para utilizar o sistema de identificação dos Estilos de Aprendizagem proposto por Felder e Silverman (1988); (ii) utilizar o *software* Face Reader para avaliar as emoções dos usuários durante a interação dos usuários diante do conteúdo adaptado ao seu Estilo de Aprendizagem; (iii) utilizar o dispositivo de *eye tracker* para capturar

e gerar informações para analisar os resultados dos Estilos de Aprendizagem em relação aos comportamentos dos usuários durante a interação com o conteúdo apresentado.

Foram levantados aspectos específicos sobre como ocorre a relação entre os estilos de aprendizagem, apresentação de conteúdos e utilização do Assistente Virtual Dóris. Especificamente, sobre a participação da agente Dóris, destaca-se que a sua interação com o estudante, durante o processo de retenção de conhecimento, demonstrou ser motivacional. Isto por que, através de suas ações inteligentes, diálogos em textos e áudios desencadeou a efetiva participação do estudante, permanecendo atento ao que estava sendo proposto.

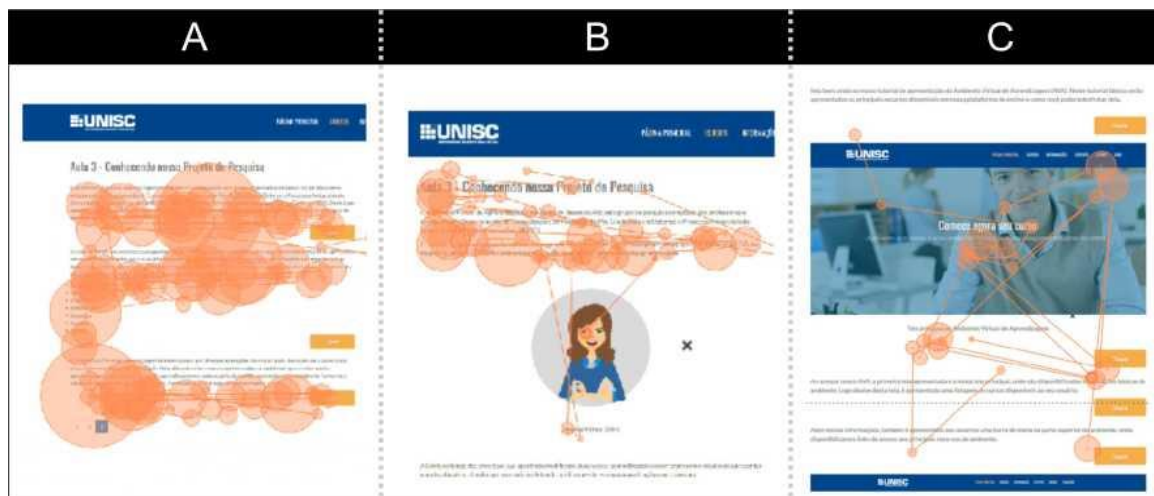
Acerca da interação usuário-ambiente, destaca-se que todos os usuários participantes (41) relataram preferir receber informações ilustrativas, relacionando imagens com explicações textuais ou ainda esquemáticos, tais como fluxogramas ou modelo mental do processo estudado. Em relação às informações teóricas, prevaleceu o relato de que são preferíveis através de uma apresentação textual, organizada em tópicos, uma vez que possibilita um melhor entendimento. Todos esses resultados apresentados ocorreram independentemente da dimensão do Estilo de Aprendizagem. Da amostragem de 41 usuários, 46% destes preferiram a aula iii, isto é, com conteúdo oposto ao seu estilo de aprendizagem identificado; 34 % dos usuários relataram preferir a aula ii, com conteúdo misto; por fim, apenas 20 % dos usuários relataram preferir a aula fortemente alinhada ao seu Estilo de Aprendizagem, a aula i. Apenas 5 usuários obtiveram concordância entre seus relatos, seus comportamentos e seus resultados do questionário de pós-teste. A preferência pelo conteúdo oposto fez mais sentido quando os usuários relataram que o conteúdo era mais interessante por ser diferente, algo que eles não conheciam. Então, observou-se que o interesse foi pelo conteúdo e não pela forma de apresentação do conteúdo.

O questionário pós-teste possibilitou o entendimento do comportamento subjetivo dos estudantes frente ao ambiente e ao conteúdo apresentado. A dimensão visual e verbal demonstrou ser de baixa relevância quando comparada à dimensão ativa e reflexiva. Isso significa que, em linhas gerais, a forma de apresentação do conteúdo está fortemente ligada ao tipo de assunto. Logo, assuntos teóricos, filosóficos, cálculos e formulações têm um melhor entendimento quando apresentados de forma textual, organizados em tópicos e de forma sequencial. Temas e assuntos de viés prático, com aplicações práticas ou passíveis de exemplificação real do processo se tornam mais atrativos e de melhor entendimento através de ilustrações, animações, fluxogramas e esquemas de passo a passo ou vídeos.

Os resultados obtidos pelo *software Face Reader*, apresentaram baixa relevância. Isso por que os usuários participantes não apresentaram expressões significativas durante as observações ou, ainda, apresentaram interferência dos resultados por conta de traços fisionômicos ou acessórios utilizados, como óculos. Considerando os resultados obtidos através dos relatos dos usuários durante o questionário de pós-teste, observou-se que os Estilos de Aprendizagem visual e verbal têm baixa relevância se comparada à dimensão ativa e reflexiva, isto é, o contexto do assunto.

Já as análises dos rastreamentos oculares dos usuários contribuiu para as observações. Através das capturas, pôde-se observar e comparar os resultados obtidos pelas preferências identificadas através dos Estilos de Aprendizagem com o real comportamento dos usuários. O exemplo apresentado na Figura 3 ilustra as observações realizadas sobre o comportamentos dos usuários

Figura 3: Resultados dos Estilos de Aprendizagem versus comportamentos dos usuários



5. Considerações Finais

O comportamento dos usuários diante de um conteúdo personalizado e adaptado às suas preferências individuais, foi avaliado na interação com o ambiente virtual personalizado. A estratégia utilizada relaciona estilos de aprendizagem e elementos multimídia adequados aos estilos de aprendizagem, conforme proposto por Felder e Silverman (1988). Como mencionado, esta avaliação envolveu o *software Face Reader* para observar as emoções dos usuários. Já a utilização do dispositivo de *Eye Tracker* buscou aferir se as preferências descritas aos Estilos de Aprendizagem são compatíveis com as reais observações registradas diante dos conteúdos personalizados apresentados.

Em relação ao levantamento bibliográfico, o estudo realizado demonstra haver poucas publicações na base de dados Scopus, reconhecida internacionalmente, correlacionando os quatro termos pesquisados. Pesquisas que abordam Ambientes Virtual Personalizados são recorrentes, entretanto, existe uma lacuna do conhecimento quando se tem em vista a aplicabilidade do software Face Reader e do dispositivo Eye Tracker. Esta investigação impulsiona o interesse em aprofundar ainda mais este campo, pois acredita-se que tanto a abrangência de adaptação de Ambientes Virtuais Personalizados, quanto a utilização do *software* Face Reader para fim de avaliação do comportamento de usuários, adaptado aos Estilos de Aprendizagem, bem como a utilização do dispositivo de *eye tracker* para investigar e verificar a atenção dos usuários durante a sua interação com o ambiente, tem muito a contribuir. Especificamente, em relação ao uso do software Face Reader, cabe aqui, sugerir que se deve levar em consideração uma pré-seleção de usuários, não incluindo usuários que façam uso de óculos, possuem fortes traços fisionômicos e/ou expressões que possam induzir o *software* ao erro.

Como possibilidade de trabalhos futuros, um ponto pertinente de ser estudado é a relação entre as faixas etárias e os tipos de conteúdo com maior receptividade. Isso por que os estudantes participantes deste estudo se mantiveram com uma média de 26 anos. Em uma breve análise, verificou-se que estes indivíduos fazem parte de uma geração com acesso à internet de altas velocidades, recursos multimídias muito elaborados e com alto grau de interatividade entre informação e usuário, com a capacidade de utilizar recursos de 3D e ainda imersão em ambientes tridimensionais com uso de realidade virtual.

Referências

- Bauman, Zygmunt. (2007) *Tempos Líquidos*. 7 ed. Rio de Janeiro: Jorge Zahar, p. 1-119.
- Behar, P.; Bernardi, M.; Frozi, A., and Kellen, K. (2007). *ROODA: desenvolvimento, implementação e validação de um AVA para UFRGS*. XII Taller Internacional de Software Educativo TISE, v. 2007, p. 321-338.
- Bijlstra, G., and Dotsch, R. (2011). *FaceReader 4 emotion classification performance on images from the Radboud Faces Database*. Unpublished manuscript, Department of Social and Cultural Psychology, Radboud University Nijmegen, Nijmegen, The Netherlands.
- Borin, M; Frozza, R.; Kipper, L. and Schreiber J. N. C. (2012) *Virtual Learning Environments with Emotional Pedagogical Agents for Training in Organizations*. In: *International Conference on Industrial Engineering and Operations Management (ICIEOM)*, Guimarães, Portugal.
- Boulic, R. and Renault, O. (1991) “3D Hierarchies for Animation”, In: *New Trends in Animation and Visualization*, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons Ltd., England.
- CAPES: Portal de Periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, disponível em: <https://www-periodicos-capes-gov-br.ez127.periodicos.capes.gov.br/index.php?> Acesso em 27 de fevereiro de 2019.
- Cardoso, Tereza; Alarcão, Isabel and Antunes, Celorico J. (2010) *Revisão da literatura e sistematização do conhecimento*. Porto: Porto.
- Cortes, E. G.; Bagatini, D. D. da S. and Frozza, R. (2016) *Métodos E Técnicas De Processamento De Linguagem Natural Para Chatterbots: Análise De Trabalhos Relacionados*. In: *XXIII Simpósio de Engenharia de Produção (SIMPEP)*. Bauru: UNESP, p. 1-12.
- Cossul, D.; Frozza, R.; Fagundes, B. J.; Ferreira, G.; Kipper, L. M. and Witczak, M. V. C. (2018) *Evolução do agente pedagógico emocional Dóris em um ambiente virtual de aprendizagem*. In: *Editora Poisson. (Org.). Gestão da Produção em Foco*. 1ª ed. Belo Horizonte: Poisson, v. 24, p. 28-38.
- Costas, Rodrigo. (2017) *Discussões gerais sobre as características mais relevantes de infraestruturas de pesquisa para a cientometria*. *Bibliometria e Cientometria no Brasil: infraestrutura para avaliação da pesquisa científica na Era do Big Data*, ECA - USP, São Paulo, p. 19-42.
- Ekman, P., and Keltner, D. (1970). *Universal facial expressions of emotion*. *California mental health research digest*, 8(4), 151-158.
- Ekman, Paul. (2011) *A linguagem das emoções*. 3. ed. São Paulo: Lua de Papel, p. 1-288.
- FaceReader disponível em: <http://www.vicarvision.nl/products/facereader/>

- Fagundes,, B. J. and Frozza, R. (2017) Desenvolvimento De Modelos De Apresentação De Conteúdo Baseados Em Estilos Cognitivos De Aprendizagem. In: Encontro Nacional De Engenharia De Produção, Joinville. XXXVII ENEGEP – Encontro Nacional de Engenharia de Produção. Joinville: ABEPRO, p. 1-16.
- Felder, R. M. and Silverman, L. K. (1988) Learning and teaching styles in engineering education. *Engineering education*, v.78, n. 7, p. 674-681.
- Fleming, N. D. (1995) I'm different; not dumb. modes of presentation (vark) in the tertiary classroom. In *Research and development in higher education*, Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA), HERDSA, v. 18, p. 308-313.
- Flores, M. S; Frozza, R.; Bagatini, D. D. S. and da Cruz, M. E. J. K. (2013) Uso de rede neural artificial para determinar o estilo cognitivo do aprendiz. In: Congresso Brasileiro de Informática na Educação. p. 717-726.
- Friesen, E., and Ekman, P. (1978). *Facial action coding system: a technique for the measurement of facial movement*. Palo Alto.
- Frozza, R.; da Silva, A. A. K; Schreiber, J. N. C.; Lux, B.; Molz, K. W.; Kipper, L. M.; Borin, M. P.; de Carvalho, A. B.; Baierle, J. L. and Sampaio, L. (2011) Agentes pedagógicos emocionais atuando em um ambiente virtual de aprendizagem. *RENOTE*, v.9, n. 1.
- Frozza, R.; Kipper, L. M.; Fagundes, T. S. and Witczak, M. V. C. (2016) Táticas De Ensino Em Ambientes Virtuais De Aprendizagem: Perspectivas em Habilidades Cognitivas. In: XII Congresso Nacional De Excelência Em Gestão & Inovarse, Rio De Janeiro. p 1-11.
- Gardner, H. (1994) *Estruturas da mente: a teoria das inteligências múltiplas*. Artes Médicas, p. 1-340.
- Griesang, G. and Frozza, R. (2013) Desenvolvimento de um Mecanismo de Comunicação entre Agentes Pedagógicos em um Ambiente Virtual de Aprendizagem. Dissertação de mestrado. Universidade de Santa Cruz Sul – UNISC.
- Harley, J. M. (2015). Chapter 5 - Measuring Emotions: A Survey of Cutting Edge Methodologies Used in Computer-Based Learning Environment Research, In *Emotions and Technology*, p. 89-114. 2016.
- Hawasly, M. A. R. K. and Mozzaquatro, P. M. (2012) Estilos e Estratégias de Aprendizagem Personalizadas a Alunos das Modalidades Presenciais e a Distância. *Revista Brasileira de Informática*, v. 20, n. 1.
- Holton, M. and Alexander, S. (1995) "Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials", *Computer Graphics: Developments in Virtual Environments*, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd., p. 449-460.

- Honey, P., and Munford, A. (1992) The manual of learning styles. p. 1-94.
- Koc-Januchta, M.; Hoffler, T.; Thoma, G. B.; Precht, H. and Lautner, D. (2017) Visualizers versus verbalizers: effects of cognitive style on learning with texts and pictures - An eye-tracking study. *Computers in Human Behavior*, v. 68, p. 170-179.
- Kolb, D.; Boyatzis, R. and Mainemelis, C. (1999) *Experiential learning theory: Previous research and new directions*. Department of organizational behavior. Weatherhead School of management. Case Western Reserve University.
- Lin, Y. Y. and Holmqvist, K.; Miyoshi, K. and Ashida, H. (2017) Effects of detailed illustrations on science learning: an eye-tracking study. *Instructional Science*, v.45, n.5, p. 557-581.
- Loijens, L. and Krips, O. (2013) *Facereader methodology*. Wageningen: Noldus Information Technology.
- Mainieri, A. G.; Frozza, R.; Schreiber, J. and Molz, K.; (2005) Educational System based on Cognitive styles and/or Learning styles. In: XXVII Annual Conference of the Cognitive Science Society. Stresa.
- Mayer, W. R. E. (2005) Principles for managing essential processing in multimedia learning: Segmenting, pretraining, and modality principles. *The Cambridge handbook of multimedia learning*, p. 169-182.
- Meirelles, D. S.; Agüero, F. H.; Leon, D.; Santos, L. A. P. dos and Francisco, R. N. D. (2017) Tecnologia e Competências Tecnológicas em Educação a Distância: um estudo exploratório. *Future Studies Research Journal*, SP, V.9.
- Mongeon, P. and Paul-Hus, A. (2016) The journal coverage of Web of Science and Scopus: a comparative analysis. *Scientometrics*, v. 106, n. 1, p. 213-228.
- Monteiro, Mara Rúbia Muniz and Pereira, Kelly Ticiano Azevedo. (2018) Educação a distância na era digital: perspectivas para pensar os novos atores virtuais - nativos e imigrantes digitais. *Congresso Internacional de Educação e Tecnologias – CIET*, p. 1-13.
- Pereira, Flávia Maria. (2017) Impactos da Utilização das Tecnologias no Processo de Aprendizagem das crianças. *Revista Multidisciplinar em Psicologia*, p. 520-529.
- Pimenta, A. A.; Portela, A. R. M. R.; De Oliveira, C. B. and Ribeiro, R. M. (2017) A bibliometria nas pesquisas acadêmicas. *Revista de Ensino, Pesquisa e Extensão – Scientia*, v. 4, p. 1-13.
- Santos, C. T. dos; Frozza, R.; Dahmer A. and Gaspary, L. P. (2001) Dóris – Um Agente de Acompanhamento Pedagógico em Sistemas Tutores Inteligentes. In: XII Simpósio Brasileiro de Informática na Educação – SBIE – UFES, p. 97-105.

- Schaefer, M.; Frozza, R.; Silveira, R. A. and Rutsatz, R. F. (2015) Treinamentos E-Learning com estratégia de ensino baseada em animações. XX Congreso Internacional de Informática Eduactiva - TISE. Santiago: Universidad de Chile, v. 11, p. 786-791.
- Van Kuilenburg, H., Wiering, M., and Den Uyl, M. (2005, October). A model based method for automatic facial expression recognition. Springer, Berlin, Heidelberg, In European Conference on Machine Learning, p. 194-205.
- Wegner, W.; Frozza, R.; Bagatini, D. D. da S. and Da Cruz, M. E. J. K. (2015) Técnica de clusterização para determinar o estilo cognitivo de usuários em um ambiente virtual de aprendizagem. In: XX Congreso Internacional de Informática Eduactiva - TISE. Santiago: Universidad de Chile. v. 11. p. 444-449.
- Yu, C. and K, C. (2017) Applying Face Reader to Recognize Consumer Emotions in Graphic Styles. Procedia CIRP, v. 60, p. 104-109.

RESUMOS ESTENDIDOS

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

Henrique M. R. Jasinski¹, Mariela Morveli-Espinoza¹, Cesar A. Tacla¹

¹CPGEL– Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro, 3165 – 80230-901 – Curitiba, Brasil

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

Abstract. *The objective of this paper is to propose an architecture and discuss the design and implementation of a Belief-Based Goal Processing (BBGP) based agent simulator. An implementation of the model could be employed on the study of the conceptual model BBGP besides simplifying the analysis and explanation of the agent's reasoning process. In this work we show how the model was adapted for the simulator implementation¹.*

Resumo. *O objetivo deste artigo é propor uma arquitetura e discutir o design e implementação de um simulador de agentes baseados em Processamento de Objetivos Baseado em Crenças (BBGP). Uma implementação do modelo poderia ser empregada em estudos do modelo conceitual BBGP além de tornar mais simples a análise e explicação do processo de raciocínio do agente. Neste trabalho mostramos como o modelo foi adaptado para a implementação do simulador¹.*

1. Introdução

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

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



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

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

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

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

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

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

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

2. Argumentação Computacional

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

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

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

3. Modelo BBGP-argumentativo

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

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

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

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

$$\begin{aligned} r_{de}^1 &: \neg has_incompatibility(g) \rightarrow chosen(g) \\ r_{de}^2 &: most_valuable(g) \rightarrow chosen(g) \\ r_{ch} &: has_plans_for(g) \wedge satisfied_context(g) \rightarrow executive(g) \end{aligned}$$

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

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

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

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

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

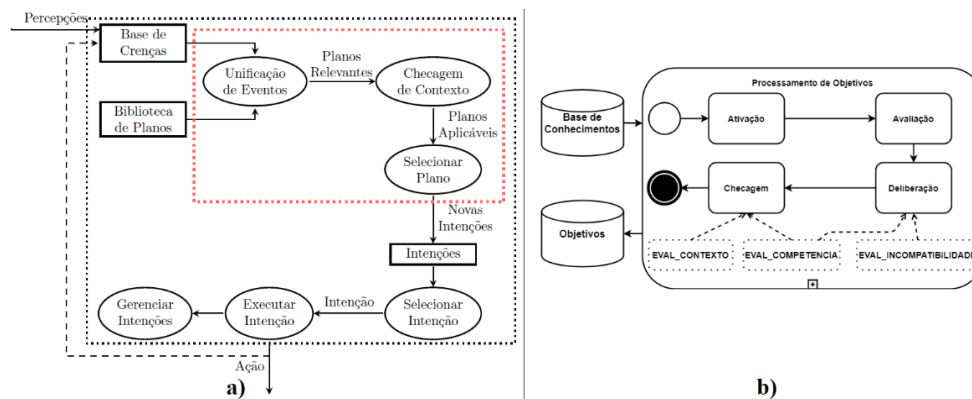


Figura 1 a) Ciclo de execução do agente. b) Processamento de Objetivos.

4. Arquitetura do Agente BBGP-argumentativo

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

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

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

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

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

5. Implementação do Modelo

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

5.1 Adaptação das Regras

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

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

$$take_hospital(X) \mapsto take_hospital(gHolder, X)$$

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

Antes: $most_valuable(take_{hospital}(X)) \rightarrow chosen(take_{hospital}(X))$

Convertido: $most_valuable(g002) \rightarrow chosen(g002)$

Algoritmo 1 Compatible Goals

```

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

```

5.2 Algoritmo de Geração de Ataques entre Objetivos

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

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

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

6. Conclusões

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

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

Referências

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

Desenvolvimento de um Framework para Modelagem e Simulação de Redes Regulatórias Genéticas usando Sistemas Multiagente *


Nilzair B. Agostinho¹, Adriano V. Werhli¹, Diana F. Adamatti¹

¹Centro de Ciências Computacionais – Universidade Federal de Rio Grande (FURG)
Av. Itália, km 8, bairro Carreiros, Rio Grande, RS, Brasil.

{nilzairmb, werhli, dianaada}@gmail.com

Abstract. *Biological systems are highly complex and they are separated into individual parts to facilitate their study. The representation of biological systems as Genetic Regulatory Networks, that form a map of the interactions between the molecules in an organism, is a standard way of representing such biological complexity. As Genetic Regulatory Networks are composed of genes that are translated into transcription factors, which in turn regulate other genes. Scientists have worked on the inference and representation of Genetic Regulatory Networks. For simulation and inference purposes, many different mathematical and algorithmic models have been adopted to represent the Genetic Regulatory Networks. Among these methods, we hypothesize that Multiagent Systems are somewhat neglected. In this paper we present the first efforts to develop a simulator using the Multiagent System to model generic Genetic Regulatory Networks. To accomplish this, we develop a Multiagent System that is composed of agents that mimic the biochemical processes of gene regulation.*

Resumo. *Os sistemas biológicos são altamente complexos e a sua separação em partes individuais facilita o estudo. A representação de sistemas biológicos como Redes Regulatórias Genéticas que formam um mapa das interações entre as moléculas num organismo é uma maneira padrão de representar essa complexidade biológica. As Redes Regulatórias Genéticas são compostas de genes que são traduzidos em fatores de transcrição, que por sua vez regulam outros genes. Os cientistas trabalharam na inferência e representação de Redes Regulatórias Genéticas. Para fins de simulação e inferência, muitos modelos matemáticos e algorítmicos diferentes foram adotados para representar as Redes Regulatórias Genéticas nos últimos anos. Entre esses métodos, acreditamos que os Sistemas Multiagentes sejam um pouco negligenciados. Neste trabalho, apresentam-se os primeiros esforços para desenvolver um simulador usando o Sistemas Multiagente para modelar Redes Regulatórias Genéticas genéricos. Neste sentido, está sendo desenvolvido um Sistema Multiagente que é composto por agentes que imitam os processos bioquímicos de regulação de genes.*

* O trabalho Desenvolvimento de um Framework para Modelagem e Simulação de Redes Regulatórias Genéticas usando Sistemas Multiagente de Nilzair B. Agostinho, Adriano V. Werhli, Diana F. Adamatti está licenciado com uma Licença Creative Commons - Atribuição 4.0 Internacional. <http://creativecommons.org/licenses/by/4.0/>

1. Introdução

Embora o dogma central da biologia afirme que a informação flui através de macromoléculas, do DNA para o RNA e do RNA para as proteínas (Figura 1), através de Redes Regulatórias Genéticas (Genetic Regulatory Networks - GRN) que por fim funcionam através de reações bioquímicas, a vida não existiria apenas com as macromoléculas. Então, para que o dogma central seja altamente descritivo, ele deve incluir pequenas moléculas que são os genes [Schreiber 2005].

Esses genes (pequenas moléculas) se traduzem em proteínas que podem ativar ou inibir a transcrição de outros genes. Portanto, identificar os padrões de GRN é de grande valia para uma melhor compreensão dos processos que ocorrem em nível molecular. Então, no cenário atual da área de Sistemas Biológicos, trabalha-se para o desenvolvimento de ferramentas que possam auxiliar na obtenção de novos conhecimentos sobre as interações moleculares. Por este motivo, tem sido desenvolvidas pesquisas com o objetivo principal de obter um detalhamento da organização funcional de Sistemas Biológicos [Gibas and Jambeck 2002, Lopes 2011].

Na Figura 1b pode-se visualizar os sítios de ligação e os Fatores de Transcrição (Transcription Factor - TF) que entram nos sítios de ligação e assim, de acordo com a sua quantidade e as regras de interação existentes, podem produzir ou inibir uma proteína, que é o produto final.

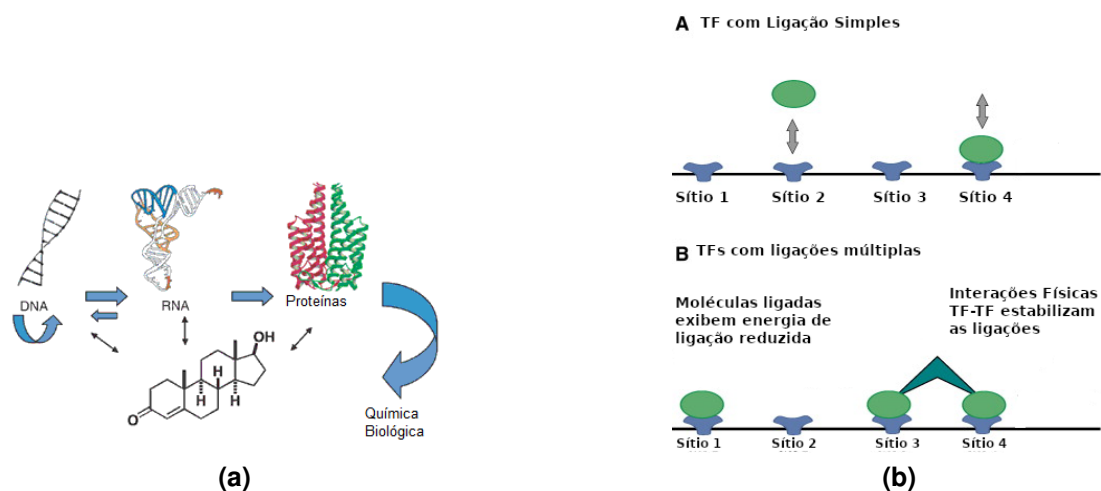


Figura 1. (a) Dogma Central da Biologia: replicação, transcrição e, finalmente o produto final (proteína) - adaptado de [Schreiber 2005]. (b) Fatores de Transcrição e Sítios de Ligação: os Fatores de Transcrição são proteínas que ajudam a ativar ou desativar genes específicos, ligando-se ao DNA próximo; os Fatores de Transcrição, que são ativadores, aumentam a transcrição de um gene, e os que são repressores diminuem a transcrição. Adaptado de [Ezer 2014].

Neste trabalho, o principal objetivo é mostrar o desenvolvimento de uma versão de um Sistema Multiagente (Multiagent Systems - MAS), como um *framework* de modelagem para representação de GRN. Para o desenvolvimento deste MAS, foi realizado um processo de Revisão Sistemática detalhado em [Agostinho et al. 2018]. A Revisão Sistemática foi realizada de acordo com [Mariano et al. 2017], e resul-

tou nos seguintes artigos: [Liu et al. 2016], [Khan et al. 2003], [Haydarlou et al. 2016], [Ghazikhani et al. 2011], [Sanfilippo et al. 2012], [Pham 2008], [Yang and Sun 2011].

Este artigo apresenta a versão inicial entre a interação de dois TF que interagem num Sítio de Ligação específico, e são apresentados alguns resultados preliminares, comparados com o processo de Michaelis-Menten [Johnson and Goody 2011].

O artigo está organizado da seguinte maneira: Operação de uma Rede Regulatória Genética, que explica a fundamentação de uma GRN; Modelo Proposto, que explica a fundamentação de MAS e como o modelo foi desenvolvido; Resultados e Conclusões.

2. Operação de uma Rede Regulatória Genética

Genes são ativados ou inibidos por proteínas chamadas TF, que são genes produtos. Uma GRN descreve a interação entre os TF e os genes que são regulados. Esse processo permite que células produzam as proteínas necessárias e apropriadas em termos de tempo e quantidade [Lopes 2011]. O mecanismo básico de controle regulatório é realizado através da ligação dos TF em um local específico, os Sítios de Ligação, que estão localizados numa região promotora de um gene como é descrito em [Schreiber 2005]. Um TF pode interagir num Sítio de Ligação e ativar ou inibir o gene associado, e, então, incrementar ou decrementar a produção de proteínas, numa taxa específica.

O processo de Michaelis-Menten foi definido há mais de 100 anos e mostra que a taxa de reação catalizada para uma enzima é proporcional à concentração do complexo substrato-enzima, conforme Equação 1.

A taxa de reação varia linearmente com a concentração de substrato [S] (cinética de primeira ordem). No entanto, em níveis mais altos [S] com $[S] \gg K_M$, a reação se torna independente de [S] (cinética de ordem zero) e se aproxima assintoticamente de sua taxa máxima V_{max} [Johnson and Goody 2011]. O processo de regulação entre os genes emerge de suas interações no Sítio de Ligação e pode ser modelado como um processo de cinética enzimática, também conhecido como cinética de Michaelis-Menten [Johnson and Goody 2011]. A taxa de reação varia linearmente com a concentração do substrato [S] (cinética de primeira-ordem). O comportamento da curva de Michaelis-Menten (Figura 2) é usado para comparações com os resultados do modelo proposto.

$$v = \frac{d[P]}{dt} = \frac{V_{max}[S]}{K_M + [S]} \quad (1)$$

3. Modelo Proposto

O modelo proposto é uma abstração do processo bioquímico subjacente e simula a interação entre TF e um gene alvo. As ferramentas MAS são muito flexíveis e são capazes de representar cada indivíduo no sistema, simplificando a avaliação de novas hipóteses sobre o modelo.

O simulador é implementado no NetLogo¹, um software orientado a agentes utilizando a linguagem Logo. Dois arquivos de configuração são definidos para configurar o modelo: definições dos agentes e definições das restrições (Tabela 1a e 1b). Importante lembrar que no NetLogo, cada etapa do tempo é representada por um “ tick ”.

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

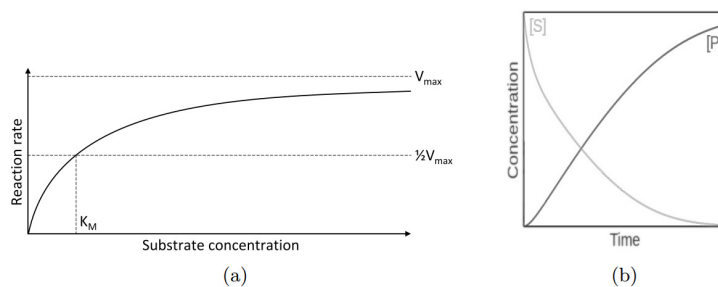


Figura 2. (a) Curva da taxa de reação de Michaelis-Menten. (b) Variação típica na concentração do substrato.

Tabela 1. Arquivos de configuração

(a)					(b)			
Id	Tam	Cor	Qtd	Gene	Regulador	Ação	Regulado	Qtd
0	1	branco	0	“X”	“Y”	1	“X”	3
1	1	vermelho	30	“Y”				

Na Tabela 1a apresenta o arquivo de entrada com as definições dos agentes, onde tem-se 0 gene “X” e 30 genes “Y”. É definido cada gene em termos de tamanho (em plano bi-dimensional), cor, quantidade e nome. Na Tabela apresenta o arquivo de entrada para as restrições entre as ações dos agentes, onde a primeira coluna define o nome do gene regulador e a terceira define o gene regulado. A segunda coluna é a definição da ação. No exemplo, a ação de “Y” em “X” é ativação (Ação 1) e 3 agentes de “Y” são requeridos para produzir cada novo gene do tipo “X”. O ambiente do simulador é apresentado na Figura 3, onde “X” e “Y” são representados, respectivamente, com círculos brancos e vermelhos. Existe um quadrado verde no ambiente que representa o Sítio de Ligação para “Y”.

4. Resultados

No atual estado de desenvolvimento do simulador, os genes (agentes) interagem em Sítios de Ligação, de acordo com as configurações definidas e o comportamento apresentado na Figura 3. No NetLogo cada passo é representado por um “tick”. A Figura 3a apresenta a simulação com os genes e o Sítio de Ligação depois de alguns “ticks” de simulação. Na Figura 3b o ambiente de simulação é apresentado após 1684 “ticks”. O painel a direita da Figura 3c apresenta a concentração de genes “X” e “Y” ao final da simulação (1684 “ticks”). Salientando que o valor 1684 é o valor no qual consegue-se observar a estabilidade no sistema, ou seja, os TF param de movimentar-se no ambiente.

Salienta-se aqui que o gráfico resultante desta simulação é representado pela concentração de TF em X *versus* “ticks” e concentração de produto em Y *versus* “ticks”.

5. Conclusões

Segundo nosso conhecimento, esta é a primeira tentativa de simular GRN usando um *framework* Multiagente. No presente estudo um MAS é definido com poucas regras e parâmetros. Depois de executar a simulação, a concentração das séries temporais dos

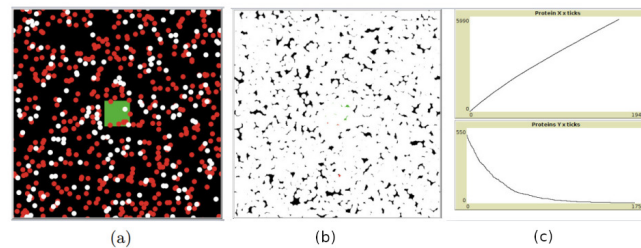


Figura 3. (a) Ambiente de simulação. Os Sítios de Ligação são representados por quadrados verdes e os agentes por círculos brancos e vermelhos. (b) Estado do ambiente de simulação após 1684 ticks. (c) Gráficos de concentração temporal dos agentes (genes).

genes foram apresentadas na Figura 3c. Essas curvas apresentam comportamento similar a curva de Michaelis-Menten apresentada na Figura 2b. Apesar de esses serem resultados preliminares, eles indicam que existe um bom local para a pesquisa.

Pretende-se validar esta abordagem comparando-a com o trabalho de [Alon 2007], cujos resultados são apresentados na Figura 4. A Figura 4 mostra o resultado do TF em Y e do produto em Z resultante da interação proveniente do circuito representado na mesma figura. Além disso, pretende-se comparar os resultados deste trabalho com os estudos realizados com o modelo do ciclo circadiano da planta *Arabidopsis thaliana* [Pokhilko et al. 2012] e com o Bio-PEPA [Haydarlou et al. 2016].

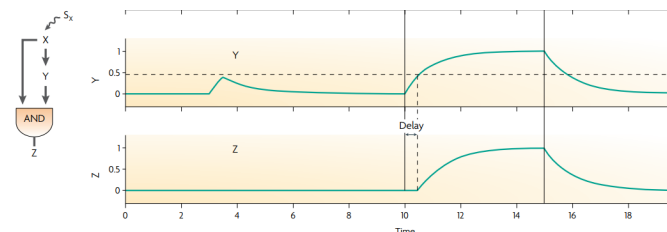


Figura 4. Circuito regulatório e concentrações resultantes de seus componentes [Alon 2007].

6. Agradecimentos

Gostaríamos de agradecer à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro pela bolsa de estudos.

Referências

- Agostinho, N., Werhli, A., and Adamatti, D. (2018). Revisão sistemática para o desenvolvimento de um ambiente de simulação multiagente para vias regulatórias. *RETEC. REVISTA DE TECNOLOGIAS (OURINHOS)*, 11.
- Alon, U. (2007). Network motifs: theory and experimental approaches. *Nature Reviews Genetics*.
- Ezer (2014). Homotypic clusters of transcription factor binding sites: A model system for understanding the physical mechanics of gene expression. *Comput Struct Biotechnol J*.

- Ghazikhani, A., Akbarzadeh, T., and Monsefi, R. (2011). Genetic regulatory network inference using recurrent neural networks trained by a multi agent system. *International Conference on Computer and Knowledge Engineering (ICCCKE)*.
- Gibas, C. and Jambeck, P. (2002). Developing bioinformatics computer skills. *Yale Journal of Biology and Medicine*.
- Haydarlou, R., Jacobsen, A., Bonzanni, N., Feenstra, K. A., Abeln, S., and Heringa, J. (2016). Bioasf: a framework for automatically generating executable pathway models specified in biopax. *Bioinformatics*, 32:i60–i69.
- Johnson, K. and Goody, R. (2011). The original michaelis constant: Translation of the 1913 michaelis-menten paper. *Biochemistry*.
- Khan, S., Makkena, R., McGeary, F., Decker, K., Gillis, W., and Schmidt, C. (2003). A multi-agent system for the quantitative simulation of biological networks. *AAMAS*.
- Liu, J., Chi, Y. Z. C., and Zhu, C. (2016). A dynamic multiagent genetic algorithm for gene regulatory network reconstruction based on fuzzy cognitive maps. *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, 24.
- Lopes, F. M. (2011). Redes complexas de expressao genica: sintese, identificacao, analise e aplicacoes. Master's thesis, Universidade de Sao Paulo.
- Mariano, D., Leite, C., Santos, L., Rocha, R., and Melo-Minardi, R. (2017). A guide to performing systematic literature reviews in bioinformatics. Technical Report RT.DCC.002/2017, Universidade Federal de Minas Gerais.
- Pham, D. (2008). Multi-agent based simulation of large random boolean network. *MSC-LES*.
- Pokhilko, A., Fernandez, A., Edwards, K., Southern, M., Halliday, K., and Millar, A. (2012). The clock gene circuit in arabidopsis includes a repressilator with additional feedback loops. *molecular systems biology*. *Molecular System Biology*, 8.
- Sanfilippo, A., Haack, J., McDermott, J., Stevens, S., and Stenzel-Poore, M. (2012). Modeling emergence in neuroprotective regulatory networks. *International Conference on Complex Sciences - springer*, pages 291–302.
- Schreiber, S. (2005). Small molecules: the missing link in the central dogma. *Nature Chemical Biology*.
- Yang, T. and Sun, Y. (2011). The reconstruction of gene regulatory network based on multi-agent system by fusing multiple data sources. *2011 IEEE International Conference on Computer Science and Automation Engineering*, 4.

Simulação baseada em agentes para a gestão de recursos hídricos *

Julia P. C. Pereira¹, Diana F. Adamatti², Raquel M. Barbosa¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - IFRS
Rua Engenheiro Alfredo Huch, 475 – CEP: 96201-460 – Rio Grande – RS – Brasil

²Centro de Ciências Computacionais (C3)
Universidade Federal do Rio Grande (FURG) – Rio Grande – RS – Brasil

{juliapcp2003, dianaada, raq.mbarbosa}@gmail.com


Abstract. *This paper aims to present the study about the participatory management of water resources from simulations developed in the multiagent environment NetLogo. In this way, the work uses the NetLogo to simulate the modeled social situation that comprises active agents such as farmers, businessmen, mayors, councilors, members of NGOs and environmental inspectors in search of the management of their professional interactions in favor of the more efficient use and distribution of the water, with a view to reducing pollution with good practices. The multiagent system presented assumes a fundamental role in solving the proposed problem, providing relevant information and indicators for the study and prevention of the impacts of agriculture on the environment.*

Resumo. *Este artigo tem como objetivo apresentar o estudo sobre a gestão participativa de recursos hídricos a partir de simulações desenvolvidas no ambiente multiagente NetLogo. Desta maneira, o trabalho utiliza o ambiente para simular a situação social modelada que compreende agentes atuantes como agricultores, empresários, prefeitos, vereadores, membros de ONGs e fiscais ambientais em busca da gestão de suas interações profissionais em prol do uso e distribuição mais eficientes da água, visando a diminuição da poluição com boas práticas. O sistema multiagentes apresentado assume papel fundamental na resolução do problema proposto, provendo informações e indicadores relevantes para o estudo e prevenção dos impactos da agricultura no meio ambiente.*

1. Introdução

Atualmente a produção agrícola é um investimento de grande rentabilidade, sendo um dos impulsionadores da economia nacional. Alguns insumos, dentre eles fertilizantes químicos, agrotóxicos e máquinas agrícolas, ajudam na viabilização de plantações de alta produtividade e lucro, bem como de alto impacto ambiental, principalmente referente à poluição das águas de mares, lagos e rios próximos [Dellamatrice and Monteiro 2014].

A poluição exacerbada da água interfere diretamente na saúde das pessoas que a consomem, como mostrado no estudo realizado por [Teixeira et al. 2014], onde provou-se

* O trabalho Simulação baseada em agentes para a gestão de recursos hídricos de Julia P. C. Pereira, Diana F. Adamatti, Raquel M. Barbosa está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-Compartilha Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

que entre os anos de 1999 e 2009 houveram registros de quase 10 mil casos de intoxicação proveniente de agrotóxicos no Nordeste do Brasil. Além dos impactos à saúde humana, a longo prazo, o agrotóxico diminui a produtividade da plantação de alimentos, como houve comprovação por [Bontempo et al. 2016], ao verificar que plantações de cenoura localizadas em áreas onde aplicou-se o herbicida tembotriona 8 meses antes tiveram sua produtividade afetada. Este problema faz necessária a sobredosagem de insumos danosos, promovendo um ciclo de reparação de difícil reversão e sem previsibilidade de impacto a gerações futuras.

Em conformidade com [Millington and Funge 2009], diz-se que a Inteligência Artificial (IA) trata sobre computadores que são capazes de executar tarefas ‘pensantes’, que humanos e animais são capazes de executar. Como uma de suas vertentes, as simulações desenvolvidas em sistemas multiagente (SMA), como a ferramenta NetLogo, possibilitam a atuação de entidades autônomas em um ambiente comum, visualizando seus impactos no universo a partir de seu pertencimento a um grupo de agentes. Os agentes podem trabalhar de forma cooperativa ou antagônica, executando tarefas previamente definidas e interagindo com outros grupos e o universo, produzindo indicadores para avaliação individual e mapeamento de decisões ideais.

Segundo [Drogoul and Ferber 1992], alguns dos objetivos de simulações baseadas em modelos multiagente são testes de hipóteses sobre a emergência de comportamentos no nível macro a partir de interações no nível micro e a construção de teorias que contribuam para o entendimento de fenômenos relacionados a comportamentos. Desta maneira, o presente estudo procura simular diferentes cenários do problema a partir da atuação de diferentes entidades, em busca da obtenção de indicadores e informações pertinentes para o mapeamento da gestão participativa de recursos hídricos na agricultura, solidificando previsões de impacto.

O presente artigo está estruturado conforme descrito a seguir: A Seção 2 apresenta a ferramenta NetLogo, utilizada para o desenvolvimento da simulação. A Seção 3 descreve a modelagem do problema abordado e sua implementação é detalhada na Seção 4. Por fim, a Seção 5 apresenta a conclusão e os próximos passos deste trabalho.

2. NetLogo

O modelo desenvolvido e apresentado neste trabalho para a realização de testes de impactos das relações provenientes da agricultura foram desenvolvidos em linguagem Logo no ambiente integrado multiagente NetLogo [Wilensky et al. 1999]. Por ser um sistema de modelagem programável que possibilita a simulação de fenômenos naturais e sociais, o NetLogo viabiliza interagir com passagem de tempo e impactos que evoluem no mesmo.

Sistemas baseados em agentes como o NetLogo permitem a criação de agentes responsáveis por suas ações, desempenhando suas funções em um ambiente comum, interagindo uns com os outros. Essa funcionalidade foi aproveitada para a criação de papéis dentro do modelo proposto, de maneira que cada entidade do sistema tem um conjunto de funcionalidades na sociedade projetada, desempenhando funções que dependem do desempenho de outro na resolução de um problema em comum, formando uma dependência muito semelhante às relações interpessoais da realidade.

Ao prover ferramentas e interface que tornam a simulação personalizável quanto a número de agentes e outras variáveis, o ambiente permite a compreensão de diversos

cenários e captura de diversos impactos possíveis, buscando a fidedignidade dos fatos ao verificar possíveis mudanças provenientes de números maiores ou menores de determinado agente, bem como a capacidade de simular fenômenos da natureza, variável importante quando nos referimos à simulação ligadas à sociedade se relacionando com o meio ambiente.

3. Modelagem do problema

De acordo com [Dellamatrice and Monteiro 2014], a contaminação da água ocorre principalmente em áreas próximas aos locais de aplicação de insumos químicos de tratamento de solo por deflúvio superficial ou contaminação do lençol freático. Alguns fatores que afetam o transporte para o meio aquático são as propriedades do agente químico em questão e as condições climáticas, como as chuvas. Dessa maneira, o cenário proposto apresenta um ambiente que aborda a agricultura e as relações geradas por ela. Os agentes modelados para resolver o problema da contaminação da água pertencem a três grupos: produtores, fiscalizadores e reguladores.

As relações de integração entre os grupos de agentes foram definidas em [Leitzke et al. 2019] e são apresentadas na Figura 1.

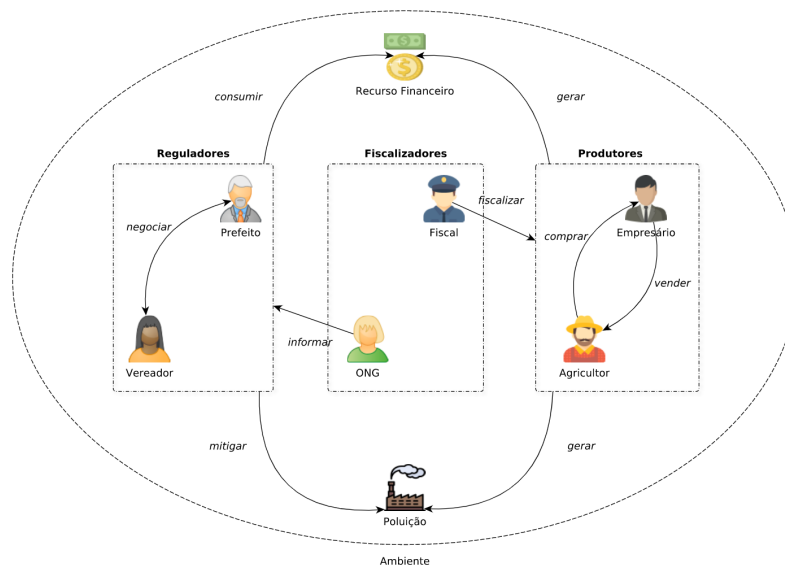


Figura 1. Diagrama de Integração. Fonte: [Leitzke et al. 2019]

No âmbito dos produtores, são especificadas duas identidades distintas, agricultores e empresários. Estes têm o papel de explorar sua área de atuação visando sucesso financeiro, de modo que suas relações entre si e com o meio ambiente envolvem a produção e uso de artefatos auxiliares que aumentam a produtividade e a rentabilidade das plantações, como agrotóxicos, fertilizantes e maquinários, trazendo, assim, uma maior quantidade de movimentação financeira e, conseqüentemente, de poluição ao ambiente que atuam em conjunto.

Para a fiscalização e punição pelo uso exacerbado de insumos danosos e poluentes provenientes das ações dos agentes produtores, precisa-se da constante vigília vinda de

membros de ONGs e fiscais, importantes para a aplicação de multas e denúncias pela poluição elevada. Por buscar a veracidade dos fatos em relação à realidade, é previsto que possa haver desonestidade por parte dos agentes fiscalizadores, levando em consideração seus interesses nas propostas feitas pelos agentes produtores.

Dentro do grupo de reguladores compreendem-se os agentes que utilizam as denúncias e multas dos agentes fiscalizadores para promulgar políticas e leis que reduzem a poluição, os prefeitos e vereadores. Estes estão encarregados de estabelecer quais são os níveis recomendados de poluição, mediando essa taxa sem comprometer as relações de produção, importantes para a renda do cenário o qual gerem, enriquecendo a dualidade de estratégias que pode ser seguida pelo agente.

Estes agentes, de diferentes organizações, são modelados para possibilitar a visualização das interações ideais para uma melhor gestão de insumos para a diminuição da poluição da água no cenário comum de duas cidades. Este cenário é composto de terras denominadas posse de agricultores aleatórios, prefeitura e fábricas, com passagens de rios entre as extensões de terra, que não são da posse de nenhum dos agentes, mas são utilizados por todas as propriedades nas imediações das cidades, tanto para a produção quanto para o descarte de resíduos, refletindo diretamente em todas as entidades.

É previsto que os agricultores possam comprar insumos dos empresários, apaziando os níveis de produtividade e poluição provenientes do uso do produto, bem como seu preço para adquirir ou não os insumos, inclinando-se ora a produzir mais, ora a poluir menos. Cada empresário oferece um produto dentre agrotóxicos, fertilizantes, sementes, máquinas, podendo haver mais representantes do mesmo produto, mas nunca nenhum.

Planeja-se que os fiscais utilizem-se dos indicadores de poluição para aplicar multas nos mesmos, baseando-se nos indicadores de poluição do consumo e produção destes agentes para verificar a quem aplicar multas baixas, médias ou altas. Ao visualizar que o nível de poluição gerada pelo agente produtor é nula ou baixa, o fiscal pode conceder selos verdes, que fazem com que o imposto pago pelo agente seja 5% mais baixo.

Os prefeitos e vereadores das cidades são responsáveis por tomar medidas de diminuição e prevenção da poluição e aumentar ou diminuir impostos, que são calculados para cada agente produtor em uma porcentagem da produção do mesmo.

O presente trabalho contribui com o domínio ao avaliar as configuração das variáveis e validação de valores projetados no ambiente, analisando o comportamento do sistema e das entidades em diferentes cenários e configurações de parâmetros, enriquecendo a modelagem e o escopo do projeto como um todo.

4. Implementação e testes

A simulação atualmente encontra-se em processo de desenvolvimento, formando rotinas de acordo com as possíveis escolhas modeladas dos agentes, com algumas funcionalidades já implementadas que permitem a retenção de informações parciais sobre a situação a partir de variáveis personalizadas pelo visualizador em cada simulação. Dentro do proposto, taxas de gravidade de poluição variam de 0 a 125, onde 0 é a poluição nula e números entre 115 e 125 representam a poluição mais elevada, causando um caos ambiental.

No primeiro teste configura-se a simulação com 4 empresários *num* –

empresarios, entidades às quais foram atribuídas rotinas de produção de insumos para venda de maneira aleatória. Esta função gera elevada poluição, devido ao descarte de resíduos provenientes das indústrias no rio, que se torna mais escuro devido à contaminação. Essa interação chegou ao nível de caos ambiental no tempo correspondente a 364 'ticks' (medidas de tempo do ambiente, relacionadas diretamente com a unidade de tempo dia).

A seguir foram realizados testes com um aumento na quantidade de empresários *num – empresarios*, que possibilitou um aumento de poluição até o nível máximo aos 231 ticks, conforme pode ser visualizado na Figura 2. Percebe-se que o caos ambiental ocorreu em um tempo substancialmente menor do que o apresentado no primeiro teste. Conclui-se, então, que um aumento no número de empresários (representando mais indústrias) gera a necessidade de uma maior produção de insumos, aumentando o índice de poluição em busca da necessidade de haver produtos melhores a serem oferecidos aos agricultores, fenômeno incitado pela concorrência entre as empresas. A tomada de decisão do agricultor na compra dos insumos ofertados pelas empresas é descrita por

$$probabilidadeCompra = \frac{producaoGerada}{poluicaoGerada + preco} \quad (1)$$

onde *probabilidadeCompra* é o valor avaliado pelo agricultor na compra de um insumo, *producaoGerada* é a produtividade gerada ao utilizar o insumo na plantação, bem como *poluicao* corresponde à poluição proveniente da utilização do insumo e *preco* é o preço proposto pelos empresários na compra do produto ofertado.

Projeta-se que um aumento na quantidade de agentes fiscalizadores proporcional ao aumento de empresários estabilize o aumento de poluição em função do tempo.

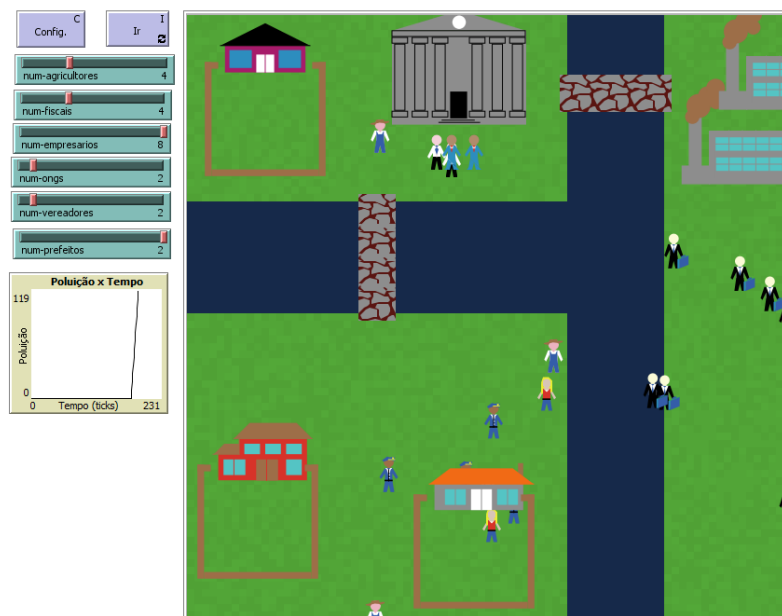


Figura 2. Simulação configurada com oito empresários.

As implementações seguintes permitirão a visualização do cenário como um todo, ao disponibilizar fiscalizações, produção, vendas e promulgações de leis partindo dos

agentes, visando não atingir o caos ambiental a partir do momento em que cada agente realize sua função.

5. Conclusão e trabalhos futuros

O presente artigo apresentou o desenvolvimento parcial de um trabalho no ambiente multiagente NetLogo, com base na temática da poluição da água a partir das relações provenientes da agricultura, vinculando temáticas sociais, ecológicas e computacionais em busca da obtenção de índices utilizáveis na mudança das gestões participativas que envolvem o bem-estar ambiental nesses meios.

O desenvolvimento encontra-se no estágio da implementação das funções modeladas, muito importantes para entender a temática abordada e as características e impactos de cada agente no universo, de maneira a nortear a simulação para a obtenção de dados da forma mais efetiva e garantindo a abstração de suas interações mais importantes. Dessa maneira, as seguintes etapas configuram-se o término das implementações no sistema integrado NetLogo, bem como a revisão da simulação ao final desta etapa, resultando na obtenção e análise de indicadores a partir do uso da simulação, enriquecedoras no aspecto de contribuição ao projeto descrito em [Leitzke et al. 2019].

Agradecimentos

Gostaríamos de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e à Agência Nacional das Águas (ANA) pelo apoio financeiro via Edital nº 16/2017.

Referências

- Bontempo, A. F., Carneiro, G. D., Guimarães, F. A., Dos Reis, M. R., Silva, D. V., Rocha, B. H., Souza, M. F., and Sediya, T. (2016). Residual tembotrione and atrazine in carrot. *Journal of Environmental Science and Health, Part B*, 51(7):465–468.
- Dellamatrice, P. M. and Monteiro, R. T. (2014). Principais aspectos da poluição de rios brasileiros por pesticidas. *Revista Brasileira de Engenharia Agrícola e Ambiental*, 18(12):1296–1301.
- Drogoul, A. and Ferber, J. (1992). Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 2–23. Springer.
- Leitzke, B., Farias, G., Melo, M., Gonçalves, M., Born, M., Rodrigues, P., Martins, V., Barbosa, R., Aguiar, M., and Adamatti, D. (2019). Sistema multiagente para gestão de recursos hídricos: Modelagem da bacia do São Gonçalo e da Lagoa Mirim. In *Anais do X Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais*, pages 87–96. SBC.
- Millington, I. and Funge, J. (2009). *Artificial intelligence for games*. CRC Press.
- Teixeira, J. R. B., Ferraz, C. E. d. O., Couto Filho, J. C. F., Nery, A. A., and Casotti, C. A. (2014). Intoxicações por agrotóxicos de uso agrícola em estados do nordeste brasileiro, 1999-2009. *Epidemiologia e Serviços de Saúde*, 23:497–508.
- Wilensky, U. et al. (1999). Center for connected learning and computer-based modeling. In *NetLogo*. Northwestern University.

Um estudo do Myers-Briggs Type Indicator (MBTI) para modelagem de sistemas multiagentes no apoio a processos de recrutamento e seleção nas empresas*

Luiz Fernando Braz¹, Jaime Simão Sichman¹

¹Laboratório de Técnicas Inteligentes (LTI)
Escola Politécnica (EP)
Universidade de São Paulo (USP)

luiz.braz@usp.br, jaime.sichman@usp.br

Abstract. *The Myers-Briggs Type Indicator, inspired by the theory of psychological types, seeks to identify the behavioral preferences of individuals around four typological classifications named as dichotomies. In this way, some studies address the use of multi-agent systems based on the personalities used in the MBTI, investigating through simulations the behavior of the agents, as well as the results of their interactions in the environment. In this paper, it is explored how new approaches can expand the simulation of agents in order to create high-performance teams, thus enabling tools to support recruitment and selection processes to be developed in the future.*

Resumo. *O Myers-Briggs Type Indicator, inspirado na teoria dos tipos psicológicos, busca identificar as preferências comportamentais dos indivíduos em torno de quatro classificações tipológicas chamadas de dicotomias. Neste sentido, alguns estudos abordam o uso de sistemas multiagentes baseados nas personalidades utilizadas no MBTI, investigando através de simulações o comportamento demonstrado pelos agentes, bem como os resultados de suas interações no ambiente. Neste trabalho, é explorado como novas abordagens podem ampliar a simulação dos agentes visando a formação de times de alto desempenho, possibilitando assim que ferramentas de apoio a processos de recrutamento e seleção possam ser desenvolvidas no futuro.*

1. Introdução

O estudo do comportamento humano no trabalho visando sua classificação em grupos de similaridade de perfis tem sido tema de diversas pesquisas ao longo dos anos. Desde a fundação do laboratório de estudos da fadiga, por Luigi Patrizi em 1889 e com o surgimento de uma nova especialização da Psicologia denominada Psicologia do Trabalho [Bastos et al. 2004], vários pesquisadores têm se dedicado a analisar problemas relacionados à produtividade visando que estes novos conhecimentos gerados pudessem ser aplicados nas organizações. Em [Koppes 2014] apresentam-se diversos estudos, principalmente da segunda metade do século 19, que tentavam maximizar o uso dos



* O trabalho Um estudo do Myers-Briggs Type Indicator (MBTI) para modelagem de sistemas multiagentes no apoio a processos de recrutamento e seleção nas empresas de Luiz Fernando Braz, Jaime Simão Sichman está licenciado com uma Licença Creative Commons - Atribuição-CompartilhaIgual 4.0 Internacional. <https://creativecommons.org/licenses/by-sa/4.0/>

recursos humanos no trabalho, buscando aplicar as noções de energia que eram aplicadas a outras áreas de conhecimento às situações de trabalho humano. Em [Thompson & Pozner 2007] também nota-se que a escala competitiva fez com que a demanda por medidas de desempenho se tornassem cada vez mais necessárias e com isso, surgiram estudos relacionados à aplicação de testes psicológicos para auxiliar as organizações na seleção de pessoas mais aderentes às suas necessidades. Com William Stern, em 1903, nasce o termo “psicotécnica” [Ibarra 2015], e a realização de pesquisas utilizando métodos experimentais permitem que diversos outros pesquisadores desenvolvam e apliquem testes psicológicos, principalmente na seleção de pessoal. Hugo Münsterberg, em sua publicação de 1913 “Psychology and Industrial Efficiency”, busca relacionar problemas encontrados no mundo dos negócios com experimentos psicológicos conduzidos em laboratório e com isso demonstrar como a psicologia poderia auxiliar com técnicas e métodos eficazes na seleção de pessoal, tentando correlacionar habilidades pessoais com a maximização da produtividade em um determinado contexto organizacional. Münsterberg também analisa neste estudo que padrões associados à fadiga, treinamento, capacidade intelectual, atitude e personalidade de um determinado indivíduo podem influenciar sua produtividade em um ambiente de trabalho [Koppes, 2014]. A partir de então, o uso de técnicas que permitam a identificação de padrões psicológicos de comportamento das pessoas tem sido amplamente utilizado nas organizações visando por meios que possibilitem principalmente a formação de times de alto desempenho.

O Myers-Briggs Type Indicator (MBTI), [Myers et al. 1998], oriundo da teoria da psicologia da personalidade humana, é amplamente conhecido e busca realizar através de um teste psicológico a classificação de diferentes tipos de personalidades e interesses das pessoas, visando sua categorização em 16 grupos distintos. Através dele, é possível identificar preferências e características pessoais que possibilitam utilizá-la em variadas situações, desde o seu uso para o próprio autoconhecimento quanto para aprimoramento de métodos educativos adaptados às características de personalidade dos estudantes [Mupinga et al. 2006]. Desta forma, é interessante imaginar como pessoas com perfis comportamentais diversos poderiam interagir entre si visando um objetivo comum, como por exemplo na realização de um trabalho em equipe.

Neste contexto, a utilização de uma abordagem baseada em sistemas multiagentes (SMA) pode proporcionar um instrumento importante para simulação de interações dentro de uma organização específica. Em [Alvares & Sichman 1997 apud Demazeau e Müller 90], os autores destacam que a “área de SMA se interessa pelo estudo de agentes autônomos em um universo multiagentes”, em que o objetivo, segundo [Alvares & Sichman 1997] é “estudar modelos genéricos a partir dos quais podem-se conceber agentes, organizações e interações”, visando os meios para que os agentes possam cooperar na resolução de um problema específico [Alvares & Sichman 1997].

O objetivo futuro deste trabalho é portanto utilizar uma abordagem de sistemas multiagentes para simular como o MBTI pode ser aplicado como ferramenta de apoio a processos de recrutamento e seleção nas empresas, visando a formação de times de alto desempenho. Na seção 2 são descritos o MBTI e suas classificações tipológicas, na seção 3 as possibilidades de sua aplicação para modelagem de agentes e, por último, na seção 4 as considerações finais e próximos passos visando a ampliação do estudo para simulações envolvendo times com múltiplas personalidades.

2. Myers-Briggs Type Indicator – MBTI

A teoria dos tipos psicológicos, concebida por Carl Jung em 1921, foi a fonte de inspiração para o desenvolvimento do MBTI. Basicamente o propósito principal era fazer com que a teoria de Jung fosse mais bem compreendida e que pudesse de fato ser aplicada na vida das pessoas [Myers et al. 1998].

Neste sentido, o objetivo de Isabel Myers e sua mãe, Katherine Briggs, no desenvolvimento do MBTI, foi fazer com que os *insights* trazidos pela Teoria dos Tipos Psicológicos fossem acessíveis a indivíduos e grupos, e para isso elas buscaram endereçar duas questões: A primeira é relativa à identificação das preferências básicas de cada uma das 4 dicotomias propostas, de forma implícita ou explícita por Jung e a segunda diz respeito à identificação e descrição de 16 tipos de personalidades distintas que resultam das interações em torno destas preferências [Myers et al. 1998]. Desta forma, foram propostas quatro classificações tipológicas, chamadas de dicotomias, que indicam quais são as preferências principais do indivíduo quanto ao seu comportamento. Elas são classificadas como Extroversão ou Introversão (E-I), Sensorial ou Intuitivo (S-N), Racional ou Sentimental (T-F) e Julgador ou Perceptivo (J-P), em que para cada uma delas é observado uma preferência principal. Estas classificações ajudam a identificar indivíduos que são por exemplo mais extrovertidos que outros, não significando que eles serão extrovertidos o tempo todo, mas sim que terão uma tendência maior a serem mais sociáveis e expressivos [Myers et al. 1998].

2.1. Extroversão-Introversão (E-I)

Esta primeira dicotomia busca explicar como os indivíduos tendem a direcionar suas energias através da interação com o mundo externo ou interno [Myers et al. 1998]. Indivíduos com preferência à extroversão tendem a direcionar suas ações para interação com o mundo externo, buscando ser sociáveis, se comunicando através da fala, aprendendo com outros indivíduos e tendo facilidade com a proatividade. Já indivíduos com preferência a introversão buscam a privacidade, preferindo se comunicarem através de outros meios, como a escrita, aprendendo individualmente e sendo proativos apenas quando necessário [Salvit 2012].

2.2. Sensorial-Intuitivo (S-N)

Esta dicotomia busca descrever como um indivíduo tende a buscar informações do mundo, seja através de mecanismos sensoriais ou de sua intuição. Indivíduos sensoriais, buscam a percepção do mundo através dos seus cinco sentidos, visão, audição, olfato, paladar e tato. Já, os indivíduos intuitivos, buscam através de sua intuição o entendimento de padrões e interações que tragam algum significado para suas percepções, sendo este um processo menos óbvio e mais sujeito a uma compreensão subjetiva do mundo [Myers et al. 1998].

2.3. Racional-Sentimental (T-F)

A terceira dicotomia busca descrever como os indivíduos tomam decisões [Salvit 2012], seja através de análises lógicas com foco em fatores objetivos, ou através de conclusões baseadas em valores pessoais ou sociais, com foco no entendimento e harmonia do ambiente em que se encontram. Os primeiros, considerados como racionais, preferem decidir de forma impessoal, baseado na consequência lógica de suas decisões, já

indivíduos sentimentais, buscam tomar suas decisões baseadas em valores pessoais ou sociais [Myers et al. 1998]. Um ponto importante é que segundo [Myers et al. 1998] o termo racional não implica em inteligência ou competência, assim como o termo sentimental, também não deve ser confundido como emoções, sendo estas expressões independentes da teoria proposta por elas.

2.4. Julgadores-Perceptivos (J-P)

Esta última dicotomia descreve como os indivíduos lidam com o mundo exterior através de suas atitudes e orientações, em que indivíduos Julgadores são aqueles que tendem a utilizar suas preferências racionais ou sentimentais quando lidam com o mundo exterior [Myers et al. 1998], segundo [Salvit 2012] isto significa que eles gostam de tomar decisões, de chegar a conclusões, de ter rotinas e planos claros para serem seguidos. Já um indivíduo Perceptivo, têm preferência de usar os seus sentidos ou sua intuição quando lida com o mundo exterior [Myers et al. 1998], também de acordo com [Salvit 2012] isto significa que eles gostam de estar com a mente aberta a novas opções, não realizando planos ou rotinas objetivas e muitas vezes chegando a conclusões de última hora.

2.5. Os 16 tipos consolidados

Com as dicotomias, observa-se que cada indivíduo tem preferência por um determinado tipo de comportamento, fazendo com que todas as combinações possíveis proporcionem $4^2 = 16$ tipos de personalidades distintas [Salvit 2012], sendo que cada uma delas é denotada como o conjunto das quatro letras identificadas para cada comportamento (ex: ESTJ, INFP)[Myers et al. 1998]. A tabela 1 (adaptado de [Salvit, 2012]) ilustra como os diferentes tipos de personalidade são compostos.

Tabela 1. Os tipos de personalidade (adaptado de [Salvit 2012])

		Sensorial (S)		Intuitivo (N)	
		Racional (T)	Sentimental (F)	Racional (T)	Sentimental (F)
Introvertido (I)	Julgador (J)	ISTJ	ISFJ	INTJ	INFJ
	Perceptivo (P)	ISTP	ISFP	INTP	INFP
Extrovertido (E)	Julgador (J)	ESTJ	ESFJ	ENTJ	ENFJ
	Perceptivo (P)	ESTP	ESFP	ENTP	ENFP

3. Modelagem de agentes com o MBTI

Em [Salvit 2012] é realizado um estudo da aplicação do MBTI para modelagem de agentes que possam representar personalidades e comportamentos seguindo a teoria dos tipos de personalidade. Nele também é proposto a extensão do BDI (*Belief-Desire-Intention*), framework para modelagem de comportamento de agentes, que permite, segundo [Rao & Georgeff 1995] representar “as informações, motivações, e estados deliberativos de um agente”, e com isso analisar como agentes, com diferentes personalidades, poderiam agir visando a conclusão de uma determinada tarefa.

No estudo de [Salvit 2012] é apresentado que agentes com diferentes tipos de personalidade, seguindo o modelo de dicotomias do MBTI, podem manifestar diferentes comportamentos na realização das mesmas tarefas, em que [Salvit 2012] ressalta que no ambiente de simulação, alguns agentes com determinados tipos de personalidade se destacam na realização de certas tarefas do que outros agentes, o que, segundo ele, é possível observar também em situações da vida real dos seres humanos. Assim, para potencialização de resultados neste ambiente simulado, pode-se esperar que seja preferível a utilização de agentes com tipos de personalidade que melhor se adequem à tarefa e ao ambiente em que se encontram, visto que nas observações realizadas estes atingem o objetivo com maior produtividade.

Com os experimentos de [Salvit 2012] pode-se observar a produtividade de um agente com um determinado perfil de personalidade em um ambiente homogêneo, em que todos os agentes têm o mesmo tipo de personalidade, ou em um ambiente heterogêneo, em que o agente em questão está situado com outros agentes de diferentes tipos de personalidade. Porém, sabendo-se que especialmente no mundo corporativo, a composição de equipes de trabalho visando alto desempenho depende da interação de pessoas entre si, em que o resultado geralmente é comum ao time (e não somente ao indivíduo) seria importante que o estudo de [Salvit 2012] possa ser também complementado com análises que contemplem não só o indivíduo, ou o agente, mas também a composição de agentes que formam uma determinada equipe de trabalho.

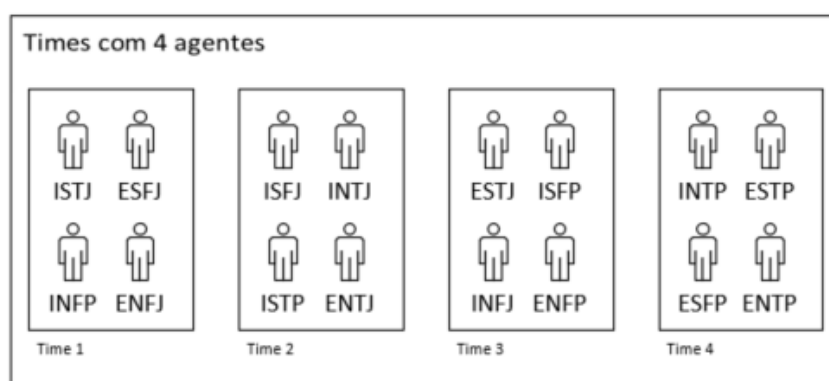


Figura 1. Times com múltiplos tipos de personalidade

Desta forma, conforme demonstrado na figura 1, times diversos poderiam ser formados, possibilitando que seja observado se equipes compostas por certos tipos de personalidade poderiam ter maior desempenho do que outras sem a mesma formação. Como exemplo, pode-se imaginar que seja preferível times de vendas formados apenas por pessoas extrovertidas, visto que este tipo de atividade geralmente necessita de pessoas que gostem de se comunicar com os outros (direcionando suas energias para o mundo exterior), por outro lado, talvez formar um time com pessoas com diferentes tipos de personalidade poderia proporcionar maior dinamismo e flexibilidade resultando em maior integração e desempenho a médio e longo prazo.

4. Considerações Finais e Próximos Passos

O presente trabalho apresentou uma revisão de estudos anteriores em que buscou-se aprofundar o MBTI, proposto por Isabel Myers e Katherine Briggs, com o entendimento

de cada uma das dicotomias concebidas na teoria. Também buscou-se compreender como uma abordagem baseada em SMA pode ser utilizada para simulação de agentes com personalidades definidas através nos tipos de personalidade utilizados no MBTI, analisando os experimentos conduzidos por [Salvit 2012] e identificando uma possível nova abordagem a partir deste estudo.

Como trabalho futuro, é possível ampliar o uso de SMA para simulações envolvendo times compostos por múltiplos tipos de personalidade decorrentes do MBTI, possibilitando desta forma um melhor entendimento de como resultados que sejam comuns a um time (e não somente a um agente específico) podem ser alcançados com sucesso. Este estudo poderia proporcionar avanços significativos principalmente em processos de recrutamento e seleção, apoiando as organizações em um melhor entendimento de como fatores relativos à diversidade de personalidades podem impactar a formação de times de alto desempenho.

Referências

- Alvares, L. O., & Sichman, J. S. (1997, August). Introdução aos sistemas multiagentes. In XVII Congresso da SBC-Anais JAI'97. In: Demazeau, Y. e Müller, J.P. eds. Decentralized A.I., Amsterdam, Elsevier, 1990.
- Bastos, A. V. B., Borges-Andrade, J. E., & Zanelli, J. C. (2004). Psicologia, organizações e trabalho no Brasil. Porto Alegre: Artmed.
- Castka, P., Bamber, C. J., Sharp, J. M., & Belohoubek, P. (2001). Factors affecting successful implementation of high performance teams. *Team Performance Management: An International Journal*.
- Conte, R., Gilbert, N., & Sichman, J. S. (1998, July). MAS and social simulation: A suitable commitment. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation* (pp. 1-9). Springer, Berlin, Heidelberg.
- Ibarra, M. F. (2015). Psicotecnia. Historia de un encuentro entre la psicología y la técnica. *Revista Internacional sobre Subjetividad, Política y Arte*, 11(2), 71-85.
- Koppes, L. L. (Ed.). (2014). *Historical perspectives in industrial and organizational psychology*. Psychology Press.
- Myers, I. B., McCaulley, M. H., Quenk, N. L., & Hammer, A. L. (1998). *MBTI manual: A guide to the development and use of the Myers-Briggs Type Indicator* (Vol. 3). Palo Alto, CA: Consulting Psychologists Press.
- Mupinga, D. M., Nora, R. T., & Yaw, D. C. (2006). The learning styles, expectations, and needs of online students. *College teaching*, 54(1), 185-189.
- Rao, A. S., & Georgeff, M. P. (1995, June). BDI agents: from theory to practice. In *ICMAS* (Vol. 95, pp. 312-319).
- Salvit, J. (2012). *Extending BDI with Agent Personality Type*. Thesis – The Faculty of the Department of Computer and Information Science.
- Thompson, L., & Pozner, J.-E. (2007). *Organizational Behavior. Social Psychology: Handbook of Basic Principles*, pp. 912–931
- Weiss, G. (Ed.). (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*.

Modelagem para processamento e coleta de dados de trajetórias temporais na sociologia *

Matheus Mattos de Almeida Cruz¹

¹ Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba – PR – Brazil

matheusacruz@alunos.utfpr.edu.br

Abstract. *In the simulation of social networks based on agents, one of the ways to investigate a social process is through graph theory and using metrics, abstracting yourself to be able. Power is a person's ability to induce someone else's behavior. The social network will be an abstraction of the network of vertex connections, represented by a graph, and Agents are abstractions of the vertices of the graph. In this work, the metrics used in the current literature in the area of social power are discussed, exemplifying them, to propose a computational model.*

Resumo. *Uma das maneiras de se investigar um processo social é através da teoria de grafos e utilizando-se de métricas, abstraindo-se para um indicador social de poder. O poder é a possibilidade de uma pessoa induzir o comportamento de outra pessoa. A rede social será uma abstração da rede de ligações de vértices, representada por um grafo, e agentes são abstrações dos vértices do grafo. Neste trabalho são abordadas as métricas utilizadas na literatura atual da área de computação do tema de poder social, exemplificando-os, para propor um modelo computacional.*


1. Introdução

No Paraná as estratégias políticas e o poder das classes dominantes aparelhadas ao Estado reproduzem seus próprios privilégios e produzem pobreza, miséria e carência, [de Oliveira 2007]. A rede social dos clãs dominantes do Paraná será uma abstração representada por um grafo $G = \{V, E\}$, e atores políticos são abstrações dos vértices.

Existem trabalhos que estudaram este tema de poder e coalizão usando a abordagem de redes sociais, grafos e as relações de poder, como [Nazareno 2005].

Uma das maneiras de se investigar um processo social é utilizando-se da teoria de grafos e de métricas computacionais, abstraindo-se para poder e suas relações.

O objetivo deste trabalho é desenvolver um modelo computacional capaz de explicitar os mecanismos de criação de ligações em grafos de redes sociais dinâmicas que representam relações de poder.

*  O trabalho Modelagem para processamento e coleta de dados de trajetórias temporais na sociologia de Matheus Mattos de Almeida Cruz está licenciado com uma Licença Creative Commons - Atribuição-CompartilhaIgual 4.0 Internacional. <http://creativecommons.org/licenses/by-sa/4.0/>

2. Trajetórias temporais (*Timelines*)

Um dos formatos utilizados para construções de relações de poder é pela antropologia analítica devido sua relação com genealogia, como por exemplo o formato GEDCOM e grafos-P (grafo de parentesco) [White et al. 1999], principalmente devido ao problema de escala e complexidade do problema.

Considerando a análise longitudinal da população sendo multidimensional, temporal, e necessidade descritiva da rede social, optou-se pela estrutura de eventos em bancos de dados temporais pela capacidade de comportar e unificar estas características [Clark 2006].

Trajetoórias temporais, mostrado na figura 1, é uma metodologia em que mostra o crescimento quantitativo dos dados, um histórico individual e visualização gráfica poderosa permitindo agregação e comparação [Plaisant et al. 2003, Gramling and Carr 2004].

Para [Do et al. 2012], trajetórias temporais são um conjunto parcialmente ordenado de intervalos de tempo, a relação de eventos em na dimensão tempo usando intervalos em sua ocorrência, sendo que uma trajetória do tempo universal são todas os eventos associados à um tempo absoluto da ocorrência do evento. Definindo os eventos e seus tipos, é possível se calcular a esperança do número de eventos, como por exemplo o número de eventos em uma sub-população de uma região [Courgeau 2007].

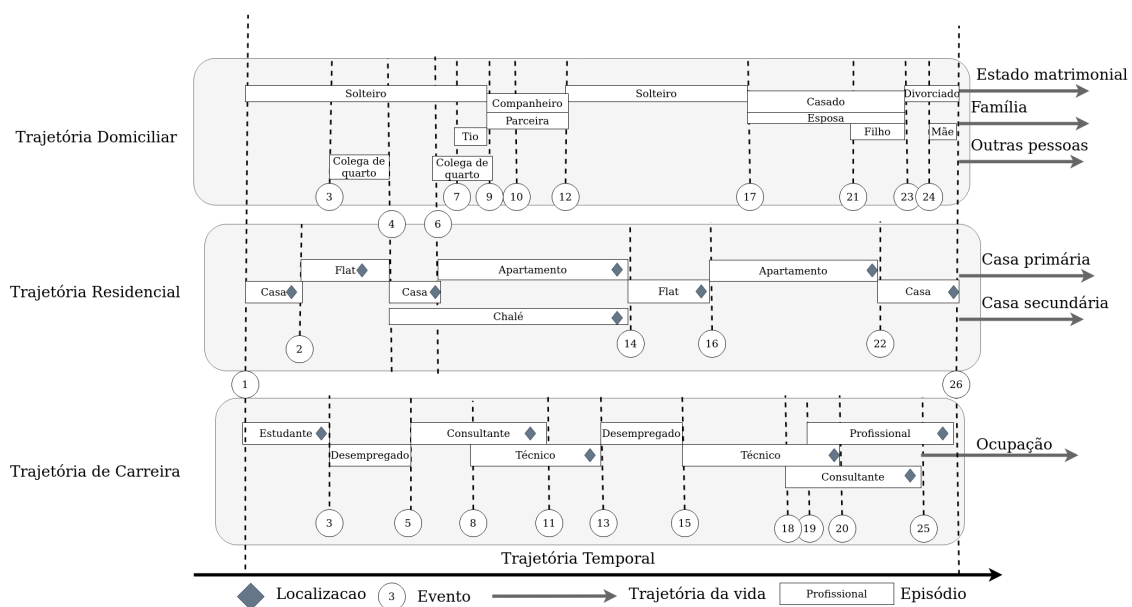


Figura 1. Trajetória temporal - Fonte: Adaptado de [Vandersmissen et al. 2009] - Tradução própria

3. Modelagem computacional de trajetórias temporais de agentes

Existem múltiplos trabalhos que criam com modelos para análise à partir dos históricos de indivíduos dando enfoque à eventos, misturando sistemas de informações geográficas (GIS), e adicionando uma dimensão de temporalidade, como mostrado em [Claramunt and Theriault 1996, Vandersmissen et al. 2009].

Para o modelo proposto neste artigo, foi observado as seguintes regras escolhidas para o modelo: Um Evento possui papéis sociais específicos dependendo do campo, um Evento deve possuir atributos diferentes dependendo do campo em que está relacionado, e todo evento acontece em local físico ou concreto dentro do Campo.

3.1. Modelo para trajetórias temporais

Para a construção do *dataset* é feita a construção da genealogia de uma classe dominante de clãs e suas relações de poder com partições representadas por campos sociais, este modelo para coleta de dados é representado na figura 2. As trajetórias temporais são representados como eventos e tipos de eventos, o campo é a categorização *a priori* feito na sociologia, assim como a classificação de um indivíduo pertencente a um clã dominante, e é representada como contexto composto de um evento.

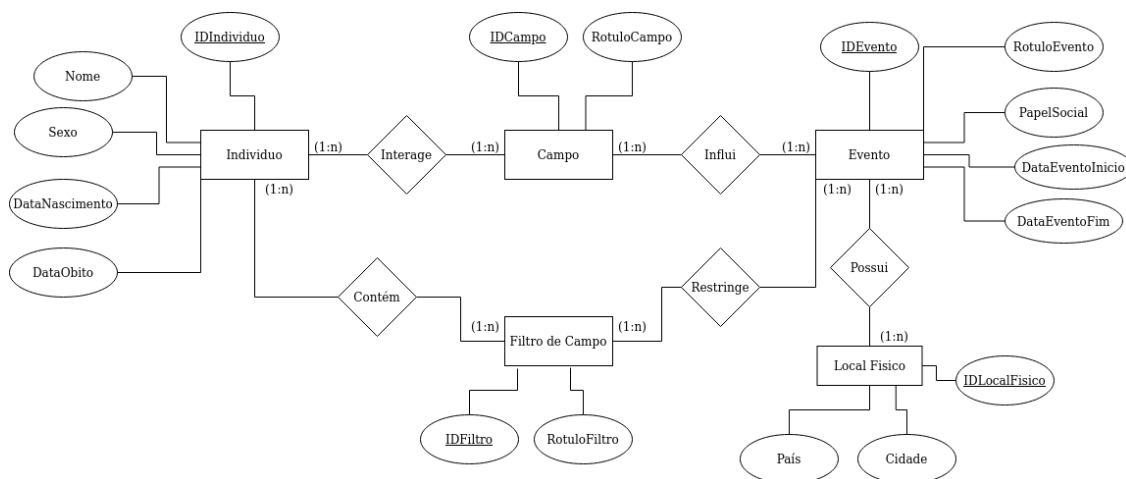


Figura 2. Diagrama ER. - Fonte: Autoria própria

3.2. Exemplo de trajetória temporal de um agente

Exemplificando o caso de dados de estudo de elites e coleta de dados para trajetória temporal, foi retirado da revista NEP dados de [Granato 2019], um estudo de caso que não fazem parte deste trabalho, mostrando na tabela abaixo em que explicita a maneira em que se pode aparecer estes dados. Isto é representado na figura 3.

Tabela 1. Exemplo de dados para ser coletado. - Fonte: [Granato 2019]

Caio Gracho Machado de Lima	Delegado Técnico do Ensino Público; Deputado Estadual 1908-1909; 1929-1930; 1935-1937. Juiz do Tribunal de Contas do Estado desde 1947
Camilo Stellfeld	Deputado Estadual, 1935-1937.
Linneu Madureira Novais	Deputado Estadual, 1935-1937; Deputado Estadual, 1947-1950. Medicina (PR)
José Manoel Ribeiro dos Santos	Deputado Estadual, 1935-1937; Diretor da Saúde Pública, Deputado Estadual, 1947-1950; Medicina (PR)

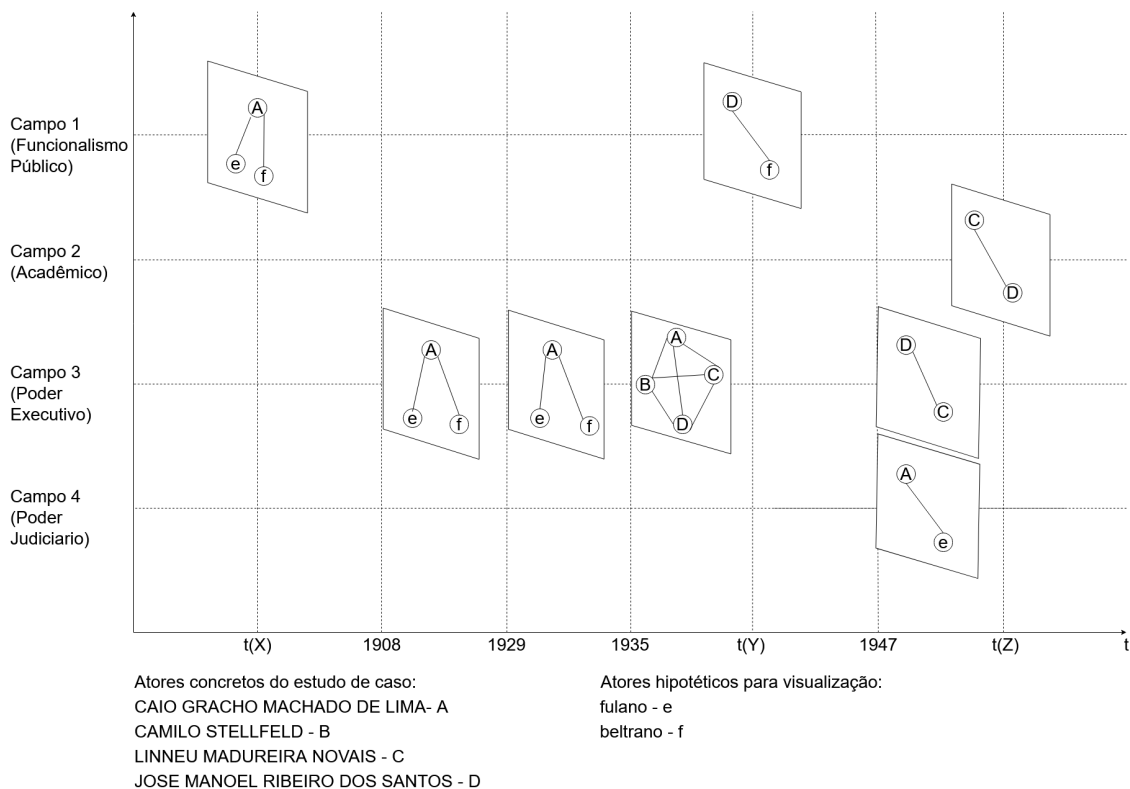


Figura 3. Trajetórias temporais dos dados de [Granato 2019] com vértices rotulados para representador atores. - Fonte: Autoria própria

4. Extração de dados e cenários futuros

Utilizando-se do cenário de um clã dominante no Paraná, na construção do *dataset* é feita a construção da genealogia de uma classe dominante tradicional e suas relações de poder com grupos diversos a partir da consulta aos periódicos do núcleo de estudos paranaenses (NEP) da UFPR ¹, consulta ao trabalho de [OLIVEIRA 2012], sendo necessário a digitalização da maioria desses dados.

V representam indivíduos e grupos (e.g. poder executivo, legislativo, empresariado, família). E representa a vínculo social entre o indivíduo para o grupo, dando assim em um grafo n-partido. Este grafo direcionado n-partido é representado na figura 4.

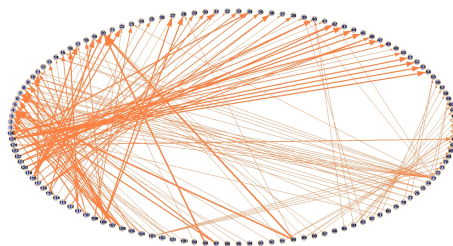


Figura 4. Dataset representado por grafo k-partido dos campos, ligando indivíduos aos campos, $k = 9$ - Fonte: Autoria própria

¹<https://revistas.ufpr.br/nep/index> - Visitado em 02/12/2019

Segundo [OLIVEIRA 2012] o governo do ator do grafo é composta por uma estrutura central de nepotismo e possui conexões com a famílias importantes. O clã trabalhado apresenta alta concentração, densidade, proximidade de poder com grande desigualdade entre atores políticos e concentração de papéis, mostrado na figura 5.

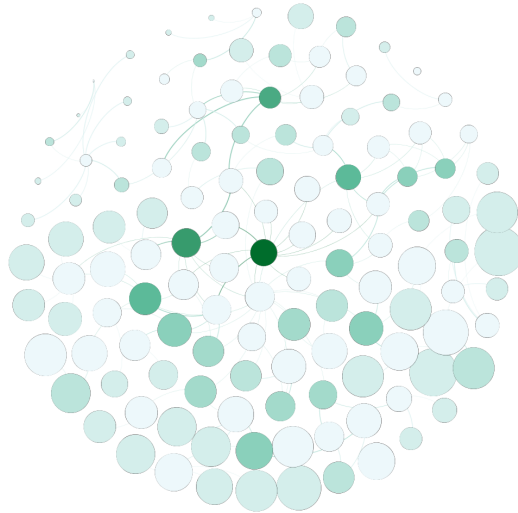


Figura 5. Centralidade de grau graduado em verde dos indivíduos no clã avaliado do dataset. - Fonte: Autoria própria

A análise implica em descobertas de fenômenos ou mecanismos não-observáveis de um grafo, assim propondo explicitar o funcionamento de um modelo de redes sociais dinâmicas (i.e. grafo dinâmico) da sociedade, bem como possa ser usado para análise de mecanismos de dinâmicas de grafo [Boccaletti et al. 2006].

De acordo com [Bijak et al. 2018], a utilização de cenário pode contribuir para explorar áreas além do horizonte do espaço de parâmetros de uma única geração, a inspiração na metodologia de demografia aplicada a uma programação baseada em modelo pode melhorar e flexibilizar as simulações computacionais.

5. Considerações finais

A criação de um modelo computacional serve para explicitar os mecanismos de formação de vínculos, assim como mudança de papéis, de um grafo de rede social das classes dominantes do Paraná gerando a contribuição na linha de pesquisa de simulação social.

O projeto dialoga com os tópicos de redes complexas, em particular redes sociais, e visualização de dados sociais além do conteúdo interdisciplinar da sociologia. No Paraná um clã de atores políticos pertencentes à esfera de interesse público apresentou alta concentração, densidade, proximidade de poder com grande desigualdade entre atores políticos e concentração de papéis, se possui papel central de poder, também possui papel como próximo ao poder e mediador deste poder em suas relações.

Referências

Bijak, J., Courgeau, D., Franck, R., and Silverman, E. (2018). Modelling in demography: From statistics to simulations. In *Methodological Investigations in Agent-Based Modelling*, pages 167–187. Springer.

- Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. (2006). Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308.
- Claramunt, C. and Theriault, M. (1996). Toward semantics for modelling spatio-temporal processes within gis. *Advances in GIS Research I*, pages 27–43.
- Clark, S. J. (2006). A general temporal data model and the structured population event history register. *Demographic research*, 15(7):181.
- Courgeau, D. (2007). *Multilevel synthesis: from the group to the individual*, volume 18. Springer Science & Business Media.
- de Oliveira, R. C. (2007). Famílias, poder e riqueza: redes políticas no paran  em 2007. *Sociologias*, 9(18):150–169.
- Do, Q. X., Lu, W., and Roth, D. (2012). Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687. Association for Computational Linguistics.
- Gramling, L. F. and Carr, R. L. (2004). Lifelines: A life history methodology. *Nursing Research*, 53(3):207–210.
- Granato, N. C. (2019). O poder legislativo paranaense no contexto da revolu o de 1930: um estudo sobre os capitais familiares e pol ticos dos deputados federais e estaduais (1930 a 1937).
- Nazareno, L. R. d. (2005). *Redes sociais e coaliz o de governo em Curitiba 1985-2004*. PhD thesis, Universidade de S o Paulo.
- OLIVEIRA, R. C. d. (2012). Na teia do nepotismo: sociologia pol tica das rela es de parentesco e poder pol tico no paran  e no brasil. *Curitiba: Insight*.
- Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., and Shneiderman, B. (2003). Lifelines: using visualization to enhance navigation and analysis of patient records. In *The craft of information visualization*, pages 308–312. Elsevier.
- Vandersmissen, M.-H., S guin, A.-M., Th riault, M., and Claramunt, C. (2009). Modeling propensity to move after job change using event history analysis and temporal gis. *Journal of geographical systems*, 11(1):37.
- White, D. R., Batagelj, V., and Mrvar, A. (1999). Anthropology: Analyzing large kinship and marriage networks with pgraph and pajek. *Social Science Computer Review*, 17(3):245–274.

Modelagem de Emoções para Tomada de Decisão de Agentes

*

Thiago Dantas¹, Patrícia Padula¹, Diana F. Adamatti¹, Cleo Z. Billa¹

¹Programa de Pós-Graduação em Computação
Universidade Federal do Rio Grande (FURG)

{thiagodantas923, padulalopes, dianaada, cleobilaa}@gmail.com

Abstract. *This paper presents a mapping of emotions using the OCC model applied to Multi-Agent Systems (MAS), with the help of the Watson tool from IBM. This mapping was performed so that Watson processes the communication between the agents and identifies the emotions expressed in each message exchanged. Emotions were identified using the OCC model. As a result, it was possible to structure a system that makes decisions based on the emotions presented to each agent.*


Resumo. *Este artigo apresenta um mapeamento das emoções utilizando o modelo OCC aplicado a Sistemas Multiagente (SMA), com auxílio da ferramenta Watson da IBM. Este mapeamento foi realizado de forma que o Watson fizesse o processamento da comunicação entre os agentes e identificasse as emoções expressadas em cada mensagem trocada. A identificação das emoções foi feita utilizando modelo OCC. Como resultado, foi possível estruturar um sistema capaz de tomar decisões com base nas emoções apresentadas por cada um dos agentes.*

1. Introdução

Existem vários sistemas os quais operam em uma lógica semelhante ao nosso raciocínio. Contudo, compreender a mente humana é um desafio que exige estudos em diferentes ramos da ciência. Neste sentido, há especial interesse em compreender a dinâmica do relacionamento entre estes sistemas.

Acredita-se que isto seja possível a partir da utilização da modelagem baseada em sistemas multiagente [Norvig and Russell 2014]. Por meio das interações que ocorrem, relacionamentos podem ser estabelecidos por trocas sociais entre estes agentes. Do ponto de vista computacional, compreender estas interações ainda é um grande desafio.

Sob esta perspectiva, este estudo procura entender e estudar a influência das emoções sobre as ações e comportamentos dos agentes, levando em consideração a memória adquirida por eles, possibilitando que situações do cotidiano humano possam ser estudadas em ambientes controlados [Marsella et al. 2010].

* O trabalho Modelagem de Emoções para Tomada de Decisão de Agentes de Thiago Dantas, Patrícia Padula, Diana F. Adamatti, Cleo Z. Billa está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-Compartilha Igual 4.0 Internacional. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Para isso, este trabalho está focado em desenvolver um sistema para modelar as emoções definidas pelo Modelo OCC [Ortony et al. 1990] através do processamento de linguagem natural. O sistema desenvolvido busca avaliar as emoções expressadas entre os agentes, bem como responder as suas mensagens utilizando as emoções captadas, dando um *feedback* desejado, através da geração da emoção presente nas interações entre agente-agente, para a compreensão de certos padrões, comportamentos e interações realizadas que são influenciados pela emoção sentida.

Para realizar este estudo, foi utilizado o Modelo OCC. Este modelo, fornece informações de como construir uma interpretação de uma situação do ponto de vista de cada agente e para qual emoção esta interpretação nos leva. Também foi utilizado o sistema de computação cognitiva da IBM, denominado Watson¹. Esta ferramenta foi utilizada para que fosse possível interpretar as interações que acontecem entre os agentes, realizar o processamento de linguagem natural (PLN) e utilizar isto para o mapeamento das emoções que são apresentadas pelo Modelo OCC.

2. Modelo OCC

Com o objetivo de tornar as ações e reações dos agentes mais semelhantes a dos humanos, vários tipos de abordagens tem sido utilizadas para estudar emoções e simplificar o seu entendimento frente a sistemas computacionais. Dentre estas, destaca-se o modelo OCC, desenvolvido por Ortony, Clore e Collins, cujas iniciais dão nome ao modelo [Ortony et al. 1990].

Esse modelo, é um modelo psicológico cuja teoria é baseada na abordagem cognitiva da emoção, ao qual é muito utilizado para o reconhecimento das emoções em ambientes computacionais e para implementar emoções em máquina, devido ao detalhismo e a simplicidade. O modelo é composto por 22 emoções. Um exemplo de emoção modelada pode ser observado a partir da Figura 1, onde cada uma das emoções apresenta uma especificação dividida em 3 partes:

Emoções de medo

TIPO DE ESPECIFICAÇÃO: (descontente com) a perspectiva de um evento indesejável.

TOKENS: ansioso, medo, nervoso, petrificado, assustado, encolhido, pavor, amedrontado, temeroso, tímido, apavorado, etc.

VARIÁVEIS QUE AFETAM A INTENSIDADE:

- 1 - O grau com que o evento é indesejável.
- 2 - A probabilidade do evento acontecer.

Exemplo: O funcionário, suspeitando que não era mais necessário, temia que fosse demitido.

Figura 1. Exemplo de especificação da emoção medo, adaptado de [Ortony et al. 1990]

- **TIPO DE ESPECIFICAÇÃO:** Descreve as condições que causam determinada emoção;

¹Disponível em: <https://www.ibm.com/watson/br-pt/>

- *TOKENS*: Uma lista de *tokens* são apresentados especificando quais palavras indicam determinada emoção. Como *anxious* (ansioso), *dread* (pavor) ou *fearful* (temeroso), são tipos de *fear* (medo);
- *VARIÁVEIS QUE AFETAM A INTENSIDADE*: Cada emoção contém uma lista de variáveis que afetam sua intensidade. Estas variáveis afetam uma única emoção, variáveis que afetam várias não são consideradas. Quanto maior o número de variáveis mais forte será a emoção.

3. Processamento de Linguagem Natural e WATSON

Atualmente, é notório o uso da IA nos mais variados seguimentos da área de Tecnologia da Informação (TI). A IA tem como objetivo fazer com que máquinas e computadores consigam imitar as ações humanas e descobrir soluções para problemas comuns [Makridakis 2017]. Esse tipo de tecnologia, utiliza métodos de aprendizagem para resolver problemas através do conhecimento adquirido e para isso, precisam ter algumas capacidades, dentre estas: a linguagem natural [Norvig and Russell 2014].

A comunicação existente entre máquinas e humanos ocorre por meio da linguagem natural, ao qual permite que os computadores, a partir de modelos estatísticos e da análise de padrões do comportamento linguístico, possam realizar a tradução das sentenças que são informadas pelos seus usuários para linguagem de máquina [da Silva 2006].

O Watson, por sua vez, é um dos sistemas de aprendizado de máquina que usa linguagem natural para manter a comunicação com os usuários, através da simulação de processos linguísticos realizada por humanos [High 2012]. Utiliza tecnologia de cognição para simular os processos realizada pela mente humana e, assim, aprender de acordo com as informações que é transmitido através do usuário. O sistema, disponível através da plataforma on-line da IBM utiliza uma arquitetura de software denominada DeepQA, que funciona da seguinte forma:

1. É identificado o assunto principal da pergunta, o tipo da questão que foi requerida, sua classificação e elementos que necessitem de algum tratamento;
2. Ocorre a busca na base de dados para a geração das hipóteses que atendam as respostas da questão, para isso, várias técnicas de procura são utilizadas neste processo;
3. Há uma seleção das melhores repostas, o que é chamado de escore de evidência;
4. Ocorre a etapa de síntese, onde as 100 melhores respostas devem ser ranqueadas até chegar nas cinco melhores;
5. Finalização do processo, onde são utilizados modelos pré-definidos gerados durante o desenvolvimento da arquitetura para obtenção da melhor resposta.

4. Integração Watson e OCC

O funcionamento do sistema desenvolvido ocorre basicamente da seguinte maneira: um texto é inserido no campo de pergunta, a qual contém alguns *tokens*. Estes *tokens* são então capturados e analisados na base de perguntas e respostas do Watson. Essa base determina qual a emoção foi identificada, tendo como base as 22 emoções presentes no Modelo OCC. Para um melhor entendimento do funcionamento do sistema é detalhado por meio da Figura 2 a ordem de processamento em relação a comunicação entre os agentes no sistema.

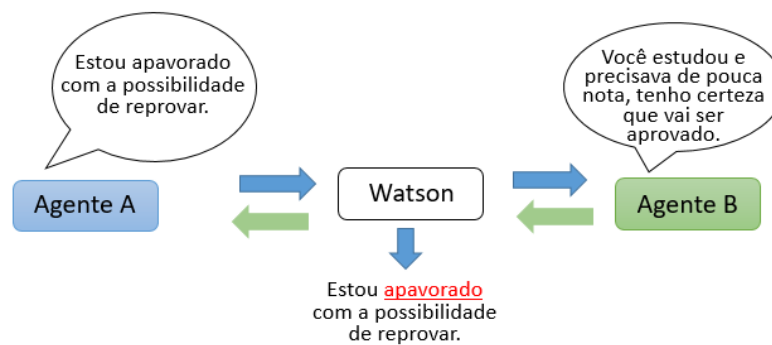


Figura 2. Exemplo de funcionamento do sistema

1. O agente A envia uma mensagem para o Agente B;
2. Esta mensagem é processada pelo Watson, que busca *tokens* e ao encontrar “apavorado” identifica a emoção expressa como medo;
3. O Watson passa a informação de que o agente A está expressando medo ao Agente B;
4. O agente B responde ao agente A com objetivo e acalmá-lo.
5. O Watson processa a mensagem do agente B (“*Você estudou e precisava de pouca nota, tenho certeza que vai ser aprovado.*”) e não encontra nenhum *token* que indica que nenhuma emoção foi expressa.

Após a modelagem do sistema e a definição dos *tokens* referentes a cada uma das 22 emoções, foi realizada a implementação no Watson. Esta implementação foi realizada utilizando o módulo do *Watson Assistant*, que normalmente é utilizado em *chat bots*. Porém, neste projeto ele foi adaptado para identificar as emoções presentes no modelo OCC, sendo necessário usar os conceitos de intenção *intent* e Entidade *entity*:

- *Intent*: são a representação do propósito que o usuário inseriu na mensagem, uma intenção é definida para cada tipo de chamada que a aplicação deseja atender. Neste projeto existe uma *intent* para cada emoção.
- *Entity*: representa o termo ou objeto que é relevante para a intenção e provem o contexto específico para a intenção. Neste projeto, as entidades são os *tokens* de cada emoção.

Como exemplo do que foi mencionado anteriormente, é apresentado na Figura 3 um exemplo de *entity* utilizado na emoção *medo* e os *tokens* presentes nesta emoção aos quais aparecem a direita na figura como sinônimos para que o sistema os identifique como *medo* e, na Figura 4 podemos observar as 2 emoções identificadas pelo sistema: *Medo* e *Tristeza*.

O sistema funciona para as 22 emoções do modelo OCC (inclusive com variações no diálogo). A única característica que é necessária é a presença de pelo menos um *token* definido por OCC. Porém, o potencial do Watson está em sua capacidade de aprender o que pode ser explorado durante o diálogo, quanto maior a troca de mensagens, melhor será sua capacidade de interpretar as emoções, além de ser possível auxiliá-lo nesta tarefa, caso ele não identifique a emoção em uma determinada frase pode-se definir manualmente que a frase representa uma emoção, e assim o Watson será treinado para poder identificá-la futuramente. Na Figura 5 é apresentado como isto ocorre:

Values (1) ↑	Type		
medo	Synonyms	amendrotado	assustado
		amendrotada	assustada
		apavorada	apavorado
		terrificada	terrificado
		susto	preocupado
		preocupada	nervoso

Figura 3. Representação de entidade no Watson para a emoção medo

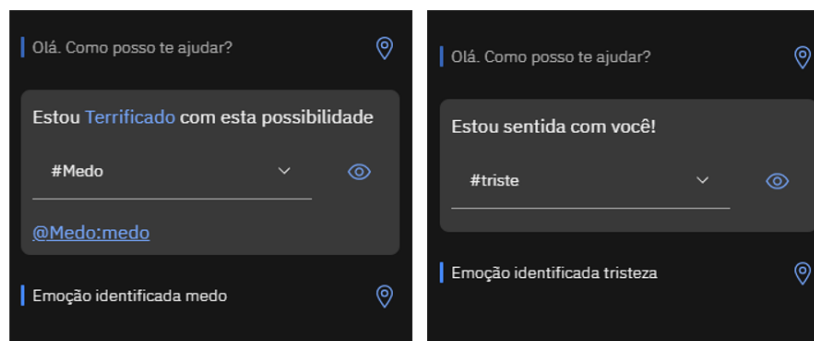


Figura 4. Identificação da emoção medo e da emoção tristeza

1. Ao apresentar a mensagem “*Estamos amedrontados com a possibilidade dela não conseguir*”, o Watson não reconheceu a emoção de medo no sistema;
2. Para resolver o problema, o Watson foi ajustado pelo próprio desenvolvedor para reconhecer esta frase como medo. Com isso foi possível treinar o modelo de IA e ajustar ao domínio da aplicação. Isto fez com que o sistema levasse alguns minutos para treinar e assim ser capaz de identificar a emoção presente nesta frase e em outras que fossem semelhantes a esta;
3. Após o treinamento do Watson, é enviada a mesma mensagem para verificar se ele é capaz de identificar a emoção, o que é feito com êxito, identificando como medo.

Este processo pode ser realizado sempre que o sistema obter um resultado diferente do esperado e assim torná-lo mais eficiente a cada interação.

5. Conclusões e Trabalhos Futuros

Neste trabalho, foi proposto um sistema para subsidiar a construção de um modelo computacional que permita a modelagem das emoções. Foi necessário a definição dos eventos possíveis gerados na utilização do ambiente Watson e do Modelo OCC, que auxiliasse o desenvolvimento de Agentes Inteligentes com capacidade de entender e expressar emoções.

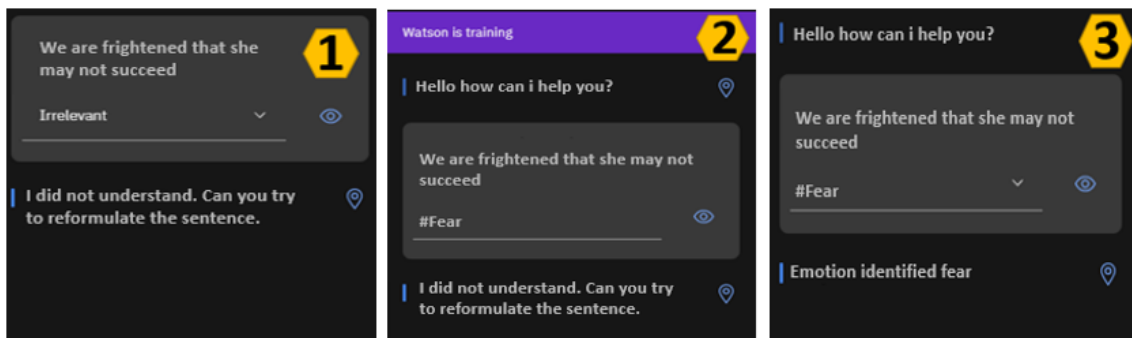


Figura 5. Treinamento do Watson para aprender a identificar uma emoção

O Modelo OCC foi escolhido como base para a estruturação das emoções, pois é baseado no princípio da diferenciação entre reações de valência positivas ou negativas e tenta simplificar o desenvolvimento de um modelo focando vários aspectos cognitivos das emoções. Já o Watson, foi escolhido porque é um sistema cognitivo que pode entender as mensagens que são trocadas entre os agentes no sistema, identificando suas emoções, aprendendo com eles e "raciocinando" a partir deles, na linguagem que foram naturalmente produzidas pelo homem (linguagem natural).

Contudo, até hoje, uma aplicação onde todas as emoções propostas pelo OCC possam ser utilizadas no Watson ainda não foi implementada. Para tanto, no modelo proposto buscou-se integrar os recursos oferecidos pelo Watson para desenvolver uma ferramenta que permitisse identificar as 22 emoções propostas pelo OCC.

Através da validação computacional do modelo apresentado, tem-se como trabalhos futuros realizar uma integração deste sistema com a ferramenta Jason e a inclusão de um modelo que represente memória em agentes para tornar o sistema mais realístico.

Referências

- da Silva, B. C. D. (2006). O estudo lingüístico-computacional da linguagem. *Letras de Hoje*, 41(2):103–138.
- High, R. (2012). The era of cognitive systems: An inside look at ibm watson and how it works. *IBM Corporation, Redbooks*, pages 1–16.
- Makridakis, S. (2017). The forthcoming artificial intelligence (ai) revolution: Its impact on society and firms. *Futures*, 90:46–60.
- Marsella, S., Gratch, J., Petta, P., et al. (2010). Computational models of emotion. *A Blueprint for Affective Computing-A sourcebook and manual*, 11(1):21–46.
- Norvig, P. and Russell, S. (2014). Inteligência artificial: Tradução da 3a edição (vol. 1).
- Ortony, A., Clore, G. L., and Collins, A. (1990). *The cognitive structure of emotions*. Cambridge university press.

Influência da Dor no Ritmo Circadiano: uma Modelagem Baseada em Sistemas Multiagente *

Angélica T. Santos¹, Catia M. Machado¹, Diana F. Adamatti¹

¹Programa de Pós Graduação em Modelagem Computacional (PPGMC)

Universidade Federal do Rio Grande (FURG)

Caixa Postal 474 – 96.203.900 – Rio Grande – RS – Brasil

{theisangelica, catiamachado, dianaadamatti}@furg.br


Abstract. *The circadian rhythm controls the unconscious activities of living beings through the biological clock. External influence, such as pain, cause dysfunction in the synchronization and desynchronization of the human body. We propose to study the mathematical and computational properties that can describe the modeling of circadian rhythm, specifically with the influence of pain. A mathematical model of two processes, composed by circadian rhythm and homeostatic rhythm. The proposed computational modeling, using a multi-agent system, will define values to the pain variable by means of a non-invasive questionnaire. Preliminary results show that pain directly influences the quality of sleep, as well as the development of daily activities.*

Resumo. *O ritmo circadiano comanda as atividades inconscientes dos seres vivos através do relógio biológico. Influência externa, como a dor, causa disfunção na sincronização e dessincronização. Nesse contexto, é proposto, estudar as propriedades matemáticas e computacionais que descrevem o modelo do ritmo circadiano com base na dor. No modelo matemático são utilizados dois processos, um composto do ritmo circadiano e o outro do ritmo homeostático. A modelagem computacional, via sistema multiagente, definirá valores para a variável dor através da aplicação de um questionário não invasivo. Resultados preliminares mostram como a variável dor influencia diretamente na qualidade do sono, bem como no desenvolvimento das atividades do cotidiano.*

1. Introdução

Processos rítmicos são intrínsecos e desempenham boa parte dos processos indispensáveis do corpo humano. Entre todos os processos rítmicos, o processo do ritmo circadiano é o que se destaca. O mesmo é caracterizado por processos biológicos que apresentam um ritmo de oscilação no período sono/vigília, sendo ajustado para 24 horas.

O estudo formal do ritmo circadiano faz parte da cronobiologia, área pertencente às ciências biológicas, que tem como objetivo estudar os relógios biológicos que controlam os ritmos e são responsáveis por atividades dos seres vivos. Sobretudo, os ritmos estão associados a funções vitais como hormônios, sistema digestivo, sensação de sono e fome, e influências externas, como dor, ansiedade ou depressão [Bruna 2019].

*  O trabalho Influência da Dor no Ritmo Circadiano: uma Modelagem Baseada em Sistemas Multiagente de Angélica T. Santos, Catia M. Machado, Diana F. Adamatti está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Os relógios biológicos ligados a modelos matemáticos e sistemas multiagente podem representar situações reais, realizar previsões e auxiliar no suporte à decisão. Os sistemas multiagentes estudam o comportamento de um conjunto independente de agentes com características diferentes, evoluindo em um ambiente [Wooldridge 2009].

Portanto, estudar e diagnosticar a dinâmica do ritmo circadiano é fundamental para a cronobiologia e para a ciência. Tem-se como objetivo a apresentação da pesquisa que está em andamento sobre a influência da dor no ritmo circadiano, bem como nas atividades do cotidiano, relacionada com indivíduos que trabalham ou estudam. Espera-se, desenvolver um ambiente multiagente que descreva de forma fidedigna o comportamento do ritmo circadiano e a dor. Assim a implementação da variável dor em modelo matemático e sistema multiagente é o diferencial deste trabalho, pois, a literatura carece de estudo de sistema multiagente usando modelo matemático de dois processos.

2. Embasamento Teórico

O relógio biológico de cada pessoa é sincronizado conforme suas atividades decorrentes do dia. Assim, a marcação horária interna eventualmente é precisa. Para que haja a regulação interna, é necessário os mecanismos de ajustes que permitem a sincronização. Essa sincronização é realizada pelo fenômeno de ajuste, chamado de “arrastamento”, esse fator externo que comanda o ajuste, denomina-se “zeitgeber”. Os “zeitgeber” são os sincronizadores do relógio biológico. Assim, o ritmo circadiano e homeostático são sincronizados pelo “zeitgebers”, de maneira que os mesmos estejam sempre interligados, para que possam estar interligados é necessário um “pacemaker” (marcapasso) [Daan et al. 1984].

Dormir pouco interfere negativamente nas atividades realizadas durante o dia, aumentando o risco para acidentes e baixo rendimento no trabalho e em atividades escolares. A curta duração do sono mostrou ter relação com irritabilidade emocional e déficit de atenção, além de ser identificada como fator de risco para a obesidade. Dados experimentais sobre os efeitos da privação do sono mostram que têm impactos negativos sobre a sonolência, o humor, no desenvolvimento das atividades cognitivas e motora, assim como em variáveis metabólicas, hormonais e imunológicas [Nixon et al. 2008]. Há mudanças bruscas na qualidade do sono com a idade. Não se sabe exatamente quais fatores ocorrem no corpo para causar essas mudanças, mas há evidência de que, à medida que envelhecemos, os ritmos circadiano e homeostático têm amplitude diferentes [Skeldon 2014].

O ritmo circadiano regula os ritmos materiais e psicológicos do ser humano, sendo controlado por um marca-passo localizado no cérebro, que é independente da vigília e do sono, com alcance do nível máximo de propagação do sono no início da manhã e o seu mínimo no início da noite. O ritmo homeostático decorrente da vigília sono é a pressão decorrente do sono acumulado durante o dia e que diminui durante a noite [Borbély and Achermann 1999].

2.1. Trabalhos relacionados

Nas pesquisas realizadas, não foram encontradas publicações científicas, envolvendo os três termos de busca juntos, sendo ritmo circadiano, dor e sistema multiagentes. Desta maneira foi realizado a inter-relação entre às áreas, onde cada item representa uma revisão e suas respectivas palavras-chave.

- Três grandes áreas de pesquisa: *Multiagent - Circadian - Pain*;

- União *Circadian - Multiagent*;
- União *Circadian - Pain*;
- União *Multiagent - Pain*;
- União *Circadian - Mathematical Model*;
- União *Multiagent - Biological System*.

A Tabela 1 apresenta os principais autores que embasam esta pesquisa, onde é proposto a inter-relação de todas as áreas dor, sistema, multiagente, ritmo circadiano, modelo matemático e sistema biológico.

Tabela 1. Trabalhos Relacionados, onde MAS: Multiagent; Circ: Circadian; Math: Mathematical Model; e Bio: Biological System

	Pain	MAS	Circ	Math	Bio
Neste trabalho	X	X	X	X	X
[Andreychenko et al. 2016]	-	X	X	-	-
[Baptista and Costa 2008]	-	X	X	-	-
[Aviram et al. 2015]	X	-	X	-	-
[Hatcher et al. 2018]	X	-	X	-	-
[Jankowski 2013]	X	-	X	-	-
[Lopez-Jornet et al. 2015]	X	-	X	-	-
[Burish et al. 2019]	X	-	X	-	-
[Borbély 1982]	-	-	X	X	-
[Daan et al. 1984]	-	-	X	X	-
[Achermann et al. 1993]	-	-	X	X	-
[Borbély and Achermann 1999]	-	-	X	X	-
[Achermann and Borbély 2003]	-	-	X	X	-
[Borbely and Achermann 2011]	-	-	X	X	-
[Borbély et al. 2016]	-	-	X	X	-
[Florioian et al. 2011]	-	X	-	-	X
[Chiacchio et al. 2014]	-	X	-	-	X
[Montagna et al. 2008]	-	X	-	-	X
[Amigoni and Schiaffonati 2007]	-	X	-	-	X
[Kefalas et al. 2005]	-	X	-	-	X

Com o auxílio dos trabalhos relacionados, esta modelagem proposta do ritmo circadiano e variável dor, sob a perspectiva de sistema multiagente vem para preencher uma lacuna na literatura.

3. Análise e Resultados Preliminares

Usando a simulação baseada em agentes, é possível mostrar uma população real de maneira artificial, onde cada indivíduo da população é apresentado por um agente e todos os agentes formam um grupo, cada um com suas próprias regras e comportamentos [Wooldridge 2009].

Os dados estão sendo coletados, através de um questionário não invasivo. Alguns dados coletados no questionário são a idade, hora de dormir, hora de acordar, nível de dor

em cada dia da semana, local da dor e produtividade no trabalho ou estudo¹.

Para cada dia da semana o indivíduo responde sobre o horário de dormir, horário de acordar e nível de dor. O horário de dormir e acordar, é definido como hora fixa, devido ao tempo constante que o mesmo leva para dormir profundamente e despertar ao acordar. Caso o horário de dormir, não conste no questionário é porque o indivíduo é considerado fora dos padrões da pesquisa.

Com os dados já coletados, total de oitenta e sete participantes 66,28% tem entre 20 e 35 anos. Foi definido pelas pesquisadoras no CEPAS - Comitê de Ética de Pesquisas na Área da Saúde, que o público alvo desta pesquisa são indivíduos entre 20 e 35 anos, pois é nessa faixa etária que o sono sofre maiores picos de dessincronização.

Ainda no questionário, foi perguntando sobre a localização da dor (Figura 1), sendo que destes 24,56% relataram sobre dor na região da cabeça, seguido de 22,81% dos participantes ter dor na região das costas. Observa-se que muitos participantes tem dor em mais de uma região do corpo, podendo ser membros superiores ou membros inferiores.

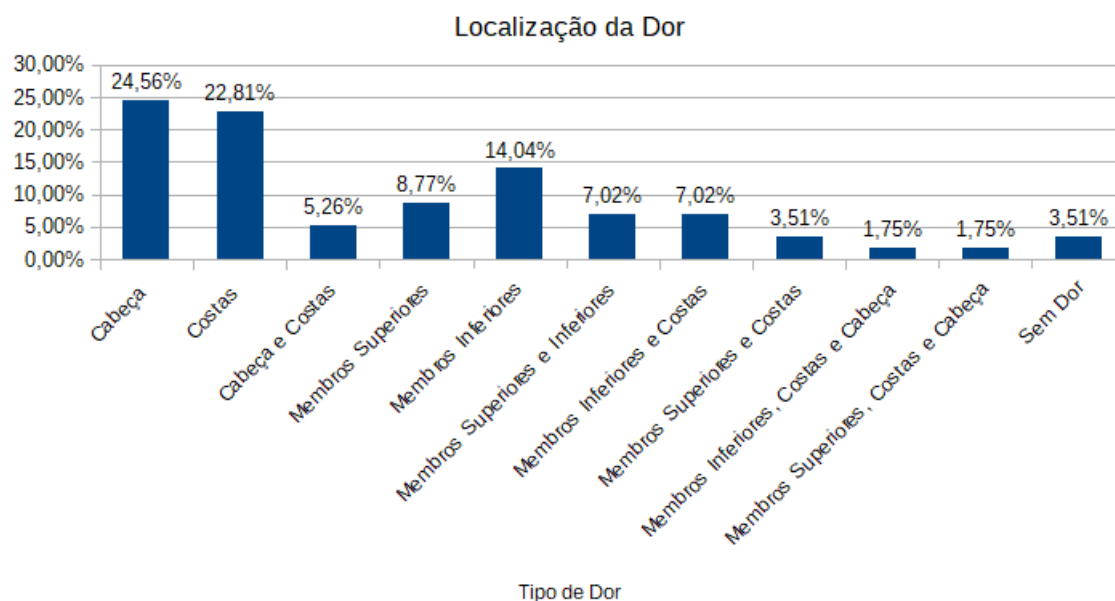


Figura 1. Região corporal da dor

Na simulação baseada em agentes é possível através de suas ferramentas, simular regras de comportamento de um determinado sistema, aproximando de um fato real. Neste estudo simulou-se o ritmo circadiano e a variável dor, conforme a Figura 2, onde foi analisado um indivíduo com horário de dormir às vinte e duas horas e acordar as sete horas, com uma variação, a cada dia no nível de dor. A escala de dor utilizada é “0, 2, 4, 6, 8 e 10”. Os picos em rosa, mostram o momento da dor, sendo “0, 6, 10, 8, 6, 4 e 2, e os períodos que o indivíduo gostaria de estar dormindo, mas está acordado devido ao nível da dor.

No Netlogo esta sendo mostrado uma aplicação real de como a dor afeta o corpo

¹Projeto aprovado pela Plataforma Brasil e Comitê de Ética em Pesquisa na Área da Saúde (CEPAS) da Universidade Federal do Rio Grande (FURG) - Protocolo CAAE: 18147119.9.0000.5324.



Figura 2. Interface do Netlogo com simulação da dor variável

humano, bem como o ritmo circadiano. Ainda pretende-se realizar testes estatístico para comprovar cientificamente que o rendimento do indivíduo diminui com a dor afetar a qualidade do sono.

4. Conclusões e Trabalhos Futuros

Na literatura, existem diversas pesquisas que utilizam de forma integrada ritmo circadiano e dor, mas nenhuma encontra-se integrada com sistema multiagente. Desta forma, torna-se importante a modelagem do ritmo circadiano e dor utilizando sistema multiagente, que está em andamento, já que se pretende, com a pesquisa, mostrar o quanto a dor implica na produtividade do indivíduo.

Como trabalhos futuros pretende-se coletar mais respostas do questionário, bem como a comprovação da equação da dor de forma fidedigna por meio de testes com as respostas dos questionários, e, para comprovar cientificamente, realizar testes estatísticos mostrando a queda da produtividade de um indivíduo que sente dor e não realizou o sono de descanso adequadamente. Pretende-se com este trabalho mostrar a interdisciplinariedade entre Inteligência Artificial, um modelo matemático e um problema real do cotidiano.

Agradecimentos

As autoras deste artigo agradecem à Fundação de Amparo a Pesquisa do Estado do Rio Grande do Sul - FAPERGS pelo recurso financeiro no desenvolvimento da pesquisa.

Referências

- Achermann, P. and Borbély, A. A. (2003). Mathematical models of sleep regulation. *Front Biosci*, 8(Suppl.):S683–S693.
- Achermann, P., Dijk, D.-J., Brunner, D. P., and Borbély, A. A. (1993). A model of human sleep homeostasis based on eeg slow-wave activity: quantitative comparison of data and simulations. *Brain research bulletin*, 31(1-2):97–113.
- Amigoni, F. and Schiaffonati, V. (2007). Multiagent-based simulation in biology. In *Model-based reasoning in science, technology, and medicine*, pages 179–191. Springer.

- Andreychenko, A., Magnin, M., and Inoue, K. (2016). Analyzing resilience properties in oscillatory biological systems using parametric model checking. *Biosystems*, 149:50–58.
- Aviram, J., Shochat, T., and Pud, D. (2015). Pain perception in healthy young men is modified by time-of-day and is modality dependent. *Pain Medicine*, 16(6):1137–1144.
- Baptista, T. and Costa, E. (2008). Evolution of a multi-agent system in a cyclical environment. *Theory in Biosciences*, 127(2):141–148.
- Borbély, A. A. (1982). A two process model of sleep regulation. *Hum neurobiol*, 1(3):195–204.
- Borbely, A. A. and Achermann, P. (2011). Sleep homeostasis and models of sleep regulation. *Principles and Practice of Sleep Medicine*, page 431–444.
- Borbély, A. A., Daan, S., Wirz-Justice, A., and Deboer, T. (2016). The two-process model of sleep regulation: a reappraisal. *Journal of sleep research*, 25(2):131–143.
- Borbély, A. A. and Achermann, P. (1999). Sleep homeostasis and models of sleep regulation. *Journal of biological rhythms*, 14(6):559–570.
- Bruna, M. H. V. (2019). Relógios biológicos. <https://drauziovarella.uol.com.br/neurologia/relogios-biologicos-artigo/>. [Online; accessed 19-February -2020].
- Burish, M. J., Chen, Z., and Yoo, S.-H. (2019). Emerging relevance of circadian rhythms in headaches and neuropathic pain. *Acta Physiologica*, 225(1):e13161.
- Chiacchio, F., Pennisi, M., Russo, G., Motta, S., and Pappalardo, F. (2014). Agent-based modeling of the immune system: Netlogo, a promising framework. *BioMed research international*, 2014.
- Daan, S., Beersma, D., and Borbély, A. A. (1984). Timing of human sleep: recovery process gated by a circadian pacemaker. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 246(2):R161–R183.
- Floroian, D., Floroian, L., and Moldoveanu, F. (2011). Multiagent system for monitoring chronic diseases. In *International Conference on Advancements of Medicine and Health Care through Technology*, pages 26–31. Springer.
- Hatcher, K. M., Royston, S. E., and Mahoney, M. M. (2018). Modulation of circadian rhythms through estrogen receptor signaling. *European Journal of Neuroscience*.
- Jankowski, K. (2013). Morning types are less sensitive to pain than evening types all day long. *European Journal of Pain*, 17(7):1068–1073.
- Kefalas, P., Eleftherakis, G., and Stamatopoulou, I. (2005). Formal modelling of the dynamic behaviour of biology-inspired, agent-based systems. In *Molecular Computational Models: Unconventional Approaches*, pages 243–276. IGI Global.
- Lopez-Jornet, P., Molino Pagan, D., Andujar Mateos, P., Rodriguez Agudo, C., and Pons-Fuster, A. (2015). Circadian rhythms variation of pain in burning mouth syndrome. *Geriatrics & gerontology international*, 15(4):490–495.
- Montagna, S., Ricci, A., and Omicini, A. (2008). A&a for modelling and engineering simulations in systems biology. *International Journal of Agent-Oriented Software Engineering*, 2(2):222–245.
- Nixon, G. M., Thompson, J. M., Han, D. Y., Becroft, D. M., Clark, P. M., Robinson, E., Waldie, K. E., Wild, C. J., Black, P. N., and Mitchell, E. A. (2008). Short sleep duration in middle childhood: risk factors and consequences. *Sleep*, 31(1):71.
- Skeldon, A. (2014). Are you listening to your body clock? <http://personal.maths.surrey.ac.uk/st/A.Skeldon/sleep.html>. [Online; accessed 25-March -2019].
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.