

**Segunda edição dos Anais do
V Workshop-Escola de Sistemas de Agentes,
seus Ambientes e apliCações**

— WESAAC 2011 —

Organizado por

**Gustavo A. Giménez Lugo
Jomi Fred Hübner
Cesar Augusto Tacla**

Curitiba, 27-29 Abril de 2011

Anais do V Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações — V WESAAC / Lugo, G.A.G.; Hübner, J.F.; Tacla, C.A. (Org) — Curitiba, 2011.

203p. :il.
ISSN 2177-2096

1. Agentes. 2. Sistemas de Agentes. 3. Ambientes para Agentes. 4. Aplicações de Agentes. I. Lugo, G.A.G. II. Hübner, J.F. III. Tacla. C.A.

CDD

Prefácio

Este documento contém os trabalhos apresentados na *quinta* edição do WESAAC (Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações). O WESAAC 2011 foi realizado na cidade de Curitiba/PR, junto a Universidade Tecnológica Federal do Paraná (UTFPR), entre os dias 27 e 29 de abril de 2011, com o apoio do Programa de Pós-Graduação em Computação Aplicada da UTFPR e da Sociedade Brasileira da Computação (SBC).

O principal objetivo desta quinta edição do WESAAC é integrar pesquisadores e estudantes de todos os níveis na área de Agentes e Sistemas de Agentes, divulgar as atividades de pesquisa dos diversos grupos de pesquisa do Brasil, possibilitando o intercâmbio de conhecimentos. Para isso, o evento é constituído de uma combinação de Oficinas e Palestras (a parte “escola”), proferidas por pesquisadores experientes, e apresentações de Trabalhos Completos e Resumos Estendidos (a parte “workshop”).

Recuperando um pouco da história deste evento, percebemos que suas edições anteriores testemunharam o crescimento constante da comunidade de pesquisadores na área de sistemas baseados em agentes. As três primeiras edições do evento tiveram uma abrangência regional (RS, em sua maioria, e alguns participantes de SC e PR). O escopo, que inicialmente foi voltado para a formação e discussão de trabalhos, passou a reforçar cada vez mais a qualidade técnica dos artigos selecionados. Tal processo levou naturalmente, na sua quarta edição, a ter uma abrangência nacional, equilibrando os aspectos de divulgação, formação e qualidade acadêmica dos trabalhos. Para esta quinta edição, o caráter nacional do evento será mantido e reforçado, com a participação de pesquisadores destacados da área de sistemas de agentes, de diversas instituições do Brasil, tais como USP, IME/RJ, UFRGS, FURG, UFSC, UCPEL, UTFPR, e do exterior, notadamente da Università de Bologna (Itália) com a oficina de programação de agentes ministrada pelo Prof. Alessandro Ricci.

Gostaríamos de agradecer a todos os pesquisadores que submeteram os seus artigos, assim como aos membros do comitê de programa, aos revisores adicionais pelo criterioso trabalho desenvolvido, às nossas instituições (UTFPR e UFSC) e ao CNPq que tornaram possível o WESAAC 2011.

Abril 2011

Gustavo A. G. Lugo
Jomi F. Hübner
Cesar A. Tacla

Apoio



*Conselho Nacional de Desenvolvimento
Científico e Tecnológico*

Organização

Organização Geral

Gustavo A. Giménez Lugo

Universidade Tecnológica Federal do Paraná

Coordenação Comitê de Programa

Jomi Fred Hübner

Universidade Federal de Santa Catarina

Organização Local

Cesar Augusto Tacla

Universidade Tecnológica Federal do Paraná

João Fabro

Universidade Tecnológica Federal do Paraná

Fabrício Enembreck

Pontifícia Universidade Católica do Paraná

Comitê Consultivo

Antônio Carlos da Rocha Costa

Universidade Federal do Rio Grande

Rejane Frozza

Universidade de Santa Cruz do Sul

João Luis Tavares da Silva

Universidade de Caxias do Sul

Diana Francisca Adamatti

Universidade Federal do Rio Grande

Comitê de Programa

Adriana Soares Pereira

UNIFRA

Alexandre Moretto Ribeiro

UCS

Ana Bazzan

UFRGS

Ana Carolina Bertoletti De Marchi

UPF

Anarosa Alves Franco Brandão

USP

André Luis Alice Raabe

UNIVALI

André Maurício Cunha Campos

UFRN

Andréa Aparecida Konzen da Silva

UNISC

Antônio Carlos da Rocha Costa

FURG

Aydano Pamponet Machado

UFAL

Carine Webber

UCS

Cecília Dias Flores

UFCSPA

César Tadeu Pozzer

UFSM

Claudia Brandelero Rizzi

UNIOESTE

Daniela Duarte da Silva Bagatini

UNISC

Diana Adamatti

FURG

Elder R. Santos

UFRGS

Elisa Boff

UCS

Evandro Costa

UFAL

Fabiana Lorenzi	ULBRA
Fabrício Enembreck	PUC-PR
Fabio Yoshimitsu Okuyama	IFRS
Fred Freitas	UFPE
Graçaliz Pereira Dimuro	FURG
Jaime Sichman	USP
Jerusa Marchi	UFSC
João Fabro	UTFPR
João Luis Tavares da Silva	UCS
Jorge Luis Victória Barbosa	UNISINOS
Luciana Nedel	UFRGS
Luciano dos Reis Coutinho	UFMA
Luis Gustavo Nardim	USP
Luis Otávio Álvares	UFRGS
Magda Bercht	UFRGS
Maíra Ribeiro Rodrigues	UFMG
Marcelo Blois Ribeiro	PUCRS
Márcia Cristina Moraes	PUCRS
Marcos Eduardo Casa	UCS
Maria de Fatima Weber do Prado Lima	UCS
Michael da Costa Mora	PUCRS
Nuno David	ISCTE/Portugal
Patrícia Tedesco	UFPE
Patrícia Alejandra Behar	UFRGS
Patrícia Brandalise Scherer Bassani	FEEVALE
Paulo Blikstein	Stanford University
Paulo Roberto Ferreira Junior	UFPEL
Rafael Heitor Bordini	UFRGS
Reinaldo Bianchi	FEI
Rejane Frozza	UNISC
Ricardo Choren	IME-RJ
Ricardo Silveira	UFSC
Ricardo José Rabelo	UFSC
Rosa Vicari	UFRGS
Silvio Cesar Cazella	UNISINOS
Viviane Torres da Silva	UFF

Sumário

I Artigos Convidados

O Nível Cultural das Sociedades de Agentes	3
<i>Antônio Carlos da Rocha Costa</i>	

II Artigos

Uma Extensão ao Protocolo Contract-Net Aplicado ao Problema de Planejamento de Transporte de Derivados de Petróleo	39
<i>Roni Fabio Banaszewski, Kelvin Elton Nogueira, Jean Marcelo Simão, Lucia Valéria Arruda e Cesar A. Tacla</i>	
Um Modelo Multiagente para Carregamento Compartilhado com Prioridades em um Terminal de Contêineres: Combinando o Problema da Mochila com PDMs	51
<i>Leonardo Martins Rodrigues, Graçaliz Dimuro, Antônio Carlos da Rocha Costa e Leonardo Emmendorfer</i>	
A quantitative study about Tardiness in Java-based Multi-Agent Systems	63
<i>Pier-Giovanni Taranti, Ricardo Choren e Carlos Lucena</i>	
Coordination of Agents in the RoboCup Rescue: A Partial Global Approach	75
<i>Andre Hahn Pereira, Luis Gustavo Nardin e Jaime Simão Sichman</i>	
Um WEB Searcher Agent Baseado em Regras Contextuais e na Colaboração entre Agentes	85
<i>Daniela Maria Uez e João Luis Tavares da Silva</i>	
Framework para Re-engenharia do Ambiente AMPLIA	97
<i>Paulo Ricardo Barros, Elton Erhardt, Marta R. Bez, Sílvio César Cazella e Cecília Dias Flores</i>	
Uma Arquitetura de Referência para Softwares Assistentes Pessoais Baseada em Agentes e SOA	109
<i>Saulo Popov Zambiasi e Ricardo Rabelo</i>	
Qualificação de Valores de Troca em Interações Sociais em Sistemas Multiagentes ..	121
<i>Cion Ayres do Nascimento, Gustavo A. G. Lugo e Murilo V. G da Silva</i>	
Criação de Fatos Institucionais a partir de Fatos Brutos em Sistemas Multiagente ..	129
<i>Maiquel Brito e Jomi Fred Hübner</i>	
Protocolo de Controle Organizacional em Sistemas Multiagente	141
<i>Mauricio Rocha, Fabiana Lorenzi e Stanley Loh</i>	

III Resumos Expandidos

Agentes Computacionais na Simulação da Dengue	155
<i>Rogério Luis Rizzi, Rodolfo Lorbieski, Carlos Henrique França, Leonardo Pereira Merlin, Claudia Brandelero Rizzi, André Luiz Brun e Reginaldo Zara</i>	
Um sistema de apoio à decisão baseado em agentes para simulação de impactos gerados pela instalação de indústrias	159
<i>Gerson Nunes, Camila Thomasi, Márcio Jogueiro e Diana Adamatti</i>	

Modelando a Organização Social de um Experimento de Gestão Social de Hortas Urbanas	163
<i>Thiago Rodrigues, Iverton Santos, Glenda Dimuro, Antônio Carlos da Rocha Costa e Graçaliz Dimuro</i>	
Uma plataforma para desenvolvimento de sistemas multiagente BDI na Web	167
<i>Maurício O. Haensch, Diogo De Campos e Ricardo A. Silveira</i>	
NetLogo e o modelo Climate Change como ferramenta para a Simulação do Efeito Estufa	171
<i>Diogo Veber Lima, Antônio Carlos da Rocha Costa e Nisia Krusche</i>	
DIMI 3D - Agente Companheiro para um Ambiente Virtual de Aprendizagem	175
<i>Michael Fernando Künzel, Andréa Konzen da Silva, Rejane Frozza, Daniela Bagatini e Beatriz Lux</i>	
SimDeCS: Arquitetura de um Sistema Multiagente para Simulação de Tomadas de Decisão em Cuidados de Saúde	179
<i>Vinícius Maroni, João Batista Moosmann, Paulo Ricardo Barros, João Marcelo Fonseca, Eduardo Zanatta, Elton Erhardt, Sílvio César Cazella, Alessandra Dahmer e Cecília Dias Flores</i>	
Uma Abordagem Baseada em Leilões Combinatórios para Resolver o Problema de Transporte de Derivados de Petróleo	183
<i>Kely Plucinski Vieira e Cesar A. Tacla</i>	
Simulação social baseada em dados de redes mediadas por blogs	187
<i>Aline Macohin, Cion Ayres do Nascimento e Gustavo A. Giménez Lugo</i>	
Desenvolvimento de um modelo de Simulação Social da cidade de Mandirituba – Brasil – Paraná	191
<i>Leonardo Andreatta De Alcantara, Fernando José Esmaniotto e Gustavo A. Giménez Lugo</i>	
Sobre Agentes e Controladores Preditivos	195
<i>Marcelo Lopes de Lima, Jomi Fred Hübner e Eduardo Camponogara</i>	
Arquitetura de agentes credíveis em entretenimento eletrônico com processamento emocional multi-nível	199
<i>Diego Prates De Andrade e Jomi Fred Hübner</i>	
Índice de Autores	203

Parte I

Artigos Convidados

O Nível Cultural das Sociedades de Agentes

(03/08/2011 – Versão 0.2)

Antônio Carlos da Rocha Costa¹

¹Centro de Ciências Computacionais
Programa de Pós-Graduação em Modelagem Computacional
Universidade Federal do Rio Grande – FURG
96.201-900 – Rio Grande – RS – Brasil
ac.rocha.costa@gmail.com

Resumo. *Este texto introduz o conceito de nível cultural no modelo PopOrg de sociedades de agentes e indica como tal conceito pode ser aplicado às sociedades de agentes onde a questão da energia se constitui em uma questão central.*

Abstract. *This text introduces the concept of cultural level in the PopOrg model of agent societies and shows how such concept can be applied to agent societies where energy constitutes a central issue.*

1. Introdução

A questão da cultura é a questão central da Antropologia e da Filosofia da Cultura, e uma questão fundamental nas Ciências Sociais em geral. Na área dos sistemas multiagentes, a complexidade dos modelos de organização desses sistemas tem aumentado progressivamente, com forte inspiração vinda daquelas ciências, buscando captar os diversos aspectos sociais dos mesmos. A questão da cultura, porém, apenas muito recentemente começou a ser enfrentada, em relação a esses sistemas.

Este artigo propõe que o conceito de *nível cultural* é capaz de dar o enquadramento correto para o tratamento da questão da cultura nos sistemas multiagentes. Ele resume as principais idéias de importantes antropólogos e filósofos da cultura a respeito dessa questão e, de modo ainda inicial, mas suficientemente elaborado para permitir a compreensão geral da proposta que está sendo feita, formaliza esses conceitos, inserindo-os no modelo PopOrg de organização de sociedades de agentes.

O texto está organizado como segue. A Seção 2 discute o conceito básico de níveis de abstração em sistemas, sobre o qual se assenta o conceito de nível de cultura de uma sociedade, e apresenta a conceituação de nível de cultura com base nas concepções antropológicas de David Schneider.

A Seção 3 dá uma definição original de sociedade de agente, enfatizando a diferença específica entre sistemas multiagentes e sociedades de agentes. A Seção 4 resume as características essenciais do modelo PopOrg, um modelo extensional mínimo de sociedades de agentes, desenvolvido para suportar a análise das dinâmicas operacional e estrutural de tais sociedades.

A Seção 5 é o núcleo do texto. Ela apresenta a formalização do conceito de nível de cultura de sociedade de agentes. A formalização é apresentada de modo incremental, tendo por base a expressão formal do modelo PopOrg.

A Seção 6 é a Conclusão.

2. Níveis de Abstração em Sistemas

A idéia da existência de níveis de abstração em sistemas é bastante antiga na ciência em geral, sendo a seguinte classificação da Realidade em níveis hierárquicos uma das mais tradicionais:

- nível físico
- nível biológico
- nível social

2.1. Níveis de Abstração em Sistemas Computacionais

Na Ciência da Computação, a utilização da idéia de níveis de abstração está presente desde a origem da área, seja na distinção entre níveis de hardware e de software, seja, por outro lado, na distinção de níveis de abstração procedural em conjuntos de subrotinas ou de níveis de abstração entre módulos de programas.

Contudo, a primeira formulação de um esquema sistemático de níveis de abstração para analisar sistemas computacionais foi a realizada por Allen Newell e Gordon Bell, em seu livro sobre Arquitetura de Computadores[3]. Esse esquema, por outro lado, serviu de base para que Newell introduzisse o conceito de “nível simbólico físico” [21] e, posteriormente, o conceito de “nível de conhecimento” [22], como os níveis operatórios fundamentais para a área da Inteligência Artificial.

2.2. Níveis de Abstração em Sistemas Sociais

Na área de sistemas sociais, o conceito de níveis de abstração também é amplamente utilizada, a mais tradicional hierarquia de níveis de abstração sendo a formulação da distinção entre o “micro-nível” social e o “macro-nível” social, cf. [6] e [1].

2.3. O Nível Cultural na Antropologia de David Schneider

Nas duas principais áreas a se ocuparem da questão da cultura, a Filosofia da Cultura e a Antropologia, também a abordagem baseada em níveis de abstração foi amplamente utilizada, embora de modo nem sempre explícito.

A obra do filósofo Ernst Cassirer, por exemplo, constituiu-se na análise da cultura como um nível de abstração da sociedade humana, análise realizada em termos de “formas simbólicas” [4].

Na antropologia, desde os trabalhos pioneiros de Bronislaw Malinowski e Radcliffe-Brown, cf. [19] e [23], está presente a idéia da cultura vista como um nível de abstração da sociedade.

Leslie White, em particular [30], frisou explicitamente o caráter “extrassomático” e a autonomia da cultura em relação à base biológica do organismo do ser humano, salientando – ao molde de Émil Durkheim (cf., e.g., [20]) – que ao nascer o ser humano se defronta com o “mecanismo” da cultura já em funcionamento na sociedade.

Foi David M. Schneider [25], porém, que explicitamente formulou uma teoria antropológica baseada na idéia da cultura como um nível de abstração existente objetivamente na sociedade humana, nível de abstração que ele chamou de “nível cultural”:

In the most general terms, then, the problem I have posed is that of describing and treating culture as an independent system and of analyzing it in its own terms; that is, as a coherent system of symbols and meanings. [25], p.7–8.

Que a expressão “sistema independente” refere-se a um nível de abstração da sociedade, pensado como existente objetivamente, fica claro em outra passagem, em que os conceitos de “construtos culturais” e de “unidades culturais” fazem as vezes das “formas simbólicas” de Cassirer::

[The proposed approach to anthropological problems] assumes that the cultural level of observation can be distinguished from all others; that cultural units and constructs can be described independently of all other levels of observation; and that the culture so isolated can be examined to see what its core symbols are (if there are core symbols); how meaning is systematically elaborated (if it is systematically elaborated) throughout its differentiated parts; and how the parts are differentiated and articulated as cultural units (if they are so articulated). [25], p.8.

assim como nesta:

I started [this book] with the point that a cultural unit or cultural construct must be distinguished from any other object elsewhere in the real world, and that the cultural unit or construct has a reality of its own. [25], p.8.

2.4. O Problema da Realidade e da Interação entre Níveis de Abstração de Sistemas

Em todos os contextos, teóricos ou práticos, em que a abordagem de níveis de abstração foi utilizada para analizar um sistema (qualquer que tenha sido a natureza desse sistema) dois problemas críticos se colocaram logo de imediato: (1) o problema a realidade dos níveis de abstração; (2) o problema das interações entre eles.

O problema da realidade dos níveis de abstração não parece ter tido, até hoje, uma solução de consenso. O problema é: os níveis de abstração são realidades existentes em si mesmas, constituintes dos sistemas em estudo, ou não tem existência real, sendo apenas conceitualizações, recursos apenas conceituais, para facilitar a descrição e análise dos sistemas?

No dia-a-dia dos trabalhos baseados nas distinções de níveis de abstração, contudo, a questão é tratada como estando resolvida, o problema recebendo usualmente uma solução pragmática simples: trabalha-se “como se” os níveis de abstração fossem realidades existentes nos sistemas.

O problema das interações entre níveis, por outro lado, vem progressivamente recebendo soluções cada vez mais consensuais, na forma da concepção de uma dupla interação. Quer dizer, entre cada dois níveis de abstração situados imediatamente um acima do outro na hierarquia de níveis de abstração, assume-se a existência de dois processos de interação, um de sentido ascendente, outro de sentido descendente, pelos quais cada um dos dois níveis influencia a estrutura e o funcionamento do outro.

Uma figura clássica, expressando essa concepção, é a figura do “barco” de Coleman [6], cf. Fig. 1, onde sucessões causais de eventos que ocorrem no macro-nível de

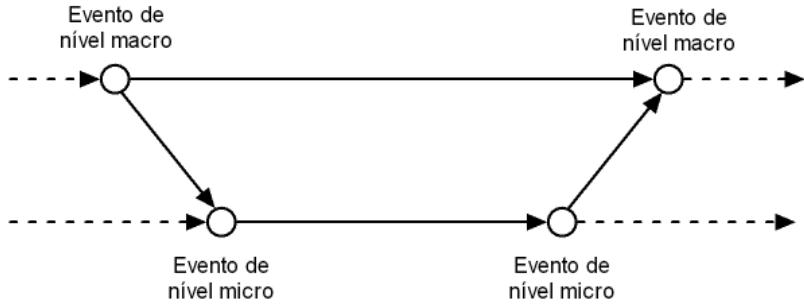


Figura 1. O “barco” de Coleman (baseado em [6], p.10).

uma sociedade interagem com sucessões causais de eventos que ocorrem no micro-nível social:

Outras concepções das relações possíveis entre dois níveis de abstração são possíveis (cf., p.ex., [7]).

No que segue, tentaremos mostrar como o conceito de “nível cultural” de Schneider pode ser aplicado à análise de sociedades de agentes. Para tanto, faremos uso do modelo PopOrg [10], um modelo de organização de sociedades de agentes fundamentado na noção de nível de abstração.

No modelo PopOrg, contudo, não se detalham as possíveis relações entre os níveis da sociedade de agentes, exceto pela determinação de que cada nível/sub-nível da sociedade “implementa” o nível/sub-nível imediatamente superior, numa forma bem especificada pelo modelo (cf. Sec. 4.4.3).

3. Sociedades de Agentes

O termo “sociedade de agentes” vem sendo utilizado há bastante tempo, em conjunto com o termo “sociedade artificial”, para designar um tipo genérico de sistema computacional composto de agentes, concebido para operar como um sistema que ou realiza um processamento para seus usuários, ou funciona como um modelo simulado de uma sociedade humana ou animal, cf. [16, 15, 14].

Neste trabalho, contudo, as sociedades de agentes recebem uma definição específica, com um propósito específico, qual seja, caracterizar um tipo específico de sistema multiagente, que acreditamos vai adquirir importância crescente num futuro próximo.

Assim, definimos:

Definição 1 *Uma sociedade de agentes é um sistema multiagente aberto e persistente.*

Por sistema multiagente *aberto*, deve-se entender um sistema em que os agentes podem entrar e sair livremente.

Por sistema multiagente *persistente*, deve-se entender um sistema cujas estruturas e funções, em cada instante, têm existência e funcionamento independente do particular conjunto de agentes que habitam a sociedade naquele instante, de modo que tais estruturas e funções são capazes de persistirem no tempo, independentemente da entrada e saída de agentes do sistema.

Abertura e persistência são, pois, as características diferenciadoras das sociedades de agentes, no conjunto dos sistemas multiagentes.

Um aspecto muito importante para o que se segue é que a persistência de uma sociedade de agentes pode ser maior que a persistência do interesse que tem por ela seus criadores, de modo que toda sociedade de agentes pode terminar tendo de se confrontar com a situação em que nenhuma das pessoas que interagem com ela tem pleno conhecimento de sua estrutura e funcionamento. Em particular, as sociedades de agentes podem ter de se confrontar com situações onde devem receber agentes cujos criadores não têm dela nenhum conhecimento.

O impacto dessas possibilidades sobre os agentes que entram na sociedade nessa situação, sejam eles recém criados ou estejam eles migrando de outras sociedades, será brevemente considerado na Conclusão do trabalho.

3.1. A Questão Energética nas Sociedades de Agentes

Uma questão fundamental, relativamente ao conceito de sociedade de agentes, é a questão da *energia* envolvida nos funcionamentos dessas sociedades.

É possível conceber sociedades de agentes em que a questão da energia envolvida nos funcionamentos dessas sociedades é desconsiderada: por exemplo, sociedades totalmente realizadas em software. A desconsideração da questão da energia pode ocorrer, nesse caso, porque ela fica implicitamente considerada na questão da energia envolvida no funcionamento do hardware que suporta o software.

Esse é, também, o ponto de vista usual da comunidade de computação em relação à questão da energia relacionada ao funcionamento de qualquer software.

É certo, por outro lado, que esse ponto de vista da comunidade de software vem mudando, um pouco, à medida que cada vez mais sistemas de software são desenvolvidos para dispositivos portáteis, energizados por baterias, que têm capacidade limitada de armazenamento de energia, dispositivos tais como computadores móveis, telefones celulares e smartphones, jogos, etc.

Contudo, mesmo com essas mudanças, a questão da energia é tratada como uma questão colateral, relegada a um “módulo de gerenciamento de energia”, cuja função essencial é evitar catástrofes, alertando os usuários da sociedade sobre a proximidade de situações de bateria com nível zero de energia, para que eles tomem as providências necessárias.

Há, porém, um tipo particular de sociedades de agentes em que a questão da energia é uma questão central, envolvendo decisões estratégicas não apenas por parte dos agentes da sociedade, como também da sociedade como um todo, não podendo de modo algum ser delegada a um “módulo de alerta” e aos usuários da sociedade.

Trata-se das sociedades de agentes em que os agentes são dotados de características robóticas, isto é, corpos físicos cuja operação requer quantidades de energia muito superiores às necessárias para a operação dos sistemas de software que os controlam, e cuja capacidade de armazenamento de energia é baixa, frente à duração das atividades que os agentes realizam na sociedade.

Especialmente nas circunstâncias em que a sociedade opera por ciclos de fun-

cionamento, de modo que a disponibilidade de energia em cada ciclo é baixa frente à demanda de energia estabelecida no ciclo pelos agentes da sociedade, a questão do uso adequado da energia disponível no ciclo ultrapassa o limite dos interesses individuais dos agentes, tornando-se um problema da sociedade de agentes como um todo.

É nesse tipo de sociedade de agentes (aberta, persistente e em que o uso adequado da energia é essencial para a continuidade do funcionamento da sociedade) que a cultura se coloca como uma questão central.

3.2. A Questão Cultural nas Sociedades de Agentes

Desde a formulação inicial da teoria de Malinowski [19], a colocação da estrutura e funcionamento da sociedade humana em relação às necessidades energéticas da vida cotidiana dos indivíduos e às necessidades enérgicas do funcionamento dos mecanismos sociais de toda ordem, materiais e simbólicos, tem sido um aspecto importante da análise antropológica.

Em síntese, a concepção Malinowskiana é a de que o suprimento de energia para as atividades dos indivíduos de uma sociedade deve ser obtida através de uma parte de suas atividades na sociedade, as quais, para tanto, requerem um *mecanismo de coordenação* que garanta não só um mínimo de eficiência na obtenção dessa energia, como também uma proporção adequada para essa atividade de obtenção de energia no contexto geral das atividades da sociedade.

A cultura da sociedade – o seu *nível cultural* – é, desse ponto de vista, o sistema que constitui e conforma os componentes essenciais daquele mecanismo de coordenação.

Neste trabalho, assumimos que esse ponto de partida Malinowskiano – a motivação energética para a gênese, estruturação e funcionamento de um nível cultural na sociedade – pode ser generalizado para todo tipo de sociedade construída por indivíduos naturais (para as sociedades animais, por exemplo), assim como para todas as sociedades mistas (sociedades humano-animais, por exemplo).

Mais ainda, assumimos que esse ponto de partida pode ser generalizado para todo tipo de sociedade construída por indivíduos artificiais (para as sociedades de agentes, por exemplo), assim como para todas as sociedades mistas envolvendo indivíduos naturais e artificiais (sociedades humano-artificiais, por exemplo).

Note-se, porém, que esse tipo de sociedade – sociedade em que o problema energético é central e que, por isso, tem um nível cultural – é um tipo de sociedade distinto do tipo de sociedade em que o problema energético não é central e que tem um nível cultural apenas porque assim foi construída: no primeiro caso, o nível cultural tem uma função fundamental a cumprir – instrumentalizar a coordenação das ações relacionadas com a questão energética –, enquanto que, no segundo caso, o cumprimento dessa função não é necessário e o nível cultural pode constituir-se livremente, sem restrições.

Note-se, por outro lado, que dizer que, em uma dada sociedade, o nível cultural tem por função instrumentalizar a coordenação das ações relacionadas com o uso da energia não é dizer que o funcionamento do nível cultural nessa sociedade tem de se restringir ao cumprimento dessa função.

Neste texto, apresentamos uma proposta de modelo organizacional abstrato para

sociedades de agentes nas quais o problema energético é um problema central e que, por isso, conforme a concepção Malinowskiana, têm de ser dotadas, necessariamente, de um nível cultural instrumentalizador da coordenação do uso da energia.

3.3. A Questão da Relação de Implementação de Ações

O que se observa a respeito das ações que um agente pode realizar em uma sociedade é que o fato de esse agente tratar um segundo agente como foco de uma ação material não impede que aquele primeiro agente esteja, concomitantemente ou paralelamente, agindo sobre o segundo através de uma ação de cunho social e, portanto, realizada no nível cultural (p.ex.: a ação de chutar o segundo agente, realizada no nível material, associada à ação de desprezá-lo, realizada no nível cultural).

A modelagem das ações dos agentes, no modelo PopOrg, portanto, procede através de múltiplos níveis de abstração: ação material, ação social, ação institucional, etc.

Esse aspecto do modelo é examinado nas seções a seguir, através da noção de *implementação* de uma ação realizada em um nível de abstração por meio de uma ou mais ações realizadas em um nível de abstração hierarquicamente inferior: dizemos que a ação de nível superior é “implementada por” essas ações de nível inferior.

Em [27], a relação entre ações que corresponde à relação de implementação de ações do modelo PopOrg é a relação de “constituição”, também denominada relação “contar como”: uma ação material “constitui” (ou, “conta como”) uma ação social. Por exemplo, chutar “conta como” desprezar¹.

A definição formal da relação de implementação de ações é intimamente relacionada à hierarquia de níveis de abstração das sociedades de agentes. Assim, ela só será apresentada após a apresentação da estrutura mínima do modelo PopOrg (cf. Sec. 4.4.3).

4. O Núcleo do Modelo PopOrg de Sociedade de Agentes

Formulado inicialmente em [12], o modelo PopOrg constitui-se em um modelo formal para a organização e a dinâmica organizacional de sociedades de agentes.

O modelo caracteriza-se por:

- ser definido em torno de um núcleo, constituído pela reunião de um conjunto mínimo de elementos extensionais, os quais se devem ser encontrados em todos os outros modelos organizacionais de sistemas multiagentes que se caracterizem como minimamente suficientes;
- distinguir entre os aspectos extensionais (observacionais) da sociedade, isto é, aqueles a que se tem acesso externamente aos agentes, e os aspectos intensionais (mentais) da sociedade, isto é, aqueles a que se tem acesso somente através da análise da mente dos agentes;
- distinguir entre aspectos estruturais e operacionais da sociedade;

¹Formalmente, porém, enquanto que a relação de constituição é dirigida do conjunto de ações A , de nível inferior, para a ação α , de nível superior, que elas constituem – p.ex.: $counts-as(A, \alpha)$ –, a relação de implementação é dirigida, de modo inverso, da ação de nível superior para o conjunto de ações que a implementam – $imp(\alpha, A)$ – cf. Sec. 4.

- distinguir dois níveis de abstração principais, o *nível populacional* (em que os agentes constituintes da sociedade são modelados, juntamente com suas interações, de modo extensional) e o *nível organizacional* (em que os elementos de organização da sociedade são modelados de modo extensional);
- no nível organizacional, distinguir dois sub-níveis, o *macro-nível organizacional* e o *micro-nível organizacional*;
- possibilitar a extensão do núcleo do modelo, em diversas direções, através do acréscimo de componentes com finalidades específicas, especialmente no que se refere aos aspectos intensionais;
- distinguir entre *modelos sincrônicos* e *modelos diacrônicos* da sociedade, os primeiros captando os aspectos operacionais da sociedade, quando ela funciona com uma estrutura organizacional fixa, os segundos captando a evolução histórica da sociedade, caracterizada por uma dinâmica da sua estrutura organizacional.

Ao longo do tempo, diversos componentes tem sido acrescentados ao núcleo do modelo, tanto componentes intensionais quanto componentes extensionais, conforme se encontra relatado em [10].

A versão do modelo PopOrg que se apresenta a seguir é a versão mais recente. Além dos principais componentes introduzidos anteriormente, ela contém os componentes necessários para a caracterização extensional do nível cultural das sociedades de agentes.

Especificamente, esta versão do modelo PopOrg se caracteriza por:

- distinguir dois tipos de ambientes em que a sociedade opera, um *ambiente material*, constituído pelos objetos materiais que os agentes manipulam durante seu funcionamento na sociedade, e um *ambiente simbólico*, constituído pelos símbolos que representam elementos tanto do ambiente material quanto do nível populacional e do nível organizacional da sociedade (tanto de nível micro quanto de nível macro, tanto elementos de caráter extensional quanto elementos de caráter intensional);
- permitir a modelagem do conjunto de *normas* que regulam a constituição e o funcionamento da sociedade através de representações simbólicas no ambiente simbólico;
- através da vinculação com símbolos pertencentes ao ambiente simbólico, modelar as *unidades organizacionais* do macro-nível organizacional como *instituições*, no sentido Malinowskiano do termo [19], o conjunto das instituições e suas interações constituindo a chamada *estrutura institucional* da sociedade de agentes;
- modelar o *nível cultural* da sociedade como a estrutura constituída pelo ambiente simbólico e pelo conjunto dos elementos do nível organizacional, do nível populacional e do ambiente material que possuem uma representação no ambiente simbólico;
- ser um modelo sincrônico, possibilitando a modelagem de sociedades que estão operando com estruturas organizacionais fixas.

A seguir, apresentam-se formalmente os diversos componentes da presente versão do modelo PopOrg.

4.1. O Ambiente Material

4.1.1. Noção Intuitiva

Intuitivamente, o ambiente natural de uma sociedade de agentes é o conjunto de objetos materiais que os agentes da sociedade usam enquanto estão operando nessa sociedade.

O conjunto de objetos materiais não se restringe a objetos estáticos: pode incluir objetos ativos naturais (animais, p.ex.) e artificiais (máquinas, p.ex.).

É claro, pode acontecer de, numa sociedade, agentes tratarem agentes como objetos, em algum momento, como quando um agente é tratado como foco da ação material de outro agente (p.ex., numa ação de chutar). Os agentes, portanto, são um tipo especial de objetos do ambiente material da sociedade de agentes.

4.1.2. Noção Formal

Formalmente, a noção de ambiente material pode ser definida como segue:

Definição 2 *O ambiente material de uma sociedade de agentes S é uma estrutura $ME_S = (Obj_S, Ags)$, onde:*

- *Obj_S é o conjunto de objetos materiais existentes na sociedade;*
- *Ags é o conjunto de agentes existentes na sociedade;*

tais que $Ags \in \wp(Obj_S)$.

O conjunto de todos os possíveis ambientes materiais de uma sociedade S é denotado por \mathcal{ME}_S .

4.2. O Ambiente Simbólico

4.2.1. Noção intuitiva

Um *símbolo* é um objeto material que *representa* um outro elemento (material ou não, extensional ou intensional), chamado *significado* do símbolo.

Intuitivamente, portanto, o *ambiente simbólico* de uma sociedade de agentes é o conjunto dos objetos materiais que tem algum significado na sociedade, isto é, que representam algum elemento da sociedade.

Símbolos são operados pelos agentes da sociedade – na qualidade de símbolos, não de objetos materiais – através de processos mentais ditos de *interpretação*: um processo de interpretação é um processo mental pelo qual um agente, tendo tido acesso ao objeto material que constitui o símbolo, acessa mentalmente o elemento representado pelo símbolo.

O processo de interpretação de um símbolo por um agente, portanto, é um processo pelo qual o agente percorre o caminho da relação de representação, acessando o objeto representado a partir do objeto representador (o símbolo).

Nessa conceituação, a *relação de representação* entre símbolos e seus significados é suposta conhecida pelo agente que interpreta o símbolo.

De um modo geral, porém, tendo acessado um símbolo enquanto objeto material, não necessariamente um agente tem conhecimento de que aquele objeto foi constituído como símbolo por outro agente, de modo que não necessariamente ele precisa se dar conta de que aquele objeto tem um significado na sociedade.

Por outro lado, quando um agente tem o conhecimento de que um determinado objeto é um símbolo, isto é, tem um significado na sociedade, não necessariamente o agente conhece a relação de interpretação tal como ela foi estabelecida pelo agente que constituiu o símbolo como símbolo.

Finalmente, cabe observar que, em geral, símbolos podem ser *ambíguos*, isto é, ter mais de um significado. Por isso a associação entre símbolos e seus significados é feita, em geral, por uma relação, não por uma função.

4.2.2. As Idéias dos Agentes

Como os símbolos são operados pelos agentes da sociedade por meio de processos mentais, é necessário diferenciá-los dos elementos mentais através dos quais os agentes representam elementos para si mesmos. Para diferenciá-los dos símbolos, conceituados acima, chamamos esses elementos mentais pelos quais os agentes representam mentalmente elementos para si mesmos de *representações mentais*, ou *idéias*.

Idéias, portanto, são objetos intensionais que, como os símbolos (objetos extensionais), estão carregados de significados, isto é, representam outros objetos (intensionais ou extensionais).

Símbolos e idéias são os elementos que possibilitam a conexão entre os aspectos extensionais (observacionais) e os aspectos intensionais (mentais) de uma sociedade, os símbolos tendo existência extensional (observacional) e as idéias tendo existência intensional (mental).

Analogamente com o que acontece com os símbolos, cabe observar que, em geral, as idéias de um agente podem ser *ambíguas*.

4.2.3. Noção Formal

Como o modelo PopOrg é um modelo cujo núcleo constitui-se em um modelo extensional de sociedades de agentes, apenas as noções de símbolo, interpretação de símbolo e ambiente simbólico serão definidos formalmente, aqui. A formalização das noções de idéia e de interpretação de idéia ficam fora do escopo do presente trabalho.

Definição 3 *Todo símbolo em uma sociedade de agentes S é um objeto material $\sigma \in Obj_S$.*

Note-se, em particular, que é possível que agentes sejam tratados como símbolos.

Definição 4 *Uma relação de interpretação de símbolos em uma sociedade de agentes S é qualquer relação não vazia $intrp \subseteq Obj_S \times X$, onde o codomínio de intrp é qual-*

quer conjunto X de elementos existentes na sociedade². O conjunto das relações de interpretação de símbolos de uma sociedade de agentes S é denotado por Intrp_S .

Note-se que, como não necessariamente se tem $\text{dom}(\text{intrp}) = \text{Obj}_S$, nem todo objeto material é um símbolo para uma dada relação de interpretação $\text{dom}(\text{intrp})$.

Definição 5 Diz-se que um objeto material $\sigma \in \text{Obj}_S$ é um símbolo para a interpretação intrp se e somente se $\sigma \in \text{dom}(\text{intrp})$. O conjunto dos objetos materiais de uma sociedade de agentes S que são símbolos para uma relação de interpretação intrp é denotado por Σ_S^{intrp} , de modo que vale que $\Sigma_S^{\text{intrp}} \subseteq \wp(\text{Obj}_S)$. O conjunto de todos os símbolos da sociedade S para todas as possíveis relações de interpretação é denotado por Σ_S .

Dadas as definições de símbolo e relação de interpretação, pode-se então estabelecer a definição de ambiente simbólico de uma sociedade de agentes:

Definição 6 O ambiente simbólico de uma sociedade de agentes S é uma estrutura $SE_S = (\Sigma_S, \text{Intrp}_S)$ onde:

- Σ_S é o conjunto de símbolos da sociedade;
- Intrp_S é o conjunto de relações de interpretação de símbolos da sociedade.

O conjunto de todos os possíveis ambientes simbólicos de uma sociedade S é denotado por \mathcal{SE}_S .

Definição 7 Diz-se que o ambiente simbólico $SE_S = (\Sigma_S, \text{Intrp}_S)$ de uma sociedade de agentes S está bem definido se e somente se $\Sigma_S = \bigcup \{\text{dom}(\text{intrp}) \mid \text{intrp} \in \text{Intrp}_S\}$, isto é, se e somente se:

- todo símbolo é interpretado por pelo menos uma relação de interpretação; e
- todo objeto interpretado por uma relação de interpretação é um símbolo.

4.3. O Nível Populacional

4.3.1. Noção Intuitiva

Intuitivamente, o nível populacional de uma sociedade de agentes é constituído pelo conjunto dos agentes que funcionam nessa sociedade e pelas interações que esses agentes têm entre si.

O modelo PopOrg prescreve que, além desses elementos, a modelagem do nível populacional das sociedades envolva um critério para a determinação da adequação das interações modeladas com as capacidades de interação dos agentes da sociedade. Para tanto, a modelagem do nível populacional deve especificar também a capacidade comportamental de cada agente e a capacidade de interação de cada par de agentes.

Com essa última especificação, dois requisitos de adequação podem ser determinados: um estabelecendo que a capacidade de interação de cada par de agentes seja compatível com as capacidades comportamentais dos agentes e, outro, que as interações especificadas sejam compatíveis com as capacidades de interação dos agentes.

²Note-se que, como estamos definindo a estrutura das sociedades de agentes de modo incremental, a compreensão plena de quais são os elementos que podem ser representados pelos símbolos de uma sociedade só será alcançada quando da conclusão da sequência de definições que estamos realizando.

Esse requisito de adequação de interações com capacidades de interação e dessas com capacidades de comportamento é um padrão de adequação que se repete em cada nível de modelagem das sociedades de agentes. É um requisito que visa garantir o critério mínimo de adequação da organização social do ponto de vista das redes de interação que podem se estabelecer em cada um dos níveis de abstração da sociedade.

Um comportamento, no modelo PopOrg, é uma sequência de conjuntos de ações, cada conjunto de ações sendo realizado em um instante de tempo. Uma especificação de comportamento indica, então, em cada instante, o conjunto de ações que “se espera” que o agente realize naquele instante, quando realiza aquele comportamento.

Em outros termos, a especificação de um comportamento no modelo PopOrg é uma definição de um “comportamento esperado”, uma “norma de comportamento”, que traz embutido nela um critério de “adequação”, o qual admite variações nas realizações corretas desse comportamento.

Assim, a especificação de comportamento viabiliza a idéia de que os comportamentos que os agentes realizam possam ser realizados tanto de modo “incompleto”, como com “acréscimos” de ações, em relação ao que é esperado na norma do comportamento, sem que isso implique que essas variações nas realizações do comportamento estejam “incorrectas”: em princípio, faltas e excessos são tratados, assim, como “inadequações”, não como incorreções.

Por outro lado, adotando um ponto de vista em que um comportamento adequado é visto como uma “especificação” de um comportamento desejado, é possível seguir uma noção semelhante à de “refinamento de traços” (proposta em [18] relativamente aos processos CSP) e dizer que um determinado comportamento dado (uma implementação) é “correto”, relativamente a um comportamento adequado (sua especificação) se ele não implementa mais do que foi especificado nesse comportamento adequado.

Quanto às interações, o modelo PopOrg utiliza a noção de “processo de troca” para modelar as interações entre os agentes [10]. Abstratamente, a noção de processo de troca reduz-se à noção de que, em cada instante, os agentes que participam de uma interação realizam, cada um, um conjunto de ações.

Desse modo, no modelo PopOrg, uma interação entre dois agentes é uma sequência de pares de ações, onde em cada instante cada agente realiza um dos conjuntos de ações do par. Uma especificação de interação entre agentes indica, então, a sequência de pares de conjuntos de ações que “se espera” que os agentes realizem, quando estão realizando a interação.

Em outros termos, a especificação de uma interação no modelo PopOrg é uma definição de uma “interação adequada”, uma “norma de interação”, que - como no caso da especificação de comportamento - admite variações nas realizações dessa interação.

Da mesma forma como acontece em relação aos comportamentos, então, uma interação é considerada “correta” se ela não realiza mais do que o especificado pela interação adequada.

Tal modelo “vetorial” de interação, em que uma interação é uma sequência de vetores de conjuntos de ações, pode ser estendido para a interação de um número qualquer de agentes, o vetor de conjuntos de ações realizado em cada instante tendo número de

componentes igual ao número de agentes participantes da interação. Por simplicidade, no que segue, somente serão consideradas apenas interações entre dois agentes.

Finalmente, cabe observar que, considerado em si mesmo (quer dizer, independentemente dos níveis organizacional e cultural que ele implementa), o nível populacional pode ser visto como uma estrutura de rede social, cujos componentes são os agentes e cujas ligações são processos de troca entre agentes.

4.3.2. Noção Formal

Para a formalização do modelo do nível populacional das sociedades de agentes, assume-se o conhecimento do conjunto de todas as ações que os agentes da sociedade podem realizar. Assume-se também que o tempo tem uma estrutura conhecida. Por simplicidade, assume-se que o tempo tem uma estrutura discreta.

A formalização começa pela definição da noção de comportamento de agente.

Seja S uma sociedade de agentes. Seja Ag o conjunto de agentes da sociedade S . Seja Act_{Ag} o conjunto de todas as ações que os agentes da sociedade de agentes podem realizar. Seja $T = \mathbb{N}$ uma estrutura temporal discreta.

Definição 8 Um comportamento de agente é uma função $b : T \rightarrow \wp(Act_{Ag})$ que, a cada instante de tempo $t \in T$ dá o conjunto de ações $b(t)$ que podem ser realizadas pelo agente que executa o comportamento b . O conjunto de todos os comportamentos possíveis de todos os agentes da sociedade é denotado por Bh_{Ag} e vale $Bh_{Ag} = [T \rightarrow \wp(Act_{Ag})]$.

Definição 9 Um processo de troca entre dois agentes é uma função $e : T \rightarrow \wp(Act_{Ag}) \times \wp(Act)$ que, a cada instante de tempo $t \in T$ dá o par de conjuntos de ações $e(t)$ que podem ser realizadas pelo par de agentes que executa o processo de troca e . O conjunto de todos os processos de troca que todos os agentes da sociedade podem realizar é denotado por Ep_{Ag} e vale $Ep_{Ag} = [T \rightarrow \wp(Act_{Ag}) \times \wp(Act_{Ag})]$.

Definidas as noções de comportamento e processo de troca, é possível definir o nível populacional da sociedade.

Definição 10 O nível populacional de uma sociedade de agentes é uma estrutura $Pop = (Ag, Bh_{Ag}, Ep_{Ag}, bc, ec)$ onde:

- Ag é o conjunto de agentes da sociedade;
- Bh_{Ag} é o conjunto de comportamentos possíveis dos agentes da sociedade;
- Ep_{Ag} é o conjunto dos processos de troca possíveis dos agentes da sociedade;
- $bc : Ag \rightarrow \wp(Bh_{Ag})$ é a função que especifica a capacidade comportamental dos agentes da sociedade, sendo $bc(ag)$ o conjunto de todos os comportamentos que o agente $ag \in Ag$ pode realizar na sociedade;
- $ec : Ag \times Ag \rightarrow \wp(Ep_{Ag})$ é a função que especifica a capacidade de troca dos pares de agentes da sociedade, sendo $ec(a_1, a_2)$ o conjunto de todos os processos de troca que os agentes a_1 e a_2 podem realizar entre si,

de modo que, para todo $t \in T$ e para todo $a_1, a_2 \in Ag$, se $e \in ec(a_1, a_2)$ vale que $prj_1(e(t)) \subseteq \bigcup\{b(t) \mid b \in bh(a_1)\}$ e $prj_2(e(t)) \subseteq \bigcup\{b(t) \mid b \in bh(a_2)\}$, onde prj_1 e prj_2 são funções de projeção.

Em outros termos, o requisito inserido na definição determina que cada processo de troca que possa ser estabelecido entre dois agentes deve ser compatível com as capacidades comportamentais desses agentes.

Note-se que um processo de troca e entre dois agentes a_1 e a_2 não é necessariamente realizado com cada agente destacando um comportamento específico para e : é possível que e seja realizado de modo que as ações de troca de cada agente pertençam a diferentes comportamentos desse agente, em cada instante.

O conjunto de todos os possíveis níveis populacionais de uma sociedade de agentes S é denotado por \mathcal{P}_S

4.4. O Nível Organizacional

O nível organizacional de uma sociedade de agentes é o nível de abstração onde se realizam as estruturas organizacionais da sociedade. É nesse nível que as noções de *papel social* e *interação de papel social* se concretizam e, com base nelas, as noções de *unidade organizacional* e *interação de unidade organizacional*.

Para melhor estruturar o nível organizacional das sociedades de agentes, o modelo PopOrg divide esse nível em dois sub-níveis, o *micro-nível organizacional* e o *macro-nível organizacional*. O micro-nível organizacional é o nível dos papéis sociais e suas interações, enquanto que o macro-nível organizacional é o nível das unidades organizacionais e suas interações.

É importante notar que essa divisão do nível organizacional em níveis micro e macro não é coincidente com a divisão da sociedade em termos de micro-nível social e macro-nível social, como é usual nas ciências sociais (cf. Sec. 2.2). O micro-nível usual das ciências sociais corresponde ao nível populacional do modelo PopOrg, enquanto que o macro-nível corresponde ao nível cultural do modelo PopOrg, definido mais adiante.

O nível organizacional do modelo PopOrg não tem correspondência com os micro e macro-níveis usuais. Por outro lado, no modelo PopOrg, o nível cultural é tratado como uma ampliação do nível organizacional (cf. Sec. 5).

4.4.1. O Micro-Nível Organizacional

Noção Intuitiva. No micro-nível organizacional, modelam-se os papéis sociais e suas interações, considerados independentemente dos agentes que os realizam.

Há uma consequência importante da decisão do modelo PopOrg de considerar os papéis e suas interações de modo independente dos agentes que os realizam: é essa decisão que possibilita o pleno tratamento do nível organizacional com um nível de abstração, independente do nível populacional que lhe subjaz.

Em consequência, é essa mesma decisão que possibilita a independência do nível cultural em relação ao nível populacional, dado o fato de o nível cultural ser tratado como uma ampliação do nível organizacional.

Cabe observar que, considerado em si mesmo (quer dizer, independentemente do modo como é capaz de implementar o macro-nível organizacional e, também, indepen-

dentemente do modo como é capaz de suportar as construções do micro-nível cultural, que o estende), o micro-nível organizacional pode ser entendido como uma rede social, cujos componentes são papéis sociais, não agentes, e cujas ligações são processos de troca entre papéis.

Por outro lado, é interessante notar que o micro-nível organizacional corresponde ao que, no modelo organizacional subjacente à metodologia Gaia foi chamado de “modelo de papéis” de um sistema multiagente [31].

Finalmente, cabe mencionar o modo como o micro-nível organizacional relaciona-se com o nível populacional que lhe subjaz, na hierarquia de níveis de abstração das sociedades de agentes. Esse relacionamento se dá através de uma “relação de implementação”.

A relação de implementação do micro-nível organizacional pelo nível populacional se dá de modo que cada papel do micro-nível organizacional é implementado por um ou mais agentes do nível populacional e cada processo de troca entre papéis é implementado por um ou mais processos de troca realizado entre agentes.

Novamente, vale a observação de que as restrições impostas sobre a correção da relação de implementação de papéis e de ligações entre papéis, conforme se pode observar a seguir, têm a intenção de determinar o conjunto de possibilidades de implementações corretas (ainda que incompletas) desses papéis e ligações, não o modo obrigatório de fazê-lo.

Noção Formal. Para formalizar a noção de micro-nível organizacional de uma sociedade de agentes, começamos por formalizar a noção de papel social.

Seja Act_ω o conjunto não-vazio das ações que os papéis sociais de uma sociedade de agentes podem realizar. O conjunto de todos os comportamentos que esses papéis podem realizar é denotado por Bh_ω e o conjunto de todos os seus possíveis processos de troca é denotado por Ep_ω .

Definição 11 Um papel social em uma sociedade de agentes é um conjunto de comportamentos $r \in \wp(Bh_\omega)$. O conjunto dos papéis sociais da sociedade é denotado por Ro_ω .

A definição de papel social como um conjunto de comportamentos é uma definição minimal, compatível com a idéia de minimalidade e extensionalidade do modelo PopOrg. O sentido da definição é o de que um papel social define o conjunto de comportamentos que um agente deve ser capaz de realizar, quando implementa esse papel, não o conjunto de comportamentos que ele deve realizar necessariamente.

Como a noção de micro-ligação social entre papéis sociais é entendida no modelo PopOrg em termos de processos de troca entre papéis, para formalizar a noção de micro-ligação social é preciso definir, antes, o conjunto dos processos de troca que os papéis sociais existentes na sociedade podem realizar entre si.

Definição 12 Um processo de troca entre dois papéis sociais é uma função $e : T \rightarrow \wp(Act_\omega) \times \wp(Act_\omega)$ que, a cada instante de tempo $t \in T$ dá o par de conjuntos de ações $e(t)$ que podem ser realizadas pelo par de papéis que executa o processo de troca e . O conjunto de todos os processos de troca que todos os papéis sociais podem realizar é denotado por Ep_ω e vale que $Ep_\omega = [T \rightarrow \wp(Act_\omega) \times \wp(Act_\omega)]$.

Como no caso dos processos de troca entre agentes, os processos de trocas envolvidos nas micro-ligações entre papéis sociais não precisam ser realizados necessariamente por um único par de comportamentos, um em cada papel.

Agora, é possível definir a noção de micro-ligação social.

Definição 13 Uma micro-ligação social *entre papéis sociais de uma sociedade de agentes é uma estrutura $l = (r_1, r_2, e)$ onde:*

- $r_1, r_2 \in Ro_\omega$ são os papéis ligados pela ligação social;
- $e \in Ep_\omega$ é o processo de troca que os papéis r_1 e r_2 realizam quando estão ligados pela micro-ligação social l .

O conjunto das micro-ligações sociais de uma sociedade de agentes é denotado por Li_ω .

Definição 14 Diz-se que uma micro-ligação social $l = (r_1, r_2, e)$ está bem definida se e somente se, para todo $t \in T$, vale que $prj_1(e(t)) \subseteq \bigcup\{b(t) \mid b \in r_1\}$ e $prj_2(e(t)) \subseteq \bigcup\{b(t) \mid b \in r_2\}$, isto é, o processo de troca que operacionaliza a micro-ligação é compatível com as capacidades comportamentais dos papéis que o realizam.

Finalmente, é possível definir a noção de micro-nível organizacional de uma sociedade de agentes.

Definição 15 O micro-nível organizacional de uma sociedade de agentes é uma estrutura $Org_\omega = (Ro_\omega, Li_\omega, lc_\omega)$ onde:

- Ro_ω é o conjunto de papéis sociais do micro-nível organizacional;
- Li_ω é o conjunto das micro-ligações sociais do micro-nível organizacional;
- $lc_\omega : Ro \times Ro \rightarrow \wp(Li_\omega)$ é a função de capacidade de micro-ligações, onde $lc(r_1, r_2)$ é o conjunto de micro-ligações sociais que os papéis r_1 e r_2 podem estabelecer entre si,

de modo que, para todo $r_1, r_2 \in Ro_\omega$, todo $l = (r_1, r_2, e) \in lc(r_1, r_2)$ e todo tempo $t \in T$ vale que $prj_1(e(t)) \subseteq \bigcup\{b \mid b \in r_1\}$ e $prj_2(e(t)) \subseteq \bigcup\{b \mid b \in r_2\}$.

Note-se que o requisito inserido na definição determina que, para cada micro-ligação social que possa ser estabelecida entre dois papéis, o processo de troca envolvido nessa micro-ligação seja compatível com os comportamentos que definem os papéis nela envolvidos.

O conjunto de todos os possíveis micro-níveis organizacionais de uma sociedade de agentes S é denotado por $\mathcal{O}\nabla}_\omega$.

4.4.2. O Macro-Nível Organizacional

Noção Intuitiva. O macro-nível organizacional do modelo PopOrg é o nível de abstração em que existem o conjunto de unidades organizacionais e o conjunto de suas macro-ligações organizacionais.

Uma unidade macro-organizacional é definida recursivamente como sendo ou uma unidade macro-organizacional básica ou um conjunto de unidades organizacionais e suas macro-ligações.

As unidades macro-organizacionais básicas são, essencialmente, papéis sociais transportados do micro-nível organizacional para o macro-nível organizacional. As macro-ligações ligam unidades macro-organizacionais através dos processos de troca que são estabelecidos entre elas.

A utilização da noção de unidade organizacional no modelo PopOrg está em consonância com trabalhos que tem buscado na Teoria da Organização uma noção estruturante que evite os problemas do emprego da própria noção de agente para uma formulação recursiva do processo de estruturação de sistemas baseados em agentes (cf., p.ex., [28] e [2]).

A noção recursiva de unidade organizacional generaliza a noção de unidade organizacional estabelecida na Teoria da Organização (cf., p.ex., [26], onde uma hierarquia fixa de quatro níveis de estruturação organizacional é proposta).

Como acontece com as micro-ligações sociais, as macro-ligações entre unidades organizacionais são concebidas como sendo realizadas por processos de troca.

As unidades organizacionais do macro-nível organizacional são a base estrutural sobre as quais se assentam as instituições do nível cultural da sociedade.

Noção Formal. Para formalizar a noção de macro-nível organizacional de uma sociedade de agentes, começamos por formalizar a noção de unidade organizacional.

Seja Act_Ω o conjunto das ações que as unidades organizacionais de uma sociedade de agentes podem realizar. O conjunto de todos os comportamentos que essas unidades podem realizar é denotado por Bh_Ω e o conjunto de todos os seus possíveis processos de troca é denotado por Ep_Ω .

Definição 16 *Uma unidade organizacional em uma sociedade de agentes é:*

- ou uma unidade organizacional básica, i.e., uma estrutura $u = (\{r\}, B, \text{imp}_{\{r\}})$ onde $r \in Ro$ é um papel social, $B \in \wp(Bh_\Omega)$ é o conjunto de comportamentos de macro-nível que o papel deve ser capaz de realizar quando está inserido como unidade organizacional no macro-nível organizacional e $\text{imp}_{\{r\}} : Act_\Omega \rightarrow \wp(Act_\omega)$ é a função de implementação de ações de macro-nível do papel r , de modo que:
 - para toda ação $\alpha \in Act_\Omega$ vale que se existe tempo $t \in T$ em que $\alpha \in b(t)$ para algum $b \in B$, então ocorre que $\text{imp}_r(\alpha) \subseteq \bigcup\{b'(t) \mid b' \in r\}$, isto é, para toda ação de macro-nível que o papel deve realizar enquanto unidade organizacional, existe um conjunto de ações de micro-nível que implementa aquela ação de macro-nível;
- ou uma unidade organizacional composta, quer dizer, uma estrutura $u = (U, B, \text{imp}_U)$, onde U é um conjunto não vazio de unidades organizacionais (básicas ou compostas) tais que $u \notin U$, $B \subseteq Bh_\Omega$ é o conjunto de comportamentos de macro-nível que o conjunto de unidades organizacionais U deve ser capaz de realizar quando está inserido como unidade organizacional no macro-nível organizacional e $\text{imp}_U : Act_\Omega \rightarrow \wp(Act_\Omega)$ é a função de implementação de ações de macro-nível das unidades organizacionais U , de modo que:
 - para toda ação $\alpha \in Act_\Omega$ vale que se existe tempo $t \in T$ em que $\alpha \in b(t)$ para algum $b \in B$, então ocorre que $\text{imp}_U(\alpha) \subseteq \bigcup\{b'(t) \mid b' \in beh(U)\}$,

onde $beh(U) = \bigcup\{B' \mid (U', B', imp_{U'}) \in U\}$, isto é, para toda ação de macro-nível que o conjunto de unidades organizacionais U deve realizar enquanto unidade organizacional, existe um conjunto de ações de macro-nível de algumas unidades organizacionais de U que implementam aquela ação de macro-nível.

O conjunto das unidades organizacionais de macro-nível de uma sociedade de agentes é denotado por U_Ω .

Agora, é possível definir a noção de macro-ligaçāo organizacional.

Definição 17 Uma macro-ligaçāo organizacional é uma estrutura $l = (u_1, u_2, e)$ onde:

- $u_1, u_2 \in U_\Omega$ são as unidades organizacionais ligadas pela macro-ligaçāo;
- $e \in Ep_\Omega$ é o processo de troca que as unidades organizacionais u_1 e u_2 realizam quando estão ligados pela macro-ligaçāo l .

O conjunto das macro-ligaçāes sociais de uma sociedade de agentes é denotado por Li_Ω .

Definição 18 Diz-se que uma macro-ligaçāo social $l = (u_1, u_2, e)$ está bem definida se e somente se, para todo $t \in T$, vale que $prj_1(e(t)) \subseteq \bigcup\{b(t) \mid b \in B_1, u_1 = (U_1, B_1, imp_{U_1})\}$ e $prj_2(e(t)) \subseteq \bigcup\{b(t) \mid b \in B_2, u_2 = (U_2, B_2, imp_{U_2})\}$, isto é, o processo de troca que operacionaliza a macro-ligaçāo é compatível com as capacidades comportamentais das unidades organizacionais que o realizam.

Finalmente, é possível definir a noção de macro-nível organizacional de uma sociedade de agentes.

Definição 19 O macro-nível organizacional de uma sociedade de agentes é uma estrutura $Org_\Omega = (U_\Omega, Li_\Omega, lc_\Omega)$ onde:

- U_Ω é o conjunto de unidades organizacionais do macro-nível organizacional;
- Li_Ω é o conjunto das macro-ligaçāes organizacionais do macro-nível organizacional;
- $lc_\Omega : U_\Omega \times U_\Omega \rightarrow \wp(Li_\Omega)$ é a função de capacidade de macro-ligaçāes, onde $lc(u_1, u_2)$ é o conjunto de macro-ligaçāes sociais que as unidades organizacionais u_1 e u_2 podem estabelecer entre si,

de modo que, para todo $u_1, u_2 \in U_\Omega$, todo $l = (u_1, u_2, e) \in lc(u_1, u_2)$ e todo tempo $t \in T$ vale que $prj_1(e(t)) \subseteq \bigcup\{b \mid b \in B_1, u_1 = (U_1, B_1, imp_{U_1})\}$ e $prj_2(e(t)) \subseteq \bigcup\{b \mid b \in B_2, u_2 = (U_2, B_2, imp_{U_2})\}$.

O requisito inserido na definição determina que, para cada macro-ligaçāo social que possa ser estabelecida entre duas unidades organizacionais, o processo de troca envolvido nessa macro-ligaçāo seja compatível com os comportamentos que aquelas unidades organizacionais são capazes de realizar.

O conjunto de todos os possíveis macro-níveis organizacionais de uma sociedade de agentes S é denotado por \mathcal{ON}_Ω .

4.4.3. A Relação de Implementação do Nível Organizacional

O nível organizacional de uma sociedade de agentes é implementado pelo nível populacional. Como o nível organizacional está dividido em dois subníveis, o micro e o macro-nível organizacional, a relação que modela a implementação do nível organizacional pelo populacional também está dividida em duas relações, uma relação de implementação do micro-nível organizacional pelo nível populacional e uma relação de implementação do macro-nível organizacional pelo micro-nível organizacional.

A composição das duas relações de implementação define a relação de implementação do macro-nível organizacional pelo nível populacional.

A Relação de Implementação do Micro-nível Organizacional Define-se, me primeiro lugar, a relação de implementação do micro-nível organizacional pelo nível populacional.

Definição 20 A implementação do micro-nível organizacional $Org_\omega = (Ro_\omega, Li_\omega, lc_\omega)$ de uma sociedade de agentes pelo nível populacional $Pop = (Ag, Bh_{Ag}, Ep_{Ag}, bc, ec)$ dessa sociedade é dado pela relação de implementação do micro-nível organizacional $imp_\omega \subseteq (Ro_\omega \times Ag) \cup (Li_\omega \times Ep_{Ag})$ onde:

- se $(r, a) \in imp_\omega$ então o papel social $r \in Ro_\omega$ é implementado pelo agente $a \in Ag$;
- se $(l, e) \in imp_\omega$ então a micro-ligação $l \in Li_\omega$ é implementada pelo processo de troca $e \in Ep_{Ag}$.

Caracterizada a relação de implementação do micro-nível organizacional pelo nível populacional de uma sociedade é importante estabelecer as condições em que a implementação dada por essa relação está propriamente definida. Para isso, primeiro define-se a noção de papel social propriamente implementado, depois a noção de micro-ligação social propriamente implementada.

Definição 21 Diz que um papel social $r \in Ro_\omega$ do micro-nível organizacional de uma sociedade de agentes está propriamente implementado por um conjunto de agentes $A \in \wp(Ag)$ do nível populacional dessa sociedade, através da relação de implementação de micro-nível organizacional imp_ω , se e somente se:

- para todo agente $a \in A$ vale que $(r, a) \in imp_\omega$, isto é, cada agente do conjunto A implementa o papel;
- para todo tempo $t \in T$ e todo $b \in r$ vale que $b(t) \subseteq \bigcup\{b'(t) \mid b' \in bc(a), a \in A\}$, isto é, cada um dos comportamentos do papel r pode ser realizado pelos agentes do conjunto A (em um modo compartilhado, possivelmente não exclusivo).

Definição 22 Diz que uma micro-ligação social $l \in Li_\omega$ do micro-nível organizacional de uma sociedade de agentes está propriamente implementado por um conjunto de processos de troca $E \in \wp(Ep_{Ag})$ do nível populacional dessa sociedade, através da relação de implementação de micro-nível organizacional imp_ω , se e somente se:

- para todo processo de troca $e \in E$ vale que $(l, e) \in imp_\omega$, isto é, cada processo de troca do conjunto E implementa a micro-ligação;

- se $l = (r_1, r_2, e)$ então, para todo $t \in T$ vale que $\text{prj}_1(e(t)) \subseteq \bigcup\{\text{prj}_1(e'(t)) \mid e' \in E\}$ e $\text{prj}_2(e(t)) \subseteq \{\text{prj}_2(e'(t)) \mid e' \in E\}$, isto é, o processo de troca que operacionaliza a micro-ligaçāo organizacional l pode ser implementado pelo conjunto de processos de troca E .

Pode-se, então, definir a noção de implementação apropriada do micro-nível organizacional de uma sociedade pelo nível populacional dessa sociedade.

Definição 23 Diz-se que o micro-nível organizacional $Org_\omega = (Ro_\omega, Li_\omega, lc_\omega)$ de uma sociedade de agentes está propriamente implementado pelo nível populacional $Pop = (Ag, Bh_{Ag}, Ep_{Ag}, bc, ec)$ dessa sociedade através da relação de implementação de micro-nível imp_ω se e somente se:

- cada papel social $r \in Ro_\omega$ está propriamente implementado por um conjunto de agentes $A \in \wp(Ag)$, através da relação imp_ω ;
- cada micro-ligaçāo social $l \in Li_\omega$ está propriamente implementada por um conjunto de processos de troca $E \in \wp(Ep_{Ag})$, através da relação imp_ω .

Se o micro-nível organizacional Org_ω está propriamente implementado pelo nível populacional Pop através da relação de implementação de micro-nível imp_ω , diz-se que imp_ω é uma relação de implementação apropriada.

A Relação de Implementação do Macro-nível Organizacional Define-se, em primeiro lugar, a relação de implementação do macro-nível organizacional pelo nível micro-organizacional.

Definição 24 A implementação do macro-nível organizacional $Org_\Omega = (U_\Omega, Li_\Omega, lc_\Omega)$ de uma sociedade de agentes pelo micro-nível organizacional $Org_\omega = (Ro_\omega, Li_\omega, lc_\omega)$ dessa sociedade é dado pela relação de implementação do macro-nível organizacional $\text{imp}_\Omega \subseteq (U_\omega \times Ro_\omega) \cup (Li_\Omega \times Li_\omega)$ onde:

- se $(u, r) \in \text{imp}_\Omega$ então a unidade organizacional $u \in U_\Omega$ é implementada pelo papel social $r \in Ro_\omega$;
- se $(l, l') \in \text{imp}_\Omega$ então a macro-ligaçāo $l \in Li_\Omega$ é implementada pela micro-ligaçāo $l \in Li_\omega$.

Caracterizada a relação de implementação do macro-nível organizacional pelo micro-nível organizacional de uma sociedade é importante estabelecer as condições em que a implementação dada por essa relação está propriamente definida. Para isso, primeiro define-se a noção de unidade organizacional propriamente implementada, depois a noção de macro-ligaçāo social propriamente implementada.

Definição 25 Diz que uma unidade organizacional $u \in U_\Omega$ do macro-nível organizacional de uma sociedade de agentes está propriamente implementada por um conjunto de papéis sociais $R \in \wp(Ro_\omega)$ do micro-nível organizacional dessa sociedade através da relação de implementação de macro-nível organizacional imp_Ω se e somente se:

- para todo papel social $r \in R$ vale que $(u, r) \in \text{imp}_\Omega$, isto é, cada papel social do conjunto R implementa a unidade organizacional;

- se $u = (U, B, \text{imp}_u)$ então, para todo tempo $t \in T$ e todo $b \in B$ vale que $b(t) \subseteq \bigcup\{b'(t) \mid b' \in r\}$, isto é, cada um dos comportamentos da unidade organizacional r pode ser realizado pelos papéis sociais do conjunto R (em um modo compartilhado, possivelmente não exclusivo).

Definição 26 Diz que uma macro-ligaçāo organizacional $l \in Li_\Omega$ do macro-nível organizacional de uma sociedade de agentes está propriamente implementado por um conjunto de micro-ligações organizacionais $L \in \wp(Li_\Omega)$ através da relação de implementação de macro-nível organizacional imp_Ω se e somente se:

- para toda micro-ligaçāo $l' \in L$ vale que $(l, l') \in \text{imp}_\Omega$, isto é, cada micro-ligaçāo do conjunto L implementa a macro-ligaçāo;
- se $l = (u_1, u_2, e)$ então, para todo $t \in T$ vale que $\text{prj}_1(e(t)) \subseteq \bigcup\{\text{prj}_1(e'(t)) \mid e' \in E\}$ e $\text{prj}_2(e(t)) \subseteq \{\text{prj}_2(e'(t)) \mid e' \in E\}$, isto é, o processo de troca que operacionaliza a micro-ligaçāo organizacional l pode ser implementado pelo conjunto de processos de troca E .

Pode-se, então, definir a noção de implementação apropriada do macro-nível organizacional de uma sociedade pelo micro-nível organizacional dessa sociedade.

Definição 27 Diz-se que o macro-nível organizacional $Org_\Omega = (U_\Omega, Li_\Omega, lc_\Omega)$ de uma sociedade de agentes está propriamente implementado pelo micro-nível organizacional $Org_\omega = (Ro_\omega, Li_\omega, lc_\omega)$ dessa sociedade através da relação de implementação de micro-nível imp_Ω se e somente se:

- cada unidade organizacional $u \in U_\Omega$ está propriamente implementado por um conjunto de papéis sociais $R \in \wp(Ro_\omega)$, através relação imp_Ω ;
- cada macro-ligaçāo organizacional $l \in Li_\Omega$ está propriamente implementada por um conjunto de micro-ligações organizacionais $L \in \wp(Li_\Omega)$, através da relação imp_Ω .

Se o macro-nível organizacional Org_Ω está propriamente implementado pelo micro-nível organizacional Org_ω através da relação de implementação de macro-nível imp_Ω , diz-se que imp_Ω é uma relação de implementação apropriada.

Finalmente, pode-se definir a relação de implementação do macro-nível organizacional de uma sociedade de agentes pelo nível populacional da sociedade.

Definição 28 A implementação do macro-nível organizacional $Org_\Omega = (U_\Omega, Li_\Omega, lc_\Omega)$ de uma sociedade de agentes pelo nível populacional $Pop = (Ag, Bh_{Ag}, Ep_{Ag}, bc, ec)$ dessa sociedade é dado pela relação de implementação organizacional composta $\text{imp}_{\omega\Omega} \subseteq (U_\omega \times Ag) \cup (Li_\Omega \times Ep_{Ag})$ onde:

- se $(u, a) \in \text{imp}_\Omega$ então a unidade organizacional $u \in U_\Omega$ é implementada pelo agente $a \in Ag$;
- se $(l, e) \in \text{imp}_\Omega$ então a macro-ligaçāo $l \in Li_\Omega$ é implementada pelo processo de troca $e \in Ep_{Ag}$.

A caracterização da relação de implementação organizacional composta como a composição das relações de micro e macro-implementação pode ser dada como segue.

Definição 29 Diz-se que a relação de implementação organizacional composta $\text{imp}_{\omega\Omega} \subseteq (U_\omega \times Ag) \cup (Li_\Omega \times Ep_{Ag})$ é a composição da relação de implementação de micro-nível organizacional $\text{imp}_\omega \subseteq (Ro_\omega \times Ag) \cup (Li_\omega \times Ep_{Ag})$ com a relação de macro-nível organizacional $\text{imp}_\Omega \subseteq (U_\Omega \times Ro_\omega) \cup (Li_\Omega \times Li_\omega)$, denotada por $\text{imp}_{\omega\Omega} = \text{imp}_\omega \circ \text{imp}_\Omega$, se e somente se:

- para cada unidade organizacional $u \in U_\Omega$ existe um conjunto de papéis sociais $R \subseteq Ro_\omega$ e uma família de conjuntos de agentes $\mathbb{A} \subseteq \wp(Ag)$ tal que a unidade organizacional u é implementada pelo conjunto de papéis sociais R através de imp_Ω e para cada papel social $r \in R$ existe um conjunto de agentes $A \in \mathbb{A}$ tal que r é implementado pelo conjunto de agentes A através de imp_ω ;
- para cada macro-ligação organizacional $l \in Li_\Omega$ existe um conjunto de micro-ligações sociais $L \subseteq Li_\omega$ e uma família de processos de troca $\mathbb{E} \subseteq \wp(Ep_{Ag})$ tal que a macro-ligação organizacional l é implementada pelo conjunto de micro-ligações sociais L através de imp_Ω e para cada micro-ligação social $l' \in L$ existe um conjunto de processos de troca $E \in \mathbb{E}$ tal que l' é implementada pelo conjunto de processos de troca E através de imp_ω .

Pode-se, então, definir a noção de implementação apropriada do macro-nível organizacional de uma sociedade pelo nível populacional dessa sociedade.

Definição 30 Diz-se que o macro-nível organizacional $Org_\Omega = (U_\Omega, Li_\Omega, lc_\Omega)$ de uma sociedade de agentes está propriamente implementado pelo nível populacional $Pop = (Ag, Bh_{Ag}, Ep_{Ag}, bc, ec)$ dessa sociedade através do micro-nível organizacional $Org_\omega = (Ro_\omega, Li_\omega, lc_\omega)$ e da relação de implementação composta $\text{imp}_{\omega\Omega}$ se e somente se:

- $\text{imp}_{\omega\Omega} = \text{imp}_\omega \circ \text{imp}_\Omega$
- a relação de implementação imp_ω é apropriada;
- a relação de implementação imp_Ω é apropriada.

Se o macro-nível organizacional Org_Ω está propriamente implementado pelo nível populacional Pop através da relação de implementação composta $\text{imp}_{\omega\Omega}$, diz-se que $\text{imp}_{\omega\Omega}$ é uma relação de implementação apropriada.

Nessas condições, vale o seguinte lema, relativo à realizabilidade do funcionamento das sociedades de agentes [9]:

Lema 1 Se o macro-nível organizacional Org_Ω de uma sociedade de agentes é propriamente implementado pelo nível populacional Pop da sociedade (através uma relação de implementação composta apropriada) então cada ação realizada por uma unidade organizacional de Org_Ω é propriamente implementada por um conjunto de ações de agentes de Pop .

O lema garante a validade do seguinte teorema sobre a consistência implementacional das sociedades de agentes [9]:

Teorema 1 (Consistência Implementacional) Se o macro-nível organizacional Org_Ω de uma sociedade de agentes é propriamente implementado pelo nível populacional Pop então:

- cada comportamento de unidade organizacional do macro-nível organizacional é propriamente implementado por um conjunto de comportamentos de agentes em Pop ;

- *cada macro-ligaçāo organizacional entre unidades organizacionais do macro-nível é propriamente implementado por um conjunto de processos de troca entre agentes em Pop.*

O teorema da consistência implementacional é importante porque garante que, em sociedades de agentes com níveis macro-organizacionais propriamente implementados, os comportamentos e interações das unidades organizacionais são consistentemente realizados pelos agentes que habitam a sociedade.

Observa-se, por outro lado, os vários conjuntos de ações Act_Ω , Act_ω e Act_{Ag} não foram correlacionados, até aqui.

Em alguns casos simples, a relação entre esses conjuntos é a inclusão $Act_\Omega \subseteq Act_\omega \subseteq Act_{Ag}$, o que foi adotado nas definições acima.

De um modo geral, o relacionamento entre esses conjuntos de ações não é de inclusão, mas sim de implementação: cada ação em cada nível é implementada por um conjunto de ações do nível imediatamente inferior.

Nessas condições, os conjuntos de ações são relacionados por funções do tipo:

- $\text{imp}_{Act_\Omega} : Act_\Omega \rightarrow \wp(Act_\omega);$
- $\text{imp}_{Act_\omega} : Act_\omega \rightarrow \wp(Act_{Ag});$

e todos os requisitos indicados acima, envolvendo inclusões de conjuntos de ações entre comportamentos de níveis de abstração diferentes, e inclusões de projeções de processos de troca de níveis de abstração diferentes, devem ser reescritos para explicitar os mapeamentos imp_{Act_Ω} e imp_{Act_ω} .

4.5. O Modelo Mínimo

Definidos todos os elementos acima, pode-se definir agora o modelo PopOrg mínimo de uma sociedade de agentes.

Definição 31 Uma sociedade de agentes S é uma estrutura $S = (Pop, Org_{\omega\Omega}, \text{imp}_{\omega\Omega})$ onde:

- $Pop \in \mathcal{P}l$ é o nível populacional;
- $Org_{\omega\Omega} = (\sqrt[S]{Org_\omega, Org_\Omega}) \in \mathcal{O}\nabla}_S$ é o nível organizacional da sociedade, onde:
 - Org_ω é o micro-nível organizacional;
 - Org_Ω é o macro-nível organizacional;
- $\text{imp}_{\omega\Omega} = (\text{imp}_\omega, \text{imp}_\Omega)$ é a relação de implementação do nível organizacional pelo nível populacional, onde:
 - imp_ω é a relação de implementação do micro-nível organizacional pelo nível populacional;
 - imp_Ω é a relação de implementação do macro-nível organizacional pelo micro-nível organizacional;

5. O Nível Cultural

5.1. As Abordagens de Radcliffe-Brown e Malinowski à Questão da Cultura

O termo cultura tem sido concebido, na Antropologia e na Filosofia da Cultura, de dois modos diferentes, um com ênfase nos seus aspectos sociais, outro com ênfase nos seus aspectos psicológicos.

Na concepção voltada para captar mais os aspectos psicológicos da cultura (aquele adotada, por exemplo, por Cassirer [4]), o termo cultura designa o sistema de símbolos que suporta atividades humanas como as da ciência, da arte, da religião, da mitologia e da linguagem, símbolos que constituem também os principais produtos dessas atividades.

Nesse primeiro sentido, cultura é o repertório de elementos simbólicos através dos quais as pessoas se mantêm em contato com as outras pessoas, passadas, presentes ou futuras.

Na concepção voltada mais para os aspectos sociais da cultura (aquele adotada, por exemplo, pelos pioneiros da Antropologia [19, 23]), o termo cultura designa o conjunto das formas de organização, interação e integração das atividades humanas em sociedade, as quais se fazem acompanhar de representações simbólicas para permitir sua objetivação e, consequentemente, sua conservação, modificação e transmissão entre gerações.

Entre as abordagens teóricas que adotam essa segunda concepção de cultura, podemos distinguir duas abordagens fundamentais, a de Malinowski e a de Radcliffe-Brown.

Na abordagem de Radcliffe-Brown ao conceito de cultura [23], a análise da cultura, considerada nos termos em que estamos apresentando este trabalho, é realizada nos níveis de abstração inferiores, quais sejam os níveis populacional e micro-organizacional das sociedades, isto é, nos níveis dos funcionamentos dos indivíduos e dos papéis que eles desempenham uns para os outros, quando interagem:

For a preliminary definition of social phenomena it seems sufficiently clear that what we have to deal with are relations of association between individual organisms [human or not, depending on the kind of society we are dealing with]. [23], p. 189.

A particular social relation between two persons [...] exists only as part of a wide network of social relations, involving many other persons, and it is this network which I regard as the object of our investigations. [23], p.191.

Assim, os aspectos culturais enfocados por Radcliffe-Brown dizem respeito aos símbolos que existem nesses níveis de abstração e que interveem nessas redes sociais constituídas pelos indivíduos e pelos seus papéis sociais.

Na abordagem de Malinowski ao conceito de cultura [19], por outro lado, a análise da cultura é realizada nos níveis de abstração superiores, quais sejam, os micro- e macro-níveis organizacionais, isto é, nos níveis dos funcionamentos das unidades organizacionais e dos papéis sociais que as constituem.

Nesses níveis de abstração, redes também são constituídas, apenas que não são redes de indivíduos, mas sim redes de unidades organizacionais, agora equipadas de elementos simbólicos, elas mesmas sendo tratadas, às vezes, como símbolos.

Malinowski chama de “instituições” as unidades organizacionais constituídas com o suporte de elementos simbólicos e, em consequência, as redes que analisa são redes de instituições.

Por outro lado, enquanto Radcliffe-Brown parte do fato imediatamente perceptível de que os indivíduos encontram-se em relação uns com os outros, Malinowski necessita

partir de uma consideração mais abstrata para poder alcançar as instituições como os elementos centrais da organização da sociedade: ele parte das necessidades energéticas dos indivíduos, passa pela necessidade de coordenação das atividades que satisfazem essas necessidades e, só então, alcança a noção de instituição como a unidade organizacional de caráter simbólico que possibilita à sociedade, ao longo do tempo, a manutenção coordenada dessas atividades de obtenção de energia.

Ao traçar esse caminho, alcança duas noções centrais em sua teoria antropológica: a noção de “função”, como sendo a satisfação de uma necessidade pela realização de uma atividade, e a noção de “estrutura”, como sendo a unidade organizacional que possibilita que aquela atividade seja executada e a função realizada.

Nesse sentido, as instituições que ele identifica na constituição das sociedades humanas são as estruturas que realizam as mais diversas funções necessárias ao funcionamento e persistência da sociedade:

The analysis just outlined, in which we attempt to define the relation between a cultural performance and a human need, basic or derived, may be termed functional. For function can not be defined in any other way than the satisfaction of a need by an activity in which human beings cooperate, use artifacts, and consume goods. Yet this very definition implies another principle with which we can concretely integrate any phase of cultural behavior. The essential concept here is that of organization. In order to achieve any purpose, reach any end, human beings have to organize. As we shall show, organization implies a very definite scheme or structure [...]. [19], p.38.

I propose to call such a unit of human organization by the old but not always clearly defined or consistently used term, institution. [19], p.39.

5.2. O Conceito Malinowskiano de Instituição

O que, precisamente, Malinowski chama de instituição pode ser visualizado através do diagrama da Fig. 2.

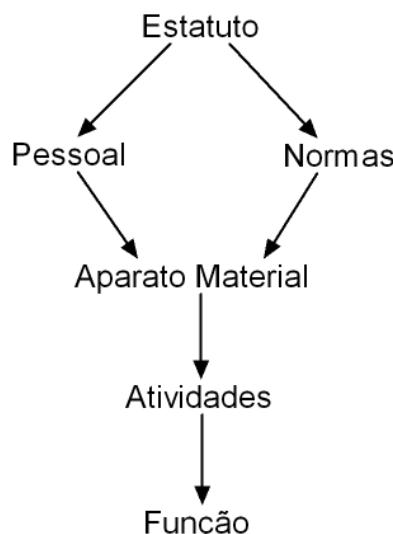


Figura 2. Componentes de uma instituição [19], p.53.

Uma olhada inicial para esse diagrama já revela a diferença existente entre o conceito de unidade organizacional e o conceito Malinowskiano de instituição.

Na concepção extensional de unidade organizacional, que definimos na Sec. 4.4, uma unidade organizacional corresponde à junção dos componentes “pessoal” e “atividades”: as atividades exigem diferenciação de papéis, para sua coordenação, de modo que “papéis sociais” são, também, um componente implícito nessa coleção de componentes.

O modelo extensional das unidades organizacionais apresentado na Sec. 4.4, portanto, constitui uma parte da noção Malinowskiana de instituição.

Se à noção extensional básica de unidade organizacional que definimos adicionarmos o componente “normas”, alcançamos a conceituação intensional básica que a noção de “organização de sistemas multiagentes” tem correntemente na comunidade científica da área [13].

Um passo a mais pode ser dado, acrescentando-se ao modelo o componente “aparato material”. Com esse acréscimo, alcança-se a noção de organização tal como ela vem sendo estendida mais recentemente, através da inclusão do conceito de “ambiente” (cf., p.ex., os trabalhos na série de workshops E4MAS [29], assim como os trabalhos que se seguiram à introdução da noção de “artefato” [24]).

Os dois tipos de componentes das instituições que requerem um exame mais acurado, para relacioná-los aos conceitos extensionais e intensionais de unidade organizacional e de instituição são os componentes “estatuto” e “função”.

A noção de estatuto de uma instituição, em Malinowski, é uma noção composta: o estatuto de uma instituição diz respeito aos “valores” que regulam a estrutura e o funcionamento dessa instituição, entre os quais encontram-se os “objetivos” que essa instituição pretende alcançar.

Pensado no que diz respeito à parte “objetivos”, reconhece-se que vários modelos de organização de sistemas multiagentes contemplam essa noção (p.ex., [17], através da noção de “missão”).

Por outro lado, pensado no que diz respeito à explicitação de outros “valores” (éticos, políticos, etc.) que regulam o funcionamento da instituição, vê-se que os modelos correntes de organização de sistemas multiagentes não contemplam adequadamente esse aspecto. Na verdade, a questão dos valores tem sido muito pouco explorado até mesmo no nível das arquiteturas de agentes [5].

Finalmente, o que é o ponto importante dessa análise das concepções de Malinowski para o presente trabalho, resta examinar a questão do componente “função” de uma instituição.

Como definido por Malinowski, a realização de uma função é a realização da satisfação de uma necessidade através da execução de uma atividade.

Nesses termos, a introdução do conceito de função no contexto das sociedades de agentes deve ser feita através da formalização da noção de “necessidade”, o que requer, por um lado, uma revisão conceitual dos fundamentos da área [8] para se poder determinar o que é uma “necessidade” de um sistema artificial e, por outro lado, requer a consideração das interações das instituições entre si, posto que a referida satisfação das necessidades se

obtém pela realização de atividades de uma instituição para outra.

Finalmente, mas não contemplado como tema do presente texto, incorporamos também ao trabalho as concepções de Cassirer [4] sobre a análise específica dos sistemas de símbolos que objetivam em nível representacional as realizações culturais.

A Fig. 3 sumariza esta análise da noção Malinowskiana de instituição e de sua relação com os modelos de organização de sociedades de agentes.

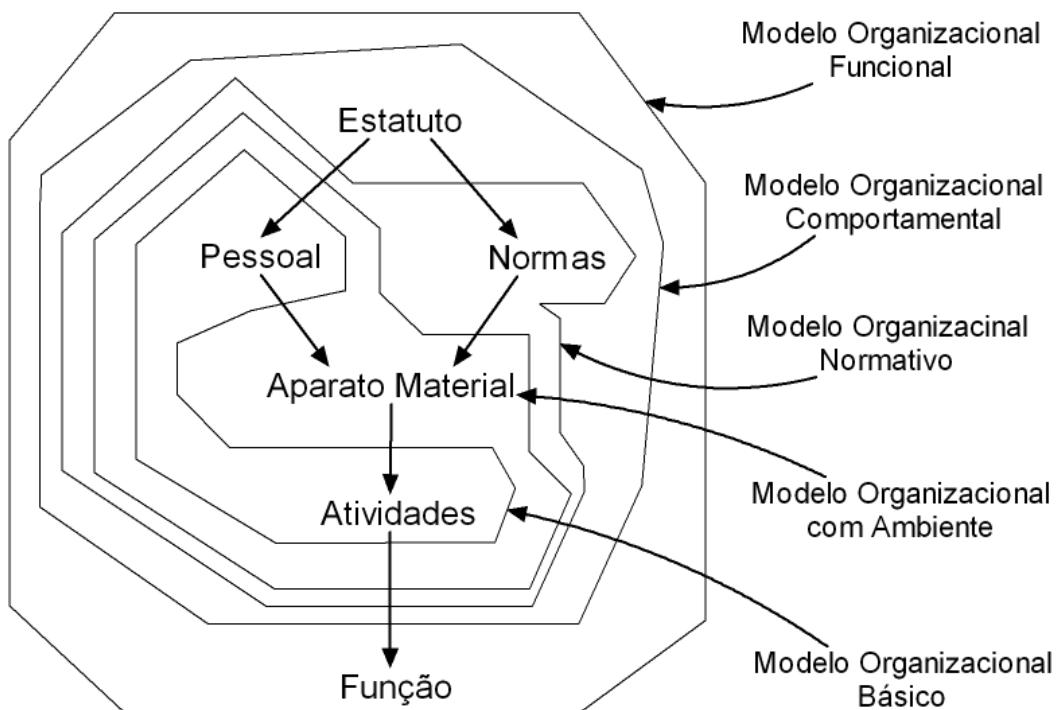


Figura 3. Modelos de Organização e a Noção de Instituição

No que segue, apresentamos uma primeira tentativa de extensão do modelo PopOrg com um nível de abstração correspondente à noção de cultura, chamado de “nível cultural”.

A extensão toma por base uma combinação das concepções teóricas de Radcliffe-Brown, para fazer o nível cultural abranger os níveis populacional e micro-organizacional, assim como as concepções teóricas de Malinowski, para fazê-lo abranger adicionalmente o nível macro-organizacional. Para tanto, fazemos uso essencial do ambiente simbólico do modelo PopOrg (cf. Sec. 4.2).

5.3. A Questão da Cultura no Modelo PopOrg

Introduzir a questão da cultura no modelo PopOrg de sociedades de agentes implica, claramente, em introduzir um aspecto essencialmente intensional em um modelo extensional. O eixo dessa ampliação do modelo deve ser o ambiente simbólico, o único capaz de dar um tratamento extensional a aspectos intensionais.

Assim, o procedimento para a introdução da cultura no modelo PopOrg deve ser o de relacionar o ambiente simbólico com cada um dos componentes do modelo aos quais

se deseja dar uma inserção cultural.

O nível cultural da sociedade de agentes resultará, então, do conjunto dessas articulações entre os componentes dos diversos níveis de abstração da sociedade com os símbolos e estruturas simbólicas residentes no ambiente simbólico.

Para tanto, introduzimos uma noção auxiliar às noções de símbolo, significado e interpretação definidas na Sec. 4.2.

Definição 32 *Seja Σ o conjunto de símbolos de uma sociedade de agentes e $\iota \subseteq \Sigma \times X$ uma relação de interpretação de Σ sobre um conjunto X de significados. Designamos por $\nabla_\iota(\sigma)$ o conjunto das interpretações possíveis do símbolo $\sigma \in \Sigma_S$, isto é, $\nabla_\iota(\sigma) = \{x \in X \mid x \in \iota(\sigma)\}$. Correspondentemente, designamos por $\Delta_\iota(x)$ o conjunto dos símbolos que simbolizam o significado $x \in X$, isto é, $\Delta_\iota(x) = \{\sigma \in \Sigma \mid x \in \iota(\sigma)\}$.*

No que segue, introduzimos os elementos minimamente necessários para caracterizar de modo genérico o nível cultural de uma sociedade de agentes nos termos do modelo PopOrg. Uma definição mais completa parece exigir a vinculação do modelo PopOrg se não a uma sociedade específica, a pelo menos a uma categoria de sociedades cujo sistema cultural tenha estruturas suficientemente especificadas para poderem ser modeladas de modo extensional.

5.4. O Micro-Nível Cultural

5.4.1. Noção Intuitiva.

O micro-nível do nível cultural corresponde à parte do nível cultural que se relaciona com os aspectos culturais dos níveis populacional e micro-organizacional.

Nesse nível, os elementos culturais dizem respeito à simbolização dos elementos envolvidos nas atividades dos agentes na sociedade, nas suas interações, nos papéis que os agentes desempenham uns para os outros quando interagem e nas ligações sociais que tais papéis estabelecem entre si.

Com esse encaminhamento, introduzimos no micro-nível cultural das sociedades de agentes elementos que tem características essencialmente extensionais no nível populacional e no micro-nível organizacional.

Não viabilizamos, contudo, a introdução dos elementos que tem características essencialmente intensionais, tais como valores, normas, objetivos, etc., associados aos agentes e seus papéis sociais.

Para isso, dispomos de duas alternativas: ou fazemos uso dos elementos que compõe as estruturas mentais dos agentes da sociedade, ou nos restringimos (ainda que provisoriamente) a um tratamento puramente sintático das representações dos elementos que têm caráter intensional.

No sentido de preservar o máximo possível o caráter extensional do modelo Po-
pOrg, não faremos aqui uso da primeira alternativa, optando (provisoriamente) pelo trata-
mento puramente sintático da representação cultural dos elementos de caráter intensional,
envolvendo exclusivamente sua forma sintática.

Definimos, então, do seguinte modo, a estrutura mínima do micro-nível cultural.

5.4.2. Noção Formal.

Primeiramente, formalizamos a definição da inclusão no nível cultural dos elementos de caráter extensional.

Definição 33 *Seja X um dos conjuntos de elementos constituintes do nível populacional, do micro-nível organizacional, do ambiente material ou do ambiente simbólico de uma sociedade de agentes S , isto é, $Ag, Act_{Ag}, Bh_{Ag}, Ep_{Ag}, Ro_\omega, Li_\omega, Obj_S$ ou Σ_S . Seja $x \in X$ um elemento de um desses conjuntos. Diz-se que o elemento x tem representação no micro-nível cultural da sociedade se e somente se existe pelo menos uma interpretação de símbolos ι na sociedade e pelo menos um símbolo $\sigma \in \Sigma$ no ambiente simbólico da sociedade tal que o elemento x seja um significado de σ pela interpretação ι , isto é, $x \in \Delta_\iota(\sigma)$.*

Agora, formalizamos de modo puramente sintático a inclusão no micro-nível cultural dos elementos de caráter intensional.

Definição 34 *Os conjuntos de valores V_S , normas N_S e objetivos G_S possíveis de vigarem em uma sociedade de agentes S são definidos como tipos especiais de símbolos do ambiente simbólico da sociedade, isto é, $V_S, N_S, G_S \subseteq \Sigma_S$.*

Definidos assim, provisoriamente, como elementos puramente sintáticos apenas existentes no ambiente simbólico da sociedade, elementos como valores, normas e objetivos só adquirem significação através de relações de interpretação que lhes estabeleçam como significados elementos existentes nas estruturas mentais dos agentes da sociedade.

Em outros termos, somente adquirem significados de caráter subjetivo, individuais aos agentes que os interpretam.

Claramente, essa solução se mostra insatisfatória, pela impossibilidade que apresenta de garantir uniformidade de interpretação dos símbolos entre a população de agentes.

Viabilizar essa possibilidade é uma necessidade em qualquer modelo de sociedade de agentes porque, por maior que seja o número de símbolos cujos significados têm caráter subjetivo, há em toda sociedade um conjunto de símbolos cujos significados necessitam ter um mínimo de caráter objetivo, pela função coletiva que tais símbolos têm na sociedade (p.ex., símbolos linguísticos, etc.).

Como essa é uma tarefa que está fora do escopo do presente trabalho, nos contentamos aqui com a seguinte definição temporária.

Definição 35 *O micro-nível cultural de uma sociedade de agentes S é uma estrutura $CL_\omega = (Org_\omega, ME_\omega, \Sigma_\omega, V_\omega, N_\omega, G_\omega)$ onde:*

- Org_ω é o micro-nível organizacional;
- $ME_\omega \in \wp(ME)$ é a parte do ambiente material relevante para o micro-nível organizacional;
- $SE_\omega \in \wp(ME)$ é a parte do ambiente simbólico relevante para o micro-nível organizacional;
- $V_\omega \subseteq \Sigma_S$ é a parte do conjunto de valores relevante para o micro-nível organizacional;

- $N_\omega \subseteq \Sigma_S$ é o conjunto de normas relevantes para o micro-nível organizacional;
- $G_\omega \subseteq \Sigma_S$ é o conjunto de objetivos relevantes para o micro-nível organizacional.

O conjunto de todos os possíveis micro-níveis culturais de uma sociedade de agentes é denotado por \mathcal{CL}_ω .

5.5. O Macro-Nível Cultural

5.5.1. Noção Intuitiva.

Na modelagem do macro-nível cultural de uma sociedade de agentes, o elemento central é o conjunto de instituições da sociedade. Assim, a macro-nível cultural abrange o macro-nível organizacional da sociedade.

Por outro lado, como as instituições são compostas de unidades organizacionais e suas ligações, as quais por sua vez são compostas por papéis sociais e suas interações, tanto o micro-nível quanto o macro-nível organizacional são componentes implícitos do macro-nível cultural.

Além disso, como visto na Sec. 5.1, o ponto crucial da existência das instituições é a realização de funções, para outras instituições e para a sociedade como um todo, de modo que, ao definir o macro-nível cultural, é essencial considerar a noção de função como uma noção central desse nível.

Intuitivamente, os seguintes elementos entram na concepção da noção de função:

- a necessidade que a função satisfaz;
- a atividade que satisfaz a necessidade;
- a instituição (ou a sociedade como um todo) cuja necessidade a função satisfaz;
- a instituição que realiza a função (i.e., a atividade que satisfaz a necessidade);
- o evento da satisfação da necessidade pela realização da atividade (quer dizer, o evento da realização da função).

Novamente, a questão intensional se instala: o conceito de necessidade é, em geral, um conceito intensional. Contudo, Malinowski procurou distinguir as necessidades que tem caráter essencialmente intensionais (que ele chamou de “necessidades secundárias”) das necessidades que, mesmo tendo um forte caráter intensional, podem ser reduzidas a um caráter extensional (as quais ele chamou de “necessidades primárias”).

Entre as necessidades primárias, a necessidade mais crucial é a necessidade de energia. Assim, como indicado na Sec. 3, tomamos a necessidade de energia como uma característica extensional das sociedades de agentes e, nela, apoiamos a definição do nível cultural.

Por fim, é preciso considerar que, como instituições são constituídas de papéis sociais e suas micro-ligações, a formalização do nível cultural das sociedades de agentes deve levar em conta as relações internas entre as instituições e seus papéis e micro-ligações constituintes.

O conjunto de instituições de uma sociedade, juntamente com as interações que realizam entre si e que viabilizam a realização de suas respectivas funções, constitui então a *estrutura funcional* da sociedade.

5.5.2. Noção Formal.

Como visto na Sec. 5.1, uma função se realiza pela satisfação de uma necessidade, através da execução da função.

Para caracterizar a ocorrência do evento da realização da função, um conjunto de índices pode ser utilizado (situação, tempo, etc.). Seguindo a perspectiva da construção de modelos mínimos, adotada na construção do modelo PopOrg, optamos por utilizar, então, o tempo como índice referencial da realização de uma função.

Formalização da Noção de Função Social Formalmente, denominamos as funções como “funções sociais” e as definimos como segue. Seja $Inst_S$ o conjunto de instituições da sociedade de agentes S (definido formalmente mais adiante).

Definição 36 Uma função social em uma sociedade de agentes é uma estrutura $sf = (nec, actv, inst, t)$ onde:

- $nec \in N_S$ é a necessidade (de algum elemento da sociedade) que a função satisfaz;
- $actv$ é a atividade que satisfaz a necessidade;
- $inst \in Inst_S$ é a instituição que realiza a função;
- $t \in T$ é o tempo em que a função sf é realizada.

Quando uma instituição $inst$ realiza a função de satisfazer a necessidade nec através da realização da atividade $actv$, no tempo t , escrevemos $inst \rightsquigarrow_{actv}^t nec$. O conjunto de funções sociais que as instituições de uma sociedade de agentes pode realizar é denotado por SF_S .

Note-se que, nessa definição, deixamos indeterminada a natureza do componente $actv$, pelo fato de que uma atividade capaz de satisfazer necessidades envolve, em geral, tanto comportamentos quanto interações e, no atual estágio inicial de formalização do nível cultural, não há indicações suficientes nem para se poder fazer uma escolha entre essas alternativas, nem para se poder combiná-las adequadamente (cf. [11]).

Contudo, reservamos o denotador FA_S para o conjunto das atividades das instituições de uma sociedade de agentes S, qualquer que seja o modo como elas sejam definidas, e chamamos essas atividades de *atividades funcionais*.

Por outro lado, para definir formalmente a questão energética das sociedades de agentes e, sobre ela, definir formalmente o conceito de necessidade, estendemos o modelo do ambiente material das sociedades de agentes com uma função que, para cada elemento do ambiente material e para cada instante de tempo, é capaz de indicar a necessidade de energia daquele elemento, naquele instante.

Definição 37 A necessidade energética de um elemento $x \in Obj$ (um agente ou objeto) do ambiente material de uma sociedade de agentes é dada, em cada instante de tempo $t \in T$, pela função $en^t : Obj \rightarrow EN$, onde EN é o conjunto de necessidades energéticas que os elementos do ambiente material podem ter.

Desse modo, uma expressão como $i_1 \rightsquigarrow_{\alpha}^t en(x)$ indica que a instituição i_1 , pela execução da atividade α , realiza a função de satisfazer a necessidade energética $en(x)$ do elemento x , no instante t .

As necessidades das instituições são denominadas, por outro lado, de *necessidades institucionais*.

Formalização da Noção de Instituição Para formalizar a noção de instituição, tomamos como referência a concepção de Malinowski [19] (cf. Sec. 5.1).

Definição 38 Uma instituição em uma sociedade de agentes é uma estrutura $inst = (u, gls, vls, rls, nrms, objs, actvs, funs)$ onde:

- $u \in U_\Omega$ é a unidade organizacional que dá estrutura operacional à instituição;
- $gls \in \wp(G)$ é o conjunto de objetivos;
- $vls \in \wp(V)$ é o conjunto de valores;
- $nrms \in \wp(N)$ é o conjunto de normas;
- $fa \in \wp(A)$ é o conjunto de atividades funcionais;
- $sf \in \wp(F)$ é o conjunto de funções sociais.

Como dito anteriormente, o conjunto das possíveis instituições de uma sociedade de agentes S é denotado por $Inst_S$.

Formalização da Noção de Ligação Funcional Seguindo a abordagem proposta em [11], consideramos que instituições interagem via realização de funções, umas para as outras. Assim, definimos formalmente o conceito de “ligação funcional” entre instituições do modo que segue.

Definição 39 Uma ligação funcional entre duas instituições de uma sociedade de agentes é uma estrutura $fl = (i_1, a, i_2, n)$ onde:

- $i_1 \in Inst$ é a instituição provedora da função, isto é, a instituição que realiza a função;
- a é a atividade realizada pela instituição i_1 e que satisfaz a necessidade institucional n da instituição i_2 ;
- $i_2 \in Inst$ é a instituição beneficiária da função, isto é, a instituição cuja necessidade é satisfeita pela função;
- $n \in N$ é a necessidade institucional da instituição i_2 .

O conjunto das possíveis ligações funcionais de uma sociedade de agentes S é denotado por FL_S .

Formalização da Noção de Estrutura Institucional Definidas as noções de instituição e de ligação funcional de instituições, pode-se, então definir a noção integradora de “estrutura institucional”.

Definição 40 Uma estrutura institucional em uma sociedade de agentes é uma estrutura $IS = (Inst, FL, ilc)$ onde:

- $Inst \in \wp(Inst_S)$ é um conjunto de instituições;
- $FL \in \wp(FL_S)$ é um conjunto de ligações institucionais;
- $ilc : Inst \times Inst \rightarrow \wp(FL)$ é a função que, para cada par de instituições indica a capacidade de ligações institucionais que elas tem.

O conjunto das possíveis estruturas institucionais de uma sociedade de agentes S é denotado por IS_S .

Formalização da Noção de Macro-Nível Cultural Dadas as definições acima, formalizamos a noção de macro-nível cultural do seguinte modo.

Definição 41 *O macro-nível cultural de uma sociedade de agentes S é uma estrutura $CL_\Omega = (IS_\Omega, ME_\Omega, SE_\Omega, V_\Omega, N_\Omega, G_\Omega)$ onde:*

- $IS_\Omega \in \wp(IS_S)$ é a estrutura institucional global do macro-nível cultural;
- $ME_\Omega \in \wp(ME)$ é a parte do ambiente material relevante para o macro-nível organizacional;
- $SE_\Omega \in \wp(SE)$ é a parte do ambiente simbólico relevante para o macro-nível organizacional;
- $V_\Omega \subseteq \Sigma_S$ é a parte do conjunto de valores relevante para o macro-nível organizacional;
- $N_\Omega \subseteq \Sigma_S$ é o conjunto de normas relevante para o macro-nível organizacional;
- $G_\Omega \subseteq \Sigma_S$ é o conjunto de objetivos relevante para o macro-nível organizacional.

O conjunto de todos os possíveis macro-níveis culturais de uma sociedade de agentes é denotado por \mathcal{CL}_Ω .

Formalização da Noção de Nível Cultural Finalmente, podemos formalizar a noção de nível cultural de uma sociedade de agentes.

Definição 42 *O nível cultural de uma sociedade de agentes é uma estrutura $CL = (CL_\omega, Cl_\Omega, imp_\Omega)$ onde:*

- $CL_\omega \in \mathcal{CL}_\omega$ é o micro-nível cultural;
- $CL_\Omega \in \mathcal{CL}_\Omega$ é o macro-nível cultural;
- imp_Ω é a relação de implementação do macro-nível cultural sobre o micro-nível cultural.

O conjunto de todos os possíveis níveis culturais de uma sociedade de agentes é denotado por \mathcal{CL} .

O Modelo Mínimo Estendido com o Nível Cultural A extensão do modelo mínimo PopOrg de uma sociedade de agentes pode, agora, ser formalmente estendido com a noção de nível cultural.

Definição 43 *Uma sociedade de agentes com nível cultural S_{CL} é uma estrutura $S_{CL} = (Pop, CL, imp_{CL}, imp_{CL}, ME, SE)$ onde:*

- $Pop \in \mathcal{P}_\Omega$ é o nível populacional;
- $CL \in \mathcal{CL}$ é o nível cultural;
- imp_{CL} é a relação de implementação do nível cultural sobre o nível populacional;
- $ME \in \mathcal{ME}$ é o ambiente material;
- $SE \in \mathcal{SE}$ é o ambiente simbólico.

6. Conclusão

Este texto apresentou, de modo preliminar, a formalização da noção de nível cultural de sociedade de agentes. Para tanto, tomou por base o modelo PopOrg de organização dessas sociedades.

A idéia central do texto é vincular a estruturação do nível cultural à estruturação do modelo PopOrg, organizando-o em termos dos dois sub-níveis organizacionais, micro e macro.

A formalização da noção de nível cultural apresentada encontra-se em estágio preliminar porque noções centrais para o trabalho ainda precisam ser melhor elaboradas e, posteriormente, formalizadas. Entre eles:

- a noção de “significado” de símbolo, quando o significado tem uma carga essencial de características intensionais; a solução da formalização apenas sintática, adotada no texto, é insuficiente;
- a noção de “necessidade” no contexto das sociedades de agentes: necessidades de agentes, necessidades de instituições; o texto deixou a noção indefinida, abrindo apenas a possibilidade de incorporação da noção em relação à energia; uma definição mais ampla e precisa é necessária;
- a noção de “atividade funcional” que satisfaz uma necessidade também restou indefinida, correspondendo à indefinição da noção de necessidade; uma vez obtido uma definição adequada da noção genérica de necessidade, a noção de atividade funcional poderá, então, também ser definida.

Por fim, cabe comentar que a formalização apresentada deixou de lado a questão da existência de elementos intensionais pertencentes à sociedade como um todo, limitando-se a propor elementos intensionais gerais (valores, normas, objetivos, etc.) apenas para as instituições que constituem a sociedade. Uma análise na linha da Filosofia das Formas Simbólicas pode, se convenientemente elaborada, ajudar talvez a resolver esse ponto.

O texto está apresentado como sendo a versão 0.1 do trabalho. Isso é uma indicação de que o autor gostaria muito de receber comentários, críticas e sugestões a respeito do trabalho.

Agradecimentos

As idéias colocadas neste texto constituem a base da palestra de mesmo título apresentada no WESAAC 2011 (5º. Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações), evento realizado na UTFPR, em Abril de 2011, sob a coordenação dos Profs. Gustavo Lugo e Jomi Hübner. O autor agradece aos organizadores o convite para a palestra e a oportunidade para a discussão das idéias aqui apresentadas.

Referências

- [1] J. Alexander, G. Bernhard, M. Richard, and J. Neil. *The Micro-Macro Link*. University of California Press, Berkeley, 1987.
- [2] Estefania Argente, Vicente Julian, and Vicente Botti. Mas modelling based on organizations. In M. Luck and J.J. Gomez-Sanz, editors, *AOSE 2008 (LNCS 5386)*, pages 16–30, Berlin, 2009. Springer.

Parte II

Artigos

Uma Extensão ao Protocolo Contract-Net Aplicado ao Problema de Planejamento de Transporte de Derivados de Petróleo

**Roni F. Banaszewski, Kelvin E. Nogueira,
Jean M. Simão, Lucia V. Arruda, Cesar A. Tacla**

Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba – PR – Brazil

{ronifabio, kelvin.elton}@gmail.com, {lvrarruda, tacla}@utfpr.edu.br

Abstract. In supply chains of petroleum industry, maintaining a proper inventory level for multiple petroleum products is a crucial issue. Basically, this chain has several elements like producer bases, consumer bases and terminals that are linked by means of a multi-modal transport network. These elements must cooperate to achieve the global goal of the system with a minimized cost. To achieve this goal, this paper presents a multiagent protocol for simultaneous negotiations based on the Contract-Net and the application of this in the chain aforementioned. As a result, this protocol proved to be very efficient to return a feasible solution in a low processing time.

1. Introdução

A cadeia de suprimentos da indústria petrolífera brasileira é constituída por refinarias, pontos de consumo, terminais para o armazenamento de derivados de petróleo e meios de transporte que conectam tais elementos. Todos estes elementos devem cooperar para movimentar produtos a um custo mínimo para que refinarias não parem de produzir por falta de espaço para estocagem e consumidores não fiquem sem produtos.

O planejamento de transportes de derivados de petróleo consiste em um problema bastante complexo. Nos últimos anos, problemas similares têm sido abordados por diferentes tipos de abordagens, tal como a otimização matemática. Em Magatão (2004), um modelo baseado em Programação Linear Inteira Mista (MILP) foi utilizado para resolver problemas de escalonamento em dutos, e em Neiro (2004) esta abordagem foi usada em uma parte da cadeia de suprimentos petrolífera. Estes trabalhos ilustram a difícil tarefa de modelar grandes problemas com modelos matemáticos. Geralmente, tais trabalhos consideram apenas parte ou abstraem um problema complexo para que o modelo seja computável. No entanto, quando os modelos são bem construídos, eles são capazes de encontrar soluções ótimas.

De forma semelhante, os sistemas multiagentes também têm sido aplicados aos problemas da cadeia de suprimentos, pois existe uma correspondência natural entre o modelo e a realidade. Em uma comparação simplista, enquanto na abordagem matemática todos os dados do problema são centralizados em um conjunto de fórmulas, em um sistema multiagente, os dados podem ser naturalmente distribuídos entre os agentes. Em alguns problemas (e.g., atribuições de tarefas e problemas de escalonamento), as abordagens multiagentes usando leilões podem encontrar soluções factíveis (sem degradação significativa da qualidade da solução) em tempo finito geralmente mais rápido do que os modelos matemáticos (Brito, 2009).

Leilões multiagentes são comumente aplicados a problemas que envolvem competição por recursos. Neste trabalho, os leilões são utilizados como um mecanismo de cooperação entre os agentes para que estes atinjam seus objetivos (i.e. manter um nível de estoque adequado para cada produto) por meio da movimentação de produtos. Mais precisamente, este trabalho propõe um novo protocolo para a execução de leilões de forma simultânea e apresenta a modelagem para o problema de planejamento de transportes de produtos para um dado horizonte usando tal protocolo. Além disso, este trabalho apresenta um experimento da modelagem em um estudo de caso.

Este trabalho é estruturado nas seguintes seções: a seção 2 descreve as particularidades do problema. A Seção 3 descreve a fundamentação teórica embasada no protocolo Contract-Net. A seção 4 descreve um novo protocolo baseado no Contract-Net e a modelagem usando tal protocolo para o problema em questão. A seção 5 apresenta os resultados do estudo de caso e finalmente, a seção 6 apresenta a conclusão.

2. Visão Conceitual do Problema

O problema de planejamento de transporte de derivados de petróleo na rede multimodal brasileira é altamente complexa devido à quantidade de elementos e as restrições de estocagem e movimentação. É praticamente impossível construir uma solução computável que atenda plenamente todos os detalhes dessa rede. Desta forma, a abstração é fundamental para se alcançar uma solução. No problema abordado neste trabalho, alguns detalhes de planejamento em nível operacional foram omitidos, sendo assim, tal problema refere-se ao planejamento em nível tático.

Basicamente, o problema consiste na alocação da capacidade de rotas para atender a demanda do mercado em um horizonte de planejamento de três meses com custo de transporte minimizado, respeitando as capacidades dos modais de transporte e os limites de estoques em produtores e consumidores. Os produtores não devem produzir mais do que as suas capacidades de estoque locais e os consumidores não podem ficar sem produtos.

Neste trabalho, os produtores e consumidores são genericamente chamados de bases e cada mês de planejamento é considerado como um período de planejamento. Desta forma, cada base apresenta um estoque inicial no início do horizonte e um valor de balanço para cada produto em cada período de planejamento. O valor de balanço consiste na soma do volume de produção e consumo de um dado produto. Estes valores podem ser positivos, negativos ou nulos para cada período, podendo diferir de um período para outro (e.g. um produtor pode produzir com sobras em um período e se encontrar com balanço negativo nos demais).

Para armazenar este volume, cada base apresenta faixas de estoque operacional e físico constituídas de limites máximos e mínimos que devem ser respeitadas. Estas faixas também podem variar de um período para outro. Assim, as movimentações de produtos devem ser executadas com precisão de modo que cada base mantenha diariamente o estoque dentro da faixa operacional.

Estas bases são conectadas por rotas registradas. As rotas são compostas por um ou mais segmentos que representam os meios de transporte. Esses segmentos não são necessariamente do mesmo tipo. Por exemplo, uma rota composta por dois segmentos pode vincular uma refinaria a um terminal através de dutos e um terminal a um ponto de consumo por meio de navios. Cada rota suporta um dado conjunto de produtos e define

um volume mínimo aceitável para transporte. Entre duas bases, pode haver zero ou mais rotas registradas e diferentes rotas podem compartilhar os mesmos segmentos. Consequentemente, as capacidades destes segmentos são também compartilhadas. No planejamento em nível tático não há preocupação com a ordem que os produtos devem ser escalonados nos segmentos. Desta forma, apenas leva-se em consideração as capacidades dos segmentos em cada período de planejamento.

3. Contract-Net

O protocolo de leilão utilizado neste trabalho é uma evolução do protocolo Contract-Net (Smith, 1980). Basicamente, o protocolo Contract-Net propõe dois papéis principais no processo de interação: o papel do leiloeiro e o papel dos participantes de um leilão.

De forma sucinta, um leiloeiro inicia um leilão enviando um anúncio para os participantes com informações sobre os produtos. Nos termos estabelecidos pelo protocolo, este anúncio é chamado de CALL-FOR-PROPOSAL (CFP). Ao receber um CFP, cada participante envia ao leiloeiro uma mensagem de PROPOSAL contendo o seu lance. Se o participante decidir não participar do leilão, ele envia ao leiloeiro uma mensagem de REFUSE. O leiloeiro espera por um prazo fixo ou o tempo necessário para receber todas as propostas dos participantes. Se o prazo expirar, o leiloeiro apenas considera as propostas recebidas até o momento e começa o processamento para definir o vencedor. O leiloeiro envia uma mensagem de ACCEPT-PROPOSAL para o vencedor e REJECT-PROPOSAL para os demais participantes.

O participante vencedor recebe o ACCEPT-PROPOSAL e retorna uma mensagem de INFORM para confirmar a negociação. No caso em que o vencedor decide desistir da negociação, ele envia uma mensagem de FAILURE ao leiloeiro. Finalmente, o leiloeiro encerra o leilão.

Os leilões também podem ocorrer simultaneamente. Basicamente, esta técnica é definida pela ocorrência simultânea de vários leilões ao mesmo tempo. Na literatura, há algumas propostas de versões simultâneas do Contract-Net, a que mais merece destaque é a extensão proposta por Aknine, Pinson et al. (2004). Esta versão altera a forma tradicional de comunicação entre os agentes por meio da substituição das primitivas de comunicação. Nesta, as mensagens PROPOSE, ACCEPT e REJECT são substituídas pelas mensagens: PRE_BID, PRE_ACCEPT, PRE_REJECT, DEFINITIVE_BID, DEFINITIVE_ACCEPT, DEFINITIVE_REJECT. Estas mensagens dividem o protocolo em duas fases: uma fase em que as alocações são provisórias, permitindo desistências e mudanças de alocações, e outra fase em que as alocações são definitivamente persistidas (Aknine, Pinson et al. 2004). No entanto, devido às alocações provisórias e consequentes desistências de acordos, este protocolo pode não apresentar bom desempenho na resolução de problemas complexos, tal como o abordado.

Em trabalhos anteriores (Banaszewski, 2010a, Banaszewski, 2010b), foi demonstrado para uma variação mais simplificada do problema de planejamento abordado, que leilões simultâneos apresentam resultados mais satisfatórios do que os leilões sequenciais (i.e. um leilão por vez, um após o outro).

4. Modelagem do Problema

Esta seção apresenta a modelagem do problema em questão com uso do protocolo proposto baseado no Contract-Net. Devido à complexidade da malha multimodal brasileira, a prática utilizada para viabilizar a busca por uma solução satisfatória em tempo hábil foi o desmembramento e organização da malha em uma hierarquia de níveis. Assim, cada nível pode ser tratado como um problema menor de planejamento e os mesmos métodos de solução, tal como a programação matemática, podem ser aplicados sem alterações em qualquer nível. Desta forma, a resolução de todo o problema ocorre primeiramente resolvendo os níveis mais altos e depois os mais baixos, considerando as movimentações realizadas em níveis mais altos.

Mais precisamente, a hierarquia apresenta três níveis: a) Nível 1: Consiste nas regiões Brasil e Exterior. Ambas são tratados como bases com balanços e limites próprios; b) Nível 2: A rede do Brasil é desmembrada em 16 regiões. Da mesma forma, cada região é representada por uma base; c) Nível 3: Neste nível estão representadas as bases propriamente ditas (i.e. refinarias, terminais e pontos de consumo).

No protocolo proposto, os leilões ocorrem simultaneamente em cada período para todos os produtos. As rodadas simultâneas ocorrem sequencialmente em relação aos períodos, ou seja, um período por vez. Neste protocolo, o papel de leiloeiro pode ser realizado tanto por bases produtoras (com balanço positivo) ou consumidoras (balanço negativo) com o objetivo de escoar produtos para bases com as quais estão conectadas por rotas registradas. O papel de participante também pode ser executado tanto por produtores e consumidores de um dado produto, no entanto, apenas na intenção de adquirir produtos. Quando uma base apresenta oportunidade de aquisição de produtos com vários leiloeiros, esta base é representada por vários participantes, cada um participando de um leilão particular. As bases são modeladas como agentes, dependendo do período de planejamento, uma base pode ser produtora ou consumidora para um mesmo produto ou produtos diferentes.

Há também os agentes que representam os segmentos das rotas e o agente Manager, o qual é responsável por controlar o sistema, realizando tarefas como inicialização/finalização dos agentes e escalonamento dos leilões. O protocolo de leilão proposto é executado em quatro fases: (i) inicialização dos agentes, (ii) ordenação dos leilões, (iii) realização dos leilões, e (iv) finalização dos agentes.

4.1 Inicialização

Na fase de inicialização, o Manager cria e inicializa os agentes com seus respectivos dados de entrada. Na sequencia, cada agente base calcula o seu estoque de cada produto para cada dia no horizonte distribuindo os balanços linearmente pelos dias dos períodos. Neste trabalho, o estoque diário no horizonte é chamado de curva de estoque. A curva de estoque também contabiliza o estoque inicial do respectivo produto. Os limites operacionais e físicos de cada produto também são definidos para cada dia.

Após a inicialização dos dados, cada agente base deve analisar a curva de estoque de cada produto e decidir se é possível escoar algum volume para o período corrente, ou seja, o primeiro período. A base decide escoar um produto apenas quando a curva deste se encontra acima do mínimo operacional em todos os dias do período corrente. Geralmente, as bases com balanço positivo de um produto realizam os leilões, mesmo quando a curva de estoque está dentro da faixa operacional, uma vez que uma

possível aquisição dos produtos deste leiloeiro pode ser mais vantajosa para algum participante em termos de custo de movimentação. Por outro lado, uma base não pode ter sua faixa operacional extrapolada e deve ter preferência no escoamento. No entanto, a própria forma como o valor do lance é calculado leva em consideração este fator.

Outrossim, mesmo que a base tenha balanço negativo do produto para o período, ela pode se candidatar a leiloeiro para escoar a quantia “excedente” do período (i.e. a diferença entre o estoque inicial e o balanço). Esta decisão é importante, pois pode haver consumidores que demandam tal produto e não apresentam oportunidades de adquiri-lo com outras bases. Porém, uma base com balanço negativo de um produto apenas se candidata a leiloeiro quando verifica que tem rota para recebimento do mesmo produto em períodos posteriores.

Quando uma base não deseja se candidatar a leiloeira no período corrente para um dado produto, ela pode definir um período posterior desejável. Desta forma, após definir os períodos para escoamento para cada produto, a base se candidata a leiloeiro para cada um dos produtos com o agente Manager. O Manager aceita apenas a candidatura a um período por vez de uma base para um mesmo produto.

4.2 Ordenação

Na segunda fase, o agente Manager recebe todas as mensagens dos candidatos a leiloeiros e classifica tais mensagens em ordem crescente de período. O Manager seleciona todos os candidatos e produtos referentes a um mesmo período e notifica tais bases para que iniciem os leilões.

Esta fase ocorre repetidamente durante a execução do sistema, uma vez que as rodadas de leilões ocorrem por período. Desta forma, quando o Manager verifica que todos os leilões do período corrente estão finalizados, ele seleciona os candidatos e produtos do período seguinte, se houver. O Manager sempre recebe candidaturas para qualquer período, tal como as recebidas na fase de inicialização e de leiloeiros que escoaram todo o volume ou que não receberam lances para o produto oferecido no período corrente e por isto decidiram se candidatar a um novo período.

4.3 Execução

Nesta fase, múltiplas bases podem realizar leilões simultaneamente para diferentes tipos de produtos. Uma mesma base pode realizar mais de um leilão simultaneamente, um para cada tipo de produto. O protocolo apresentado neste trabalho define o agente Manager como um elemento de decisão durante a execução dos leilões. Assim, quando um leiloeiro define o lance vencedor (apenas é selecionado um lance vencedor por leilão) ele solicita uma permissão ao Manager para enviar o ACCEPT-PROPOSAL. O Manager determina a ordem na qual os leiloeiros enviam as mensagens de ACCEPT-PROPOSAL, sendo que apenas uma é enviada por vez, ou seja, apenas um leilão é finalizado por vez. A Figura 1 apresenta as interações entre os agentes neste protocolo.

4.3.1 Concepção do CFP

Basicamente, um leiloeiro envia um anúncio (CFP) para as bases conectadas a ele. Este CFP contém informações sobre o produto, período corrente, dia em que a curva de estoque passa acima do limite operacional máximo e físico máximo até o último dia do período corrente, quando passar, e o volume diário passível de escoamento.

O volume ofertado no anúncio engloba todos os dias desde o primeiro dia do horizonte de planejamento até o último dia do período corrente. Desta forma, cabe ao participante escolher o dia de saída para o montante desejável do produto levando em conta o tempo de movimentação pela rota escolhida e o espaço disponível para armazenamento do montante. No entanto, como as rodadas de leilões ocorrem por período, o participante apenas pode requisitar uma quantia que dê para receber até o último dia do período corrente, uma vez que a resolução é incremental em relação aos períodos.

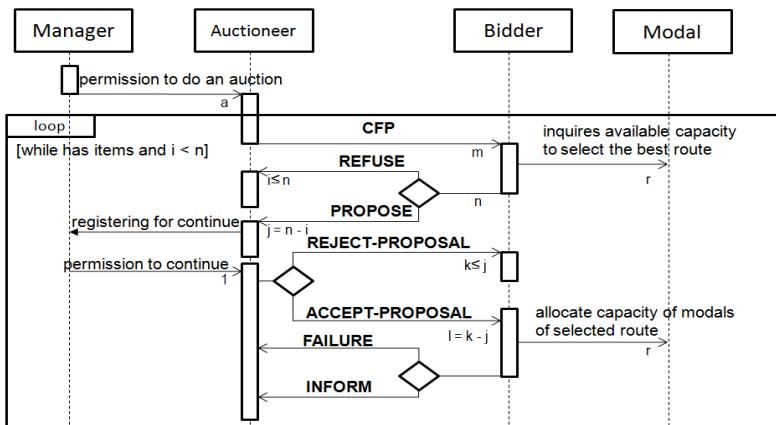


Figura 1. Fase de Execução do Protocolo Proposto

No entanto, esta forma incremental de resolução não impede que movimentações interperiódos venham ocorrer, mesmo havendo prioridade para movimentações intraperíodos. Por exemplo, se o período corrente for o segundo período, o anúncio conterá a quantidade do segundo período e também a quantidade não escoada no primeiro período. Esta redundância é intencional para permitir que participantes que enviaram um REFUSE para o leiloeiro no primeiro período por não terem capacidades disponíveis em rotas, espaço para armazenamento ou mesmo por causa do tempo muito longo de transporte, possam calcular um lance considerando os recursos do período corrente. Assim, pode naturalmente ocorrer de um montante ser escoado no primeiro período e chegar ao destino apenas no segundo período.

Basicamente, um anúncio contém duas listas/vetores de volume diários. O primeiro é calculado para cada dia de escoamento por meio da diferença da curva de estoque para o limite mínimo operacional e de forma semelhante, o segundo é calculado pela diferença da curva de estoque para o limite mínimo físico. No entanto, para calcular o volume disponível para escoamento em um dia, usa-se o montante disponível no final do dia anterior ou estoque inicial quando for o primeiro dia, ou seja, não considera a produção do dia em questão, uma vez que o volume pode estar em produção porque não se pode prever qual o momento do dia que o produto será escoado. Também, não oferece qualquer volume para o dia em que a curva de estoque se encontra acima do limite físico máximo, uma vez que seria uma situação irreal por não haver espaço suficiente para armazenamento em tal dia. Assim, à medida que o volume é escoado em dias precedentes ao dia de extração, a curva de estoque é reduzida para os dias posteriores e assim poderá retardar o dia de extração do limite físico.

O uso de dois vetores é importante para garantir que as bases se mantenham dentro da faixa desejável de estoque. Geralmente, apenas o primeiro vetor é utilizado pelos participantes para o cálculo do lance. Porém, quando um participante tem a curva

de estoque de um produto abaixo do mínimo físico (estoque negativo do produto) em um dado dia (ou em um intervalo de dias) e não consegue calcular a entrega para este dia ou algum dia anterior por falta de produtos no primeiro vetor para o intervalo de dia desejável ou porque o volume é menor do que mínimo exigido pelas rotas, o participante faz uso do segundo vetor. Este vetor oferece o volume entre os limites mínimos para saída em dias mais prévios do leiloeiro.

Neste caso, uma possível movimentação poderia manter ambas as bases ainda na faixa, mesmo que um deles ou os dois fiquem na faixa física. No entanto, como será visto mais adiante, o lance terá seu valor penalizado, o que afetará o escalonamento do leiloeiro para envio do ACCEPT-PROPOSAL, e desta forma a movimentação apenas irá ocorrer se a base participante não conseguir o volume desejado com outro leiloeiro que esteja com um estoque mais alto e por isso com maior prioridade. Mesmo que esta movimentação ocorra e o leiloeiro venha ficar abaixo do mínimo operacional, é mais preferível que o participante esteja na faixa operacional do que o leiloeiro, porque geralmente o leiloeiro apresenta uma curva crescente de produção.

4.3.2 Concepção do Lance

Cada base que apresenta rota para aquisição de produtos recebe pelo menos um CFP. Se uma base está conectada a mais de um leiloeiro, então ele participa de todos os leilões simultaneamente. Para cada CFP recebido, a base participante deve preparar um lance. Este lance pode ser um PROPOSE ou um REFUSE. Um participante pode mandar um REFUSE nas seguintes situações: (a) quando a sua curva de estoque se encontra em um nível (i.e. acima do operacional máximo, na faixa operacional, abaixo do operacional mínimo, abaixo do físico mínimo) acima da curva do leiloeiro; (b) quando o volume oferecido no anúncio é menor do que o volume mínimo de todas as rotas com o leiloeiro; (c) quando o volume não tem como chegar no participante até o final do período corrente; (d) quando não há espaço para armazenamento nos dias até o final do período corrente; (e) quando não há capacidade nos modais nos dias até o final do período corrente; (f) quando identificado um possível loop de negociações. Um loop ocorre quando uma base A atua como leiloeira escoando um volume em um determinado período para uma base B e depois atua como participante de um leilão de B adquirindo outro volume do mesmo produto no mesmo período. Um loop também ocorre entre mais de duas bases.

Caso as situações acima não ocorram, o participante começa a preparação do lance. Primeiramente, ele calcula o volume desejável para aquisição. A definição do volume inicial não envolve um volume absurdamente alto e desnecessário, isto é feito intencionalmente para manter equilíbrio de estoque entre os participantes ao dar oportunidade para que todos obtenham um dado volume para manter seus níveis de estoque. Em seguida, o participante deve determinar a melhor rota para receber o tipo de produto. As rotas já se encontram ordenadas por menor custo. Desta forma, outras rotas somente serão avaliadas quando a rota de menor custo: (a) não tiver capacidade disponível, (b) o volume ofertado for menor do que o seu volume mínimo e (c) quando a chegada prevista do volume no participante ocorrer após o seu dia de extração do mínimo operacional. As rotas são avaliadas uma por vez até encontrar uma que entregue os produtos antes do dia de extração do mínimo operacional. Se esta rota não for encontrada, a rota com menor custo entre as que entregam com atraso será selecionada.

Na avaliação de uma rota, o participante seleciona o dia de saída do leiloeiro conforme os vetores de anúncio enviados no CFP de forma que o dia previsto de chegada ocorra em dias em que tenha espaço para armazenamento e preferencialmente em dias anteriores ao dia de extração do mínimo operacional, quando ocorrer. Em rotas constituídas por dutos, o cálculo do tempo de trânsito de um dado volume considera as suas vazões e volumes, a movimentação paralela pelos dutos e os tempos de espera do volume em pulmões quando as vazões de segmentos adjacentes são diferentes. Em rotas constituídas por navios, o tempo de movimentação já é conhecido e a capacidade é considerada infinita.

Com posse destas informações, o participante calcula o valor do lance pela junção de 3 variáveis: a) Grau de urgência para movimentação (GUM): esta medida tenta priorizar os consumidores com menores oportunidades para receber os produtos. Quanto menor a quantidade de um produto que um consumidor pode adquirir com seus produtores (considerando o volume oferecido e a capacidade das rotas para movimentação) em uma rodada de leilão simultâneo, maior é o GUM para receber este tipo de produto. Esta heurística ajuda a prevenir que segmentos compartilhados sejam ocupados por movimentações de bases que apresentam maiores opções para recebimento. Todos os participantes de leilões que representam o mesmo produto de uma base terão sempre o mesmo GUM; b) Grau de urgência do nível de estoque (GUE): este valor prioriza participantes que estão com a curva de estoque mais baixa. Maior será o GUE quanto menor for o nível de estoque; c) Custo da rota selecionada: Este valor auxilia na definição de qual leiloeiro uma base consumidora movimentará o volume necessário de um produto. Esta heurística prioriza as movimentações de menor custo dentre as movimentações possíveis entre os leiloeiros.

4.3.3 Definição do Lance Vencedor

Ao receber os lances dos participantes, cada leiloeiro normaliza os valores de graus de urgência para movimentação e os custos de transporte (CT) para uma escala linear. As normalizações do GUM (nGUM) e CT (nCT) são calculadas conforme a equação (1) e (2), dividindo-se o GUM ou CT de um lance particular respectivamente pela soma de valores de GUM ou CT de todos os lances recebidos na rodada.

$$nGUM = GUM / \sum_{n=1}^N GUM_n \quad (1)$$

$$nCT = CT / \sum_{n=1}^N CT_n \quad (2)$$

A fim de determinar o vencedor, o leiloeiro classifica os lances de todos os participantes de acordo com a equação (3), onde nGUM, nGUE e nCT são respectivamente ponderados por wGUM, wGUE e wCT. O nGUE é enviado pelo participante para ser usado como um adicional na escolha do lance vencedor. O participante que obtiver o maior valor de lance é o vencedor.

$$\text{lance} = nGUM * wGUM + (1 - nCT) * wCT + nGUE * wGUE \quad (3)$$

4.3.4 Escalonamento no Manager

No entanto, após determinar o participante vencedor, os leiloeiros não enviam imediatamente a mensagem de ACCEPT-PROPOSAL para o vencedor. Primeiramente, os leiloeiros pedem autorização ao Manager para o envio do ACCEPT-PROPOSAL. O Manager libera uma autorização por vez para os leiloeiros, levando em conta primeiramente o GUM e em caso de empate o custo de transporte de seus lances

vencedores, ambos sem normalização. Esta sequencia de mensagens de ACCEPT-PROPOSAL é importante para priorizar os transportes de produtos para bases com maior grau de urgência e para evitar inconsistências na alocação final de capacidade em segmentos de transporte compartilhados.

Ainda, como as alocações ocorrem na mesma rodada, o GUM dos participantes pode mudar devido à ocupação de segmentos compartilhados (apenas quando o segmento for um duto). Desta forma, cada segmento que recebe uma alocação persistente notifica todas as bases participantes que apresentam rotas que o contem. Após o recebimento das notificações, as bases atualizam seus GUM e notificam os seus respectivos leiloeiros com uma mensagem de atualização de lance. Após a atualização do lance vencedor, o leiloeiro atualiza a sua prioridade com o Manager para enviar o ACCEPT-PROPOSAL. Esta prática é eficiente, uma vez que as atualizações são enviadas apenas quando os segmentos compartilhados são alocados persistentemente.

Portanto, existem dois pontos importantes nesta fase de execução: (a) o lance para determinar o vencedor de cada leilão e (b) o sequenciamento dos leiloeiros para o envio do ACCEPT-PROPOSAL. A ordem do envio de mensagens de ACCEPT-PROPOSAL é bastante importante, uma vez que os participantes geralmente pedem apenas a quantidade de produtos necessária para atingir um certo nível de estoque. Se um leiloeiro tem o envio do ACCEPT-PROPOSAL adiado, pode ocorrer do participante vencedor não querer mais o volume para o qual deu o lance e desta forma, desistir do leilão porque a base a qual representa pode ter adquirido o produto com outro leiloeiro.

4.3.5 Fechamento de um leilão

Quando um participante recebe um ACCEPT-PROPOSAL, ele verifica se ainda deseja o volume para o qual enviou o lance e se a rota selecionada ainda apresenta a capacidade para movimentação. Se as verificações forem realizadas com sucesso, ele envia uma mensagem de INFORM ao leiloeiro, caso contrário, envia uma mensagem de FAILURE. Antes de enviar uma mensagem de INFORM, este deve persistir a alocação dos produtos com os segmentos que compõem a rota selecionada e atualizar a sua curva de estoque para os dias de recebimento do volume.

No entanto, quando um participante atualiza a curva de estoque do produto, ele verifica se tem quantidade suficiente para realizar um leilão, pois pode haver outras bases que podem estar precisando do produto e não tiveram oportunidade de adquirir com nenhuma outra base. Ao realizar o leilão, a base participante/leiloeiro recebe os lances e registra para envio do ACCEPT-PROPOSAL no Manager com uma prioridade baixa, pois pode ocorrer das bases participantes manter rotas com outras bases produtoras e desta forma, estas teriam preferência de escoamento.

Quando o leiloeiro recebe uma mensagem de INFORM ele também atualiza a sua curva de estoque e verifica se ainda apresenta um volume suficiente para ser escoado no período corrente. Se esta verificação for confirmada, ele realiza outro leilão. Este leilão não ocorre simultaneamente com qualquer outro. No entanto, este leilão segue o mesmo processo dos leilões simultâneos, incluindo a intervenção do Manager para enviar o ACCEPT-PROPOSAL. Desta forma, o leiloeiro tem a oportunidade de disputar a preferência dos consumidores com outros leiloeiros para tentar escoar o volume remanescente no período.

4.4 Finalização dos Leilões

Quando um leiloeiro recebe apenas mensagens de REFUSE, ele considera que os participantes estão satisfeitos para o tipo de produto e que não há mais possibilidade de movimentação para eles para o período corrente. Quando isso ocorre, o leiloeiro seleciona um novo período para se candidatar a leiloeiro e informa ao Manager. No entanto, pode ocorrer de um leiloeiro não desejar mais se candidatar a nenhum período.

Desta forma, quando o Manager não tiver mais pedidos pendentes para envio de ACCEPT-PROPOSAL e não houver mais leiloeiros candidatos para fazer leilão para qualquer produto em qualquer período, a fase de finalização é executada. Nesta fase, o Manager envia mensagens a todos os agentes para que estes finalizem suas execuções.

5. Estudo de Caso

As heurísticas apresentadas foram experimentadas em um cenário com dados aproximativos dos reais referentes à malha brasileira devido à dificuldade para a aquisição destes dados. Como exemplo, as bases no cenário de estudo apenas apresentam um limite máximo e mínimo, sendo o mínimo sempre igual à zero. Assim, considera-se que os limites operacionais e físicos apresentam os mesmos valores. Neste cenário de estudo, aborda-se o planejamento de movimentação de apenas um único produto, o GLP, para um horizonte de três períodos (de um mês cada) no nível 2 de planejamento. Neste nível, há 16 bases, cada uma representando uma região brasileira.

Estas bases são conectadas por meio de 72 rotas. Cada rota é formada por apenas um segmento não compartilhado. Estas rotas são apresentadas na Tabela I. Em cenários do nível 3, as rotas são compostas por mais de um segmento, podendo estes segmentos serem compartilhados por diferentes rotas e representarem diferentes tipos de modais de transporte. Porém, ainda não há dados suficientes para gerar experimentos neste nível.

Tabela I - Rotas Registradas

		RECEBIMENTO															
		BA	REG	PR	RJ	SP	RS	REM	PA	MA	LUB	RPC	PB	PE	AL	SE	ES
ESCOAMENTO	BA	---	X	X	X	X	X	X	X	X	X			X			
	REG		---														
	PR	X		---	X		X	X						X			
	RJ	X		X	---	X	X		X	X	X		X			X	
	SP	X	X		X	---								X			
	RS	X		X	X		---	X		X	X			X			
	REM	X		X			X	---	X	X	X			X			
	PA	X			X			X	---					X			
	MA	X			X		X	X		---				X			
	LUB	X			X		X	X			---	X		X			
	RPC										X	---					
	PB												---				
	PE	X		X	X	X	X	X	X	X			---		X		
	AL													---	X		
	SE												X	X	---		
	ES				X											---	

No experimento sobre o cenário corrente, todas as bases terminaram os períodos com a curva de estoque dentro da faixa limitante. Os resultados são apresentados na Tabela II. Nesta tabela, para cada base e período são apresentados os limites máximos e os pontos da curva de estoque inicial e final no último dia de cada período.

Como observado na Tabela II, entre as regiões do presente cenário, certamente a região de São Paulo (SP) apresenta a maior demanda por período. Por isto, SP será analisada mais detalhadamente. Conforme a Tabela I, a região de SP apenas tem rotas com as regiões da Bahia (BA), Pernambuco (PE) e Rio de Janeiro (RJ). Com a execução do sistema, o volume recebido por SP por período está apresentado na Tabela III.

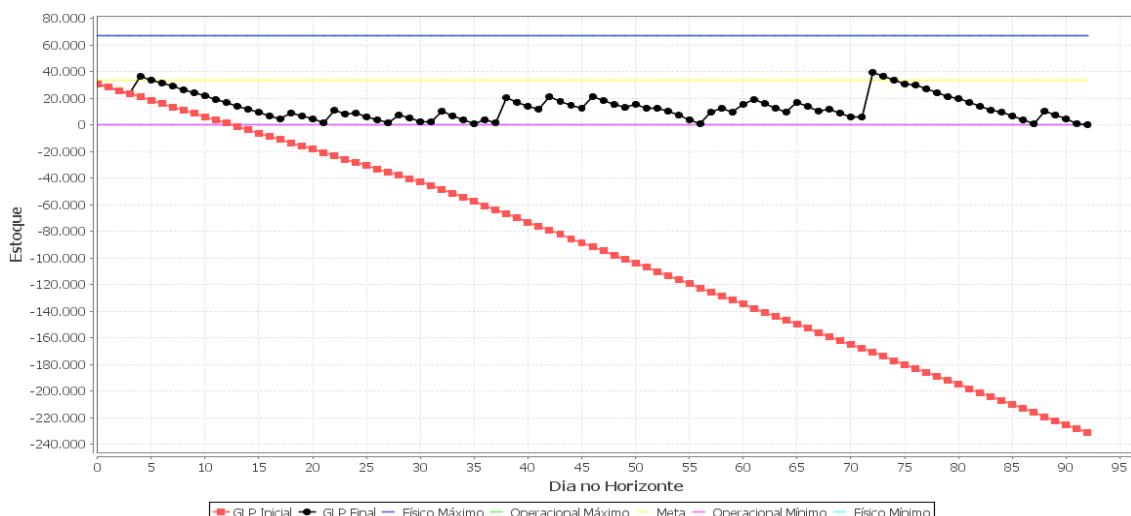
Tabela II – Dados Iniciais e Finais para todas as Bases do Cenário de Estudo

	Per.	Lim. Max	V. Inicial	V. Final		Per.	Lim. Max	V. Inicial	V. Final
BA	1º	13948	36680	0	MA	1º	765	-1300	31
	2º	10500	97681	0		2º	765	-1300	0
	3º	5389	112181	1401		3º	5145	-11590	1781
REGAP	1º	8755	6380	6380	LUBNOR	1º	9000	-17320	4437
	2º	8755	4780	4780		2º	7800	-32920	4310
	3º	8755	4780	4780		3º	10650	-54220	6814
PR	1º	9656	3460	0	RPCC	1º	0	0	0
	2º	9656	24460	7986		2º	0	0	0
	3º	9656	42360	9140		3º	600	1200	266
RJ	1º	16200	13330	0	PB	1º	0	0	0
	2º	16200	20030	0		2º	0	0	0
	3º	16200	26130	13300		3º	0	0	0
SP	1º	66878	-45400	2127	PE	1º	40970	58460	0
	2º	66878	-137900	18786		2º	40970	102360	0
	3º	66878	-231300	162		3º	40970	139660	3609
RS	1º	9180	-14760	8389	AL	1º	3500	5000	366
	2º	9180	-19160	5597		2º	3400	200	0
	3º	9180	-30760	4501		3º	3300	4480	2276
REMAN	1º	16750	38940	0	SE	1º	2300	-4600	1186
	2º	15250	69440	0		2º	2350	-9300	0
	3º	18900	107240	8533		3º	2670	-14640	800
PA	1º	4000	-6100	1389	ES	1º	550	-100	500
	2º	4000	-14100	1958		2º	530	-160	440
	3º	8750	-31600	5026		3º	530	-160	440

Tabela III – Movimentações para a Região de São Paulo

	Período 1	Período 2	Período 3
Rota BA→SP	42013	36959	22534
Rota PE→SP	8597	65314	47747
Rota RJ→SP	12957	6891	7104

Pela análise da tabela, pode-se perceber que um maior volume foi movimentado de regiões mais distantes e consequentemente de maior custo. Isto se deve pelas diferenças entre os volumes de produção das respectivas bases produtoras, sendo que o RJ apresentou uma baixa produção em comparação com as outras duas. Mesmo assim, as três bases terminaram o 1º e 2º período com um estoque zero para atender toda a demanda de SP. Pela análise das movimentações, percebe-se que BA escoou mais do que produziu no 1º período, isto ocorreu certamente por causa do recebimento de produtos de outra base. Também, vale ressaltar a ocorrência de movimentações entre períodos na rota BA→SP. Mais precisamente, o montante de 10269 saiu no dia 28 do 1º período da BA e chegou a SP no dia 1 do 2º período.

**Figura 2. Curva de Estoque Inicial e Final da Base que representa a Região de São Paulo**

Por meio destas movimentações SP manteve a sua curva de estoque dentro da faixa em todos os dias do horizonte de planejamento, conforme a Figura 2. Nesta Figura, o conjunto de pontos vermelhos representa a curva de estoque inicial e o conjunto de pontos pretos representa a curva de estoque final. As linhas azul e rosa representam respectivamente os limites máximos e mínimos.

Em uma análise final, apesar de toda a complexidade da malha, o tempo de execução do sistema pode ser considerado bastante satisfatório para o seu uso esperado. Em um computador Intel Core 2 Duo, 2GHz, 3GB de RAM, o tempo de execução foi de aproximadamente 9 segundos. Um tempo baixo de execução é um requisito do sistema.

6. Conclusão

Este artigo apresentou a modelagem para o problema de planejamento de movimentação de derivados de petróleo sobre as diretrivas do protocolo proposto. Este protocolo vem apresentando resultados satisfatórios em se tratando da complexidade da malha multimodal brasileira e principalmente em tempo hábil.

No entanto, devido à esta complexidade, o protocolo demanda evoluções para melhor adaptação às particularidades do problema. Exemplos de tais evoluções podem se referir à atualização e reenvio do anúncio quando o seu conteúdo é alterado por alguma movimentação e a possibilidade de desistências de acordos já concretizados, uma vez que devido à imprevisibilidade futura, um agente pode encontrar uma oportunidade mais satisfatória em dias ou períodos posteriores ou então precisar realmente desfazer uma movimentação para solucionar conflitos na alocação de modais ou de espaço para armazenamento. Com estas alterações, a solução apresentará maior iteratividade entre os agentes e contribuirá para o refinamento da qualidade da solução.

7. Referencias

- Aknine, S., S. Pinson, et al. (2004). An Extended Multi-Agent Negotiation Protocol. *Autonomous Agents and Multi-Agent Systems* 8(1): 5-45.
- Banaszewski, R. F., Tacla C. A., Pereira F. R., Arruda L. V. R. e Enembreck, F. (2010), Planning Transport of Crude Oil Derivatives with Simultaneous Auctions. In: 2010 IEEE Conference on Systems, Man and Cybernetics, Istambul, Turquia. SMC2010,
- Banaszewski, R. F., Pereira F. R., Arruda L. V. R., Simão, J. M., Tacla C. A. (2010), Simultaneous Auctions in Transport Planning of Multiple Derivatives of Oil in Multi-Modals Networks. In: 42º Simpósio Brasileiro de Pesquisa Operacional (SBPO).
- Brito, R. C. and Tacla, C. A. (2009), Um Sistema Multiagente Auto-Interessado para Auxiliar nas Decisões Logísticas de Alocação de Petróleo em Portos, III Workshop Escola de Sistemas de Agentes para Ambientes Colaborativos, Caxias do Sul, Brazil.
- Magatão, L., Arruda, L.V.R. and Neves-Jr, F. (2004), A mixed integer programming approach for scheduling commodities in a pipeline. *Computers and Chemical Engineering*, v. 28, pp. 171–185.
- Neiro, S. and. Pinto, J.M. (2004), A general modeling framework for the operational planning of petroleum suply chain. *Computers and Chemical Engineering*, 28, 871-896.
- Smith , R. G., The contract net protocol: High level communication and control in a distributed problem solver, *IEEE Transactions on Computers* C-29 (1980) 1104–1113.

Um Modelo Multiagente para Carregamento Compartilhado com Prioridades em um Terminal de Contêineres: Combinando o Problema da Mochila com PDMs

**Leonardo M. Rodrigues, Graçaliz P. Dimuro, Antônio C. R. Costa,
Leonardo R. Emmendorfer**

¹Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Avenida Itália, Km 8 – Campus Carreiros – 96.201-900 – Rio Grande – RS – Brazil

Abstract. *This paper proposes the modeling and implementation of a multiagent system for the shipment operation with priorities in the port terminal of loading/unloading of containers, located in Rio Grande City. We faced a problem of optimal policy formulation for teams of resource-limited agents in stochastic environments, which is composed of two strongly-coupled subproblems: a resource allocation problem and a policy optimization problem. However, the containers are required to be shared between the load agents (or loaders), considering the following constraints: the containers have different weight and load priorities and the load agents have a maximum load that they can support during a journey. We show how to combine the two problems, by formulating a policy optimization problem over the load operations for each loader, after the allocation of the shared containers considering the above constraints. To solve the problem of load distribution between the load agents we use a reduction of Knapsack Problem. Then, we use Markov Decision Processes (MDP) to decide on the actions (e.g., load and unload) of loaders in the environment.*

Resumo. *Este trabalho propõe a modelagem e implementação de um sistema multiagente para a operação de carregamento com prioridades no terminal de carga/descarga de contêineres do Porto da cidade do Rio Grande. Trata-se um problema de formulação de política ótima para times de agentes com limitação para recursos em ambientes estocásticos, que é composto por dois subproblemas fortemente acoplados: um problema de alocação de recursos e um problema de otimização de política. Entretanto, os contêineres devem ser distribuídos entre os agentes carregadores segundo as seguintes restrições principais: os contêineres têm valores de peso e prioridades quanto ao carregamento e os carregadores têm um limite de carga que podem suportar durante uma jornada de trabalho. Mostra-se como combinar os dois problemas, pela formulação de um problema de otimização de política sobre as operações de carga de cada agente carregador, depois de realizada a alocação dos contêineres compartilhados, segundo as restrições acima mencionadas. Para tanto, para resolver o problema da distribuição de carga entre os carregadores foi utilizada uma redução do Problema da Mochila (Knapsack Problem). Em seguida, o sistema utiliza os Processos de Decisão de Markov (PDM) para decidir sobre as ações de movimentação (tais como: carga e descarga) de cada carregador dentro do ambiente.*

1. Introdução

Diversos trabalhos têm discutido o problema de alocação eficiente de recursos em diferentes áreas (p.ex., [Huang et al. 2011, Yang et al. 2011]), alguns no contexto de sistemas multiagentes (p.ex., [Guo and Conitzer 2010, Dolgov and Durfee 2004, Meuleau et al. 1998, Parr 1998]). Outras áreas que também utilizam esses conceitos são: Economia, Sistemas Operacionais, Engenharias e etc.

Um particular problema relacionado é o da alocação de recursos compartilhados para grupos ou times de agentes autônomos, com recursos limitados, que operam em ambientes estocásticos [Dolgov and Durfee 2004, Meuleau et al. 1998].

Vários aspectos deste problema têm sido abordados, alguns direcionados ao problema combinado de decidir como os recursos compartilhados limitados devem ser distribuídos aos agentes e que políticas eles devem adotar, tal que o bem estar social do grupo seja maximizado [Dolgov and Durfee 2004, Meuleau et al. 1998, Parr 1998].

Neste tipo de problema, descobrir o valor de uma alocação de recursos específica exige a solução de um problema de otimização de política estocástico, e, portanto, a alocação de recursos e problemas de otimização de políticas são fortemente acoplados [Dolgov and Durfee 2004]. Ou seja, é inviável tentar melhorar a alocação de recursos sem alterar a otimização do problema em geral e vice-versa.

Uma proposta para resolver este problema é formular o processo de alocação de recursos e a operação real dos agentes como um grande PDM multiagente [Boutilier 1999] sobre o espaços de estados e ações em conjunto. No entanto, este método cresce exponencialmente com relação ao espaço de estados, e, portanto, torna-se inviável para grupos de agentes de tamanho razoável [Dolgov and Durfee 2004].

Outras formas de abordar o problema são baseadas na decomposição do problema [Dolgov and Durfee 2004, Boutilier et al. 1997, Dean et al. 1995, Meuleau et al. 1998, Singh and Cohn 1998, Pereira et al. 2008, Dimuro et al. 2008], onde um PDM global é decomposto em diversos sub-PDMs independentes ou sub-PDMs fracamente acoplados. Estes sub-PDMs são usualmente resolvidos independentemente e as políticas resultantes são então combinadas para obter uma solução (talvez subótima) para o PDM global original. Em [Dolgov and Durfee 2004], essas técnicas de decomposição/composição não são utilizadas, sendo apresentado um método que formula somente um problema de otimização de política, com restrições que garantem que as políticas não consomem em excesso os recursos compartilhados, reduzindo-o a um problema de programação linear inteira mista (MILP).

Neste trabalho, foca-se no problema de distribuição de carga entre agentes carregadores, juntamente com o problema do carregamento de navios, no terminal de carga/descarga de contêineres do Porto da cidade de Rio Grande, RS. No mundo real, existem inúmeros contêineres espalhados no terminal de forma que a posição de cada um é conhecida. Existem também alguns carregadores (neste caso, os próprios agentes), geralmente dois ou três, que servem para transportar os contêineres até o navio cargueiro. Sendo assim, quando um navio atraca no cais, a jornada de trabalho dos agentes carregadores deve ser iniciada, de forma que todos os contêineres a eles atribuídos sejam colocados no navio. A forma como os contêineres são colocados no navio é abstraída neste trabalho, importando somente como os contêineres são distribuídos entre os agentes

carregadores, bem como o método que indica quando os carregadores devem parar de transportar os contêineres até o navio.

Este problema pode ser entendido como um problema de alocação de recursos compartilhados (os contêineres) por um grupo de agentes carregadores, que fazem o carregamento dos navios. Cada contêiner deve ser associado a apenas um agente carregador, de forma que, quando essa associação seja realizada, o contêiner selecionado não esteja disponível para os outros carregadores do ambiente. Assim, cada agente carregador fica responsável pelo transporte de um conjunto fixo de contêineres e não existe a possibilidade de ocorrerem condições de disputa com relação a um determinado contêiner.

Existem duas restrições principais: os contêineres têm valores de peso e prioridades quanto ao carregamento e os carregadores têm um limite de carga que podem suportar durante uma jornada de trabalho (veja Seção 4).

[Dolgov and Durfee 2004] salienta que, neste domínio, os agentes carregadores operam quase que independentemente, mas seus problemas de otimização de políticas são acoplados via os contêineres que eles compartilham, os quais ficam depositados no pátio do terminal. Observe que, uma vez que um contêiner é atribuído a um carregador ele já não fica disponível para os outros carregadores. Esse fraco acoplamento torna esse problema bem adequado para os métodos de solução mencionados nos parágrafos anteriores. Entretanto, alguns desses métodos de decomposição/composição não evitam a completa enumeração do espaço conjunto de estados e ações (p.ex., [Singh and Cohn 1998]), outros somente fornecem soluções aproximadas ou não garantem a solução (p.ex., [Meuleau et al. 1998]).

Neste trabalho, inspirado pela abordagem proposta em [Dolgov and Durfee 2004], propõe-se uma solução combinada para o problema do terminal de contêineres. Para resolver o problema da distribuição de carga entre os carregadores utiliza-se uma redução ao Problema da Mochila (Knapsack Problem) [Sniedovich 2011]. Em seguida, o sistema utiliza um PDM [Russel and Norvig 2003] para decidir sobre as movimentações (p.ex., carga, descarga) de cada carregador dentro do terminal. Para a programação do algoritmo de solução do Problema da Mochila e da execução do método de Iteração de Valor foi utilizada a Linguagem de Programação C. Já para a criação do ambiente de um terminal portuário, foi utilizado o programa de simulação multiagente NetLogo. Neste último, serão realizadas simulações para avaliar se os agentes estão se comportando como desejado, bem como se eles estão carregando os contêineres corretos de acordo com as respectivas prioridades e se estão terminando seu processo no momento certo.

Este artigo está organizado como descrito a seguir. A Seção 2 apresenta conceitos básicos sobre PDMs e PDMs fracamente acoplados. A Seção 3 discute brevemente o Problema da Mochila. Na Seção 4, são apresentados os detalhes do problema das operações de carga/descarga do terminal de contêineres, e, na Seção 5, o modelo proposto, combinando o problema da mochila com PDMs. Na Seção 6, são apresentados os resultados obtidos com as simulações no software NetLogo, bem como alguns detalhes de implementação. A Seção 7 é a Conclusão.

2. Processos de Decisão de Markov Fracamente Acoplados

A especificação de um problema de decisão sequencial para um ambiente completamente observável com um modelo de transição de Markov e recompensas aditivas é chamada de Processo de Decisão de Markov (PDM) [Russel and Norvig 2003]. Por ser matematicamente bem estabelecido e fundamentado, este formalismo facilita o estudo do apren-

dizado por reforço. Por outro lado, assume uma condição simplificadora, conhecida como Condição de Markov, que reduz a abrangência das soluções, mas que é compensada em grande parte pela facilidade de análise [Ribeiro 2002].

A *Condição de Markov* impõe que o estado do sistema não depende do seu histórico. Com isso, temos que o estado em que o sistema se encontrará no próximo instante ($t + 1$) é uma função dependente apenas da situação em que se encontra o sistema no estado atual e da ação que o agente toma neste estado, no tempo t .

Um PDM pode ser descrito como um processo estocástico e definido formalmente pela quádrupla $\langle S, A, T, R \rangle$, onde:

- S : é um conjunto finito de estados do ambiente.
- A : é um conjunto finito de ações que o agente pode realizar.
- $T : S \times A \times S \mapsto [0, 1]$ é uma função de transição que dá a probabilidade de o sistema passar para um estado $s' \in S$, dado que o processo estava em um estado $s \in S$ e o agente decidiu executar uma ação $a \in A$. Representa-se uma transição como: $T(s' | s, a)$.
- $R : S \times A \mapsto R$ é uma função de recompensa que gera um custo (positivo ou negativo) por tomar uma decisão $a \in A$ quando o processo está em um estado $s \in S$.

É possível definir um conjunto de ações (A_s) possíveis em cada estado do sistema ($s \in S$). Por exemplo, se um determinado sistema conta com um conjunto de três estados e três ações possíveis, é completamente aceitável que em um dos estados seja possível executar apenas duas das três ações disponíveis. Portanto, se ao longo do tempo conseguirmos computar uma política $\pi : S \times A$ que maximiza uma função (normalmente a recompensa recebida), teremos a resolução de um PDM. Esse mapeamento de estados e ações é denominado de política ótima.

Utilizando o conhecido algoritmo de Iteração de Valor [Russel and Norvig 2003], pode-se encontrar uma política ótima. A idéia básica é calcular a utilidade (U) de cada estado e depois empregar as utilidades de estados para selecionar uma ação ($a \in A$) ótima em cada estado ($s \in S$) [Russel and Norvig 2003]. Para o cálculo, utiliza-se a Equação de Bellman, que parte do pressuposto de que: “a utilidade $U(s)$ de um estado s é a recompensa ($R(s)$) imediata correspondente a esse estado, somada à utilidade descontada esperada do próximo estado, supondo-se que o agente escolha a ação ótima”:

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s') U(s'). \quad (1)$$

sendo $0 < \gamma \leq 1$ o fator de desconto.

Essa equação gera um sistema de equações, uma para cada estado. Porém, estas equações são *não-lineares* por causa da existência do operador “max”. Devido a esse fato, os cálculos não podem ser realizados utilizando técnicas de álgebra linear, mas sim, uma abordagem *iterativa*, denominada de atualização de Bellman, onde as utilidades dos estados são atualizadas de acordo com as utilidades dos seus vizinhos até que haja convergência para um ponto fixo:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s, a, s') U_i(s'). \quad (2)$$

sendo i a iteração atual da atualização de Bellman.

O problema de um PDM é que ele se torna computacionalmente inviável se o ambiente for parcialmente observável, ou a quantidade de estados, ações e recursos a

serem compartilhados forem muito grandes [Kaebling et al. 1998]. Entretanto, alguns problemas podem ser simplificados, dada as características especiais do ambiente ou a forma como são distribuídas as tarefas/recursos.

Para o problema de alocação de recursos, tal como o problema explorado neste trabalho, foi proposto em [Dolgov and Durfee 2004] um modelo denominado Processo de Decisão de Markov Fracamente Acoplado (PDMFA). Modelos desse tipo, adaptados para aplicações específicas, também foram utilizados em [Meuleau et al. 1998, Parr 1998].

Considerando, como discutido na Seção 1, que para este tipo de problema existem dois subproblemas *fortemente acoplados*, que são: o problema da alocação de recursos e o problema da otimização da política, pode-se decompor a estrutura reconhecendo que existem limitações de recursos. Assim, se um dos agentes toma um dos recursos disponíveis, os outros agentes não terão acesso a esse recurso, pois o mesmo já está alocado. Com isso, ações selecionadas por um PDM restringem as ações disponíveis para outros. Portanto, os outros agentes terão acesso a tal recurso somente quando o agente que o possui devolver o mesmo à base de recursos. Essa idéia define bem o termo *fracamente acoplado*, pois os recursos podem ser tratados separadamente uns dos outros. Os agentes poderão apenas, então, compartilhar os recursos que estão disponíveis no momento.

3. O Problema da Mochila

O problema da mochila (*Knapsack Problem*) [Sniedovich 2011] é um problema NP-completo, sendo um problema de optimização combinatória, onde a idéia é preencher uma mochila com um determinado limite de peso de forma que caiba a maior quantidade possível de objetos de acordo com seu valor.

Assim, considere n items $i = 1, 2, \dots, n$, cada um com um peso w_i (peso do contêiner) e um valor v_i (prioridade de embarque do contêiner). Então, dada uma mochila que suporta um peso W como restrição, quais ítems devem ser inseridos nessa mochila para maximizar o seu valor total? A definição formal deste problema é dada como:

$$\begin{aligned} \text{Maximize: } & \sum_{i=1}^n v_i x_i \\ \text{Sujeito à restrição: } & \sum_{i=1}^n w_i x_i \leq W, x_i \in \{0, 1\}. \end{aligned}$$

Neste projeto, optou-se pela utilização do conceito do Problema da Mochila *Booleana*. Neste caso, existe a variável x_i que recebe os valores 1 (um) ou 0 (zero), indicando, respectivamente, se o objeto i foi ou não inserido na mochila em questão. Para solucionar esse problema, pode ser utilizado Programação Dinâmica (PD). Este método trata o problema da mochila resolvendo uma equação funcional relacionando soluções ótimas com sub-problemas do problema original [Sniedovich 2011].

4. O Problema das Operações de Carga/Descarga do Terminal de Contêineres

No terminal portuário de carga/descarga de contêineres, existe um pátio onde são colocados os contêineres de maneira que a posição de cada um é conhecida. A cada contêiner é associado o seu respectivo peso (w) e o seu valor de prioridade de embarque (v). Esse valor de prioridade dependerá do tipo de carga existente no contêiner.

Suponha que existem L carregadores no ambiente, de forma que, quando um navio atraca no porto, esses agentes devem transportar os contêineres para o navio cargueiro. Cada carregador pode lidar com um peso máximo (W) por jornada de trabalho. Neste caso, uma jornada de trabalho significa o carregamento de todos os contêineres associados a um carregador. Ou seja, se um agente carregador deve transportar cinco contêineres para o navio, sua jornada de trabalho terminará somente quando todos os contêineres sob sua responsabilidade estiverem em algum navio.

Para tornar mais eficiente o transporte dos contêineres para o navio, os carregadores devem dividir o trabalho de forma que cada um transporte o máximo de contêineres possível, respeitando as prioridades dos mesmos. Nesse sentido, os carregadores podem transportar qualquer um dos contêineres espalhados pelo terminal. É importante ressaltar, também, que não existe uma área determinada para o trabalho de cada carregador. Ou seja, os carregadores podem transportar qualquer contêiner dentro da área do terminal¹, sendo um por vez. Em teoria, isso diminuirá o tempo de espera do navio no porto, aumentando a eficiência dos carregamentos.

O navio cargueiro conta com uma restrição de peso. Ou seja, existe um peso máximo (W_{max}) que ele pode transportar. Existem duas situações em que o navio pode deixar o cais: na primeira, as jornadas de trabalho dos carregadores foram finalizadas; na segunda, o peso máximo suportado pelo navio foi atingido. Ao término da jornada de trabalho, cada agente carregador deve voltar para a garagem para realizar inspeções/manutenções técnicas rotineiras. Isso indica o fim do processo.

5. O Modelo Proposto: combinando o problema da mochila com PDMs

O modelo proposto para a resolução deste problema utiliza um algoritmo de solução para o Problema da Mochila para alocação dos contêineres compartilhados aos carregadores, considerando a limitação de carga máxima de cada carregador e as prioridades de carregamento dos contêineres. Um PDM é então utilizado para as movimentações dos carregadores no pátio do terminal. As fases do processo podem ser observadas na Fig. 1.

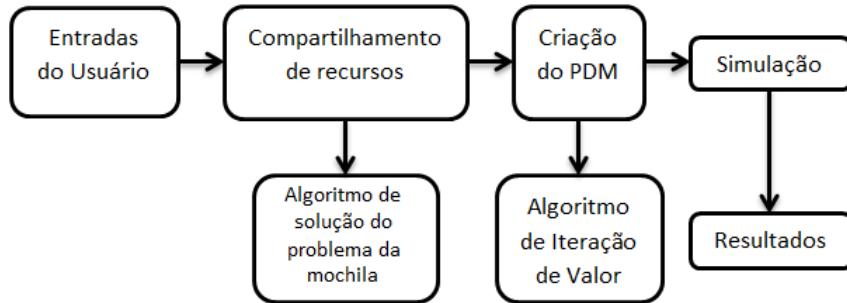


Figura 1. Fases de solução do problema proposto.

Na *primeira fase*, o usuário (operador do sistema) indica o número de carregadores disponíveis e o peso total que cada carregador pode transportar em uma jornada de trabalho (com a chegada de um navio no cais, por exemplo).

Na *segunda fase* do processo, denominada “Compartilhamento de recursos”, é executado um algoritmo de solução do Problema da Mochila. Com isso, a distribuição

¹Os agentes carregadores trafegam somente pelas ruas determinadas no ambiente. Essas ruas tem posicionamento conhecido pelos agentes.

dos contêineres entre os carregadores é realizada conforme as prioridades de embarque e pesos atribuídos aos contêineres (conforme discutido na Seção 3), considerando a carga máxima que cada carregador pode transportar antes da parada para inspeção. A cada carregador é então atribuído um conjunto de contêineres, que serão transportados para o navio um de cada vez. Nesse trabalho, só é possível distribuir os contêineres entre os agentes quando todos os agentes carregadores estão disponíveis. A seguir, a Fig. 2 mostra um pseudo-algoritmo da função utilizada, chamada “mochila”.

função mochila(w, v, W, n, x) **retorna** melhor combinação de acordo com a tabela t .

Entradas:

w (Um vetor contendo o peso de cada item)

v (Um vetor contendo a prioridade de cada item)

W (O peso total suportado pela mochila)

n (Quantidade de itens disponíveis para seleção)

x (Vetor booleano para indicar quais itens foram inclusos na mochila)

Variáveis locais:

t (Uma matriz de $n+1$ linhas e $W+1$ colunas)

peso (Peso utilizado na mochila em questão)

```
t = criarTabelaT (w, W, n);
verificaObjetosUtilizados (n, W, t, x, w, v);
verificaPesoTotal (peso, W);
retornar MelhorCombinação (t, x, w, v);
```

fim

Figura 2. Pseudo-algoritmo da função “mochila”, utilizada para distribuir os contêineres entre os carregadores.

Na *terceira fase*, são criadas as definições do PDM, que dependem do número de contêineres atribuídos aos carregadores. São assim descritos o conjunto de estados, o conjunto de ações, as recompensas e as transições dos PDMs para cada carregador.

Na Fig. 3 é possível observar os conjuntos de estados S e de ações A , criados para a solução de um problema genérico, onde considera-se que o carregador deve transportar n contêineres. Neste caso, o conjunto de estados do PDM é dado por:

$$S = \{0_{n-i} \mid 0 \leq i \leq n\} \cup \{1_{n-i} \mid 0 \leq i \leq n\} \cup \{2\}, \quad (3)$$

onde 0_n representa “agente sem contêiner”, 1_n representa “agente com contêiner” e 2 representa “Agente parado (na Garagem)”, sendo este um estado terminal, e i é um contador de contêineres descarregados. O conjunto de ações é dado por $A = \{A_0, A_1, A_2\}$, onde A_0 significa “carregar contêiner”, A_1 significa “descarregar contêiner” (incrementando o contador de contêineres i) e A_2 significa “locomover (ir para a garagem)”.

Em cada estado, os agentes podem executar qualquer uma das ações disponíveis, porém receberão recompensa negativa caso executem uma ação não prevista na política determinada pelo algoritmo de Iteração de Valor. As ações mostradas na Fig. 3 indicam a política ótima (π^*) para este PDM. Assim, segundo esta política, o carregador deverá carregar/descarregar os n contêineres, percorrendo os estados do tipo 1/0, e somente irá para a garagem quando já tiver levado todos os contêineres atribuídos a ele para o navio.

A Fig. 4 mostra o modelo do PDM simplificado, que representa as transições possíveis, porém com o desacoplamento da variável que controla a quantidade de contêineres que ainda devem ser transportados por cada carregador (Fig. 4). Nessa generalização, os estados 0_n e 1_n foram “unidos” em classes de estados 0 e 1, respectivamente, tal qual sugere a definição do conjunto S dada pela Eq. (3). Os agentes sempre

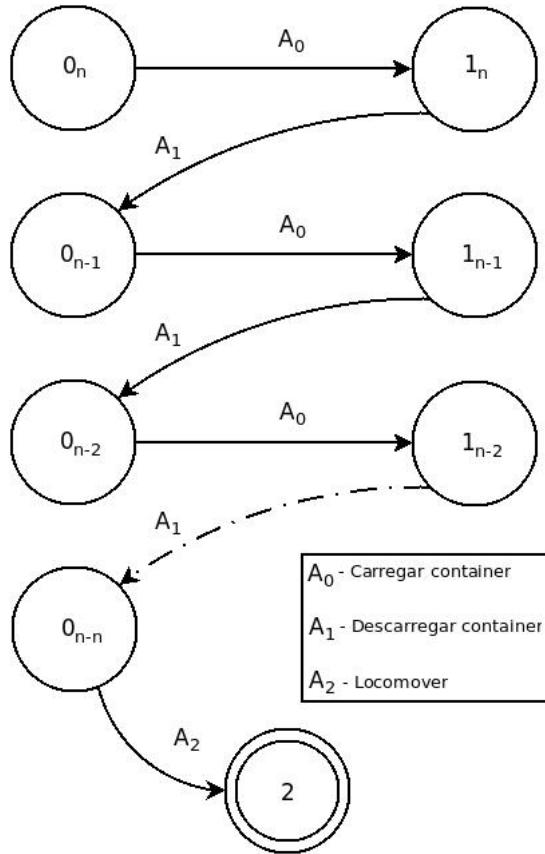


Figura 3. Estados e ações do PDM sob uma perspectiva genérica com n contêineres.

partem do estado 0, significando que estão sem nenhum contêiner, e vão para o estado 1 ao executar a ação A_0 e, neste momento, passam a ter um contêiner carregado. Ao descarregar esse item, executando a ação A_1 , o agente volta para o estado 0 e decrementa a variável que indica a quantidade de contêineres. Isso é feito até que esta variável seja igual a zero, fazendo com que o agente termine seu processo e se encaminhe para o estado 2. Na Fig. 4, p representa a probabilidade de transição de estado quando é realizada a ação indicada em um determinado estado e r a recompensa por fazer tal ação naquele estado.

O PDM é, então, utilizado para verificar o fato de ainda existirem contêineres para os carregadores transportarem. Com isso, cada carregador executa o seu respectivo PDM para verificar quando deve terminar o seu processo. Por fim, na *quarta fase* são executadas as simulações, onde é possível verificar os resultados. Com isso, pode-se verificar se os agentes estão transportando os contêineres corretos, ou seja, é possível saber se um agente carregador não está transportando um contêiner que deveria ser transportado por outro agente. Outra análise realizada através das simulações está no fato de ser possível analisar o momento em que cada carregador termina sua jornada de trabalho, ou seja, é possível verificar se o PDM funcionou de acordo com o que se esperava.

6. Simulações e Discussão de Resultados

Para implementação da função “mochila” e do método de Iteração de Valor para PDMs, utilizou-se a Linguagem de Programação C, e as simulações foram implementadas no ambiente de simulação multiagente NetLogo. No ambiente NetLogo foi criado todo o

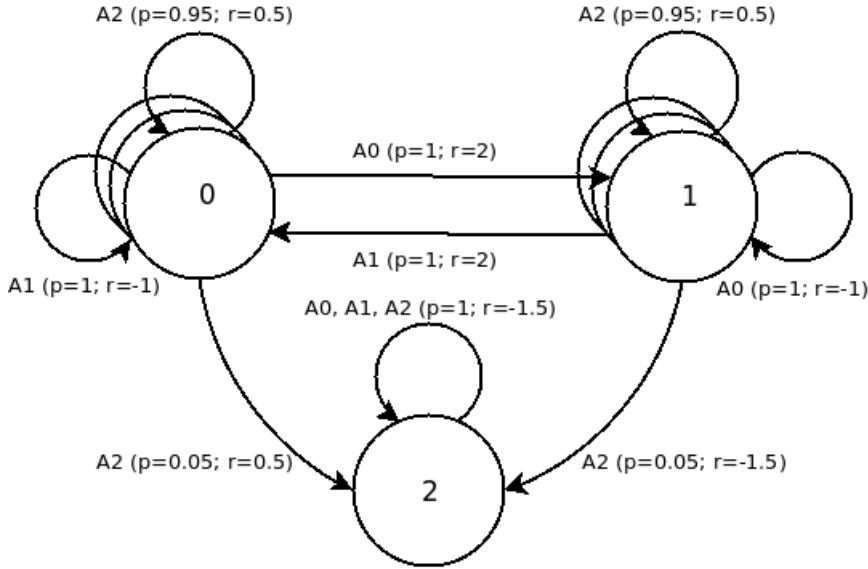


Figura 4. Estados e ações do Processo de Decisão de Markov simplificado.

contexto de um terminal portuário, como: o cais onde atracam os navios, os contêineres e as ruas para os deslocamentos dos carregadores. A Fig. 5 mostra o terminal portuário desenvolvido no *software NetLogo*.



Figura 5. Terminal de carga/descarga de contêineres desenvolvido no NetLogo.

Neste esquema, os carregadores se movimentam pelas ruas² demarcadas em escurinho e transportam os contêineres para o navio³, que fica atracado à direita do ambiente. Ao iniciar a simulação no NetLogo, os agentes já têm conhecimento de quais contêineres devem transportar, pois essa divisão foi realizada no programa em C. Assim, utilizou-se arquivos como controle de fluxo de informações entre o programa em C e o ambiente NetLogo. Através dessas informações, os agentes podem controlar o momento correto

²O tratamento de congestionamentos é trabalho futuro.

³Nesta modelagem, podem ocorrer descargas simultâneas no navio.

de término do seu respectivo processo, pois, através do PDM, recebem as instruções de quantos contêineres ainda faltam para transportar.

Como resultado positivo, desta-se a utilização do algoritmo de solução do Problema da Mochila combinado com um PDM, que conseguiu resolver o problema proposto de maneira satisfatória, utilizando a idéia de um PDMFA. Da mesma forma, a redução do problema inicial ao Problema da Mochila e a adaptação das características do problema, como a associação de um peso e uma prioridade aos contêineres, facilitaram a utilização do algoritmo de resolução do Problema da Mochila. Esse método de resolução utilizou a PD, o que contribuiu para a eficiência do compartilhamento de recursos.

A Fig. 6 mostra um gráfico que relaciona o crescimento da quantidade de contêineres pelo tempo de execução (em segundos) da função “mochila”. Observa-se que, mesmo que quarenta contêineres sejam utilizados por cada carregador, o tempo de execução da função *mochila* é de cerca de 120ms. Porém, no NetLogo podem ser utilizados apenas quatorze contêineres (por se tratar de uma pequena simulação), o que torna o algoritmo de solução do Problema da Mochila muito rápido, aproximadamente 40ms.

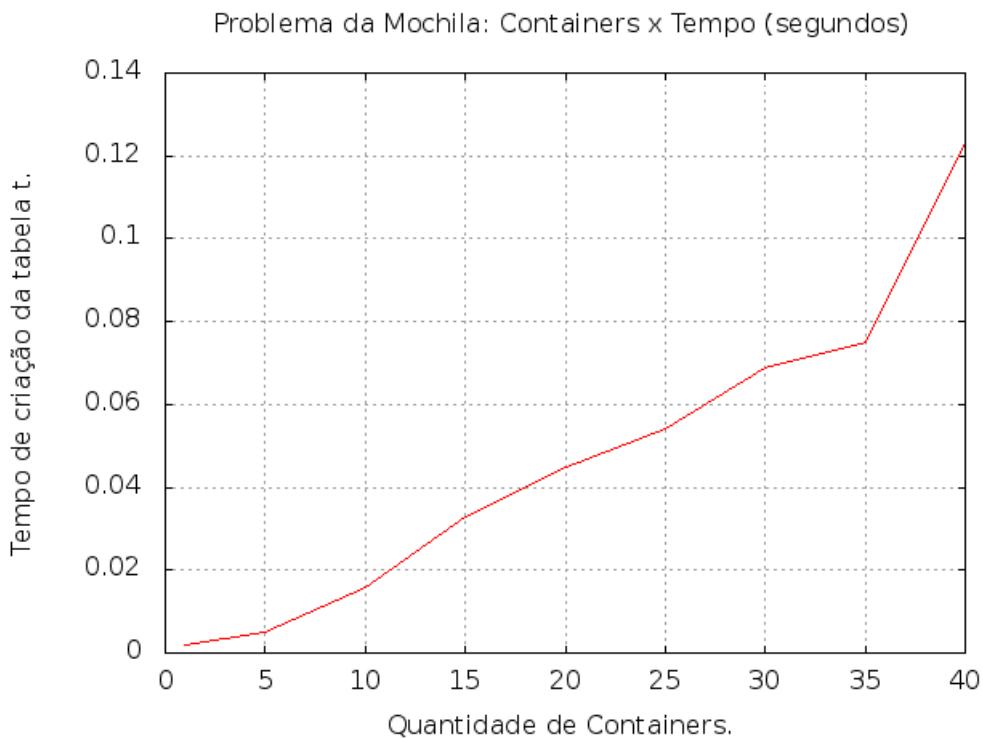


Figura 6. Relação entre quantidade de contêineres pelo tempo de execução da função “mochila”.

Um outro detalhe, relevante à simulação no ambiente NetLogo, diz respeito ao tempo de carregamento dos *containers* conforme são inseridos mais carregadores no ambiente. Em testes realizados, notou-se uma diferença de aproximadamente 50% de redução do tempo em alguns casos. Os testes foram realizados em um ambiente contendo, no máximo, 8 *containers*. Com isso, foram inseridos 3 agentes carregadores no ambiente, um a um. Quando a simulação contava com somente um carregador, cronometrou-se o tempo (ticks do simulador NetLogo) de carregamento de um *container*, depois de dois

containers e assim por diante. Logo em seguida, adicionou-se mais um carregador, totalizando 2 carregadores. Da mesma forma, os tempos foram medidos conforme aumentava o número de *containers* transportados. Por fim, adicionou-se o último carregador e, então, foram medidos os tempos. A Fig. 7 mostra um gráfico indicando a melhora na eficiência quando foram inseridos mais carregadores no ambiente. A linha em *vermelho* indica os tempos obtidos quando existia apenas um carregador na simulação. A linha *verde* mostra os tempos obtidos quando o ambiente contava com dois carregadores. Já a linha em *azul* mostra os tempos obtidos quando três carregadores estavam presentes no ambiente.

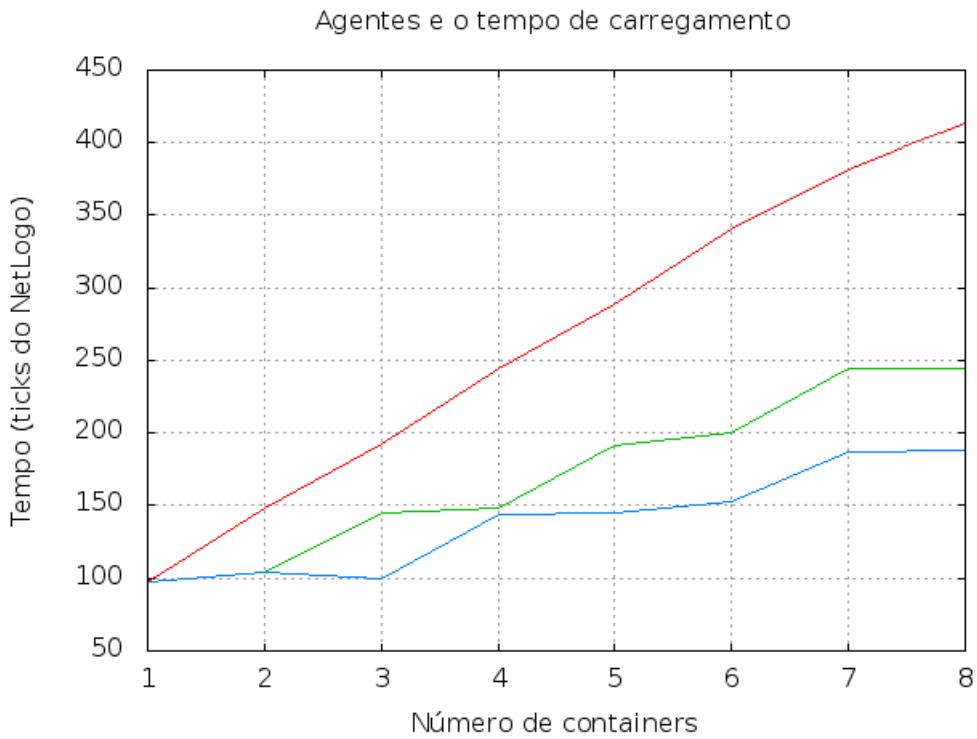


Figura 7. Relação entre a performance e a quantidade de agentes.

7. Conclusão

Este trabalho apresentou uma proposta para um sistema multiagente para a operação de carregamento com prioridades no terminal de carga/descarga de contêineres do Porto da cidade do Rio Grande. Deparou-se com um problema de formulação de política ótima para times de agentes com limitação para recursos em ambientes estocásticos, que é composto por dois subproblemas fortemente acoplados: um problema de alocação de recursos e um problema de otimização de política.

Ao contrário do que fizeram [Dolgov and Durfee 2004], onde foi utilizado o conceito de MILPs, a principal contribuição deste trabalho foi uma proposta de combinar os dois problemas, pela formulação de um problema de otimização de política sobre as operações de carga de cada carregador, depois de realizada a alocação dos contêineres compartilhados, segundo as restrições acima mencionadas, garantindo que as políticas não exigem carregamento em excesso por parte dos carregadores e as prioridades de embarque são respeitadas. Para tanto, para resolver o problema da distribuição de carga entre os carregadores foi utilizada uma redução do Problema da Mochila (Knapsack Problem).

Em seguida, o sistema utiliza os Processos de Decisão de Markov para decidir sobre as ações de movimentação (tais como: carga e descarga) de cada carregador dentro do ambiente. Uma restrição a essa proposta é o número de variáveis que podem ser consideradas na abordagem do Problema da Mochila (no caso, peso e prioridade de cada contêiner). Como trabalho futuro pretende-se avaliar essa condição para generalizar o modelo.

Agradecimentos. Financiado pelo CNPq (Proc. No. 305131/10-9, 483257/09-5, 560118/10-4, 304580/07-4) e FAPERGS.

Referências

- Boutilier, C. (1999). Sequential optimality and coordination in multiagent systems. In *Proc. Intl. Joint Conf. on Artificial Intelligence*, pages 478–485.
- Boutilier, C., Brafman, R., and Geib, C. (1997). Prioritized goal decomposition of Markov Decision Processes: Towards a synthesis of classical and decision theoretic planning. In Pollack, M., editor, *Proc. 15th Intl. Joint Conf. on Artificial Intelligence*, pages 1156–1163, San Francisco. Morgan Kaufmann.
- Dean, T., , and Lin, S.-H. (1995). Decomposition techniques for planning in stochastic domains. In *Proc. Intl. Joint Conf. on Artificial Intelligence*.
- Dimuro, G. P., Costa, A. C. R., and Gonçalves, L. V. (2008). Recognizing and learning observable social exchanges strategies in open societies. In *Proc. 1st Braz. Work. on Social Simulation - BWSS 2008/XIX Braz. Symp. on Artificial Intelligence - SBIA 2008, Salvador*, pages 140 –152, Porto Alegre. SBC.
- Dolgov, D. A. and Durfee, E. H. (2004). Optimal resource allocation and policy formulation in loosely-coupled Markov Decision Processes. In Zilberstein, S., Koehler, J., and Koenig, S., editors, *Proc. 14th Intl. Conf. on Automated Planning and Scheduling (ICAPS 2004)*, pages 315–324, Whistler. AAAI.
- Guo, M. and Conitzer, V. (2010). Optimal-in-expectation redistribution mechanisms. *Artificial Intelligence*, 174(5-6):363 – 381.
- Huang, Z., van der Aalst, W., Lu, X., and Duan, H. (2011). Reinforcement learning based resource allocation in business process management. *Data & Knowledge Engineering*, 70(1):127 – 145.
- Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134.
- Meuleau, N., Hauskrecht, M., Kim, K., Peshkin, L., Kaelbling, L. P., Dean, T., and Boutilier, C. (1998). Solving very large weakly coupled Markov Decision Processes. In *Proc. 15th Nat. Conf. on Artificial Intelligence (AAAI-98), 10th Conf. on Innovative Applic. of Artificial Intelligence (IAAI-98)*, pages 165–172, Menlo Park. AAAI Press.
- Parr, R. (1998). Flexible decomposition algorithms for weakly coupled Markov Decision Problems. In Cooper, G. F. and Moral, S., editors, *Proc. 14th Conf. on Uncertainty in Artificial Intelligence (UAI-98)*, pages 422–430, San Francisco. Morgan Kaufmann.
- Pereira, D., Gonçalves, L. V., Dimuro, G. P., and da Rocha Costa, A. C. (2008). Towards the self-regulation of personality-based social exchange processes in multiagent systems. In Zaverucha, G. and Costa, A. L., editors, *Advances in Artificial Intelligence*, volume 5249 of *LNAI*, pages 113–123. Springer, Berlin.
- Ribeiro, C. (2002). Reinforcement learning agents. *Artificial Intelligence Review*, 17(3):223–250.
- Russel, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, 3 edition.
- Singh, S. and Cohn, D. (1998). How to dynamically merge Markov Decision Processes. In Jordan, M. I., Kearns, M. J., and Solla, S. A., editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.
- Sniedovich, M. (2011). Dynamic Programming - Foundations and Principles (Second Edition).
- Yang, R., Bhulai, S., van der Mei, R., and Seinstra, F. (2011). Optimal resource allocation for time-reservation systems. *Performance Evaluation*, In Press.

A quantitative study about Tardiness in Java-based Multi-Agent Systems

Pier-Giovanni Taranti¹, Ricardo Choren², Carlos José Pereira de Lucena¹

¹PUC-Rio, Rua M. de São Vicente 225 – Rio de Janeiro/RJ, Brazil

²SE/8 - IME, Pça General Tibúrcio, 80 – Rio de Janeiro/RJ, Brazil

{ptaranti, lucena}@inf.puc-rio.br, choren@ime.eb.br

Abstract. Time constraints are designed as non-functional requirements which are implemented using programming language-dependent time schedulers. Multi-Agent System (MAS) applications are frequently developed using a Java-based agent platform. Thus, this Java-based MAS applications rely on native Java time schedulers. However, the Java documentation clearly states that there are no guarantees about the time accuracy during system execution. As a consequence, delays may occur at run time, and the correctness of the system execution can be compromised, which is even more critical in real-time or simulation applications. This paper presents a quantitative study on this issue, aiming to establish relationships between these delays and multi-agent paradigm abstractions. Such study is important to obtain data to indicate development solutions to mitigate problems that may arise from execution delays.

1. Introduction

Multi-Agent Systems (MAS) are systems composed of multiple interacting software agents that are situated in some virtual environment, and that are capable of autonomous action in order to meet its design objectives [Wooldridge and Jennings 1995, Wooldridge 2002]. The MAS paradigm intends to provide software engineers with concepts and tools for analyzing, designing and implementing complex software systems [Jennings 2001].

A number of toolkits, frameworks and platforms has been presented to support the development of MAS, as can be seen in [Bordini et al. 2006]. A representative subset of these technologies are based on Java, such as Jack [Everts et al. 2004], JADE [Bellifemine et al. 2007], Cougaar [Helsinger et al. 2004], Jadex [Pokahr et al. 2005] and Jason [Bordini et al. 2007]. Systems developed using these tools are executed on Java Run-time Environments (JRE).

A number of MAS applications, developed using these tools based on Java or using a hard-coded approach, has been presented in literature, such as systems for air-traffic control, factory process control, logistics and simulations. In MAS developed to control dynamic processes, some actions need to be firmly scheduled over time, in order to meet time-constraints, since the logical correctness of such systems will depend on their capacity to achieve the expected times for task execution.

However, in MAS, it is difficult to ensure the computation time spent by agents' tasks: the information exchange between agents is carried out using agents' messages, which can be asynchronous. Additionally, agents are not supposed to agree and comply

with all the received messages. These MAS characteristics make it difficult for software engineers to predict the system states at design-time and also to ensure the time needed for the system perform its tasks.

To overcome the uncertainty about the time of execution, engineers have used time-schedulers to stipulate when the developed MAS must perform some actions. A usual approach is the use of time-step and time-to-execution control mechanisms.

Some java-based frameworks and toolkits for developing MAS, such as Jade, Jack and Coogar, have support to schedule actions, using time-control mechanisms associated with MAS paradigm abstractions, like agent behaviors or actions. If this support is not available, programmers are forced to adopt a had-coded approach to schedule actions to be performed in a instant in the future. Java SDK technology provides some support to implement time schedulers through the use of time-control mechanisms that consider the system time to verify if it is time to execute some action.

The issue that arises from relying on Java-based schedulers is that Java makes no guarantees about time-accuracy: the scheduled time indicates only that an action will occur after the programmed instant. As a result, unpredictable delays may occur and systems may achieve unexpected and wrong states due to these delays, and if the logical correctness of a system depends on its timeliness, the system will fail. This uncertain about the instant when a scheduled action will be performed is consequence of the Java Garbage Collector (GC) and also is consequence of the threads execution approach adopt by standard Java and its JRE implementation. Java programmers can indicate the precedence level for threads, since the Java JRE uses a preemptive scheduler. However, there are different approaches to implement the interface between the Java JRE and the Operational System (OS), which has the final decision about when a thread will be started or suspended [Silberschatz et al. 2005, Oaks and Wong 2004]. The OS may leave a thread suspended during a period when a programmed execution instant exists, due to the OS own approach to handle threads, and this will cause a delay.

Despite the unpredictable character of these delays, we consider that it is possible to develop MAS mitigating the risk in not achieve the programed times. A possible mitigation could be using a design that reduces delays by construction. However, there is no work that explicit relates the delays at execution time, i.e the system tardiness, with MAS abstractions such as agents, actions and behaviors and its properties. This information is needed to support future approaches.

This paper presents a quantitative study about the timeliness in Java-based MAS, when using time-control mechanisms. This study aims to establish relationships between delays at execution time and MAS abstractions, in order to obtain outputs that may indicate future solutions to timeliness in MAS. Before presenting the quantitative study, this paper presents a conceptual review to show why delays may occur in Java-based MAS applications and also to show the relation between such delays and MAS abstractions.

The paper is organized as follows: the section 2 presents an overview on time control issues in Java-based MAS and Tardiness and Timeliness concepts and the relationship with MAS. Section 3 presents a quantitative study that aims to establish relationships between MAS concepts and Tardiness at run-time. General comments and some mitigation actions are presented in section 4. Section 5 presents the conclusions.

2. Issues in Time-Control for Java Based MAS

Java technology is largely used in industry and academy, and also MAS researchers have used Java to develop their solutions. A number of MAS developed using Java can be verified in literature, such as Multi-Agent Based Simulations (MABS), traffic control systems, systems to support logistic activity and auction systems (e.g. electricity market). These MAS applications have, at different levels, temporal constraints, i.e the system correctness depends on the system timeliness, or, to be more precise, these systems have real-time characteristics: a real-time system is one whose logical correctness is based on both the correctness of the outputs and their timeliness [Laplante 2004].

To handle with this temporal constraints, Java-based MAS can be developed using a hard-coded multi-thread programming, where the time control may be implemented using schedulers available in *java.util.concurrent* package. However, there are a number of mature toolkits, frameworks and platforms for MAS, developed on top of Java, such as Cougaar, Jack and Jade, that have time-scheduler implementations using agents abstractions (e.g actions, behaviors). These time-schedulers support periodic executions or a unique execution to be performed in a specific moment, which are time-step and time-to-execution approaches, respectively. The implementation is a loop that verifies if the start time has passed and, if it is true, executes some system action. The loop implementation often includes some waiting time between verifications.

The available mechanisms the standard Java SDK technology to deal with time constraints are time-schedulers. These schedulers provide basic support to implement time-dependent actions and they act on the Java threads execution. This support includes the ability to schedule actions or to suspend some execution flow during a specific period, and this was included in the Java 5 (The current implementation is available in the *java.util.concurrent*). According the Java documentation, scheduled or delayed tasks execute no sooner than they are enabled, but without any real-time guarantees about when, after they are enabled, they will commence [Java 2011]. The difference between the programmed instant and the real instant when the instruction starts is an intrinsic error of the schedule mechanisms, i.e their *tardiness*.

The reason why it is not possible to guarantee the timeliness of the scheduled actions when using standard Java is related to the GC mechanism and to the JRE implementation for threads control:

The GC can suspend the overall execution of the Java-based MAS, which will appear unresponsive during the period when the GC are scanning the memory heap, in order to free memory. This GC execution is controlled by the JRE, without influence of the executed MAS, and may cause delay all scheduled threads.

To control threads execution, Java uses a preemptive approach with different 10 priority levels in Solaris, as an example. JRE providers have different implementations for such preemptive mechanism and for the way in which Java threads are mapped to the OS kernel threads (e.g many-to-one or many-to-many). Additionally, the number of priority levels for threads in the OS often are greater than the number of levels established to Java. If a host experiments moments of intensive CPU usage, like back-up routines or load of new services, kernel threads will compete for resources and delays in the mapped Java threads may occur. In [Oaks and Wong 2004] an experiment shows that the same

code executed in different Java threads had performed with different duration.

As a resume, the standard use of Java technology does not provide a way to avoid delays caused by GC and to assure that the thread that are executing a scheduler will be active in the expected instant to start a programmed action. Thus, it is not possible to ensure the timeliness in MAS that are implemented in Java, because of tardiness caused by the Java Technology design. The consequence is that the programed events may be delayed and, if the logical correctness of the MAS rely also on its timeliness, undesirable states may be reached and then the MAS will fail at execution time. That is to say the system can fail because of the tardiness increase.

The problem that arises from this scenario is the need to ensure a timeliness level for Java-based MAS, when deployed in the production environment. As showed, it is not possible to assure a no-tardiness situation, thus it is necessary to tame the MAS tardiness to acceptable levels for each MAS application, so that the system correctness will not be compromised by delays at run-time. However, taming tardiness is not a simple task, considering that it can only be measured after the system is fully implemented at production hosts.

Since tardiness is being discussed for Java-based MAS, it is necessary to establish relationship between the tardiness variation and MAS design abstractions, such as agents and behavior. This information will help software engineers to construct more reliable software, once they would consider these relationships at design-time to mitigate the tardiness risk. Moreover, such relationships may be used as a start point to devise solutions to tame the tardiness in MAS applications.

For example, if there is hard evidence that the number of agents is proportional to the tardiness level in a MAS application, then the developer can try to design and/or implement a solution that tries to reduce the number of concurrent executing agents in the system. Nevertheless, the body of knowledge about this issue is very limited with no experimental works that explore such a kind of issue.

Performing a quantitative study is a good approach to find any relationships that there may be between MAS design abstractions and tardiness. This because the final result does not include personal impressions: an experiment is performed and only statistic analysis is provided as output. This avoidance of personal impressions is important in that case, since tardiness may be so small that it cannot be perceived by humans.

The next section presents the quantitative experiment performed, that had the purpose of showing that some specific factors associated to agent abstractions can affect tardiness in MAS.

3. Tardiness in Java-Based MAS: A Quantitative Study

The intention of this study is to identify and show that some MAS abstractions can be related with the system tardiness level in Java-based MAS. After this has been verified, the tardiness taming would be performed in a higher abstraction level, preventing the need to handle with low level Java code.

In future works, this information could support more detailed approaches and methodologies to develop MAS with time constrains using Java technology. In order to prospect this information, a statistical experiment with a huge amount of data for ana-

lyzing was designed and performed. The results of this study have confirmed the influence of determined MAS concepts on tardiness, and a summary of this experiment is presented in next subsections.

3.1. Experiment Design

The question of interest was if the variation in properties of agents abstractions in a MAS, such as agents and behaviors, has influence on the tardiness variation in a MAS. To design the experiment, the initial task was to choose the factors to be analyzed, i.e, the MAS abstractions properties which were candidates to influence the tardiness. After some experiments, presented in [Taranti et al. 2010b], it was decided to use the three following factors: *Number of agents active in the MAS*; *Duration between two consecutive executions of agents' repetitive action or behavior*; and *The load of agents' actions or behaviors (measured in time duration)*. This set does not intend to be conclusive, however these factors are elements that should be dealt with at design-time through the use of distributed architecture, modifications in the time constrains and the use of lighter tasks.

The experiment has considered each factor isolated since the cross influence between factors could vary in different MAS. The analysis was performed using the Analyze of Variance Methodology (ANOVA), presented in [Montgomery 2009]. The Initial Hypotheses (H_0) for each factor is that its variation does not have influence on the tardiness, or: $H_0 : \mu_1 = \mu_2 = \dots = \mu_a$. The Alternative Hypotheses (H_1) for each factor is that its variation has influence on the tardiness, or: $H_1 : \mu_1 \neq \mu_2 \neq \dots \neq \mu_a$.

The outputs of the data analysis are the Significance Level (i.e, α), and *Power*. α means the probability of rejecting the initial hypothesis when this hypothesis is true and it is expressed as $P(\text{reject } H_0 | H_0 \text{ is true})$, where P means probability. *Power* means the probability of rejecting initial hypothesis when this hypothesis is false and it is expressed as $P(\text{reject } H_0 | H_0 \text{ is false})$.

The data analysis was performed using R [Kabacoff 2009], which is a free software environment for statistical computing and graphics. This software is a powerful tool to analyze a huge amounts of data. However, the R plugin to conduct ANOVA tests does not provide the *Power* value. To calculate Power, the *Balanced One-way Analysis of Variance*, another R plugin, was used. With α and *Power*, it is possible to have a confidence level of the ANOVA test. Low levels of α and values near 1 for Power are expected to accept the test result.

The threshold level assumed in the experiment to Reject H_0 and Accept H_1 was: $\alpha < 0.0001$; and *Power* > 0.9999 . These values are greater than the usual, which is 0,05 and 0,95, respectively. The decision of raising the confidence level was due to the need to obtain a reliable data.

A MAS was designed for producing the data to be analyzed. This MAS has a number of agents that are all instances of the same agent class. In order to avoid interferences in the measures, it was decided not to implement interactions between instantiated agents. The MAS agents have two periodic behaviors which collect data in each execution. That is, implemented agents have only behaviors for data logging.

The MASP platform [Taranti et al. 2010a] was used to implement such MAS. MASP is a platform developed to support MABS experiments and is based on JADE,

which makes it both FIPA-compliant [FIPA 2010] and time-schedule supported. The MASP support for time-scheduler is based on the *TickerBehaviour.class* and *WakerBehaviour.Class* from JADE.

The two behaviors used in the experiment have the same period for consecutive executions and they use the non-preemptive multitask model from JADE, in which behaviors are objects of agents, and each agent of a MAS has its own thread of control [Bellifemine et al. 2005].

A total of 93 instances of this MAS were created, varying the number of agents and the period for consecutive executions used for the behaviors. To obtain different levels of load in behaviors, an artificial load was inserted in the behaviors code and this load varies linearly between 0 and 200 milliseconds from the beginning to the end of the MAS application execution. Each MAS execution were programed to be carried in 25 minutes.

Before executing the experiment, it was necessary to define how to measure delays in the system, i.e the Tardiness. After some tests, it was decided to use a non-dimensional value which indicates how much the execution duration is greater than the programed. The used tardiness formula is:

$$\text{Tardiness} = \frac{T_{executed}}{T_{programed}} - 1$$

3.2. Collecting and Analyzing the data

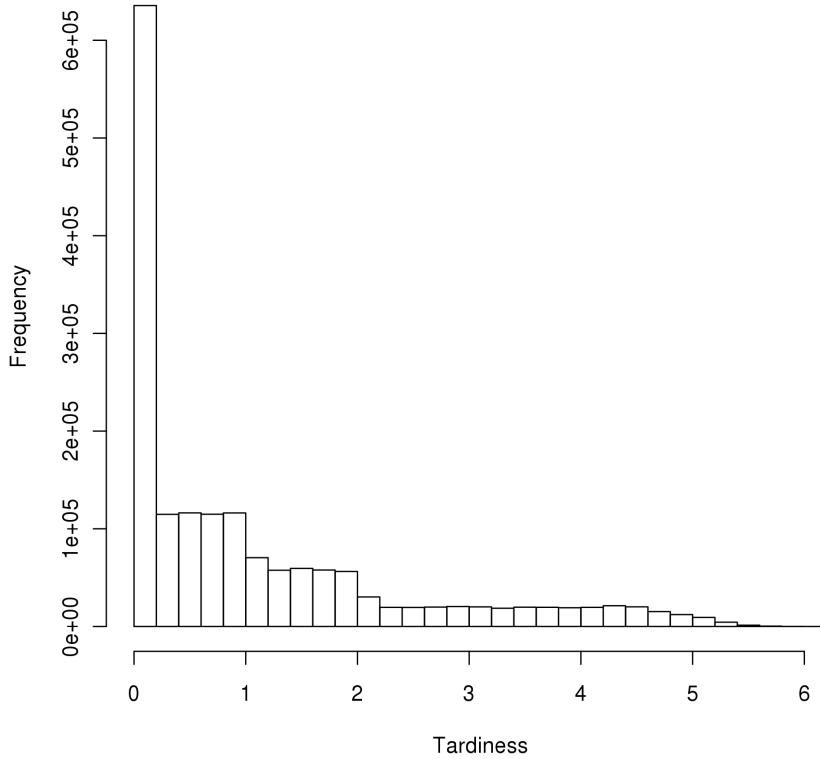
The MAS executions were all performed in the same environment. A computer was dedicated to run the experience, without network connections or other software being executed at the same time. Since there is a lot of noise at start-up, mostly due to the agents instantiation, the data related to the first 80,000 milliseconds in each execution was disregarded.

Since the total number of samples had exceeded 55 million, even the R used with the database was not able to process all the gathered data in the analysis environment. Thus, it was decided to use only data samples from 50 agents in each of the 93 instantiated MAS. This number overcomes the minimum number of samples for ensuring a power greater than 99.9% in each experiment. The calculus for this was performed using the function *power.anova.test*. The tardiness histogram is presented in the figure 1.

The analysis was carried for each factor isolated, since to measure the influence between factors will not produce a useful data: this influence will vary from a MAS to other.

First Factor: Number of Agents in the MAS – the hypothesis (H_0) for this analysis was that the number of agents running in a MAS does not affect the tardiness in the system, and the alternative hypothesis (H_1) was that this number affect the tardiness. To perform the test, a total of 30 different levels of number of agents were used, with 57,000 samples in each level. Figure 2 illustrates the sample behavior.

Second Factor: time interval between executions – the second analyzed sample had as hypothesis (H_0) that the interval between two periodic executions of repetitive agent

**Figure 1. Tardiness Histogram**

behaviors in a MAS should not affect the tardiness in the system, and the alternative hypothesis (H_1) was that this number affects the tardiness. To perform the test, a total of 3 different levels of interval duration were used (1, 2 and 3 seconds), with 580,000 samples in each level. Figure 3 illustrates the sample behavior.

Third Factor: Time for Computing Individual Actions – the third and last analyzed sample had as hypothesis (H_0) that the load, or computation time, of each individual behavior in a MAS does not affect the tardiness in the system, and the alternative hypothesis (H_1) was that this value affects the tardiness. To perform the test, a total of 19 different load levels were used, with 90,000 samples in each level. The boxplot graph illustrates the sample behavior (Fig. 4).

Statistical Analysis: for all three factors, the ANOVA analysis was performed using the specific sample, and the result confirmed the rejection of the initial hypotheses and the acceptance of the alternative (H_1), with $\alpha = 2.2e - 16$. The Balanced One-Way Analysis Of Variance Power Calculation shows for all factors that $Power = 1$ for the obtained significance level.

4. General Comments about Tardiness and Risk Mitigation

The experiment shows that tardiness in MAS has relationship with the selected factors related to MAS abstractions. The same experiment allows one to infer that a well done

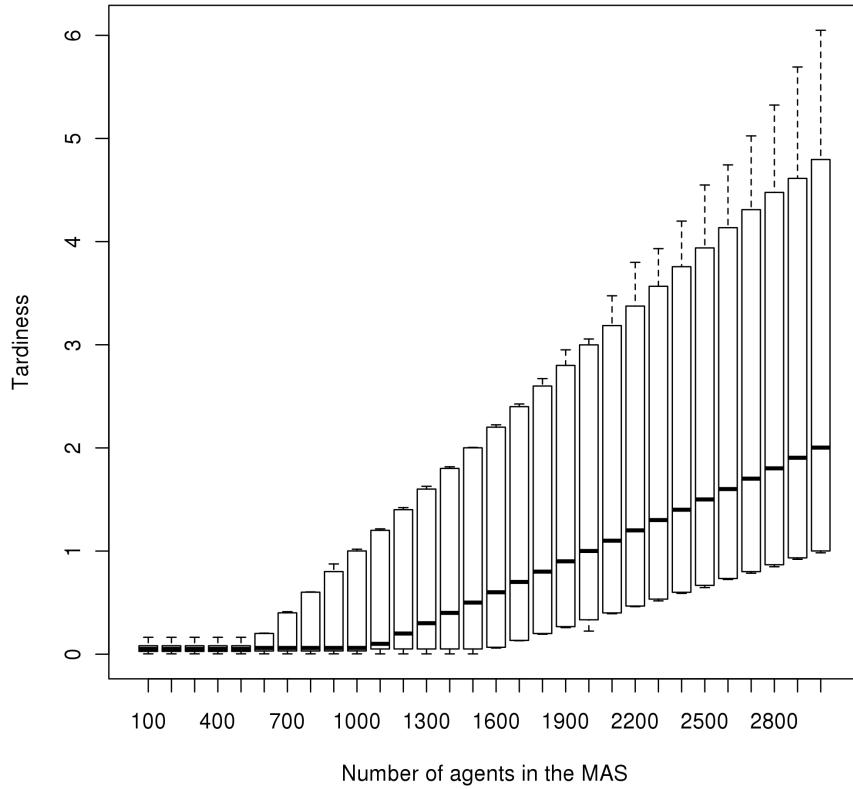


Figure 2. Boxplot of the first factor

software design could reduce the tardiness in MAS. Some possible decisions could be the use of flexible periods for consecutive executions in periodic behaviors, or spreading the load in some concurrent behaviors or reducing the numbers of agents acting simultaneously in the MAS. Another approach could use a distributed architecture in order to spread agents in case of overload.

All iterations of the presented experiment were performed in the same environment (i.e hardware and operational system) to reduce external influences. However, experiments carried out in other hardware environment with the same operational system and JRE version, have produced different level of tardiness, indicating that the tardiness is environment-dependent. This was expected, since network connections, hardware, operational system, and memory have influence on tardiness, in addition to the factors tested in the presented quantitative experiment. These factors have influence on the latency associated to computer hardware, network infrastructure and operating system. In addition, the memory heap size have influence on the GC behavior.

Since the tardiness is also environment-dependent, it is not possible to guarantee the timeliness in a production environment without performing specific non-functional tests to verify the system performance. This verification is an important mitigation action to reduce the risks of this system not achieving the expected time constrains, increasing the system reliability. In addition, the tests performed during the verification could provide the operational limits for operating the MAS, within a confidence interval.

Therefore, these tests are an *ad-hoc* activity, since there is an absence of tools and

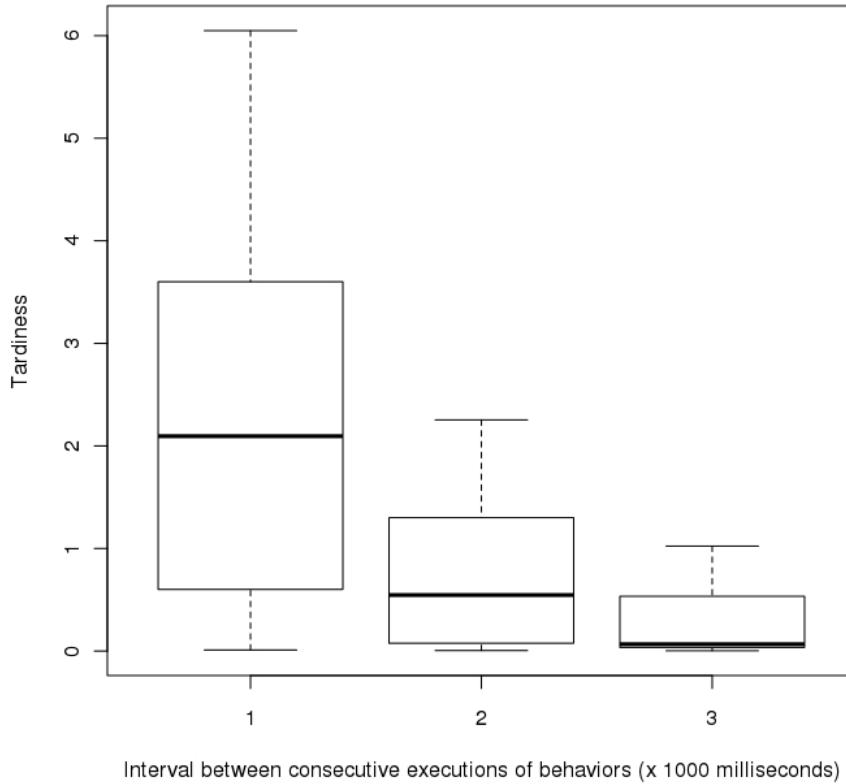


Figure 3. Boxplot of the second factor

theory for supporting the task, which is a costly one. MAS testing still is a challenge, although the active researches about the theme. Besides, the existent works commonly focus on the functional aspects of the MAS testing, not covering non-functional aspects as timeliness.

An option to support this testing activity may be instrumenting the code with executable asserts, that can measure the tardiness and launch a corrective action or exception when this tardiness pass the established critical level. These launched corrective actions, as a example, could suspend non-critical activities of the MAS, in order to prioritize the critical ones and reducing the tardiness level. In virtualized systems, these triggers could ask for resources to the host system.

The discussed mitigation actions do not intend to cover all possible solutions to tame the tardiness. In fact, these actions are very simple and limited solutions to tardiness issue. More robust approaches and methods are needed to treat the problem, and, as consequence, new researches are expected.

5. Conclusions

This paper presents a quantitative experiment about the tardiness in Java-based MAS considering the influence on tardiness of three different factors: the number of agents in the MAS, the load of the scheduled actions and the time interval between executions of periodic behaviors. This tardiness can cause logical errors at execution time, if the MAS depends on its timeliness to perform, i.e., if the MAS has real-time characteristics.

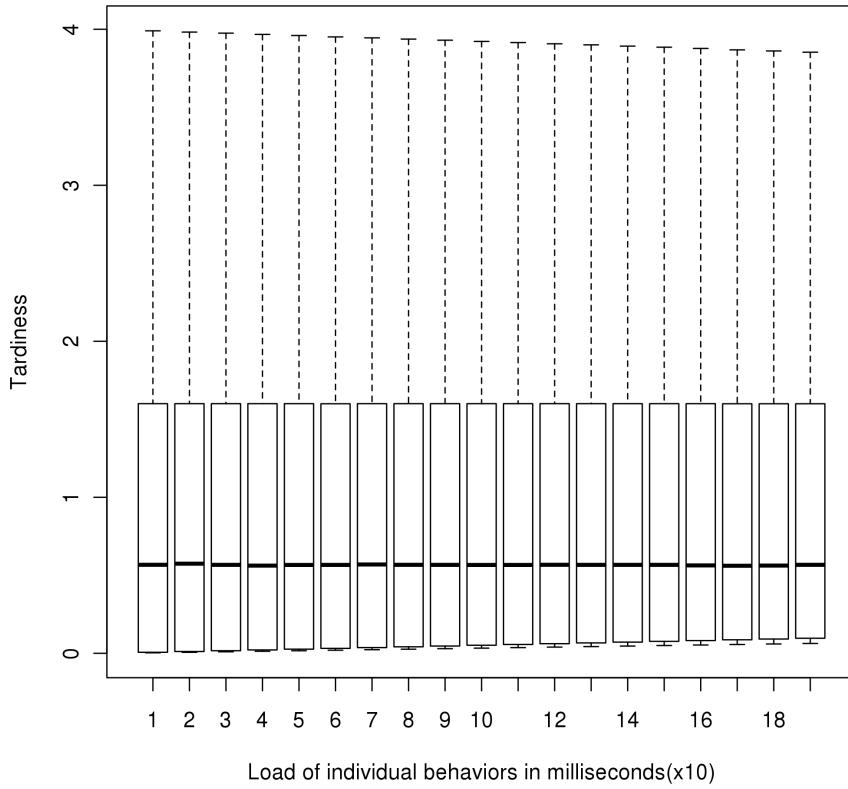


Figure 4. Boxplot of the third factor

Time schedulers are an important resource to control the data flow at execution time and this work does not intend to recommend its non-use. However, the experiment indicates that, when using time schedulers to control critical functions in a Java-based MAS, it is necessary to perform non-functional tests in an environment similar to production environment in order to measure and define the operational limits in which the MAS timeliness can be guaranteed.

The risk of tardiness occurrence, above the established critical level, must be mitigated, since this tardiness may compromise the MAS execution correctness. Some possible mitigation actions were also presented, such as performing non-functional tests at development and execution time, including executable asserts to be run at execution time to verify the error level and designing the system with flexible parameters in order to be able to tame the tardiness at run-time.

The work intends to indicate and support possible approaches to tame the tardiness. For this reason, a substantial statistical analysis was presented in this paper, and some experimental observations were included in the text, too.

References

- Bellifemine, F., Bergenti, F., Caire, G., and Poggi, P. (2005). *Multi-Agent Programming: Languages, Platforms and Applications*, chapter JADE – a java agent development framework, pages 125–148. Number 15 in Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer.

- Bellifemine, F. L., Caire, G., and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. John Wiley & Sons.
- Bordini, R., Braubach, L., Dastani, M., El, A., Seghrouchni, F., Gomez-sanz, J., Leite, J., Pokahr, A., and Ricci, A. (2006). A survey of programming languages and platforms for multi-agent systems. *Informatica*, (30):33–44.
- Bordini, R. H., Hubner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Willey & Sons.
- Everts, R., Fletcher, M., Jones, R., Jarvis, J., Brusey, J., and Dance, S. (2004). Implementing industrial multi-agent systems using jack. In Dastani, M., Dix, J., and El Fallah-Seghrouchni, A., editors, *Programming Multi-Agent Systems*, volume 3067 of *Lecture Notes in Computer Science*, pages 18–48. Springer Berlin / Heidelberg.
- FIPA (2010). Foundation for intelligent physical agents. <http://www.fipa.org/>.
- Helsingier, A., Thome, M., and Wright, T. (2004). Cougaar: a scalable, distributed multi-agent architecture. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*.
- Java (2011). <http://download.oracle.com/javase/6/docs/>.
- Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Commun. ACM*, 44:35–41.
- Kabacoff, R. I. (2009). *R in Action: Data Analysis and Graphics with R*. Manning Publications Co.
- Laplante, P. A. (2004). *Real-Time Systems Design and Analysis, 3rd Edition*. Wiley-IEEE Press.
- Montgomery, D. C. (2009). *Design and Analysis of Experiments*. John Wiley & Sons, Inc., 7 edition.
- Oaks, S. and Wong, H. (2004). *Java Threads*. O'Reilly Media, 3th edition.
- Pokahr, A., Braubach, L., and Lamersdorf, W. (2005). Jadex: A bdi reasoning engine. In Weiss, G., Bordini, R., Dastani, M., Dix, J., and Fallah Seghrouchni, A., editors, *Multi-Agent Programming*, volume 15 of *Multiagent Systems, Artificial Societies, And Simulated Organizations*, pages 149–174. Springer US.
- Silberschatz, A., Galvin, P. B., and Gagne, G. (2005). *Operating System Concepts*. Willey, 7th edition.
- Taranti, P.-G., Breitman, K. K., Lucena, C. J. P., and Choren, R. (2010a). An approach to reduce the gap between conceptual and execution models in agent-directed simulations. In *Proceedings of the 2010 Spring Simulation Multiconference*, SpringSim '10, pages 4:1–4:8, New York, NY, USA. ACM.
- Taranti, P. G., Choren, R., and Lucena, C. J. (2010b). A quantitative study about the hidden risk of using time-scheduler mechanisms to control execution flow in jade-based mas. In *Proceedings of I Workshop on Autonomous Software Systems (AutoSoft) at First Brazilian Conference on Software: Theory and Practice (CBSOFT)*, pages 61–70. SBC.

Wooldridge, M. and Jennings, N. (1995). Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 10(2):115–152.

Wooldridge, M. J. (2002). *An Introduction to MultiAgent Systems*. John Wiley & Sons, Inc.

Coordination of Agents in the RoboCup Rescue: A Partial Global approach

André H. Pereira¹, Luis G. Nardin¹, Jaime S. Sichman¹

¹Laboratório de Técnicas Inteligentes - EP/USP
Av. Prof. Luciano Gualberto, 158 – trav. 3
05508-970 – São Paulo – SP – Brasil

{andre.hahn, luis.nardin}@usp.br, jaime.sichman@poli.usp.br

Abstract. Coordination is one of the key issues in cooperative multiagent systems and it also plays an essential role in disaster management. Task allocation is an important part of the coordination problem since the decomposition of the objective into tasks is the most natural way to organize work among agents. Therefore, in this paper we propose a task allocation approach that considers the existence of local and global information in order to coordinate the agents in the RoboCup Rescue. Moreover, we present a comparative analysis between the results obtained by the agents which implement our approach and the sample agents provided in the RoboCup Rescue simulator.

Resumo. Coordenação é um dos principais problemas em sistemas multiagente cooperativos e ela também desempenha um papel essencial na gestão de desastres. A alocação de tarefas é uma parte importante do problema de coordenação, já que a decomposição de um objetivo em tarefas é a forma mais natural para se organizar o trabalho entre agentes. Portanto, nesse trabalho é proposta uma abordagem de alocação de tarefas que considera a existência de informações locais e globais com o objetivo de coordenar os agentes no RoboCup Rescue. Além disso, é apresentada uma análise comparativa entre os resultados obtidos com os agentes que utilizam nossa abordagem e os dos agentes que acompanham o simulador RoboCup Rescue.

1. Introduction

Coordination is one of the key issues in cooperative multiagent systems (MAS) since it ensures that agents will not accidentally interfere with each other's subgoals while attempting to achieve a common goal [Wooldridge 2009]. Therefore, coordination can be defined as the act of handling the inter-dependencies between the activities of the agents in order to maintain the system coherence.

Disaster may be defined as a crisis situation causing wide spread damage which far exceeds our ability to recover. Its management is normally critical since it involves a very large number of heterogeneous entities in a complex and dynamic environment, where the information is incomplete and uncertain, and decision must be taken in a timely fashion. In those situations, coordination and cooperation play an essential role in the assistance provisioning.

Therefore, in order to promote research and development of efficient response policies to disaster scenarios, the RoboCup Rescue Agent Simulation (available at [http:](http://)

//www.robocuprescue.org) competition was proposed in 2001. The competition involves scoring competing multiagent teams coordination in a city where an earthquake has just happened. These teams are comprised of platoon agents (Ambulance Teams, Fire Brigades and Police Forces) which interact directly with the world, and center agents (Ambulance Center, Fire Station and Police Office), which cannot interact directly with the world, but only communicates with platoon agents of its same kind and with the other centers.

In this paper, we propose a task allocation approach, based on *Partial Global Planning* [Durfee 1988], to coordinate agents in the RoboCup Rescue simulation environment. Such approach considers and handles the existence of local and global information. Therefore, while the agents keep only local information about the environment, the centers keep a consolidated view from all the agents. Moreover, the agents prefer to act using their local information and they request information to the centers only when they are idle. Our main idea is that the centers should not directly coordinate the agents, but should take advantage of the global view to help in directing their actions when needed.

The rest of the document is structured as follows. Section 2 presents an overview of the RoboCup Rescue simulator. The agents' communication and coordination strategy proposed is presented in section 3 followed by the description of the proposed agents' strategy in section 4. Thus, in section 5 is presented a comparative analysis between the agents using our approach and the ones following a random strategy. Finally, we present our conclusions in section 6.

2. The RoboCup Rescue Simulator

The RoboCup Rescue Agents simulator aims to provide a simulation of a post-earthquake scenario, simulating real environment objects' behavior and humans' capabilities and limitations. In order to make the simulation even more realistic some issues such as heterogeneity, limited information, limited communication channels, planning, and real-time are considered, characterizing this as a complex multiagent domain [Kitano et al. 1999].

The RoboCup Rescue simulator is comprised of one kernel and some sub-simulators, each assigned to a specific task within the simulation [Morimoto 2002]. There are sub-simulators responsible for the fire simulation, the traffic simulation, the collapse simulation, the blockage simulation and one responsible for miscellaneous simulations, such as humans' damage and buriedness and road clearing times. The simulator kernel is responsible for managing the interaction among the sub-simulators through the network.

The simulator was designed to provide a realistic vision of a disaster scenario and in order to explain its specification we subdivide it into 3 distinct parts, described in the sequence: Environment (section 2.1), Agents' Features and Capabilities (section 2.2) and Messages (section 2.3).

2.1. Environment

The environment represented in RoboCup Rescue simulator is composed of many different objects. The most important objects in the simulation are: buildings, roads, blockages, refuges and humans [Morimoto 2002].

Buildings The buildings have some properties, such as area, building material, fire fieriness and brokenness. All these information are used by the simulator to calculate

the fire propagation and the building's collapse. The building material defines how fast the building will burn and how it will collapse. The fieryness is a value which indicates how strong the fire is, ranging from 0 to 7, where 0 indicates the building is not on fire and 7 indicates the building is completely burnt. The brokenness property indicates how destroyed is the building and how likely it is to crumble.

Roads The roads are the entities used to move along the map. Some of their properties are length, number of lanes and blockages. If a road has no free lanes, i.e. the lanes are blocked by blockages or by other agents parked in the way, it is not possible to move along it.

Blockages Blockages are entities which block roads, making the lanes they are blocking impassable. They have the property of repair cost, which defines how hard it is to remove the blockage. Blockages are created by collapsed buildings and they can only be cleared by police forces (described in section 2.2).

Refuges Refuges are a special kind of building that keep the rescued civilians and provide water to fire brigade (described in section 2.2) agents. The refuge cannot get on fire and do not collapse.

Humans Humans are humanoid agents, which interact with the environment. They are described in section 2.2.

2.2. Agents' Features and Capabilities

The agents in the RoboCup Rescue Agent Simulation are divided in two main types: rescue agents and victims (civilians). The rescue agents are classified into moving (platoon) agents and fixed (center) agents.

The platoon agents are responsible for performing some actions in the environment like clearing roads, rescuing victims and extinguishing fires. They are subdivided into three kinds: Ambulance Teams, Fire Brigades and Police Forces.

Ambulance Teams Ambulance teams are the agents responsible for rescuing civilians. Their exclusive capabilities are unburying buried agents and rescuing hurt agents. The ambulances can rescue any type of agent, not only civilians, although their main goal is to take civilians safely to the refuge.

Fire Brigades Fire Brigades are the agents responsible for extinguishing fires, therefore their exclusive capability is fire extinguishing. Nevertheless, they have water tanks with limited capacity and there is a maximum distance they can reach with their fire hoses. To refill their water tanks the fire brigades need to go to a refuge, in which a certain amount of water is refilled at each turn.

Police Forces Police Forces are the agents responsible for clearing blocked roads. Their exclusive capability is clearing roads. If a road is blocked, agents cannot cross it, so they may get stuck or may need to take a much longer route to get to their destination. To clear a blockage the police force needs to stay close to the blockage and send clear commands to the kernel, the time taken to clear a blockage varies, depending on the repair cost of the blockage.

On the other hand, the center agents can only interact with the environment through voice messages and they can only send messages to the rescue agents of its kind.

Therefore, there are also three kinds of centers: Ambulance Center, Fire Station and Police Offices, which are respectively responsible for the Ambulance Teams, Fire Brigades and Police Forces.

The rescue agents interaction with the simulator is performed through the simulator kernel, and each agent runs as a separated process, similarly to the sub-simulators, which also run as independent processes. This setting allows for the computational load to be distributed among several computers, what helps the simulation workload distribution.

Table 1 presents the agents available capabilities in the simulator; it may be noticed that civilians do not have any special abilities. Further information about the agents' capabilities are found in [Morimoto 2002].

Table 1. Agents' capabilities

Type	Capabilities
Civilian	Sense, Hear, Move, Say
Ambulance Team	Sense, Hear, Move, Say, Tell, Rescue, Load, Unload
Fire Brigade	Sense, Hear, Move, Say, Tell, Extinguish
Police Force	Sense, Hear, Move, Say, Tell, Clear
Center	Hear, Say, Tell

All agents, except the centers, have one initial amount of health points, which may decrease if they are hurt or if they stay inside a burning building. They can also be buried, in which case they are trapped and cannot move until they are unburied by an ambulance team.

The only perfect information each agent has about the environment is the map of the city, as it is known since the beginning and it is static. In order to perceive current information about the environment, the agents have the function *sense*, through which the kernel sends information about the surroundings, which include current situations of buildings, roads and other agents that are in the perception radius of the agent. The other way agents can discover informations about the world is through voice messages sent by other agents.

In order to move around, agents have to send a *move* command to the kernel. They have a finite speed and may be unable to pass through some route if the road is blocked or if there is another agent blocking the way. It is possible to move around roads and buildings, entering and exiting the latter by their entrances.

The Ambulance Teams are capable of rescuing victims (Rescue), loading them into the ambulance (Load) and unloading them at the refuges (Unload). Fire Brigades have the ability to extinguish fires (Extinguish), while Police Forces are able to clear blocked roads (Clear). There is one important difference between Police Forces, Fire Brigades and Ambulances: while it makes no difference to have more than one Police Force to clean the same blocked road, this is not the case for the other two types of agents. A fire will be extinguished faster if more Fire Brigades are acting on it, and victims are going to be rescued faster if more Ambulance Teams are allocated to their rescue. The only limitation of the Ambulance Team is that it cannot carry more than one victim simultaneously.

2.3. Messages

There are two types of voice messages agents can use: the *tell* and the *say*. The *tell* type sends message through a radio to all agents of the same kind of the sender and to its center, if it is a platoon, and to all its agents and to the other centers if the sender is a center. The *say* type sends message to all agents in a defined radius from the sender, the default radius used in the competition is 30 meters.

Voice messages have a maximum size of 256 bytes, if an agent tries to send a message bigger than that the message gets dropped. There is also a limitation on the amount of messages one agent can send or receive at each turn. This limit is four received and four sent messages for platoon agents and $2n$ received and $2n$ sent messages for centers, where n is the amount of platoon agents the center has. The only information available to decide to hear the message or not is the sender id.

3. Agent Coordination and Communication: A Partial Global approach

In this section it is presented the agent coordination and communication strategy employed in the experiments with the RoboCup Rescue Agent Simulation performed at section 5.

The main idea of the strategy proposed was based on the *Partial Global Planning* approach which cooperating agents exchange information in order to reach common conclusions about the problem-solving process [Wooldridge 2009]. It is considered *partial* because the agents do not have all the environmental information available in order to make the best decision. On the other hand, it is *global* because the agents can exchange and obtain information through the centers which consolidates the environment information from all the agents.

Initially, our strategy considers that platoon agents can act independently from their centers, nonetheless they are limited to act based solely on their own local information or on information received from other close agents. This primary strategy was defined to handle situations in which either the platoon agents cannot contact their centers or the centers are not available at the simulation. In these situations, the platoon agents behavior is to walk randomly through the environment looking for victims, burning buildings or blockages. When they find any of these situations, they may act (if they have the required ability) and keep the information in their memory in order to provide it to another agent when they meet in the near future.

However, since in most situations the communication to the centers is available, our other communication strategy emphasizes the information flow concerning victims, burning buildings and blockages from platoon agents to their center and vice-versa. In such scenario, in the beginning of the simulation the platoon agents walk randomly in order to provide information about the environment to their center. If they do not find any interesting situation to act upon after a short period of time, they request information from their center. If the centers do not have any information to provide, the agents continue to walk randomly for another short period; however, if the centers provide an information, then they try to discover the shortest path to the target, they move to the location and start to act on the situation. When the action is finished, they update their centers with the current information. In the sequence, if the platoon agents does not have or cannot sense

any other situation to act, they request again information to their centers and the cycle restarts.

Even if the centers cannot directly coordinate the platoon agents, they can perform a sort of coordination when providing information to them. Our strategy considers that the centers will receive all information provided by the platoon agents, and based on these information, the centers can determine the priority of actions to be taken by the platoon agents and, in this way, coordinate the platoon agents' work.

The Ambulance Center agent, for instance, can use the information about the remaining life of the victims, the damage and distance to determine the priority for Ambulance Teams rescue. On the other hand, the Police Office Center agent can consider the rescue agent types to determine the cleaning blockage priority. In our case, the priority sequence is (1) Ambulance Teams with victims, (2) Ambulance Teams without victims and (3) Fire Brigade.

The Fire Station Center agent continuously determines where is the location with more fire spots based on the fire focuses information received from the platoon agents and redirects more Fire Brigades to that location since it has more possible victims and higher number of buildings burning.

The Police Office Center receives information about blockages from the other centers and sends it to the Police Force agents when requested.

4. Agents' Strategy

In this section, the strategy of each kind of rescue agent (Fire Brigade Agent, Policy Force Agent and Ambulance Team) is described in more detail.

Reliable information about the simulated environment is vital for efficient action planning. In the context of RoboCup Rescue simulation, the information mentioned are those related to blocked roads, victims and buildings on fire. In order to keep the agents well informed, updated and reliable information must be available for them and the best information source in this scenario are the agents themselves. However, as the agents have limited sensing and communication capabilities, information sharing is not trivial, as it is needed to deal with these limitations. So we decided to use a strategy of centralizing information, however, keeping the rescue agent autonomous to perform its tasks if it cannot contact its center. Our idea is to hold a global and centralized information repository where all relevant data sensed by the agents are kept. This information is provided by the agents themselves, and then processed and redistributed to all the other agents when requested.

The following subsection provides a brief description of each type of rescue agent implemented in this work.

4.1. Fire Brigade Agent

When the simulation starts, the Fire Brigades seek for fire spots randomly. After finding a fire spot, they try to move to that location to extinguish the fire. If they are able to get within the range to start extinguishing the fire, they do so until either the fire is extinguished or they run out of water.

In case they are unable to reach the destination within a defined amount of time, the current target is considered unreachable and is ignored, and therefore, the agent starts the search for a new target.

When the building that the Fire Brigade is targeting has its fire extinguished or burns completely the its target is reset to null and the searching process restarts.

When an agent runs out of water, it tries to find a path to the nearest refuge to refill its water tank, without changing its target building. Once the Fire Brigade is refilled, it tries to find a way back to its last target to resume its firefighting. This means that the agents are persistent with their goals, as they do not change its current goal until it is deemed impossible to reach or is no longer a valid goal.

4.2. Police Force Agent

Clearing ways to refuges is crucial, as Fire Brigades need to get to refuges to refill their water tanks and Ambulance Teams need to have fast ways to reach the refuge when transporting civilian. Hence, the initial objective of the Police Forces is to clear a way to the refuges. In order to perform this task, at the beginning of the simulation, they try to reach the closest refuge clearing all the blockade roads they find on the way. Once one refuge has been reached at least one way to that refuge is supposed to exist and the Police Force resume its operation to a general strategy.

The Police Forces' general strategy consists of clearing the closest blockage they have information of, if they know of any and they have not received any orders from the Police Office. While clearing a blockage, the Police Forces send *say* messages to warn close Police Forces the blockage is already being cleared.

In case the Police Force has received any orders from the Police Office, it stops doing whatever it is doing and moves to the target received from its center, clearing blockages on the way if needed.

If the Police Force does not know of any blockages and it has not received any order from the Police Office, then it asks the Police Office for directions and, while waiting for the response, moves randomly.

4.3. Ambulance Agent

Civilians may die if not rescued fast enough, hence, efficiency is vital. Therefore, the strategy adopted by Ambulance Teams has to be carefully planned. The strategy employed by our Ambulance Teams has some phases. At the beginning of the simulation, they search randomly for victims.

When a victim is found, the Ambulance Team moves to that location and starts the rescue procedure. Firstly, it analyzes if the building is on fire, then if the building is on fire unfortunately the ambulance can do nothing, since it can die with the fire as well. In this case, the Ambulance Team informs its center about the fire which may inform the Fire Station Center, since the Ambulance Team cannot communicate directly to other centers.

Nevertheless, if the building is not on fire then the Ambulance Team checks if the civilian is buried, in case it is, the Ambulance Team starts the unburying process. After the victim has been unburied the Ambulance Team picks it up (load it) and starts transporting

it to the closest refuge. Once they arrive at the refuge, the civilian is unloaded and the Ambulance Team starts to search for another victim.

The reliability of the information about civilians is taken into account when deciding which civilian to rescue. Hence, if the information is too old, based on a threshold of time, then the Ambulance Team ignores it, since the information is considered not reliable.

On a future work, we intend to improve this strategy in order to make the Ambulance Teams consider the buildings' fieriness where the civilian is before it moves to its location.

5. Comparative analysis

The main objective of these experiments was to compare the simulation score obtained when using the sample agents, which were provided with the RoboCup Rescue platform, with the ones using the coordination approach presented in section 3. The sample agents from RoboCup Rescue platform is simple since the agents move randomly and perform the tasks as they found it. They do not exchange information at all and they do not implement any kind of optimization in order to select the best task to perform at the moment.

We have run these experiments using four maps that are provided together with the RoboCup Rescue simulator version 0.49, namely VC, Foligno, Kobe and Kobe4. In order to run the experiments it was considered three kinds of agents (Fire Brigade, Police Force, and Ambulance Team) as well as all tasks (extinguishing fire, unblocking streets, and rescuing civilians). Moreover, it was considered the existence of one available center of each kind (Fire Station, Police Office and Ambulance Center).

In order to enable the comparison between the experiments, the parameters used in all the maps simulations were the same for both strategies. Each simulation is set to perform 300 cycles, where each cycle should take 1 second, 0.5 second is for the agent reasoning and 0.5 second for the simulation execution. The maximum message size an agent can transmit is 256 bytes and its perception can sense 10 meters away. The Fire Brigades can extinguish fire at a distance of 30 meters.

Table 2. Maps configuration parameters

	VC	Foligno	Kobe	Kobe4
Civilian	80	70	72	67
Ambulance Team	8	8	6	6
Fire Brigade	6	10	10	12
Police Force	11	10	8	9
Refuge	5	4	3	2
Fire Spot	5	4	6	7

To reach the goal, we considered the execution of 30 simulations for each map for each strategy with 300 cycles each. Each simulation was composed of number of agents specified at Table 2. All the experiments were run in a PC with processor Intel i3 (2.26 GHz) and 4 GB RAM.

To measure performance we use the standard RoboCup Rescue score which is computed in equation 1, where P is the quantity of alive civilians, H is a measure of the health condition of alive civilians, H_{init} is a measure of the health condition of the civilians in the beginning of the simulation, B is the building area left undamaged, and B_{init} is the total initial building area.

$$V = \left(P + \frac{H}{H_{init}} \right) * \sqrt{\frac{B}{B_{init}}} \quad (1)$$

Table 3 presents the mean value and the standard deviation obtained in the execution of the experiments. The full results and the script in using R Statistics (available at <http://www.r-project.org>) can be downloaded at http://www.lti.pcs.usp.br/~gnardin/wesaac11_data.tar.gz.

Table 3. Experiments results

	Sample Agent	LTI Rescue Agent
VC	34.29 ± 2.41	38.88 ± 2.78
Foligno	54.06 ± 6.41	64.96 ± 3.12
Kobe	37.43 ± 6.37	54.50 ± 7.16
Kobe4	54.04 ± 4.42	51.67 ± 3.02

The methodology used to analyze the data generated by the experiments is based on a statistical hypothesis testing [Boslaugh and Watters 2008]. The statistical hypothesis test adopted is the nonparametric statistical hypothesis test named *Wilcoxon's Rank Sum Test* [Boslaugh and Watters 2008]. This hypothesis test was selected because the data did not follow normal distribution and we could not ignore such data characteristic.

In order to compare the result of the two strategies, it was verified if the mean value of the scores obtained by using the sample agents (MV_S) was greater than the mean value obtained by using the LTI Rescue agents implemented with the approach presented in this paper (MV_O). The underlying idea was that if these results were statistically different, then it would mean that our approach is better than the random approach. Thus, using Wilcoxon's Rank Sum Test, the following test was performed:

$$H_0 : MV_O \leq MV_S$$

$$H_1 : MV_O > MV_S$$

When the hypothesis test is applied to the four maps considering the risk level (α) of 0.05 and the degree of freedom of 28, those hypotheses generate the results presented in Table 4. Those results indicate that the H_0 is rejected for three of the maps, except for the Kobe4, and demonstrate that our approach implemented in the LTI Rescue Agents is better than the one implemented in the sample agents for most of the maps.

We noted that for map Kobe4 there is no statistically significant differences between the strategies. This happens because the map covers a much larger area and the number of agents used is perhaps too low. This makes the approach presented perform as the random one since the rescue agents could not easily find the civilians and the burning builds in order to perform their tasks and to inform the centers about it.

Table 4. Wilcoxon hypotheses test results

	p-value
VC	$2.018e^{-9}$
Foligno	$5.422e^{-12}$
Kobe	$6.858e^{-13}$
Kobe4	0.9922

6. Conclusions

This paper presented a task allocation approach based on the *partial global planning* approach which considers the existence of a local and global information in order to coordinate the agents in the RoboCup Rescue. While the local information is maintained in the agents, the centers maintain a global consolidated view of the environment that was provided by all the agents.

Besides, some experiments were presented indicating that the LTI Rescue agents which implement the proposed approach performed better than the sample agents provided by the RoboCup Rescue platform for most of the maps. No statistical significance difference was observed in the Kobe4 map, this happened because the map is too large and there are few civilians and burning buildings, thus the rescue agents could not find them, therefore performing randomly.

As future work, we intend to improve the strategy presented in order to subdivide the agents into specific parts of the map depending on the number of fire spots and civilians identified. Moreover, we intend to implement rescue agents that perform their tasks totally decentralized, thus not requiring the existence of centers in the environment for the propagation of information.

Acknowledgement

André H. Pereira is supported by PIBIC/CNPq/USP. Jaime S. Sichman is partially supported by CNPq/Brazil.

References

- Boslaugh, S. and Watters, D. P. A. (2008). *Statistics in a nutshell*. O'Reilly & Associates, Inc., Sebastopol, CA, USA.
- Durfee, E. H. (1988). *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shinjou, A., and Shimada, S. (1999). Robocup rescue: Search and rescue in large-scale disasters as adomain for autonomous agents research. In *Proceedings of the IEEE international conference on systems, man, and cybernetics (SMC)*, volume 6, pages 739–743, Tokyo, Japan.
- Morimoto, T. (2002). How to develop a robocuprescue agent.
- Wooldridge, M. (2009). *An Introduction to Multiagent Systems*. John Wiley & Sons Ltd.

Um *Web Searcher Agent* Baseado em Regras Contextuais e na Colaboração entre Agentes

Daniela Maria Uez¹, João Luis Tavares da Silva¹

¹Departamento de Informática - Universidade de Caxias do Sul (UCS)
Rua Francisco Getúlio Vargas, 1130 - CEP 95070-560 - Caxias do Sul – RS – Brasil

dani.uez@gmail.com, jltsilva@ucs.br

Abstract. This paper presents a study to develop an Web searcher agent for semantic information retrieval on the Web. This agent uses semantic rules defined by an ontology to filter the search context. For this study was made a bibliographic search and the agent was developed to test the results. The bibliographic search provides the pillars to develop the agent. The agent was developed using the SemantiCore framework and Jena API. SemantiCore framework was used to develop the agents and its interactions. The Jena API enabled the reading and manipulation of the ontology used for semantic filtering. This study concludes that searching using semantic rules defined by ontologies is a viable alternative to improve web search results.

Resumo. Este documento apresenta um estudo para desenvolvimento de um agente que realizará a busca de informações na Web utilizando filtragem semântica. O agente descrito utiliza regras para filtragem dos resultados que são criadas com base em uma ontologia do contexto no qual deve ser realizada a busca. Este estudo foi desenvolvido através de uma pesquisa bibliográfica e da implementação do agente para teste dos resultados. A pesquisa bibliográfica forneceu as bases para o desenvolvimento do agente, que foi implementado utilizando o framework SemantiCore e a API Jena. O SemantiCore foi utilizado para o desenvolvimento dos agentes e das interações entre os agentes. A API Jena permitiu que fosse realizada a leitura e a manipulação da ontologia utilizada para filtragem semântica. Através deste estudo chegou-se a conclusão de que a busca baseada em regras semânticas definidas por ontologias é uma alternativa viável para melhoria dos resultados apresentados pelas ferramentas de busca atuais.

1. Introdução

As ferramentas de busca são *sites* especiais desenvolvidos para ajudar na recuperação de informações armazenadas em outros *sites* [GUIMARÃES 2002]. Quando um usuário faz uma solicitação de busca, a ferramenta compara as palavras-chave digitadas pelo usuário com as entradas do índice de *sites* em sua base de dados e depois classifica o resultado de acordo com algoritmos próprios de classificação. As ferramentas de busca não conseguem filtrar com eficiência as informações porque utilizam como método básico de busca a comparação sintática entre os termos e não tratam casos em que as palavras-chave têm mais de um significado nem as relações semânticas entre os termos. Por isso o resultado apresentado normalmente não é satisfatório. Uma busca simples retorna uma

quantidade muito grande de *sites*, sendo que muitos não tem relevância para o usuário ao mesmo tempo em que *sites* importantes podem ser ignorados devido aos processos de classificação que a ferramenta utiliza.

Muitas alternativas estão sendo pesquisadas para melhorar os resultados apresentados pelos *sites* de busca. O principal objetivo deste trabalho é realizar um estudo para definição de um sistema multiagentes que permita a recuperação de informações na *Internet* utilizando regras contextuais. Nessa abordagem, os vários agentes enviados na realização da busca possuem regras ligadas ao contexto do assunto a ser buscado. Os resultados encontrados individualmente pelos agentes serão filtrados através da combinação das regras que cada agente possui. Dessa forma, a busca levará em conta também a semântica dos termos envolvidos e o contexto para o qual a busca está sendo realizada. O resultado apresentado, portanto, tende a ser mais refinado e próximo das necessidades do usuário. O agente foi criado utilizando-se o *framework* Semanticore para manipulação dos agentes e a API Jena para manipulação e leitura da ontologia. Neste artigo serão apresentados os conceitos utilizados para criação do agente, a definição do agente e os resultados obtidos no estudo de caso.

2. Conceitos utilizados

2.1. Recuperação de Informações na *Internet* e *Web Semântica*

Os mecanismos de busca ou *search engines* são ferramentas criadas para facilitar a busca por informações armazenadas nas páginas da *Internet*. Essas ferramentas executam quatro passos básicos para atender as requisições dos usuários [NORVIG 2004]:

1. Enviam programas chamados *spiders* para percorrer os links da *Internet* em busca de páginas
2. Indexam as páginas encontradas pelos *spiders*
3. Quando há uma requisição de busca, varrem o índice em busca de páginas que contenham as palavras informadas, ordenando-as de acordo com critérios de classificação próprios de cada ferramenta
4. Mostram os resultados ordenados por ordem de importância, de acordo com a classificação utilizada pela ferramenta.

Para os mecanismos de busca é muito importante que os resultados apresentados nas páginas iniciais sejam relevantes. Contudo, é muito comum o retorno de *sites* que não possuem nenhuma relação com a busca do usuário. Isso ocorre porque os mecanismos utilizam-se da comparação sintática entre as palavras-chave digitadas pelo usuário e as entradas nos índices para determinar se uma página deve ou não ser apresentada. Esse tipo de análise não leva em conta os diversos significados que uma mesma palavra pode ter em linguagem natural e também não permite que o mecanismo identifique palavras sinônimas durante a busca.

Numa tentativa de melhorar os resultados apresentados, alguns mecanismos armazenam em seus índices informações léxicas ou linguísticas sobre as palavras ou realizam a busca automática por termos ou expressões relacionadas às palavras-chaves que foram utilizadas em buscas anteriores. A *Internet* atual, porém, não inclui em suas páginas informações que permitam aos mecanismos de busca interpretar corretamente o texto que estão contido nestas páginas. Para solucionar esse tipo de problema o W3C (*World Wide*

Web Consortium) idealizou a *Web Semântica*, que incluirá informações sobre os conceitos, as relações entre os conceitos e sobre a semântica dos conteúdos apresentados nas páginas da *Internet*, independente do formato que utilizam (som, texto, figura, vídeo, etc). Essa nova estruturação da *Web* permitirá aos sistemas computacionais ler, compreender e analisar os conceitos apresentados para, a partir destes, inferirem novos conceitos. Nesse novo ambiente, a tarefa de busca por conteúdo será facilitada pois as informações necessárias para que os mecanismos encontrem exatamente o que o usuário procura estarão na própria estrutura das páginas da *Internet*.

2.2. Agentes e Sistemas Multiagentes

Um agente é uma entidade humana ou artificial que está situada em um ambiente, pode percebê-lo e agir sobre ele. Os agentes tem objetivos específicos e raciocinam com base no conhecimento disponível e nas suas percepções sobre o ambiente para tomar a melhor decisão e atingir seus objetivos [BARROS 2002]. Segundo [WOOLDRIDGE 2002], os agentes tem como características básicas a reatividade (capacidade de responder às mudanças que ocorrem no ambiente), a proatividade (capacidade de executar ações para atingir um objetivo e não somente como resposta a eventos ocorridos) e sociabilidade (capacidade de agir de forma cooperada com outros agentes). O autor também atribui aos agentes outras propriedades, como a capacidade de se mover através dos ambientes (mobilidade), a capacidade de aprendizagem e de adaptação, a racionalidade, a benevolência e a veracidade.

Um sistema multiagentes (SMA) é um sistema composto por múltiplos agentes que interagem para resolução de um determinado problema. Os agentes em um SMA são autônomos, não dependendo de intervenção externa para existir, e podem ter graus diferentes de capacidades para resolver problemas. Assim, os agentes cooperam através da utilização de protocolos de interação social para resolver problemas que não podem ser resolvidos por um agente sozinho. Os agentes e sistemas multiagentes são muito utilizados no desenvolvimento de aplicações para *Web*.

2.3. Ontologias

As ontologias foram criadas para permitir a descrição formal dos termos, dos conceitos e das relações entre os conceitos utilizados em um domínio. Assim, mesmo agentes que não foram criados especificamente para trabalhar em conjunto poderão trocar informações e trabalhar cooperativamente sem que haja entendimento ambíguo dos termos empregados.

Uma ontologia pode ser definida como "uma especificação explícita de uma conceitualização compartilhada". [GRUBER]. Esta definição, segundo [GUIMARÃES 2002], mostra algumas características das ontologias: estas devem ser explícitas, formais e compartilhadas. Explícitas pois os conceitos e termos devem ser descritos de forma clara e explícita. Formal pois deve ser descrita utilizando-se linguagem formal que permita sua compreensão pelos sistemas de computador. Compartilhada pois deve descrever conceitos aceitos por um grupo de pessoas.

Basicamente as ontologias são compostas por

- Um conjunto de conceitos hierárquicos (taxonomias);
- Um conjunto de relacionamentos e de funções que podem existir entre os conceitos.

- Um conjunto de regras que sempre serão válidas no contexto da ontologia (axiomas);
- Um conjunto conhecimentos prévios inerentes ao domínio do conhecimento (instâncias).

O objetivo principal da *Web* semântica é explicitar o relacionamento entre os conceitos de diversos domínios de conhecimento e permitir que os agentes de software possam trabalhar sobre esses relacionamentos, compreendê-los e a partir deles inferir novos conceitos. As ontologias são um dos pilares da *Web* semântica pois fornecem um mecanismo eficiente para representação e compartilhamento de conhecimentos, através de um vocabulário que permite a descrição e a representação exata dos conceitos de um domínio. Esse vocabulário, por ser expresso em linguagem formal, não permite termos ambíguos e pode ser estendido e mapeado, permitindo que uma ontologia genérica seja readequada de forma a expressar um domínio específico de conhecimento e que essa ontologia seja expressa em várias línguas sem alterações de conceitualização.

3. Trabalhos Relacionados

Muitas aplicações utilizam agentes como forma de lidar com a complexidade da estrutura da *Web* e facilitar a busca por informações específicas na rede. [CISCON], por exemplo, propõe uma aplicação em três etapas com objetivo de organizar informações sobre o agronegócio do café. O sistema utiliza agentes para monitorar *sites* pré-selecionados e extrair deles as informações necessárias que serão armazenadas no sistema de banco de dados. Um outro agente é utilizado para disponibilizar essas informações para consulta dos usuários.

Os agentes são utilizados também por muitos mecanismos de busca para realizar a varredura da *Internet* de forma distribuída, permitindo que um número maior de páginas sejam indexados em um tempo menor. Outros mecanismos, chamados metabuscadores, utilizam os agentes para permitir que uma busca seja realizada simultaneamente em diversos mecanismos de busca. Ou seja, ao receber uma solicitação de busca de um usuário, o metabuscador irá utilizar agentes pra realizar essa mesma requisição em diversos outros mecanismos de busca. Dessa forma um número maior de páginas é incluído na consulta, já que o metabuscador não se restringe à estrutura de índices de um único mecanismo. A ferramenta *Copernic Agent* [Inc.] desenvolvida pela empresa *Copernic Inc* é um exemplo de metabuscador. O *Copernic* reúne os resultados colhidos pelos agentes em diversos mecanismos de busca, remove as entradas e apresenta os resultados restantes para o usuário.

4. Arquitetura Proposta

O *Web Searcher Agent* é um sistema multiagentes que tem por objetivo realizar uma busca na *Web* utilizando regras semânticas para filtragem dos resultados. Para o desenvolvimento do *Web Searcher* foram utilizados o *framework* SemantiCore e a ferramenta Jena. O SemantiCore facilita o desenvolvimento de sistemas multiagentes para a *Web* semântica, permitindo que os agentes manipulem ontologias com facilidade. O Jena permite leitura e a criação de ontologias e é utilizado pelo SemantiCore para prover as facilidades de manipulação de ontologias pelos agentes.

O *Web Searcher* foi desenvolvido utilizando dois tipos diferentes de agentes: o agente Gerente e o agente Buscador. O agente gerente é o responsável pelo gerenciamento

da busca e pela criação dos agentes buscadores. Os agentes buscadores executam a busca das informações solicitadas pelo gerente. Na Figura 1 pode ser visto o diagrama das classes utilizadas para implementação do *Web Searcher*.

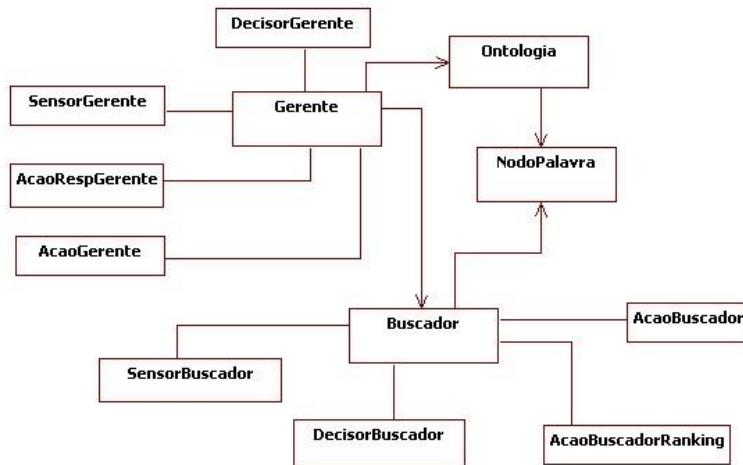


Figura 1. Arquitetura geral do WebSearcher com as principais classes representando agentes e suas capacidades

4.1. O Agente Gerente

O agente Gerente é responsável pelo gerenciamento da busca. O Gerente recebe a requisição de busca juntamente com o contexto ao qual a busca se refere. Cada contexto está associado a uma ontologia e cada termo da ontologia terá um peso de acordo com a relevância deste termo dentro da busca. Assim, termos diretamente relacionados aos termos informados pelo usuário da busca terão pesos maiores. Para cada termo encontrado na ontologia, o Gerente cria um agente Buscador. O Gerente também é responsável pelo controle da pontuação das páginas, somando as pontuações encontradas por cada Buscador. Por fim, quando os Buscadores finalizarem seu trabalho, o Gerente ordena as páginas encontradas e apresenta os resultados para o usuário. O funcionamento do Gerente pode ser descrito pela sequência de passos abaixo:

1. Recebe requisição de busca;
2. Lê a ontologia associada ao contexto da busca a ser realizada;
3. Para cada termo existente na ontologia, cria um agente de busca;
4. Enquanto houverem páginas a ser avaliadas:
 - (a) Recebe a página encontrada pelo agente de busca com sua respectiva pontuação;
 - (b) Repassa a página para ser avaliada pelos demais agentes no ambiente;
5. Soma as pontuações que os agentes atribuíram para a página;
6. Ordena a lista de páginas pela pontuação recebida;
7. Apresenta os resultados.

4.2. O Agente Buscador

Os agentes Buscadores são responsáveis pela varredura das páginas da *Web* em busca dos termos apresentados pela ontologia. Quando é criado, o Buscador recebe do Gerente uma

palavra da ontologia juntamente com o peso atribuído a essa palavra e uma lista de páginas a serem visitadas chamada de semente. O Buscador realiza basicamente duas funções: a busca pelas páginas e a valoração da página. A ação de busca das páginas é iniciada no momento em que o Buscador é criado. O algoritmo abaixo ilustra o funcionamento da ação de busca.

1. Recebe do Gerente a semente, a palavra que deve ser buscada e o peso atribuído a ela.
2. Para cada página:
 - (a) Dispara o *spider*;
 - (b) Determina a pontuação da página ;
 - (c) Envia a página e a pontuação encontrada para o Gerente;
3. Finaliza a execução das tarefas.

O Buscador envia para o Gerente as páginas que encontra com sua respectiva pontuação. A pontuação da página é calculada pela fórmula $k * p$, onde k representa a quantidade de vezes que a palavra aparece na página e p representa o peso da palavra. Cada Buscador procura na página somente a palavra que recebeu no momento de sua criação. Para que as páginas sejam corretamente filtradas, o Gerente envia cada página recebida para que os demais Buscadores pontuem a página de acordo com suas palavras. Assim, o Buscador é responsável também pela análise das páginas lidas pelos outros Buscadores e pelo cálculo da pontuação dessas páginas com base em sua palavra específica. A ação de valoração da página é conceitualmente simples, conforme o algoritmo abaixo:

1. Recebe a página a ser analisada;
2. Determina a pontuação da página;
3. Envia a página encontrada para o Gerente.

4.3. O *Spider*

O *Spider* é responsável pela busca e pela análise das páginas da *Web*. O *spider* lê a página a procura dos links existentes. Esses links são adicionados à lista de páginas a serem pesquisadas pelo Buscador. O *Spider* também é responsável pela análise que vai determinar a pontuação da página. O funcionamento do *Spider* pode ser ilustrado através da seguinte sequência de passos:

1. Recebe uma página a ser acessada;
2. Copia a página;
3. Procura pelos links existentes na página;
4. Calcula a pontuação da página;
5. Retorna os dados encontrados.

Para calcular a pontuação da página, o *Spider* procura por todas as ocorrências da palavra recebida pelo Buscador. Para cada ocorrência da palavra, o peso correspondente é somado à pontuação já calculada para aquela página, conforme o algoritmo abaixo:

1. A página inicia com pontuação zero;
2. Enquanto encontrar a palavra na página;
3. Soma o peso da palavra à pontuação da página;
4. Retorna pontuação calculada pela página.

4.4. O Filtro semântico

O filtro semântico é a parte principal do processo de busca. O filtro é realizado em dois passos

1. Leitura da ontologia associada ao contexto;
2. Cálculo da pontuação de cada página.

A leitura da ontologia é feita pelo agente Gerente. O Gerente utiliza a API do Jena para percorrer a ontologia e procurar pelas palavras solicitadas pelo usuário. O Jena fornece classes específicas para leitura das ontologias em formato OWL. Cada palavra encontrada na ontologia receberá um peso que determina a importância dessa palavra no contexto da busca. Por exemplo, seja a ontologia que apresentada na Figura 2, representando o contexto da astronomia:

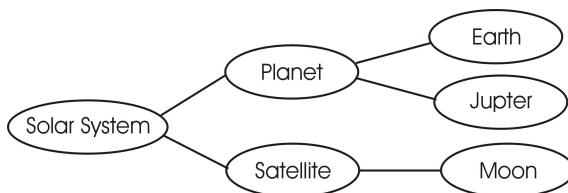


Figura 2. Representação gráfica de uma ontologia do contexto de astronomia.

Supondo que seja solicitada a busca pelas palavras "Earth" e "Solar System". O Gerente fará a leitura das palavras diretamente relacionadas com aquelas solicitadas pela busca ("Solar System", "Planet" e "Earth") terão peso 10. As demais palavras constantes na Ontologia ("Satellite", "Jupiter" e "Moon") terão peso 5. O Gerente criará um Buscador para cada uma das palavras retornadas pela ontologia e cada Buscador será responsável por visitar uma lista de *sites* inicial. Cada página visitada pelo Buscador receberá uma pontuação de acordo com a quantidade de vezes que a palavra aparece nessa página. Por exemplo, se o Buscador responsável pela busca da palavra "Earth", cujo peso é 10, encontrá-la 10 vezes em uma página, a pontuação referente à página será igual a 100. Essa página passará pela avaliação dos outros Buscadores. O Gerente somará todas as pontuações atribuídas por cada buscador e, no final, irá organizar as páginas de acordo com a pontuação conseguida por cada página. O diagrama de sequências apresentado na Figura 3 mostra o fluxo de mensagens entre os agentes.

5. Estudo de Caso

Para testar o mecanismo de busca foram feitos testes utilizando-se uma ontologia do contexto de astronomia. Essa ontologia foi criada extendendo-se a ontologia desenvolvida pelo Departamento de Astronomia da Universidade de Maryland [umd]. De um modo geral, as ontologias disponíveis hoje são muito genéricas para serem utilizadas com sucesso no filtro semântico. Por isso optou-se por alterar a ontologia já existente, incluindo nela termos específicos do contexto de busca. Essa ontologia, descrita em linguagem OWL, relaciona termos do contexto de astronomia. A Figura 4 apresenta uma representação gráfica da ontologia utilizada.

Para criação da semente foram utilizados endereços de *sites* diversos. A semente conta com uma lista de 125 *sites* apresentados como resposta às pesquisas de termos

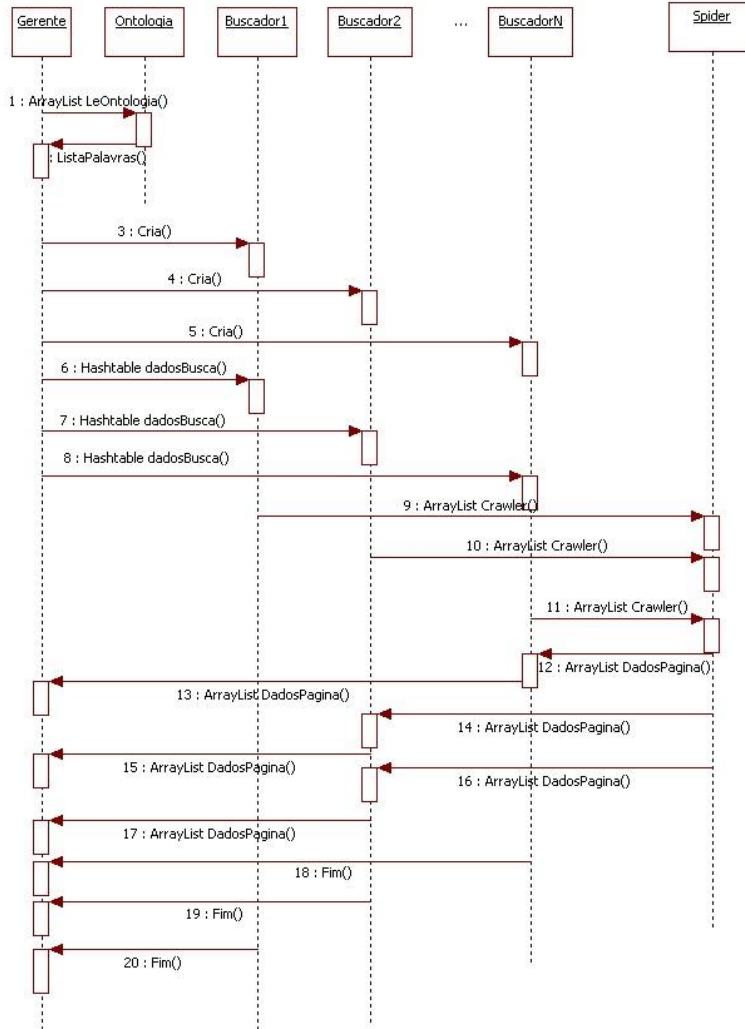


Figura 3. Diagrama de Sequência de Eventos apresentando o fluxo de mensagens trocadas entre os agentes.

constantes na ontologia pelos mecanismos de busca tradicionais. *Sites* que continham arquivos PDF, Flash, pps, doc ou outros tipos de arquivo diferentes de páginas da *Internet* (HTML) foram ignorados na busca.

Uma requisição de busca pelas palavras Terra e Lua foi feita ao *Web Searcher*. O agente Gerente realizou a leitura da ontologia e retornou as seguintes palavras: galaxia, via-lactea, estrela, sistema planetario, cometa, planeta, satelite, asteroides, halley, planeta-anao, natural, plutao, lua, io, europa, mercurio, venus, terra, marte, juptier, saturno, urano, netuno, sistema solar, sol. Analisando a ontologia as palavras que possuem maior relação com os termos requisitados para a busca, e que portanto terão peso 10 durante a busca, são: sistema planetario, planeta, terra, satelite, satelite natural, lua. As demais palavras terão peso 5.

Para cada uma dessas palavras o mecanismo de busca criou um agente Buscador. Cada Buscador é responsável pela busca de um termo em um subconjunto dos endereços constantes no arquivo semente. O primeiro agente de busca, por exemplo, analisará um

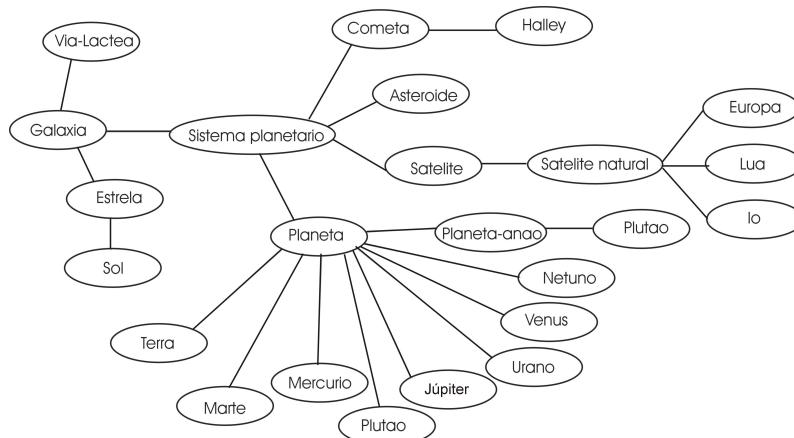


Figura 4. Representação gráfica da ontologia exemplo utilizada. Nesta representação pode-se ver a relação entre os termos da ontologia.

subconjunto de cinco endereços da semente em busca da palavra galaxia. Cada *site* analisado pelo primeiro Buscador é enviado para que os demais buscadores procurem por suas palavras. Ao final, a pontuação do *site* é calculada com base na quantidade de vezes que cada uma das palavras da ontologia foi encontrada naquela página.

Após essa pesquisa os resultados foram ordenados e mostrados para o usuário. Os *sites* de melhor pontuação foram:

1. Fases da Lua: <http://astro.if.ufrgs.br/lua/lua.htm>;
2. A Lua é uma estrela? : <http://br.answers.yahoo.com/question/index?qid=20090127075344AA2KnCg>;
3. Eclipse: <http://pt.wikipedia.org/wiki/Eclipse>;
4. Sistema Terra-Lua: <http://zenite.nu/05/3-ter.php>;
5. Influência da Lua nas marés da Terra: <http://curiofisica.com.br/influencia-da-lua-nas-mares-da-terra>;
6. A Terra, a Lua, o Sol: os nossos relógios: <http://calendario.incubadora.fapesp.br/portal/textos/professor/ptexto05> ;
7. Terra e Lua? Luli e Luciana (letra) : <http://letras.terra.com.br/luli-lucina/376840>;
8. Alguns dados sobre o planeta Terra: <http://cfh.ufsc.br/planetar/textos/terra-bege.htm>;
9. Lua de Mel ? Terra ? Lua de Mel: http://mulher.terra.com.br/noivas/interna/0,,OI524523-EI48_83,00.html;
10. Observação do Céu: <http://nautilus.fis.uc.pt/astro/hu/obser/corpo.html>.

Como pode ser visto, os resultados apresentados pelo *Web Searcher Agent*, em sua maioria, são resultados relevantes para a busca. Neste exemplo, das 10 páginas apresentadas pelo agente, 8 contém informações relevantes no contexto da astronomia. Para efeito de comparação de resultados, a mesma pesquisa foi feita no mecanismo de busca Google. Dos 8 principais resultados apresentados pelo mecanismo, cinco são resultados relevantes para o contexto de astronomia.

6. Considerações Finais

O estudo realizado para desenvolvimento do agente de busca semântica apresentado neste trabalho mostrou que a filtragem semântica é uma alternativa viável para melhorar os

resultados apresentados pelos mecanismos de busca na *Web*. O trabalho foi realizado em duas fases, sendo a primeira uma fase de pesquisa para definição dos termos e conceitos utilizados. A pesquisa forneceu a base para que o agente fosse desenvolvido corretamente. A segunda fase foi a implementação do agente e a realização de testes para determinar a relevância dos resultados apresentados.

O *Web Searcher Agent* descrito apresentou resultados satisfatórios na busca de informações, apesar de sua simplicidade de implementação. A filtragem utilizando ontologias facilita a recuperação de informações relevantes pois permite que as páginas sejam filtradas por uma série de termos ligados ao contexto da busca e não somente pelos termos informados pelo usuário. Porém, essa abordagem requer que a ontologia utilizada seja específica do contexto a ser pesquisado e especifique as entidades pertencentes a este conceito. Para permitir a busca contextual, o mecanismo deve especificar o domínio de conhecimento onde a busca será executada.

O agente realiza a busca direto na *Web*. Dessa forma para verificar se uma página é relevante ou não o agente precisa fazer a leitura da página e analisar os termos encontrados. Essa abordagem, apesar de ser de simples implementação, torna a busca lenta e não permite que um número grande de páginas seja visitado. Uma alternativa seria a criação de índices como aqueles utilizados pelos mecanismos de busca atuais. O agente de busca contextual pode ser utilizado para realizar a busca e a filtragem neste índices, o que tornaria a busca mais rápida e abrangente. Outras melhorias que podem ser feitas no agente incluem:

- Definição de ontologias específicas para serem utilizadas juntamente com o agente de busca. Essas ontologias devem apresentar detalhes referentes ao contexto, como entidades específicas, ao invés de apresentar somente as classes.
- A abordagem atual de busca é centralizada, utilizando um agente Gerente para organizar a troca de informações entre os agentes Buscadores. O uso de uma abordagem distribuída para implementação dos agentes de busca permitiria que a busca fosse feita com mais rapidez. Atuando de forma distribuída, os agentes Buscadores precisariam trocar informações entre si para que os *sites* fossem corretamente filtrados;
- Cada agente de busca pode ser disparado tendo conhecimento completo sobre a ontologia e não somente de uma palavra. Assim cada agente teria a capacidade de pontuar sozinho as páginas visitadas, não precisando solicitar que outros agentes analisem a página. Nessa abordagem torna-se necessária a definição de regras específicas que determinarão como os agentes irão coordenar as páginas encontradas, como será feita a remoção de resultados duplicados e a ordenação dos resultados que serão apresentados.

Um dos principais objetivos de Berners-Lee [BERNERS-LEE 2001] com a *Web* semântica é tornar o conteúdo da *Web* comprehensível pela máquina, permitindo que agentes e aplicações acessem uma enorme gama de recursos heterogêneos buscando automatizar suas atividades com o mínimo de interação ou interpretação humana. Por enquanto, os esforços tem se concentrado em arquiteturas que buscam uma homogeneização dos recursos através de XML, RDF, RDFS, OWL, entre outras representações. Possivelmente a evolução destes padrões realmente direcione designs arquiteturais sofisticados da *Web*, mas é possível também que soluções surpreendentemente simples sejam alternativas mais

elegantes para a "significação" na Web, preconizada por Berners-Lee. Embora o presente trabalho não pretenda em momento algum solucionar estes problemas, procuramos estabelecer um estudo exploratório sobre o uso de uma abordagem multiagentes em uma espécie de trabalho colaborativo inferencial baseado em contextos controlados.

Referências

- BARROS, F. (2002). A inteligência artificial sob o olhar dos agentes inteligentes. Master's thesis, Universidade Federal de Pernambuco.
- BERNERS-LEE, Tim; HENDLER, J. L. O. (2001). The semantic web. *Scientific American Magazine*.
- CISCON, Leonardo Aparecido, A. R. M. Desenvolvimento de um sistema web de busca inteligente para suporte à tomada de decisões no agronegócio do café. In *Convibra - Congresso Virtual Brasileiro de Administração*.
- GRUBER, T. R. T. Towards principles for the design of ontologies used for knowledge sharing. In *International Journal of Human and Computer Studies*.
- GUIMARÃES, F. J. Z. (2002). Utilização de ontologias no domínio b2c. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro.
- HÜBNER, Jomi F.; SICHMAN, J. S. Organização de sistemas multiagentes. In *III Jornada de Mini-Cursos de Inteligência Artificial*.
- Inc., C. Copernic agent family. Disponível em <http://www.copernic.com/>.
- JENA. Jena ? a semantic web framework for java. Disponível em <http://jena.sourceforge.net/index.html>.
- NORVIG, P. (2004). *Computer Science: Reflections on the field, Reflections from the field*, chapter Internet Searching. Disponível em <http://www.norvig.com/InternetSearching.pdf>.
- RIBEIRO, Marcelo Blois, E. M. d. S. C. R. (2007). Using agents and ontologies for application development on the semantic web. *Journal of the Brazilian Computer Society*.
- RIBEIRO, Marcelo Blois, E. M. L. A. P. (2006). Semanticore 2006 ? permitindo o desenvolvimento de aplicações baseadas em agentes na web semântica. *Second Workshop on Software Engineering for Agent-Oriented Systems*.
- WOOLDRIDGE, M. (2002). *An Introduction to Multiagent Systems*. John Wiley e Sons Ltd, 15th edition.

Framework para Re-engenharia do Ambiente AMPLIA

Paulo Ricardo M. Barros¹, Elton Erhardt¹, Marta R. Bez², Sílvio Cazella¹, Cecilia D. Flores¹

¹Universidade Federal de Ciências da Saúde de Porto Alegre, Porto Alegre – RS

²Universidade Feevale, Novo Hamburgo – RS

pbarros1979@gmail.com, eerhardt@terra.com.br, martabez@feevale.br,
silvioc@ufcspa.edu.br, dflores@ufcspa.edu.br,

Abstract. *The AMPLIA - Probabilistic Multiagent Learning Environment is an educational tool aimed at creating an additional resource for assistance to medical education, supporting the development of diagnostic reasoning. The first version has some limitations when it comes to access only local network (LAN) as a function of applied technologies, and provide students with an unfriendly interface. The proposed framework aims to enable the use of Internet resources, and to propose an interface for a computer game, we developed a new technology framework that will enable the provision of knowledge to other university*

Resumo. *O AMPLIA – Ambiente Multiagente Probabilístico de Aprendizagem, é uma ferramenta pedagógica com o objetivo de tornar-se um recurso adicional para o auxílio à Educação Médica, apoiando o desenvolvimento do raciocínio diagnóstico. A primeira versão possui algumas limitações no que tange ao acesso apenas em rede local (LAN) em função das tecnologias aplicadas, além de oferecer ao aluno uma interface pouco amigável. A proposta do framework tem o intuito de possibilitar o uso de recursos da internet, assim como propor uma interface de um jogo computacional, foi desenvolvida uma nova estrutura tecnológica que possibilitará a disponibilização do conhecimento às demais universidades.*

1. Introdução

Atualmente, existem inúmeros recursos pedagógicos à disposição de educadores e educandos, através do uso de ambientes virtuais de aprendizagem colaborativa. Cada ferramenta tem como foco uma determinada área do conhecimento, em virtude de adequar-se melhor ao resultado obtido.

Um ambiente virtual de aprendizagem colaborativo tem como principal objetivo disponibilizar um espaço de cooperação para construção de conhecimento, com o intuito de desenvolver atividades educativas. Atualmente, os principais recursos de informática utilizados para desenvolver este raciocínio lógico são as listas de discussão, chats, teleconferências (FLORES, 2005).

Na formação da área médica isto não é diferente, contudo, nesta, o conhecimento envolve domínios incertos, onde o aluno deve construir um raciocínio lógico traçado através de variáveis, sintomas apresentados pelo paciente, a fim de definir um possível diagnóstico da doença apresentada. Atualmente, os principais

recursos de informática utilizados para desenvolver este raciocínio lógico são as listas de discussão, chats e teleconferências (FLORES, 2005).

Com o objetivo de criar um recurso adicional para auxílio à educação médica, apoiando o desenvolvimento do raciocínio diagnóstico, desenvolveu o AMPLIA. Este é um ambiente aberto, em que o aluno constrói um modelo gráfico da representação de sua hipótese, que leva ao diagnóstico para um dado caso clínico.

O AMPLIA é um ambiente multiagente constituído por três tipos de agentes cognitivos (Agente APRENDIZ, Agente MEDIADOR e Agente de DOMÍNIO). Estes mantêm uma comunicação com um servidor (ComServer) (FLORES, 2005). Esta estrutura vale-se também do uso de uma interface de comunicação com um editor de redes bayesianas – o SEAMED (FLORES, 2002). Os agentes fazem uso dos recursos de comunicação em rede, viabilizados através de uma biblioteca denominada FACIL (GLUZ, 2002). Esta biblioteca foi desenvolvida baseada nos padrões FIPA-ACL (comitês técnicos responsáveis pela elaboração dos padrões para uma linguagem de comunicação entre agentes) (GLUZ, 2005), sendo a responsável pela troca de mensagem entre os agentes.

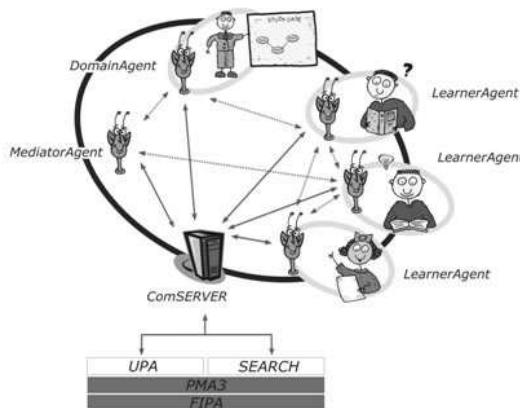


Figura 1 – Estrutura multiagente do Amplia (FLORES,2005)

Nesta estrutura (Figura 1), o Agente APRENDIZ é responsável por enviar informações geradas pelo aluno, o qual desenvolve uma rede bayesiana através do seu raciocínio lógico por meio do caso clínico apresentado. Já o Agente de DOMÍNIO tem a função de comparar a rede construída pelo aluno com a rede construída pelo especialista, que deve identificar os prováveis conflitos. O resultado gerado é então encaminhado para o Agente MEDIADOR, que será responsável pela seleção das estratégias pedagógicas mais convenientes para auxílio ao correto diagnóstico (FLORES, 2005).

Esta estrutura está limitada ao uso em uma rede local em função das tecnologias aplicadas. Contudo, a proposta é a sua re-engenharia para uso de tecnologias que comportem os recursos desejados, para o novo ambiente. Com isto, aumenta-se a abrangência e alcance, possibilitando expandir o conhecimento às demais universidades e encurtando distâncias, permitindo a criação de uma grande rede de repositórios com diagnósticos e casos clínicos à disposição de acadêmicos da área da saúde.

Outra grande limitação no ambiente diz respeito aos agentes APRENDIZ e de DOMÍNIO, que fazem uso de recursos diretamente ligados à construção e avaliação de

das Redes Bayesianas. Partindo de um relato textual de um caso clínico, o educando constrói seu raciocínio diretamente através da confecção da estrutura topográfica da rede, criando as possibilidades diagnósticas, o mesmo processo é realizado pelo especialista do domínio para construção da rede de referência, este processo exige um conhecimento prévio de diagramas de relações causais de Redes Bayesianas tanto do especialista quanto do educando. A necessidade deste tipo de conhecimento técnico acaba por ser um ponto de dificuldade na sua utilização, uma vez que o público alvo é formado por profissionais da área da saúde, não habituados a representações gráficas como grafos acíclicos orientados.

Nas seções seguintes são apresentadas as tecnologias adotadas na nova versão do ambiente AMPLIA e que, a partir de agora denominar-se-á SimDeCS (**S**imuladores **I**nteligentes para a **T**omada de **D**ecisão em **C**uidados de **S**aúde), o protótipo desenvolvido e as conclusões.

2. Alterações necessárias

A nova versão do AMPLIA, denominada SimDeCS, deve manter quase todas as funcionalidades e estruturas já empregadas no projeto original; porém, é desejado que a nova estrutura utilize ferramentas de desenvolvimento livre que comportem os requisitos e facilitem a interação tanto do educando com o ambiente, quanto do especialista na modelagem dos casos clínicos. Para isto propõem-se a utilização de uma linguagem específica de domínio própria para a construção de casos clínicos (VR-MED)(MOSSMANN, 2010), e a substituição da linguagem de programação, que no sistema original foi DELPHI, para JAVA (SUN, 2009); com intuito de facilitar a comunicação através da web, já que esta linguagem traz um aporte de recursos técnicos que comportam todos os requisitos mencionados.

Java é uma linguagem de alto nível, com sintaxe extremamente similar a de outras linguagens conhecidas como C++, muitas das suas características são herdadas de outras linguagens, como Smalltalk e Modula-3. É antes de tudo uma linguagem simples, fortemente tipada, independente de arquitetura e ambiente a ser empregados, robusta, segura, extensível, bem estruturada e distribuída sob licença de Software Livre, sob os termos da GNU *General Public License* (GPL).

O sistema AMPLIA, apesar de possuir grande volume de publicações, teses, dissertações, trabalhos de conclusão e “*papers*”, carece de documentação técnica. Outro fator importante a ser destacado é o fato de ter sido desenvolvido em partes, por “muitas mãos”, sem uma padronização.

A seguir são apresentadas algumas tecnologias e estudos detalhados de recursos sugeridos para a seqüência do projeto, como: Framework Hibernate (utilizado para persistência e comunicação com o banco de dados), IDE'S para desenvolvimento (ferramenta utilizada para codificação do software) e o uso do framework Jade, para a comunicação entre os agentes.

3. Framework de comunicação com banco de dados

Uma das características desejáveis para o novo sistema refere-se ao fato de ser uma aplicação independente do banco de dados, justamente por não se saber qual banco será

utilizado ao final do projeto. Desta forma, não poderá haver regras do negócio aplicadas ao banco, fato que restringiria o ambiente de forma expressiva; uma das soluções encontradas foi trabalhar na camada de persistência com um Framework que facilite e abstraia facilmente a camada da aplicação da camada do banco de dados.

O projeto Hibernate é ambicioso e pretende ser uma solução completa para o problema de gerenciamento de dados persistentes com JAVA. Ele cria uma interação do aplicativo com um banco de dados relacional, deixando o desenvolvedor livre para se concentrar nas regras de negócio (BAUER E KING, 2005).

O Mapeamento Objeto/Relacional (ORM) é uma solução automatizada para aproximar os dois contextos: orientado a objetos e o relacional. O Hibernate é um framework de desenvolvimento da camada de persistência totalmente implementado em Java que adota a abordagem ORM. Tem como objetivo proporcionar uma elaboração de software, orientado a objetos para uma camada do software considerada crítica a qualquer sistema que necessite de uma persistência (FERREIRA, 2006).

O principal diferencial de uma aplicação que utiliza os métodos tradicionais de comunicação através de uma API JDBC (JDBC, 2009) e com o uso do Framework Hibernate, diz respeito à produtividade, manutenção e à independência do banco de dados, pois este atua diretamente entre a aplicação e o banco de dados (Figura 2).

O banco de dados dará suporte a todos os agentes que compõem o sistema, em virtude de concentrar todas as informações das redes bayesianas e os casos clínicos reais utilizados para avaliação qualitativa e quantitativa das redes.

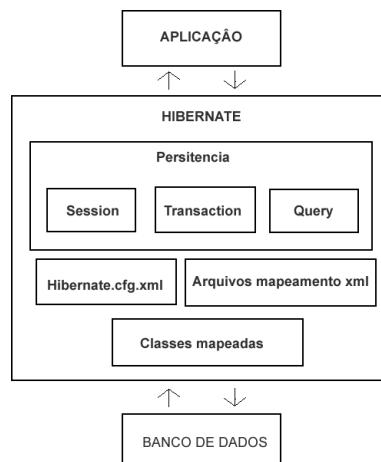


Figura 2 – Estrutura básica do funcionamento do Hibernate

O uso do Hibernate garante uma maior produtividade e menor manutenção, pois possibilita menor número de códigos escritos em função de não ter uma preocupação do programador quanto à persistência destes dados. Por consequência, sua manutenção é facilitada e a possibilidade de erros reduzida.

Com o uso desta tecnologia, o programador pode trabalhar com o banco de dados com os conceitos de orientação a objetos (Herança e Polimorfismo), mesmo sendo um banco de dados puramente relacional.

Para realizar a consulta aos dados, o HIBERNATE oferece como recurso uma linguagem própria para consulta o HQL, HIBERNATE Query Language. O HQL é uma linguagem de consulta muito semelhante ao SQL, porém, quando efetuada uma consulta

HQL, está sendo realizada uma pesquisa na classe correspondente à tabela a ser pesquisada, e desta maneira, o HIBERNATE se encarrega de traduzir e alimentar a classe com os dados pesquisados.

A quebra de paradigma talvez seja o ponto culminante para o uso desta tecnologia, pois o programador tem que alterar a forma tradicional de raciocínio, quando se trata de acesso a dados em um banco de dados. Desta maneira, não acessa mais diretamente as tabelas, o ingresso ocorre apenas a Classe e, por conseguinte, o HIBERNATE se encarrega de gerenciar e acessar a respectiva tabela do Banco de dados.

4. IDE para desenvolvimento

Partindo do princípio do qual a linguagem para desenvolvimento, JAVA, já havia sido definida pelo grupo de pesquisa, houve a necessidade de escolher uma IDE de desenvolvimento. Esta escolha foi baseada nas facilidades e no perfeito casamento entre a linguagem e os Frameworks que serão utilizados.

As ferramentas Netbeans e Eclipse são consideradas, em muitos fóruns e artigos publicados, ferramentas muito procuradas dentre os evangelistas da linguagem Java, não sendo as únicas, mas sim as principais.

Dentre as ferramentas escolhidas existem algumas particularidades diferenciais que podem facilitar e até mesmo definir o sucesso do desenvolvimento. Para sua escolha, levaram-se em consideração alguns aspectos que seriam de vital importância para que o desenvolvimento fosse executado da melhor forma possível.

A escolha da mesma foi baseada nos benefícios encontrados nas ferramentas nos quesitos de maior interesse para o futuro desenvolvimento, portanto, aqui serão apresentados os principais benefícios da ferramenta escolhida para o desenvolvimento do agente.

O NetBeans possui suporte para execução em ambiente Windows, Linux e MacOS. Sua configuração é extremamente simples, sendo facilmente instalado e executado. Possui uma ampla documentação, em diversos idiomas, inclusive com suporte direto.

Como destaque em seus recursos para desenvolvimento, faz-se importante citar seu excelente editor para desenvolvimento de aplicações visuais - Swing ou AWT, como também para aplicações web (JSP, Servlets). Conta com uma interface rica em recursos e de fácil utilização, oferecendo componentes de interface, que podem ser “arrastados para o Form”, posteriormente configurados com uma interface gráfica e utilitários que facilitam e agilizam o desenvolvimento, seguindo o mesmo padrão para uso de ferramentas pagas muito populares, como o Visual Estúdio da Microsoft.

A ferramenta ainda oferece um recurso interessante, que vale ser destacado, o gerenciador de módulos *plug-ins*. Um recurso que permite o desenvolvedor realizar a atualização, alteração ou exclusão dos módulos integrados ao ambiente de desenvolvimento de forma dinâmica, mantendo a consistência, dinamismo e compatibilidade do ambiente de desenvolvimento, com isto garantindo a estabilidade do projeto como um todo.

Através do gerenciador de módulos, a ferramenta realiza uma conexão com o centro de atualizações do grupo NetBeans.org e executa uma verificação da existência de atualização em algum módulo previamente instalado na ferramenta do usuário. Caso alguma alteração de versão seja detectada ou um novo módulo seja inserido no repositório, o desenvolvedor é alertado e deve marcar a opção de atualização ou não. Dessa forma, caso selecionada a opção de atualização, a ferramenta se encarrega de atualizar a versão do módulo automaticamente para o usuário, realizando o download e instalação do mesmo (CARVALHO; BATISTA; ULRICH, 2007 apud SEVERO, 2009).

A interface com o usuário do NetBeans é muito semelhante ao do Eclipse, porém, nativamente ele oferece uma gama muito maior de recursos para o desenvolvimento em Java, como conectividade e manipulação de servidores web e de Data Bases. O uso de um bom editor gráfico para desenvolvimento de interfaces facilita e agiliza o desenvolvimento, eliminando perda de tempo com trabalho extremamente braçal e mecânico.

Um detalhe interessante observado foi a facilidade de integração e uso do plug-in do Hibernate, completando com o recurso de Wizard, oferecendo uma excelente ferramenta de configuração e de engenharia reversa para extração das tabelas e definições do banco de dados

5. Framework de comunicação JADE

Um ponto crucial e de extrema relevância para o desenvolvimento, servindo como motivação para a proposta de reestruturação do ambiente AMPLIA, é a comunicação entre agentes com o uso de protocolos através da internet. No projeto original esta comunicação era limitada à rede local, devido a limitações tecnológicas da biblioteca FACIL, citada anteriormente.

Para corrigir esta limitação, é sugerido o uso da biblioteca JADE (*JAVA Agent DEvelopment framework*), um framework de comunicação por troca de mensagens para agentes. Jade propõe ser uma solução completa para comunicação simplificando o desenvolvimento e fornecendo um framework completo de métodos e ferramentas que trata da comunicação, monitoração e execução de atividades entre os agentes. A biblioteca foi desenvolvida na universidade de Parma na Itália, é distribuída de forma open source, sob licença LGPL (*Lesser General Public License*).

Sistemas Multiagentes são sistemas constituídos de múltiplos agentes que interagem ou trabalham em conjunto de forma a realizar um determinado conjunto de tarefas ou objetivos. Esses objetivos podem ser comuns a todos os agentes ou não. Os agentes dentro de um sistema multiagente podem ser heterogêneos ou homogêneos, colaborativos ou competitivos. Ou seja, a definição dos tipos de agentes depende da finalidade da aplicação que o sistema multiagente está inserido (SILVA, 2003).

A JADE foi escrita em JAVA devido a características particulares da linguagem, particularmente pela programação orientada a objeto em ambientes distribuídos heterogêneos. Foram desenvolvidos tanto pacotes JAVA com funcionalidades prontas para uso, quanto interfaces abstratas para se adaptar de acordo com a funcionalidade da aplicação de agentes (BELLIFEMINE 2003, apud SILVA 2003).

A biblioteca JADE cumpre todas as especificações e requisitos desejáveis para a substituição da biblioteca FACIL, sem que haja perda de funcionalidades exigidas para a comunicação entre os agentes do ambiente AMPLIA, uma vez que a biblioteca é neutra na definição do agente, dando toda e qualquer liberdade para sua definição.

Seu uso facilita o desenvolvimento e a implementação do sistema multiagente, garantindo um padrão de interoperabilidade entre o sistema Amplia, onde toda a sua comunicação é feita por troca de mensagens, garantindo uma independência entre os agentes.

Outro requisito importante na pesquisa, foi a sua fácil implementação e utilização dentro da IDE de desenvolvimento, facilitando seu uso e exploração de recursos, uma vez que os testes e implementações são facilitados pela sua vasta documentação e simplicidade de implementação, e que podem ser distribuídos por estações e controlados remotamente através de sua ferramenta de *debugging*.

6. VR-MED

O SimDeCS é um ambiente de aprendizado multiagente na área da saúde. Sua utilização parte da formulação de casos de variáveis graus de complexidade pelo tutor (educandos) através de ambiente web, trabalhando com a linguagem de domínio específico (DSL) de alto nível VR-MED.

A DSL VR-MED (MOSSMANN, 2010) foi concebida para que programadores e projetistas, apoiados por uma notação própria e simples, especifiquem características do caso de estudo em questão. Essa notação procura representar as características presentes do domínio nos casos clínicos da área da saúde e, além disso, prover o suporte para a execução destes, tal como um jogo de computador (Serious Game).

Subjacente à linguagem, as situações clínicas são expressas na forma unitária de redes bayesianas múltiplas seccionadas (MSBN). Durante o processo de modelação da situação de estudo por parte do tutor, a linguagem VR-MED provê a interface necessária para a escolha das redes bayesianas (BN) existentes no repositório, a vinculação dessas redes com os personagens do jogo e a vinculação entre diferentes BN.

7. Protótipo

Com a utilização das tecnologias mencionadas, pode-se desenvolver um novo protótipo do ambiente sob uma nova estrutura tecnológica (Figura 4).

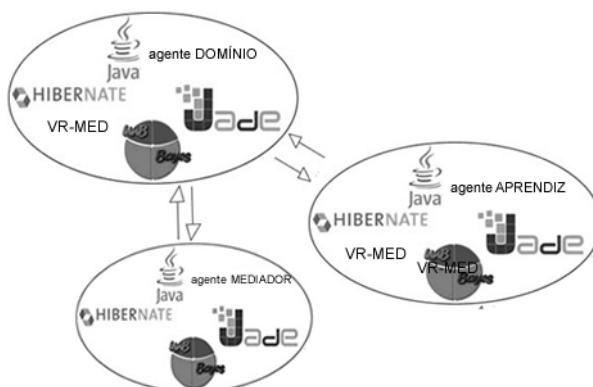


Figura 4 – Estrutura tecnológica.

A seguir serão apresentadas algumas interfaces desenvolvidas com as tecnologias mencionadas.

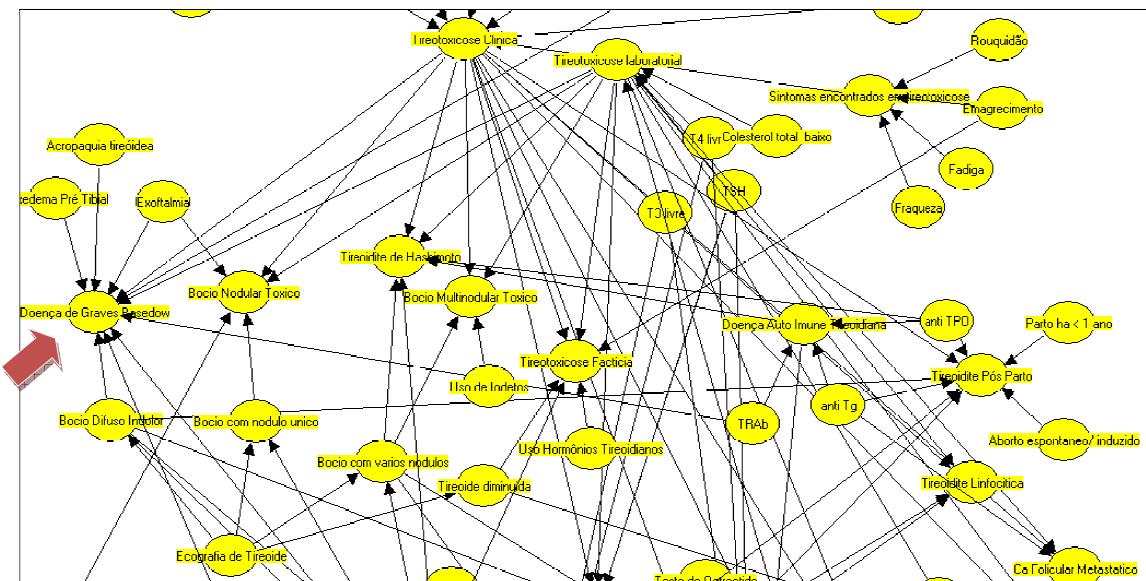


Figura 5 – Doenças da tireóide

A rede bayesiana desenvolvida pelo especialista (Figura 5) e a base de dados com casos reais, formam o conhecimento ao qual o agente de DOMÍNIO fará uso. Desta forma, este agente é responsável pela avaliação da rede bayesiana desenvolvida pelo aluno. Esta avaliação é feita através de verificações qualitativas e quantitativas da rede gerada pelo aluno.

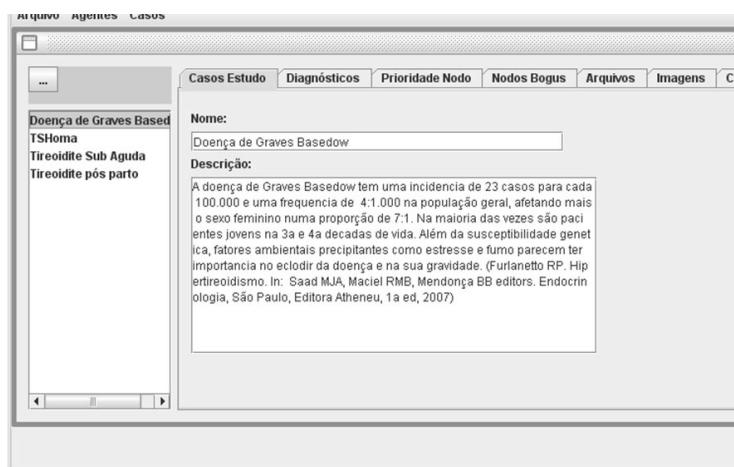


Figura 6 – Dados gerais - interface de seleção do caso de estudo

A tela de Dados Gerais (Figura 6) apresenta as características descritivas do caso, bem como o local onde se encontra a rede gerada. Através dela é possível visualizar o nome do caso e uma breve descrição. No exemplo abaixo, apresenta-se a descrição do diagnóstico *Doença de Graves Basedow*, assinalado na Figura 5. Estes campos são puramente descritivos, servindo apenas como informativo para o especialista e para o aluno.

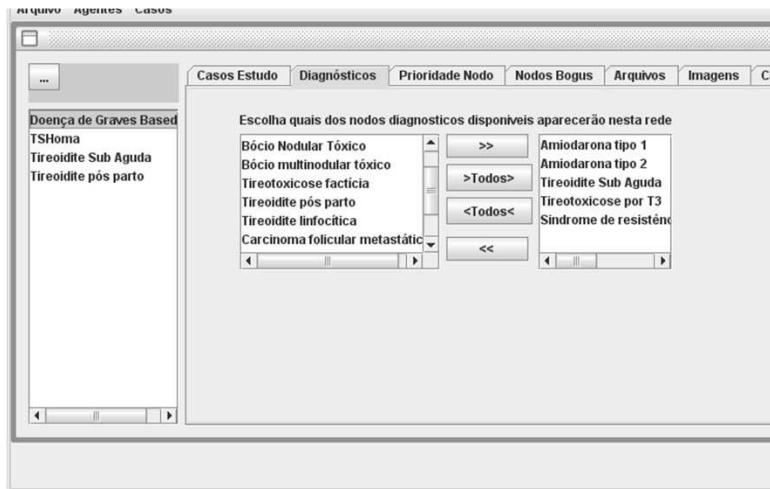


Figura 7 – Diagnosticos possíveis - interface de seleção de diagnósticos

A tela de diagnósticosPossíveis (Figura 7) apresenta os diagnósticos que estarão disponíveis para o aluno que irá modelar sua rede bayesiana. Através desta tela o especialista poderá definir quais os diagnósticos possíveis para um determinado caso de estudo.

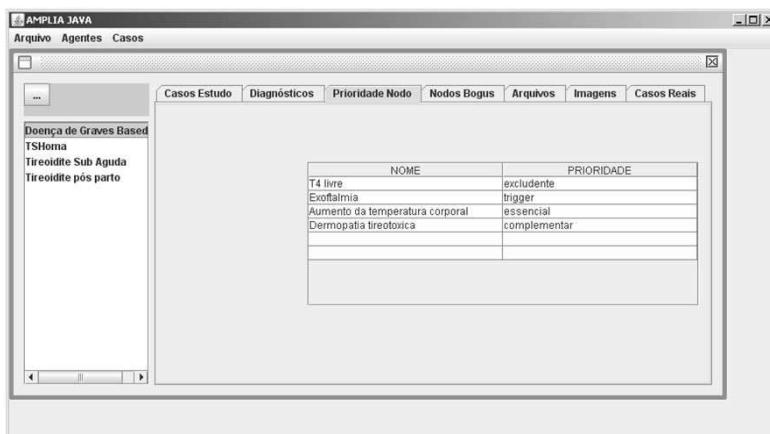


Figura 8 – Prioridade nodo - interface de seleção e definição de prioridade dos nodos

Através da interface Prioridade Nodo (Figura 8), serão definidas as propriedades que assumirá o nodo na rede, como complementar, essencial, excludente ou *trigger*.

No Agente de DOMÍNIO é realizada a modelagem do problema (casos clínico) por um especialista de domínio (médico, cirurgião dentista, agente de saúde, entre outros.) através da DSL VR-MED utilizando uma notação visual na forma de um diagrama o qual dispensa profundos conhecimentos da área de informática por parte do especialista (Figura 9). Neste diagrama é possível especificar os detalhes do caso clínico em questão, assim como os personagens (pacientes, médicos, familiares) que participam do caso.

A partir deste diagrama originam-se Redes Bayesianas Múltiplas Seccionadas as quais constituem um repositório a ser utilizado pelos demais agentes. Este repositório, quando da modelagem do caso de estudo, poderá originar uma ou mais BN (um ou mais problemas) para um mesmo personagem ou para diferentes personagens do mesmo

exercício proposto pelo tutor. A rede modelada previamente servirá para balizar as consultas do Agente MEDIADOR durante a execução do exercício pelo educando e, baseado nessa comparação, disparar as estratégias pedagógicas.

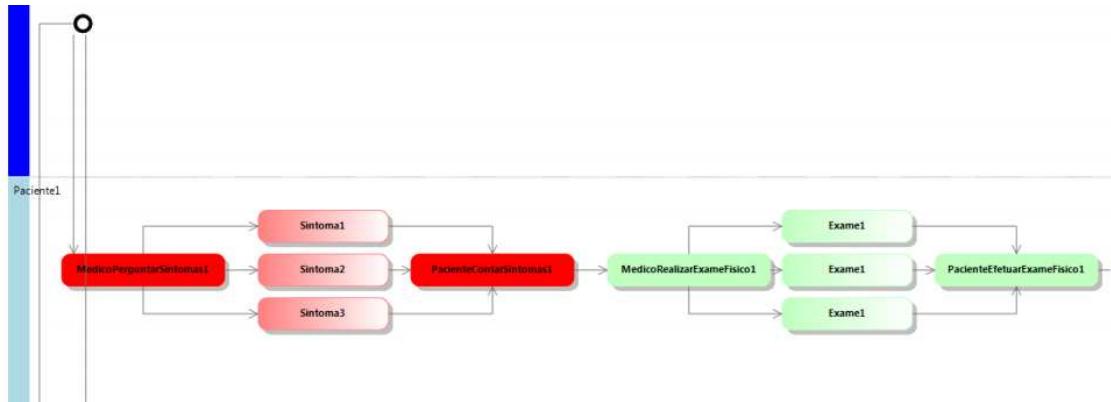


Figura 9 - Diagrama VR-MED utilizado pelo Agente de DOMÍNIO

O Agente APRENDIZ é representado através de um jogo computacional (Serious Game) em que, na perspectiva do aluno, há um caso representado no ambiente virtual sob o qual ele possui liberdade de ação dentro das possibilidades de atuação sobre os diferentes personagens modelados previamente pelo tutor (educandos) (Figura 10).



Figura 10 - Serious Game do Agente APRENDIZ

A escolha das ações a serem tomadas (anamnese, exame físico, exame complementar, atuações de outras naturezas) pode ou não disparar diferentes estratégias pedagógicas na interface do Agente APRENDIZ originadas por parte do Agente MEDIADOR, destacando-se as estratégias de orientação ou retomada de rumo, as de reforço. Essas estratégias podem estar ocultas na forma de atitudes ou diálogos dos diferentes personagens da interface.

Os Serious Game, relatados acima, consistem em jogos computacionais aplicados ao ensino. Sendo sua principal característica ensinar conteúdos específicos de disciplinas ou treinar habilidades tanto operacionais como comportamentais (MORAIS, 2010).

6. Conclusão

Inicialmente se buscou apporte teórico nos dados sobre o projeto AMPLIA, ao qual este trabalho está inserido, sendo destacados pontos importantes sobre o mesmo, como seu funcionamento, sua estrutura e os princípios educacionais envolvidos na sua construção.

Faz-se necessário destacar que as tecnologias sugeridas não são únicas, porém foram as que apresentaram maior eficácia para o desenvolvimento, devido à grandiosidade do projeto, e com desenvolvimento distribuído, faz-se necessário uma padronização para que seja alcançado sólido alicerce. O mercado atual oferece inúmeras possibilidades de ferramentas para o desenvolvimento, porém através deste estudo procurou-se trilhar um caminho para facilitar e padronizar, garantindo com isto o sucesso no seu desenvolvimento.

O resultado obtido neste trabalho foi então a proposta de reestruturação tecnológica do Ambiente AMPLIA para ser executado com recursos de comunicação através da web, utilizando a IDE Netbeans, com o framework Hibernate, usando a plataforma de comunicação de agentes, Jade e a utilização da plataforma VR-MED para interface dos agentes.

Referências

- CARVALHO, Lucas Simões de; BATISTA, Marcel Cunha;ULBRICH,Vinicius . ANÁLISE DE FERRAMENTAS PARA DESENVOLVIMENTO DE APLICAÇÕES PARA SISTEMA OPERACIONAL SYMBIAN. 2007. - Universidade do Estado de Santa Catarina/Centro de Ciências Tecnológicas – UDESC/CCT , Santa Catarina.
- FERREIRA, Paulo André. DESENVOLVIMENTO DE SISTEMA DE INFORMAÇÃO WEB PARA O CONTROLE INTERNO DE PROTOCOLOS DA ESCOLA POLITÉCNICA DE PERNAMBUCO - ESCOLA POLITÉCNICA DE PERNAMBUCO. Disponível em: <<http://dsc.upc.br/~tcc/20061/PauloAndreFerreira.pdf>> Acesso em: jun. 2009
- FIPA – Foundation For Intelligent Physical Agents. Disponível em <<http://www.fipa.org>> Acesso em: maio. 2010
- FLORES, C. D. Fundamentos dos Sistemas Especialistas Organizado por: Dante Augusto Couto Barone Sociedades Artificiais: A Nova Fronteira da Inteligência nas Máquinas:ed. 1 ed., Porto Alegre:, Bookman (ArtMed), 2002, v. 1, p. 127-154.
- FLORES, Cecília D. Negociação Pedagógica Aplicada a um Ambiente Multiagente de Aprendizagem Colaborativa. 2005. 121p. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, PPGC / UFRGS, Porto Alegre.
- GLUZ, J. C. A Biblioteca FACIL (FIPA-ACL Interface Library): Uma Avaliação das Plataformas de Comunicação FIPA e Especificação de uma Interface de Programação FIPA Independente de Linguagem de Programação. Porto Alegre:

- PPGC - Instituto de Informática - UFRGS, 2002 (Trabalho Individual em Nível de Doutorado).
- GLUZ, J. C. Formalização de Comunicação de Conhecimentos Probabilísticos em Sistemas Multiagentes: Uma Abordagem Baseada em Lógica Probabilística. 2005. 237f. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- JADE. Java Agent Development Framework. Disponível em: <<http://jade.tilab.com/>>. Acesso em: mar. 2009.
- JDBC – API JDBC – Disponível em: <<http://java.sun.com/products/jdbc/download.html>>. Acesso em: out. 2010.
- LOZANO F. Persistência com Hibernate. Java Magazine, Ed. 28, p. 18-28, 2006.
- MORAIS, A. M., et. al. Serious Games na Odontologia: Aplicações, Características e Possibilidades. In: XII Congresso Brasileiro de Informática em Saúde 2010. Porto de Galinhas/PE. Anais. 2010. CD-ROM.
- MOSSMANN, J. B., MARONI, V.; DAHMER, A.; FLORES, C. D.; PINHO, M. VR-MED: Linguagem de Domínio Específico para Ambientes Virtuais Aplicados à Educação Médica. In: XII Congresso Brasileiro de Informática em Saúde 2010. Porto de Galinhas/PE. Anais. 2010. CD-ROM.
- NETBEANS. – NetBeans open-source and free IDE. Disponível em: <<http://www.netbeans.org/>>. Acesso em: jul. 2010.
- SEAMED. Sistemas especialistas para a área médica. Disponível em: <<http://www.inf.ufrgs.br/~dflores/seamed/default.htm>>. Acesso em: mar. 2010.
- SILVA, Carolina Fernanda. - ANÁLISE E AVALIAÇÃO DO FRAMEWORK HIBERNATE EM UMA APLICAÇÃO CLIENTE/SERVIDOR. Disponível em : <<http://bibdig.poliseducacional.com.br/document/?down=8>> em 15/10/2010> Acesso em: set. 2009.
- SILVA, Leonardo Ayres de Moraes. Estudo e Desenvolvimento de Sistemas Multiagentes usando JADE: Java Agent Development framework - Monografia de Conclusão de Curso - Universidade de Fortaleza – UNIFOR
- SUN Microsystems. Java. Disponível em: <<HTTP://java.sun.com/>>. Acesso em: 09 nov. 2008.

Uma Arquitetura de Referência para Softwares Assistentes Pessoais Baseada em Agentes e SOA

Saulo Popov Zambiasi¹, Ricardo J. Rabelo¹

¹Centro Tecnológico – Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil
{popov,rabelo}@das.ufsc.br

Abstract. Much efforts have been made to create computer systems to provide assistance for people in their daily activities, at home and in business. However, they usually follow different approaches. However, they usually follow different approaches that often do not respect the required compatibility and interoperability with each other besides not being connected with legacy systems and new software services. In this way, this paper presents an open reference architecture for personal softwares assistants, allowing the derivation, personalization and deployment of interoperable instances of assistants regarding users' and companies' processes requirements.

Resumo. Vários esforços vêm sendo feitos para a criação de sistemas computacionais com o objetivo de fornecer assistência às pessoas em suas atividades diárias, em casa e na empresa. Porém, esses geralmente seguem por abordagens distintas, que normalmente não mantêm a compatibilidade e a interoperabilidade uns com os outros, além de não se preocuparem em estarem conectados com os sistemas empresariais legados e serviços novos de software. Neste sentido, este trabalho apresenta uma arquitetura de referência aberta para softwares assistentes pessoais, permitindo que instâncias interoperáveis possam ser derivadas, personalizadas e implantadas consoante as características das pessoas e processos das organizações.

1. Introdução

O conceito de softwares que fornecem assistência às pessoas não é novo e tem influenciado a imaginação de muitos escritores de ficção científica e cientistas da computação [Markoff 2008]. Essa ideia se firma em softwares baseados em conhecimento e que funcionam através da Internet na forma de um tipo de secretário, auxiliando em tarefas como pagamentos de contas, organização de viagens, localização de informações em bibliotecas virtuais na internet, etc. [Michaell 1994].

Durante os últimos anos, diversas pesquisas têm sido efetuadas nesse sentido. Contudo, uma observação feita após uma extensa revisão bibliográfica na área de softwares assistentes é a de que as propostas existentes atacam pontos isolados e que não há nenhuma que seja aberta para permitir a sua integração à ambientes empresariais, i.e. a processos de negócios da empresa [Zambiasi 2010]. Ainda, algumas perspectivas devem ser destacadas. A primeira é a de que assistentes são implementados

como sistemas independentes dos demais da empresa, sem interação ou interoperabilidade alguma, fazendo com que o usuário tenha que navegar por ambientes diferentes, além de que o assistente usualmente atua como um sistema puramente reativo à interação do usuário, e não de forma pró-ativa, executando as tarefas pelo/para ele.

A segunda perspectiva, associada a Tecnologias de Informação e Comunicação, é a interoperabilidade. Os assistentes pessoais costumam ser softwares fechados. Isso porque foram concebidos para realizarem suas atividades sem uma preocupação de projeto de transacionarem com outras aplicações. Essa perspectiva é essencial na ótica de se integrar o assistente ao ambiente geral de execução de processos da empresa, uma vez que normalmente os seus sistemas são distribuídos e heterogêneos.

A terceira perspectiva é quanto a flexibilidade das ações de um assistente. Isso envolve aspectos como adaptabilidade e escalabilidade. No escopo de transações empresariais, cada negócio tem inúmeras particularidades, tanto de contexto como de ambientes computacionais envolvidos. Portanto, há necessidade de maior agilidade geral. Aquelas mesmas transações comerciais não são estáticas. A empresa, seus sistemas e processos de negócios alteram-se ao longo do tempo. Dessa forma, há necessidade de se lidar com a escalabilidade do assistente, que deve estar preparado para flexibilizar suas ações consoante a mudanças e/ou introdução de novos processos de negócios. O aspecto final é que as empresas, embora tradicionalmente tenham seus sistemas implantados todos localmente, começam a conviver com ambientes largamente distribuídos, fazendo uso de sistemas externos, disponibilizados por outras empresas que são, principalmente, provedores externos de serviços. Tais provedores são tipicamente formados por *software-houses* e por dispositivos computacionais distribuídos. Nesse sentido, o assistente deve poder acessar tais serviços consoante aos requisitos funcionais associados aos processos de negócios ora em execução, o que pode levar ao caso da necessidade de composição de diferentes serviços para que o comportamento desejado para o assistente possa ser montado.

A quarta perspectiva é a de que nenhum dos projetos estudados sugere especificações ou padronizações para interoperabilidade com outras aplicações de forma aberta. E por fim, uma quinta perspectiva, é a de que nenhum dos trabalhos avaliados apresenta um modelo ou arquitetura de referência para o desenvolvimento de softwares assistentes pessoais com padrões suficientes para manter a interoperabilidade.

Dessa forma, este trabalho propõe uma arquitetura aberta de referência para softwares assistentes pessoais, que possa gerar implementações interoperáveis e customizáveis para se adequarem aos processos de negócios da empresa e que possam auxiliar os usuários em suas tarefas diárias.

2. Softwares Assistentes Pessoais

Os conceitos de softwares para assistência pessoal confundem-se em diversas premissas com a tecnologia de agentes de software. Com isso, o presente trabalho se apropria desses conceitos de agentes para tratar dos softwares assistentes pessoais. Um assistente, tal como um agente, pode perceber seu ambiente através de recursos (como a agenda do usuário, preferências, preferências de outros usuários, reputações) e responde com ações de assistência ao usuário. Além disso, um agente de software pode substituir

parcial ou totalmente uma pessoa em determinadas tarefas e em várias situações diferentes, assumindo o papel de um assistente. Estes podem trocar informações automaticamente por meio de negociações eletrônicas [Weiss 1999].

Para Bocionek (1994), os assistentes devem ser autônomos para poder oferecer auxílio no gerenciamento das atividades dos seus usuários, inclusive atividades conflitantes. Devem saber negociar, aprender, ter portabilidade e mobilidade, principalmente no cenário da computação móvel.

Segundo Hunhs (1998), um software assistente pessoal não pode ser visto apenas como um programa de computador personalizado. Isto porque um assistente deve ser baseado em rede, ser interativo, adaptativo, de propósito geral, de execução autônoma e deve poder interagir com outros assistentes. Com a apropriação dessas especificidades, Hunhs (1998) também faz um paralelo dos softwares assistentes pessoais com a tecnologia de agentes, apresentando os assistentes pessoais como softwares que podem representar as pessoas na Internet, auxiliando os usuários nas atividades do dia a dia, especialmente nas que envolvem recuperação de informação, negociação ou coordenação. Ainda, Hunhs (2002) sugere a utilização de serviços web, servindo como uma fonte de investigação para assistentes e agentes. Essa tecnologia, além de possuir características facilmente aplicáveis a agentes, possui flexibilidade e capacidade de trabalhar em conjunto em um ambiente interoperável de empresas.

Na parte de implementação, tanto iniciativas privadas como iniciativas de sistemas *opensources* têm tido influência e destaque no caminho da assistência via software de computador. O Assistente Pessoal da Shelltoys, por exemplo, é um sistema proprietário e fechado para auxílio em tarefas do usuário. Contudo, se limita a apenas gerenciar as tarefas do usuário via uma agenda de compromissos. Este assistente possui uma lista de tarefas e gera lembretes de compromissos quanto a aniversários, atividades, reuniões e outras pequenas funcionalidades [SHELLTOYS 2011].

O projeto Sandy [Mann 2011] foi desenvolvido para auxiliar o usuário a lembrar de compromissos, listas, pendências, contatos, projetos e trabalha com troca de mensagens via SMS, Twitter e até mensagens de e-mail. Para o usuário utilizar o Sandy, basta ele enviar mensagens do tipo: "Lembrar de pegar meu carro em 15 minutos", "O telefone do meu pai é 9999-9999", "lembra de comprar mantimentos ovos", etc. Contudo, Sandy é um projeto proprietário, fechado e foi descontinuado em 2008.

O projeto Narval (sob licença GNU LGPL) pode ser executado no próprio computador pessoal ou em um servidor remoto, e se comunica por meios normais, como e-mail, web, telnet, celular, GUI específica, etc. O sistema executa sequências de ações descritas pelo usuário, para uma ampla gama de tarefas, como preparar o jornal da manhã, ajudar a navegar na web por meio de filtro de anúncios e lixo eletrônico, cuidar de tarefas repetitivas como responder e-mail, negociar data e hora de reuniões, etc. O software utiliza Inteligência Artificial, Agentes, Redes de Petri, sistemas baseados em regras, programação de contratos, planejamento, aprendizagem automática. Podem ser desenvolvidos *plugins* em Python [Chauvat 2000], [Thenault, 2011]. Contudo, o projeto não tem tido atualizações recentes, indicando uma provável descontinuação.

O Siri (2011) é um projeto bastante recente. Ele se utiliza de informações de preferências pessoais dos indivíduos e de um histórico de interação para ajudá-lo a

resolver tarefas específicas. É baseado na web e links de resultados de buscas, sua interação na forma de conversa com o usuário. O usuário diz para seu assistente o que quer fazer e este faz pesquisa em fontes de informações para poder auxiliar o usuário. O software evolui com a experiência de interação. Ele se utiliza também como base o contexto pessoal (lugar, horário, histórico) do usuário. O Siri foi adquirido pela Apple e é um sistema fechado e proprietário.

Um projeto bastante relevante é o projeto PAL (2011), financiado pela agência americana DARPA. Este projeto tem por objetivo auxiliar os usuários nos trabalhos com sistemas computacionais e automatizar tarefas rotineiras de forma a liberar as pessoas para as tarefas mais importantes [Markoff 2008]. A intenção é criar um sistema cognitivo que pode raciocinar, aprender e lidar com situações imprevistas como forma de fornecer assistência em situações militares. Além disso, o sistema deve poder se adaptar a mudanças em seu ambiente, aos objetivos dos seus usuários e as suas tarefas sem a necessidade de alterar sua programação ou de intervenção técnica. Este projeto está sendo executado por um conjunto de cientistas da computação e pesquisadores em inteligência artificial, percepção, aprendizado de máquina, processamento de linguagem natural, representação de conhecimento, diálogo multimodal, ciberconsciência, interação homem-máquina e planejamento flexível [PAL 2011].

Isso posto, pode-se resumir as características de assistentes pessoais (incluindo a consideração dos requisitos da proposta deste trabalho) em [Bocionek 1994], [Hunhs 1998] e [Zambiasi 2010]:

- Atuar com certa autonomia em suas tarefas de forma que estes possam verificar a situação dos seus usuários e que possam responder apropriadamente a cada situação que aparece;
- Ser flexível para atuar diante de novas situações e cenários de negócios;
- Ser adaptável ao usuário, conforme informações, preferências, necessidades, situação atual, e evolução das informações;
- Interagir com o usuário;
- Ser baseado em rede, para buscar informações na Internet e interagir com outros sistemas;
- Ser de propósito geral, ou seja, não específico à apenas uma atividade ou um grupo de usuários, adaptando-se aos propósitos do seu usuário;
- Se adaptar ao contexto;
- Ser integrado e interoperável aos processos de negócios da empresa.

3. Proposta

A concepção de uma arquitetura de referência é vista como uma forma de apresentar um padrão genérico para um projeto e deve abordar os requisitos para o desenvolvimento de soluções, guiado pelo modelo de referência e por um estilo arquitetural, de forma a atender as necessidades do projeto [MacKenzie 2009]. Um modelo de referência é uma divisão de funcionalidades, juntamente com o fluxo de dados entre as partes, e possuem características de domínios maduros, decorrente da experiência sobre este domínio. Ele se caracteriza como um padrão de decomposição do problema. Já o estilo arquitetural descreve os tipos de elementos e suas relações, juntamente com um conjunto de restrições sobre como eles podem ser utilizados, além de padrões de interação entre os

elementos. Tais restrições, sobre a arquitetura e sobre o sistema em si, são vistas na forma de uma imagem da utilização do sistema como um todo [Bass 2003].

3.1. Modelo de Referência

Na Figura 1 é apresentado o modelo de referência para assistentes pessoais proposto neste trabalho.

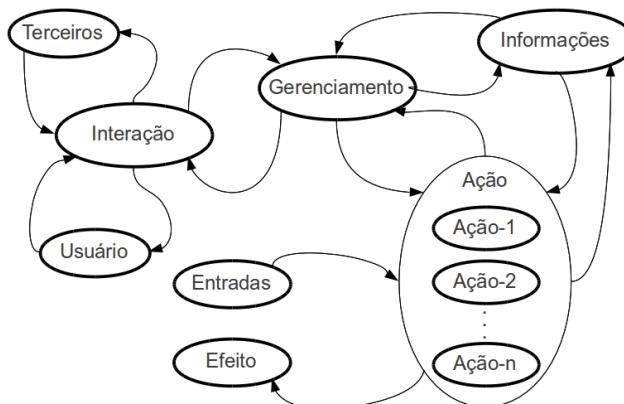


Figura 1. Modelo de Referência para Softwares Assistentes Pessoais.

Entre os elementos principais estão a Ação, que se refere aos comportamentos do assistente, a Interação para com o usuário ou outros assistentes ou softwares e o Gerenciamento, para gerenciar a execução do assistente, organizar as informações e o fluxo dessas entre os demais elementos.

A Ação é a forma como o assistente age, que é a composição do conjunto de n ações que podem ser utilizadas pelo assistente pessoal. As Informações são os elementos úteis para a execução de uma determinada ação ou de um conjunto de ações. As Entradas, tal qual um agente sente seu ambiente por meio de sensores, refere-se ao conjunto de informações que chegam de elementos externos ao assistente e que servem para iniciar uma determinada ação do assistente. Essa ação reage com um Efeito, que pode ser tanto interno do próprio assistente, como externo, com parceiros, outros assistentes, outros softwares, ou mesmo pela interação com o usuário. A Interação é a forma como se dão as negociações entre o usuário e módulo de Gerenciamento do assistente, ou entre o módulo de Gerenciamento e outros elementos.

O Gerenciamento é responsável pela organização das informações do usuário, pela execução das ações e pelo fluxo dessas informações entre os elementos do modelo. Um conjunto de informações acerca do usuário do assistente e das ações são necessárias para que o assistente gerencie sua execução. Elas devem evoluir com o tempo em concordância com a evolução do usuário, tarefas, preferências, etc. Por meio delas, o Gerenciamento deve decidir quando e como iniciar uma ação, ou seja, tornar uma ação operacional quando certas condições forem satisfeitas.

3.2. Arquitetura de Referência

A existência de uma arquitetura de referência para assistentes pessoais é relevante, mas não suficiente para fazer frente àqueles requisitos anteriormente expostos. É importante que o assistente possa ter a característica de comportamentos “plugáveis”, se manter em

um padrão de comunicação e ainda trabalhar com a questão de serviços de software, propiciando assim as bases para a flexibilidade e escalabilidade desejadas. Assim, os comportamentos do assistente pessoal são visto neste trabalho pela invocação de serviços web, distribuídos na Internet, na forma de uma federação de serviços web [Zambiasi 2010]. O usuário pode então escolher os serviços web que melhor se enquadram nas necessidades das tarefas que se quer auxílio. Nesse sentido, os comportamentos do assistente devem poder ser criados ou excluídos de forma flexível. Ainda, isso dá ao usuário do assistente a possibilidade de escolha entre serviços com a mesma funcionalidade, fornecido por diferentes empresas, organizações ou desenvolvedores. Quando um serviço, utilizado por um comportamento, não mais satisfizer as necessidades do usuário, este pode então trocá-lo por outro, que seja mais condizente com sua nova situação/contexto. Ainda, tais serviços devem poder ser utilizados sob variados modelos de negócios, gratuitamente, alugados ou vendidos por terceiros. Com isso o presente trabalho sugere, na Figura 2, uma Arquitetura de Referência para Assistentes Pessoais com estilo arquitetural na Arquitetura Orientada a Serviços (SOA – *Service Oriented Architecture*).

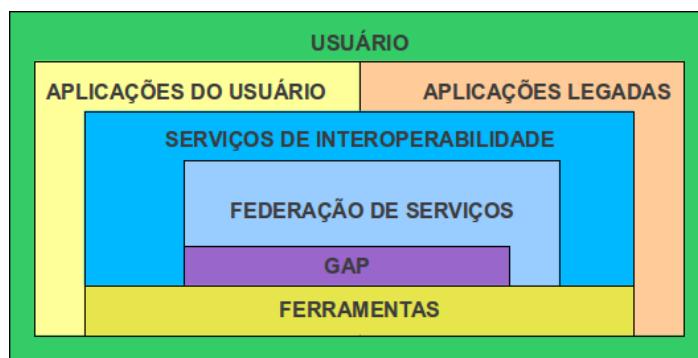


Figura 2. Arquitetura de Referência para Softwares Assistentes pessoais.

- **Aplicações Legadas:** Softwares de terceiros, ou desenvolvidos pela empresa, e que o usuário necessita que o assistente gerencie.
- **Aplicações do Usuário:** Aplicações que podem servir para interação com o assistente pessoal. Pode ser, conforme no sistema Sandy (2011), serviços de e-mails, mensagens de Twitter, sistemas de *Instant Messaging*, ou SMS.
- **Ferramentas:** Aplicações de configuração de assistentes pessoais e comportamentos ou ambientes de desenvolvimento.

A Figura 2 mostra uma visão de como o usuário está em contato com o assistente pessoal. O usuário pode estar em vários níveis de conhecimento:

- **Usuário Simples:** cadastra um assistente pessoal, configura comportamentos e as informações dos comportamentos do seu assistente.
- **Usuário Avançado:** pode compor comportamentos mais complexos no assistente, via uma interface de algoritmos ou fluxogramas.
- **Desenvolvedor de serviços web:** Desenvolve serviços web que podem ser utilizados para compor comportamentos.
- **Desenvolvedor de serviços de interoperabilidade:** Desenvolve interfaces de serviços web baseados em SOA da OASIS para sistemas fora desse padrão.
- **Desenvolvedor de Assistentes Pessoais:** Desenvolvedor avançado de

interfaces para configuração de assistentes pessoais, e núcleos de execução.

A **Federação de Serviços** é conjunto de serviços web distribuídos na Internet e que podem ser utilizados nos comportamento do assistente pessoal isoladamente ou podem servir para compor outros serviços web (orquestração e composição de serviços).

Os **Serviços de Interoperabilidade**, por sua vez, são criados para servirem de interface à outros sistemas que não estão no padrão SOA.

O **Gerenciador de Assistentes Pessoais** é responsável pela coordenação da execução dos comportamentos e suas informações. Os comportamentos podem ter três níveis básicos, apresentados aqui de forma simplificada:

- **Comportamento Simples:** Composto de informações para a chamada do serviço web. Este é executado toda vez que o comportamento é chamado, enviando os parâmetros de entrada e retornando um resultado.
- **Comportamento Condisional:** Composto de informações e especificações condicionais para a execução do comportamento.
- **Comportamento Complexo:** Necessita de um conhecimento mais avançado do usuário. O usuário pode modelar o comportamento na forma de um fluxograma, ou algoritmo, com laços, condicionais e outros elementos.

Os comportamentos são modelados em uma interface de configuração do assistente pessoal, e a forma como isso é feito depende de cada desenvolvedor.

4. Verificação

Uma das formas de verificação da Arquitetura de Referência é a criação de uma instância implementada baseada na proposta. Para tal, foi gerado um protótipo completo em cima de um estudo de caso específico. Neste estudo de caso, é considerado o cenário em que um funcionário de uma empresa tem a função de efetuar o gerenciamento do estoque de produtos. Quando o estoque de um produto passa de um limite mínimo, fica a cargo desse funcionário efetuar nova compra. Conforme produtos são retirados do estoque, sua quantidade diminui. No final do dia, o funcionário faz um relatório das atividades realizadas durante todo o dia, desde o início do processo de uma compra, até as modificações em uma ordem de compra e fechamento da ordem.

A possibilidade da utilização de um assistente pessoal é agora inserida no contexto. Neste caso, o assistente deve ser utilizado para auxiliar o funcionário a gerenciar o sistema de controle de estoque e para a criação do relatório das atividades diárias. Ainda, é considerado que o usuário deseja se comunicar com seu assistente pessoal por meio de aplicações que não precisam ser instaladas em seu computador e que podem ser acessadas de qualquer lugar, inclusive do celular. Neste caso, o usuário pretende se comunicar com seu assistente pessoal como se fosse uma outra pessoa por meio de mensagens via Twitter, Gtalk, e-mail, ou mensagem de texto no celular.

4.1. Protótipo

Sob o cenário apresentado, e a Arquitetura de Referência proposta, pode-se sugerir a modelagem de alguns elementos de implementação que podem compor o assistente

desse estudo de caso (Figura 3), e que são descritos em sequência. Ou seja, a arquitetura de referência ser instanciada (“populada”) com diferentes serviços (i.e. funcionalidades), modelos e ferramentas de implementação.

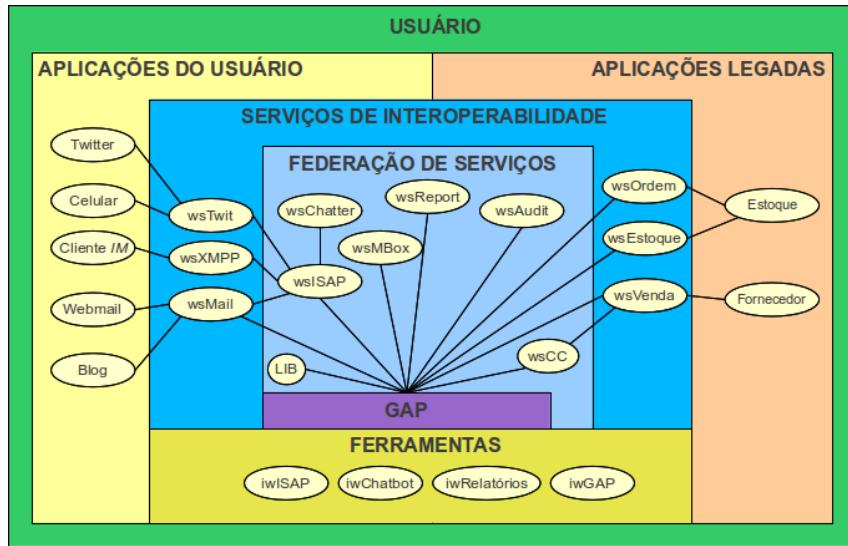


Figura 3. Estudo de caso para a implementação baseada na proposta.

As implementações feitas são aqui apresentadas conforme sua classificação na Arquitetura de Referência proposta. Todos os sistemas *daemons* foram desenvolvidos em linguagem de programação Java, os serviços web em linguagem PHP (no estilo SOA da OASIS) e as interfaces em PHP e HTML via web.

- **Aplicações Legadas**
 - **Estoque:** Sistema de controle de estoque. Possui informações dos produtos, fornecedores, quantidades máxima, atual e mínima dos produtos que são utilizadas para iniciar um processo de compra.
 - **Fornecedor:** Controle de produtos no fornecedor. Informa os produtos, quantidades, e preço. Vários fornecedores podem ser cadastrados.
- **Aplicações do Usuário**
 - **Twitter:** Serviço de publicação de mensagens de até 140 caracteres. Acesso via web ou aplicativos especiais para cada sistema operacional.
 - **Celular:** Via Twitter, o usuário pode habilitar receber/enviar mensagens via mensagem de texto no celular.
 - **Cliente IM:** Para troca de mensagens com o assistente pessoal. O usuário vê seu assistente pessoal como uma pessoa de seus contatos conectada ao serviço. É necessário uma conta no Gmail para o assistente pessoal.
 - **Webmail:** Para troca de mensagens de e-mail com o assistente.
 - **Blog:** O assistente envia postagens à um blog para publicar relatórios.
- **Serviços de Interoperabilidade:** Estes são serviços que fazem interface com outros sistemas que não possuem interfaces de serviços web SOA da OASIS.
 - **wsEstoque:** Acessa informações do sistema de controle estoque.
 - **wsOrdem:** Gerenciamento das ordens de compra.
 - **wsVenda:** Acessa o sistema automático de vendas, um *daemon*

executando do lado do sistema de fornecedores.

- **wsTwit:** Acessa uma conta no serviço de microblog Twitter.
- **wsXMPP:** Acessa um *daemon* de conexão ao serviço de IM Gtalk.
- **wsMail:** Acessa os e-mails de um usuário em um servidor.
- **Federação de Serviços**
 - **wsCC:** Simula transações com cartão de crédito.
 - **wsAudit:** Utilizado para armazenar informações de auditoria.
 - **wsReport:** Geração automática de relatórios. Recebe mensagens referente as atividades e quando requisitado retorna o relatório compilado.
 - **wsMBox:** Gerencia mensagens provindas do ISAP.
 - **wsChatter:** Gera respostas para o usuário e quem mais se comunicar com o assistente pessoal tal como um *chatbot*.
 - **wsISAP:** Serviço de interface com o usuário. Se utiliza de serviços como o wsChatter, wsXMPP e wsMail para trocar mensagens com o usuário. Há um sistema *daemon* que faz o gerenciamento das mensagens. Quando um usuário se comunica com a Interface Social para Assistentes Pessoais (ISAP) a mensagem é avaliada e enviada ao GAP. Ela também decide a melhor forma de enviar uma mensagem provinda do GAP ao usuário.
 - **LIBS:** Conjunto de bibliotecas de funções que são utilizadas para criar os comportamentos no GAP (*string, math, datetime, kqml, myMessageTable*).
- **Ferramentas:** Programas de interfaces web para configuração de alguns sistemas.
 - **iwISAP:** Configuração da ISAP do assistente pessoal.
 - **iwChatbot:** Configuração das conversas do assistente pessoal.
 - **iwRelatorios:** Configuração do gerador automático de relatórios.
 - **iwGAP:** Ambiente de configuração do assistente pessoal e seus comportamentos. Os comportamentos são aqui configurados na forma de um algoritmo. O GAP (Gerenciador de Assistentes Pessoais) é um *daemon* que gerencia a execução autônoma dos assistentes pessoais.

Havendo esse cenário de softwares e serviços web, foi criado um assistente pessoal e alguns comportamentos para efetuar o gerenciamento do sistema de controle de estoque, de responsabilidade do usuário. Em verdade, neste protótipo existem dois caminhos distintos para a criação do comportamento automático de compra:

1. **Usuário avançado:** pode criar comportamentos na forma de algoritmos na interface de configuração do Assistente Pessoal;
2. **Usuário nível básico:** Requisita a um desenvolvedor da empresa um serviço web que efetua o processo automático de compra. O usuário utiliza este serviço web no seu comportamento de compra automática;

Aqui, neste estudo de caso, é considerado que o usuário é do tipo avançado e configurou as informações e algoritmos que vai fazer a compra automática. Neste caso, os seguintes comportamentos foram criados:

- **AtualizaHora:** Invoca operações do serviço web datetime para atualizar data e horário para serem utilizadas por outros comportamentos.
- **ISAPalive:** Envia, a cada pouco, uma mensagem ao ISAP, informando que

ele deve manter a interface do assistente funcionando.

- **mbox:** Invoca operações do wsISAP para verificar e buscar mensagens do usuário para o GAP. As mensagens são identificadas e armazenadas no wsMBox, com informações de qual comportamento essa é destinada.
- **Report:** É verificado se é um horário específico. Caso afirmativo, são requisitados os relatórios das últimas atividades. Relatórios públicos são enviados ao Blog e privados são enviados ao usuário por e-mail.
- **Compra-ordem:** Se um produto está com estoque baixo, gera uma ordem de compra. Essa ordem é enviada ao fornecedor escolhido pelo sistema gerenciador de estoques. Cada nova ordem criada é enviada uma mensagem ao usuário e uma atividade é enviada ao wsRelatorio.
- **Compra:** Se há uma ordem aberta, verifica o estado dela no fornecedor. Se esta foi alterada, altera-a também no wsOrdem e envia uma mensagem ao usuário do assistente pessoal, requisitando confirmação. Se a ordem foi aceita pelo usuário, efetua o pagamento, finalizando a compra. Quando uma ordem é cancelada, o sistema seleciona outro fornecedor para o produto.
- **Compra-altera:** Aguarda a confirmação do usuário de uma ordem alterada. Se o usuário confirma, então a ordem volta a ficar aberta. Se for cancelada, então cancela a ordem no wsOrdem e no wsVenda.

O GAP executa os comportamentos por ciclos. Cada comportamento pode conter diversas atividades e em cada ciclo uma atividade de cada comportamento é executada. Essas atividades podem ser atribuição de valor à uma informação, invocação de uma operação em um serviço web, verificação de um condicional ou de um laço.

4.2. Teste

No sistema de estoque, unidades do produto mouse (código 001), são vendidas, ficando suas informações da seguinte forma: quantidade 5, quantidade mínima 10 e quantidade máxima 20. Ou seja, quando a quantidade deste produto estiver a baixo de 10, são comprados os produtos necessários para chegar à quantidade máxima. Este produto é fornecido pelos fornecedores *John Doe Ltda* e *Sora Konpyuuta* ao valor de 25 reais e 28 reais a unidade, respectivamente. Considere também que o primeiro fornecedor possui apenas 10 produtos em estoque para venda, e que o segundo possui 50. O fornecedor selecionado inicialmente para a compra é o *John Doe Ltda*. As trocas de mensagens entre assistente pessoal e usuário para efetuar o gerenciamento das ordens de compra são vistas na Figura 4.

Quando o assistente pessoal verifica que o estoque está baixo, é iniciado o processo de compra. Como a quantidade do produto 001, do fornecedor *John Doe Ltda* não é o suficiente para o total da ordem de comprar, ou seja, 15 produtos, então a ordem é modificada e o assistente requisita confirmação da alteração. Considere o caso que o usuário necessita comprar todos os produtos para repor o estoque no máximo. Dessa forma, a ordem deve ser cancelada e uma nova criada, a outro fornecedor. Tendo corrido tudo corretamente com a segunda ordem criada, o pagamento é efetuado e a ordem é fechada, atualizando os estoques no fornecedor e no cliente. Por fim, um relatório completo é enviado por e-mail no horário definido pelo usuário.

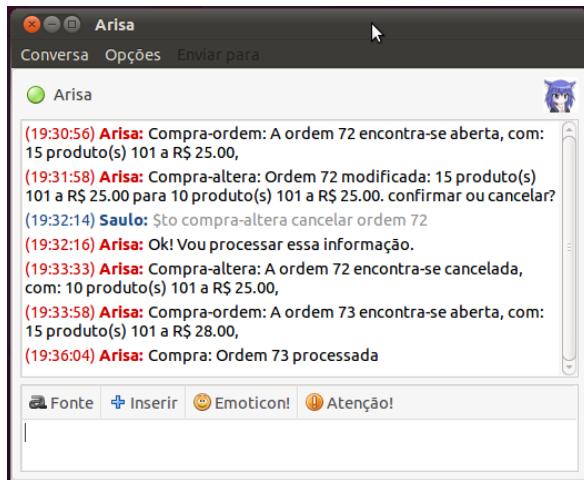


Figura 4. Troca de mensagens entre o assistente pessoal e o usuário.

Quanto a essa execução, é importante se observar alguns pontos. Primeiramente, pode não ser interessante que o usuário fique recebendo informações do assistente pessoal a todo momento, ou mesmo, o usuário pode querer que esse cancelamento de uma ordem de compra, para a geração de uma nova para outro fornecedor, seja automático. Contudo, neste exemplo várias mensagens são trocadas com o usuário para fins de testes e verificação do funcionamento do comportamento. Outro ponto a se observar é que este teste foi realizado com o usuário conectado em seu usuário no Gtalk. Caso o usuário não estivesse conectado, as mensagens seriam trocadas via mensagem de texto do celular.

5. Considerações Finais

Este artigo apresentou uma proposta para uma Arquitetura de Referência para Softwares Assistentes Pessoais. Essa proposta se apropriou da Arquitetura Orientada à Serviços para servir de estilo arquitetural e do conceito de Agentes.

Com base nos resultados de testes sob a implementação final, verificou-se que esta instância se comportou conforme o que foi proposto na arquitetura de referência e executou corretamente as ações associadas aos comportamentos do processo de negócio envolvido no estudo de caso. Todavia, ainda se constata que a área de assistentes pessoais tem ainda enormes desafios à serem enfrentados, incluindo ao que se refere a limitações das TIs atuais para certos propósitos (por exemplo, o problema da interoperabilidade semântica), os impactos organizacionais, a complexidade do problema de composição de comportamentos dentro da abordagem SOA, etc. Dessa forma, este trabalho veio no sentido de oferecer uma contribuição para a área de assistentes pessoais mais flexíveis, mais interoperáveis, e mais integrados ao mundo das empresas/processos de negócios.

6. Agradecimentos

Este trabalho é parcialmente financiado pela CAPES – Coordenação de Pessoal de Nível Superior (<http://www.capes.gov.br>).

Referências

- 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.
- Bass, Len; Clements, Paul and Kazman, Rick. “Software architecture in practice”. 2nd ed. Addison-Wesley, 2003.
- Bocionek, S. “Software secretaries: learning and negotiating personal assistants for the daily office work In Systems, Man, and Cybernetics”, In: Humans, Information and Technology. 1994 IEEE International Conference on, 12 vol.1. 2-5 Oct. 1994.
- Chauvat, N. “Narval, the intelligent personal assistant or how the french Linux Gazette is built”, In: Linux Gazette, issue 59. Nov, 2000. Disponível em <<http://linuxgazette.net/issue59/chauvat.html>>, acessado em 14/02/2011.
- SHELLTOYS. “Personal assistant - day planner and personal information manager”. Disponível em: <http://www.shelltoys.com/personal_assistant/>, acessado em 14/02/2011.
- Huhns, M.N.; Singh, M.P., “Personal assistants”, In: IEEE Internet Computing, Vol. 2, Issue 5, pp. 90-92. Sep./Oct. 1998.
- Huhns, M.N. “Agents as Web services”, In: Internet Computing, IEEE 6, 93-95. 2002.
- MacKenzie, C.; Laskey, K.; McCabe, F. at all. “Reference Model for Service Oriented Architecture 1.0”, In: OASIS Standard, 12 October 2006. <<http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>> acessado em Fev/2009.
- Mann, H. “Free Personal Assistant - Meet Sandy”. Disponível em <<http://honestholly.com/free-personal-assistant-meet-sandy/>>, acessado 14/02/2011.
- Markoff, J. “A Software Secretary That Takes Charge”, In: New York Times, 13/12/2008. Disponível em: <http://www.nytimes.com/2008/12/14/business/14stream.html?_r=1&scp=7&sq=personal%20assistant&st=cse>. Acessado em: 17/03/2009.
- Michaell, T.; et all. “Experience With a Learning Personal Assistant”, In: Communications of the ACM, July, 1994.
- PAL, PAL Project, <<http://pal.sri.com/>>, acessado em 14/02/2011.
- SIRI, disponível em <<http://siri.com/>>, acessado em 14/02/2011.
- Thenault, Sylvain. Narval-moved. Logilab Project. Disponível em: <<http://www.logilab.org/908/>>, Acessado em 14/02/2011.
- Weiss, G. “Multiagent systems: A Modern Approach to Distributed Artificial Intelligence”. MIT Press, 1999.
- Zambiasi, S.P.; Rabelo R.J.. “Uma arquitetura de referência para softwares assistentes pessoais baseada na arquitetura orientada à serviços”, In: I2TS'2010: 9th International Information and Telecommunication Technologies Symposium, Rio de Janeiro (RJ), Brasil, 2010.

Qualificação de Valores de Troca em Interações Sociais em Sistemas Multiagentes

Cion Ayres do Nascimento¹, Gustavo Alberto Gimenez Lugo¹, Murilo V. G. da Silva¹

¹Departamento Acadêmico de Informática
Programa de Pós-Graduação em Computação Aplicada
Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro, 3165 – Rebouças CEP 80230-901 – Curitiba – PR – Brasil

cion.ayres@gmail.com, gustavogl@utfpr.edu.br, m.v.g.dasilva@gmail.com

Abstract. This article describes a method of qualification of exchange values in social interactions in multiagent systems. The method can function as basis of a model that can be used to project social links between agents. Such projections can help the agents to identify unfavorable situations to the system that can exist because of undesired states that can be achieved by, for example, persistent concentrations of power in groups of agents.

Resumo. Este artigo descreve um método para a qualificação de valores de troca em interações sociais em sistemas multiagentes. O método pode servir como base de um modelo que será utilizado para projetar as ligações sociais realizadas entre os agentes. Tais projeções podem auxiliar os agentes a identificarem situações desfavoráveis para o funcionamento do sistema devido a estados indesejados que podem ser alcançados através de, por exemplo, concentrações persistentes de poder em grupos de agentes.

1. Introdução

As pesquisas na área de sistemas inteligentes tem evoluído de maneira muito significativa desde a criação do framework PRS¹ na década de oitenta [Trobe and Melbourne 1995]. Modelos como o BDI² ganharam grande importância e, apesar das críticas, são estudados e melhorados até hoje. Tais modelos promovem a interligação entre as áreas de humanas e exatas, em que pesquisadores de sistemas inteligentes voltam-se para estudos envolvendo filosofia, sociologia e psicologia de maneira a procurar novos modelos necessários para a solução de problemas em sistemas inteligentes.

Os sistemas multiagentes são compostos por diversos agentes engajados em atividades sociais. É um ambiente de desenvolvimento que traz uma série de dificuldades para o projetista, pois como os agentes são entidades autônomas, não há um controle total sobre o que estes podem realizar [Bordini et al. 2007]. Sendo assim, nem sempre um agente sozinho pode alcançar seus objetivos e ser completamente autônomo, seja por falta de recursos ou planos necessários para o objetivo. Torna-se necessário então que o agente possa obter informações a respeito de outros agentes do sistema, de maneira a poder formar conclusões e tomar decisões sobre suas atividades sociais, criando uma relação de dependência com outros agentes [Sichman et al. 1994].

¹Procedural Reasoning System - um dos primeiros frameworks a implementar o modelo BDI

²Modelo de agência que define os conceitos de Belief(Crenças),Desires(Desejos) e Intentions(Intenções)

Como os agentes estão em um ambiente dinâmico, as ligações feitas entre os mesmos e as relações de dependência formadas tornam-se imprevisíveis e certos estados indesejados podem ocorrer, como acontece quando há uma concentração de poder em um grupo de agentes. Não necessariamente essas concentrações de poder social são negativas, porém ligações deste tipo que são persistentes podem trazer problemas para a conduta dos agentes uma vez que grupos de agentes acabam servindo de HUBs na rede social [Franks et al. 2008]. Um exemplo pode ser visualizado na figura 1, em que o grupo B o responsável pela comunicação dos agentes do grupo A e C. Neste caso, o grupo B tem um poder social muito grande no papel da comunicação entre A e C, podendo interferir negativamente em benefício próprio.

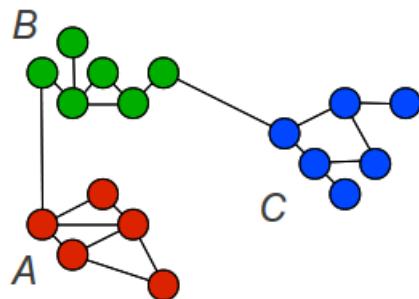


Figura 1. Exemplo de centralização de poder

A questão é como identificar estes pontos de concentração de poder. Através de cálculos associados a um mecanismo de reputação é possível que um agente analise quais as motivações de outro agente vizinho, porém é necessário realizar projeções para segmentos de grupos sociais, permitindo que se obtenha informações sobre quais tipos de ligações serão formadas entre quais agentes. Desta maneira seria possível identificar o aparecimento dessas figuras centralizadoras de poder sugerindo, quando possível, novas conexões entre os membros da rede.

Este artigo propõe a utilização da sociologia econômica de Pierre Bourdieu através de um método de qualificação de valores em trocas sociais entre agentes. A Seção 2 descreve os estudos sobre estruturas organizacionais com o modelo PopOrg, o qual utiliza-se das teorias de trocas sociais de Jean Piaget. A Seção 3 descreve a necessidade da qualificação dos valores de troca e o seu papel na formação das normas sociais através das teorias de Bourdieu. A Seção 4 explica como pode ser alcançado o equilíbrio em uma troca social e apresenta um método de classificação das qualificações dadas pelos agentes para os valores de troca. Finalmente na Seção 5 tem-se a conclusão sobre a importância da qualificação simbólica dos valores no processo de interação social entre os agentes.

2. Estudos Relacionados

Sistemas multiagentes precisam ser interpretados de maneira análoga ao comportamento social humano. Desta maneira são necessárias definições que especifiquem como estes agentes se organizam em estruturas sociais colaborativas. Estas estruturas sociais tem como objetivo o de melhorar a flexibilidade, extensibilidade, robustez e confiança do sistema como um todo.

A característica de autonomia dos agentes não garante que os mesmos sempre farão as melhores escolhas ao formar ligações sociais ou que sempre trabalhem de forma

cooperativa [Boella et al. 2006]. As teorias organizacionais visam descrever a estrutura interna de uma organização de agentes e procuram especificar como estes agentes realizam alianças estratégicas entre diferentes organizações com objetivos em comum. Estes modelos abstratos e a comparação com o mundo real são importantes no desenvolvimento de Sistemas Multiagentes (SMAs).

Um dos modelos que procura estudar as ligações sociais entre os agentes é o modelo População-Organização (PopOrg), cujo o desempenho dos papéis pelos agentes e a criação de ligações sociais entre os mesmos tornam-se mecanismos chave para a implementação de uma estrutura organizacional. O objetivo é prover para um sistema uma estrutura organizacional que é criada pelos próprios agentes através do conceito de valores de trocas sociais [Rocha Costa and Dimuro 2008].

Este modelo toma como base a teoria de valores de troca de Jean Piaget, cujos estudos nas áreas de psicologia, sociologia e biologia trouxeram grandes contribuições para os modelos atuais de interação social em SMAs. Para Piaget, é necessário que haja um equilíbrio na troca dos valores [Rodrigues et al. 2003].

Inicialmente é necessário definir o conceito de valor, que pode ser interpretado como qualquer coisa que possibilite uma troca. Os valores podem ser serviços, objetos ou até mesmo sensações e são na verdade uma avaliação mental qualitativa realizada sobre os elementos que são envolvidos em uma interação.

Dois indivíduos trocando serviços entre si geram valores e débitos morais, responsáveis por iniciar e manter uma interação entre estes indivíduos. É importante então que os agentes em um SMA possam qualificar os valores que estão sendo utilizados para as trocas a fim de se relacionarem no ambiente social. Para que isso seja possível, cada agente deve saber o valor qualitativo do serviço que pode oferecer bem como o valor do serviço que está necessitando. Esta qualificação de valores torna-se complicada pois um agente não pode assumir aleatoriamente a qualificação de cada valor utilizado na troca. Faz-se necessário um modelo que regule a qualificação destes valores e que permita a definição de normas para avaliá-los.

3. Sociologia Econômica de Pierre Bourdieu

O modelo PopOrg utiliza-se das teorias sociais de Piaget para explicar um mecanismo de valores que são trocados entre os agentes de maneira a promover uma interação social e manter o equilíbrio da sociedade. Piaget preocupa-se também com a qualificação dos valores que estão sendo trocados, porém para que exista um sistema de valores de troca assume que os indivíduos envolvidos possuem a mesma noção de avaliação dos valores, de maneira a garantir a compatibilidade da avaliação feita por ambos a respeito do valor trocado [Rodrigues et al. 2003].

A teoria ainda descreve a possibilidade de existirem situações de desequilíbrio nesse sistema de trocas, como quando um agente expressa algum valor errôneo a respeito de um serviço executado, por exemplo. Questiona-se porém o que pode ser considerada uma avaliação errônea por parte de um agente, uma vez que este apenas age de acordo com as suas crenças e observações do ambiente.

Faz-se necessário um modelo que leve em consideração a qualificação desses valores de troca por parte dos agentes e as teorias da sociologia econômica de Pierre Bourdieu

podem ser utilizadas para prover uma base sociológica para definição desse modelo.

Um processo de troca de valores pode ser observado na Figura 2 em que um agente α solicita algum valor do agente β .

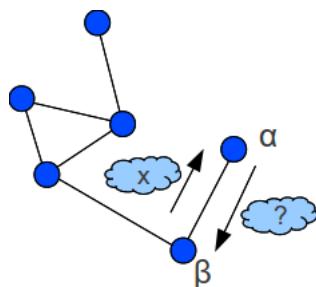


Figura 2. Exemplo de troca de valores

Não se pode assumir que α e β possuam necessariamente a mesma avaliação do valor solicitado. Levantam-se então os seguintes questionamentos:

- Qual a avaliação do valor pelo agente α ;
- Qual a avaliação do valor pelo agente β ;
- Como α e β realizam a avaliação do valor em questão.

Bourdieu descreve a cultura e os sistemas simbólicos como instrumentos de poder que podem ser usados na legitimação da ordem vigente no ambiente [Bourdieu 2007]. Os sistemas simbólicos possuem a função de organizar a representação do mundo natural e social enquanto que a cultura possui uma função ideológica e em certos casos, política. A cultura é uma estrutura cuja representação do mundo social define a estrutura das relações sócio-econômicas, as quais passam a ser percebidas pelos membros da sociedade como sendo naturais e não impostas.

Ao exemplo da Figura 2, pode-se entender que a avaliação dos valores trocados pelos agentes depende de uma avaliação que é considerada natural pela sociedade formada pelos mesmos. Isto define portanto, que o consenso que os agentes possuem a respeito dos valores trocados pode conduzir a um sistema de regras na sociedade dos agentes. Se o agente α obtém a informação do agente β que o valor requisitado vale x , é possível que α comunique-se com outros membros da sociedade questionando a informação passada por β , por exemplo. Se a grande maioria dos agentes concordar que o valor requisitado vale x , há então uma grande chance de α aceitar a avaliação, inclusive podendo adicionar à sua base de crenças que o valor x é aceitável.

Para Bourdieu, os indivíduos passam a ser agentes sociais que estão além da mera função de suporte das estruturas sociais, podendo realizar práticas que possam inclusive modificar a sociedade. Esta ideia em SMAs pode definir um sistema em que as normas estabelecidas não estão pré-definidas mas sim são criadas e emergem da própria sociedade dos agentes.

Bourdieu define ainda alguns conceitos para explicar a sua teoria, são eles o *habitus* e o *campo*. Define-se *habitus* como sendo um processo que é incorporado pelo agente social produzindo as chamadas interiorização da exterioridade e a exteriorização da interioridade. Para um agente em um SMA, pode-se relacionar a primeira com uma adoção

às normas estabelecidas pela sociedade enquanto que a segunda como sendo as ações do agente e as consequências das mesmas para a sociedade. O outro conceito, o do *campo*, define uma situação social em que os agentes realizam as suas práticas de acordo com o *habitus* aprendido.

As teorias da sociologia econômica de Bourdieu podem ser utilizadas de maneira a fazer uma ligação entre as Estruturas Organizacionais e os Sistemas Normativos, procurando explicar como os agentes podem qualificar os valores trocados a partir de conceitos como o *habitus* e o *campo* e permitindo que estes mesmos agentes criem estruturas para definição das normas da organização.

4. Valores de Troca

Ao analisar a Figura 3 vê-se um agente *a* entrando em uma sociedade formada por outros agentes. Cada um dos agentes troca valores entre si e precisa saber como qualificar cada valor. A esta situação social dá-se o nome de *campo* em que cada agente inserido normalmente se comporta de acordo com as normas estabelecidas para aquele *campo*.

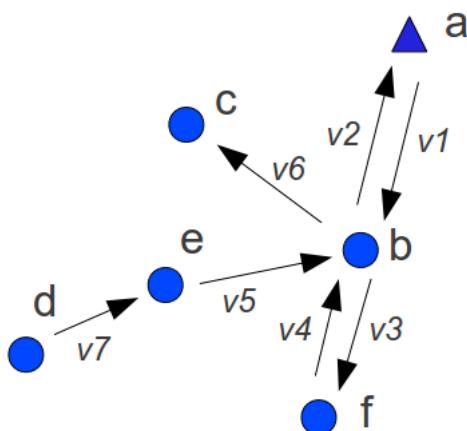


Figura 3. Troca de Valores em uma Sociedade de Agentes

Segundo a teoria de trocas sociais de Piaget, é necessário que exista um equilíbrio entre a renúncia e a recompensa entre os agentes [Rodrigues et al. 2003]. Para demonstrar o equilíbrio tem-se *r*, *s*, *t* e *v* como valores simbólicos sendo:

- $r \rightarrow$ uma renúncia;
- $s \rightarrow$ uma satisfação;
- $t \rightarrow$ um reconhecimento;
- $v \rightarrow$ uma recompensa.

Como exemplo, ao analisar uma interação social entre os agentes *a* e *b*, tem-se o agente *b* solicitando um serviço do agente *a*, neste cenário, *a* faz uma renúncia em favor de *b* gerando imediatamente uma satisfação em *b* e um reconhecimento para com *a*. O agente *b* faz então uma renúncia em favor de *a* gerando uma recompensa e equilibrando a troca.

Neste exemplo tem-se um processo em dois estágios:

- no primeiro há renúncia(*r*) feita pelo agente *a* e a recompensa(*v*) em forma de reconhecimento(*t*) do agente *b*;

- no segundo há o reconhecimento(t) do agente b gerando a renúncia(r) em b e recompensa(v) para com o agente a .

Depois do fim dos dois estágios há uma satisfação(s) nos dois agentes. Este processo é analisado em [Rodrigues et al. 2003], demonstrando o equilíbrio obtido na troca entre os agentes a e b através das equações:

$$(ra) + (sb) + (tb) + (va) = 0$$

$$(va) + (tb) + (rb) + (sa) = 0$$

Porém, para que seja possível qualificar os valores que estão sendo trocados é necessário um detalhamento do que está sendo trocado nesta negociação. A renúncia dos agentes a e b pode ser definida como uma composição de atributos sendo:

$$ra = a_1 \alpha_1 + a_2 \alpha_2 + \dots + a_n \alpha_n$$

$$rb = b_1 \beta_1 + b_2 \beta_2 + \dots + b_n \beta_n$$

Os atributos α e β representam valores qualificados pelos agentes a e b respectivamente e cada índice representa um valor diferente utilizado na troca que pode significar uma renúncia de 1 a n valores. Esta definição de atributos é importante pois o par de agentes busca o equilíbrio na troca social, porém pode possuir qualificações diferentes e valores diferentes para a negociação.

Não é o objetivo deste artigo demonstrar como são calculados ou compostos os valores simbólicos de qualificação feitos pelos agentes. Além disso, a qualificação dos valores também faz parte de um *campo* específico, ou seja, cada *campo* pode possuir qualificações diferentes para um mesmo tipo de valor.

Para este exemplo em específico, assume-se que os agentes da sociedade em questão conhecem os valores simbólicos de cada valor trocado. Assim sendo, para cada valor de troca, cada agente deve obter a qualificação com os outros agentes, preenchendo uma tabela de qualificação simbólica de valores. Para um determinado valor, tem-se uma matriz de valores simbólicos levando em consideração a relação social de cada agente e as suas respectivas qualificações de um determinado valor conforme mostra a Figura 4.

		<i>Requisitado</i>					
		a	b	c	d	e	f
<i>Requisitante</i>	a	8	6	-3	5	6	1
	b	9	10	6	4	3	1
	c	6	4	-10	4	2	1
	d	7	0	3	8	6	1
	e	3	0	7	2	0	1
	f	2	4	1	1	8	9

Figura 4. Matriz de Qualificação de Valor

Os valores numéricos utilizados na matriz da Figura 4 são apenas representações simbólicas relacionadas à qualificação dada por cada agente ao valor a ser trocado,

variando neste exemplo de uma avaliação negativa, representada pelo índice *-10* a uma avaliação positiva, representada pelo índice *10*. Ainda, a Figura 4 demonstra a qualificação de apenas um valor de troca para um *campo* específico e é interessante notar que o valor simbólico para dois pares de agentes pode ser diferente. Esta diferença entre valores simbólicos se deve a uma relação social entre pares de agentes em que um agente pode possuir um custo simbólico maior ao realizar uma renúncia em relação a outro agente.

Na matriz exibida na Figura 4, pode-se observar que os agentes *a* e *b* possuem qualificações diferentes para um mesmo valor. No exemplo, quando *a* requisita o valor de *b*, a qualificação simbólica de *a* é inferior a qualificação realizada caso o agente *b* estivesse solicitando o mesmo valor. Caso os agentes *a* e *b* iniciem uma troca social, pode-se visualizar uma situação de desequilíbrio.

É importante reforçar que esta matriz representa a qualificação simbólica de um valor para um *campo*, então para cada *campo* tem-se uma matriz multidimensional, como mostra a Figura 5 com a qualificação de cada um dos valores de troca utilizados na intereração social entre os agentes.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>a</i>	8	6	-3	5	6	1
<i>b</i>	9	10	6	4	3	1
<i>c</i>	6	4	-10	4	2	1
<i>d</i>	7	0	3	8	6	1
<i>e</i>	3	0	7	2	0	1
<i>f</i>	2	4	1	1	8	9

Figura 5. Matriz Multidimensional de Qualificação de Valor para um campo

A matriz multidimensional faz parte do *habitus* do agente, que é composto por esta qualificação de valores e pelas normas estabelecidas pela sociedade dos agentes. Para cada *campo* em que o agente se encontra poderá existir um novo conjunto de normas e o agente precisa manter um registro completo de *habitus* diferentes para cada *campo*.

Essas matrizes representam então a qualificação simbólica completa dada pelos agentes para todos os valores de troca em um *campo* específico. Porém, uma renúncia é composta por um ou mais atributos que especificam quais valores estão sendo utilizados em uma negociação. Então, além de uma matriz utilizada na qualificação individual de cada valor entre todos os agentes que estão em determinado *campo*, é necessária a criação de uma segunda matriz comparando diferentes tipos de valores de maneira a armazenar uma comparação qualitativa de cada agente entre dois tipos diferentes de valores para todos os valores utilizados dentro de um *campo*.

5. Considerações Finais

A construção de sistemas autônomos procura imitar o comportamento humano para criar softwares com capacidade de aprendizado e adaptação em ambientes dinâmicos. O estudo

das relações humanas torna-se então essencial para qualquer pesquisa nesta área.

A troca de valores entre agentes pode gerar situações desfavoráveis para um agente ou um grupo de agentes, como é o caso da qualificação dos valores de troca. A qualificação simbólica dos valores tem um papel importante na formação de ligações sociais entre agentes e pode ser utilizada para auxiliar na identificação de pontos de concentração de poder permanentes. As relações de dependência entre agentes podem apontar para essas figuras centralizadoras de poder social permitindo a sua identificação bem como possibilitando que sejam sugeridas novas ligações sociais caso seja necessário.

Os conceitos de *campo* e *habitus* das teorias de sociologia econômica de Bourdieu são essenciais para o desenvolvimento de um modelo que leve em consideração as trocas realizadas entre os agentes bem como a qualificação dos valores que estão sendo trocados, uma vez que esta interação tem o poder de influenciar a normas da sociedade formada pelos agentes, bem como o seu comportamento dentro do grupo.

Referências

- Boella, G., Torre, L., and Verhagen, H. (2006). Introduction to normative multiagent systems. *Computational and Mathematical Organization Theory*, 12(2-3):71–79.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. Wiley.
- Bourdieu, P. (2007). A economia das trocas simbólicas: Introdução, organização e seleção sérgio miceli.
- Franks, D. W., Noble, J., Kaufmann, P., and Stagl, S. (2008). Extremism propagation in social networks with hubs. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 16(4):264–274.
- Rocha Costa, A. C. d. and Dimuro, G. P. (2008). Semantical concepts for a formal structural dynamics of situated multiagent systems. In *Proceedings of the 2007 international conference on Coordination, organizations, institutions, and norms in agent systems III*, COIN’07, pages 139–154, Berlin, Heidelberg. Springer-Verlag.
- Rodrigues, M. R., Costa, A. R., and Bordini, R. H. (2003). A system of exchange values to support social interactions in artificial societies. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems - AAMAS ’03*, page 81.
- Sichman, J. S., Conte, R., Demazeau, Y., and Castelfranchi, C. (1994). A social reasoning mechanism based on dependence networks. In *11th European Conference on Artificial Intelligence, 1994, Amsterdam. Proceedings of 11th European Conference on Artificial Intelligence*, pages 188–192, Amsterdam.
- Trobe, L. and Melbourne, S. (1995). From: Proceedings of the First International Conference on Multiagent Systems. Copyright ©1995, AAAI (www.aaai.org). All rights reserved. *Practice*.

Criação de Fatos Institucionais a partir de Fatos Brutos em Sistemas Multiagente

Maiquel de Brito, Jomi F. Hübner

¹Departamento de Automação e Sistemas - Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 91.501-970 – Florianópolis – SC – Brazil

{maiquel, jomi}@das.ufsc.br

Resumo. *Estudos em sistemas multiagente têm apontado a demanda de uma infraestrutura institucional que garanta que os agentes tenham o comportamento desejado nas interações realizadas e também auxilie a execução de suas tarefas na instituição. Em certos cenários, a interação entre agentes e instituição pode ser intermediada pelo ambiente em que os agentes estão inseridos, seguindo a teoria da construção da realidade social de John Searle. Neste artigo, são analisados alguns modelos que propõem a criação de fatos institucionais a partir de fatos brutos ocorridos no ambiente. Como resultado, são levantados alguns pontos relevantes verificados em tais modelos.*

1. Introdução

Em sistemas multiagente (SMA) abertos, em que as características dos agentes não são conhecidas de antemão e novos agentes podem interagir com o sistema ao longo do tempo, é preciso estabelecer mecanismos que garantam que o comportamento dos agentes esteja de acordo com o estabelecido pela instituição e, ao mesmo tempo, auxiliem os agentes na execução de tarefas relacionadas à instituição. Para tal, estudos apontam para a utilização de infraestruturas institucionais em SMA.

Em algumas situações, no entanto, pode ser desejável ou até mesmo imperativo que os agentes não tenham acesso direto à infraestrutura institucional do sistema. Nesse caso, uma alternativa seria utilizar o ambiente como meio de interação entre agentes e instituição. Essa abordagem segue a teoria de construção da realidade social proposta por John Searle [Searle 1995], que trata da criação de fatos relacionados à instituição a partir de fatos relacionados ao ambiente.

As ideias propostas por Searle têm exercido grande influência nas pesquisas relacionadas a sistemas baseados em agentes [Báez-Barranco et al. 2007]. Este artigo analisa alguns dos modelos que seguem tais ideias e está organizado da seguinte maneira: inicialmente são explanados alguns aspectos relacionados à realidade social, segundo a teoria de Searle. Em seguida, são analisados alguns modelos que utilizam os conceitos de tal teoria e, por fim, é feita uma síntese da análise de tais modelos.

2. A realidade social

Searle afirma que a realidade perceptível às pessoas é composta por duas categorias de fatos: fatos brutos (*brute facts*) e fatos institucionais. Os fatos brutos são aqueles que independem de qualquer crença ou opinião. Pode-se citar, como exemplo, o fato de que a água é composta por oxigênio e hidrogênio.

Os fatos institucionais, por sua vez, são aqueles que se concretizam mediante à concordância coletiva e a existência de uma instituição que os suporte. Um pedaço de papel, para ser compreendido como tal, não necessita de concordância coletiva e nem de uma instituição. Este mesmo pedaço de papel pode, no entanto, representar um valor monetário qualquer, caso um grupo de pessoas concorde com isso. O fato de este pedaço de papel ser considerado dinheiro é um exemplo de fato institucional pois é originado através de concordância coletiva e é suportado por uma instituição (Banco Central). A instituição que suporta um fato institucional não precisa ser, necessariamente, uma instituição formal. Uma pedra, por exemplo, é um objeto pertencente à realidade bruta. Suponha-se porém, que esta pedra seja uma pedra preciosa, cujo valor, dado em função de sua preciosidade, pode ser expresso em grandezas monetárias. O fato de a pedra ser preciosa é decorrente da concordância coletiva de que tal objeto possui algum valor e, se em algum momento, a coletividade deixar de concordar que aquele objeto tem algum valor, a pedra não será mais considerada preciosa.

Os fatos institucionais que compõem as instituições são originados a partir de fatos brutos ou de outros fatos institucionais através da aplicação de *constitutive rules*. Uma *constitutive rule* tem o formato *X count as Y in C* e pode ser interpretada como “X representa Y no contexto C”. De acordo com essa expressão, o termo Y confere ao termo X um *status* (estado, função, poder ou propriedade) que este não possui originalmente. O termo C da expressão define o contexto em que o termo Y confere novo status ao termo X. Como exemplo, pode-se citar a instituição de um casamento. O ato de declarar que duas pessoas estão casadas (X), se executado por um sacerdote reconhecido, em cerimônia adequada (C), é socialmente aceito como um ato que confere aos noivos o status de casal (Y). Se, no entanto, o contexto, composto por elementos como sacerdote, cerimônia e outros, não for adequado ao que é aceito coletivamente, os noivos não serão considerados social ou legalmente casados.

3. A realidade social em sistemas multiagente

Alguns tipos de problema, como comércio eletrônico, por exemplo, mostram-se propícios à aplicação de agentes de software atuando em conjunto e formando “sociedades computacionais”. Estas sociedades têm como característica a heterogeneidade, possíveis conflitos entre objetivos individuais, confiabilidade limitada e alta probabilidade de inconformidade com as especificações [Artikis et al. 2002]. Diante destas características, é necessário que os sistemas tenham mecanismos que garantam sua integridade e coerência, utilizando conceitos relacionados às organizações humanas, como normas e protocolos [Campos et al. 2009]. Dessa forma, verifica-se, em SMA, uma dimensão institucional a que podem ser aplicados conceitos similares aos propostos por Searle nos estudos relacionados à construção da realidade social. Alguns estudos na área de SMA, apresentados a seguir, utilizam tais conceitos.

3.1. O poder institucionalizado segundo [Jones and Sergot 1996]

O conceito de poder institucionalizado refere-se à característica dos SMA de que agentes podem produzir fatos institucionais. Conforme [Jones and Sergot 1996], ações ou estados de um agente podem ter impacto em alguma instituição. A partir de regras particulares da instituição, é possível inferir fatos institucionais a partir de ações ou estados de um agente em nível “bruto”. Ou seja, é possível, mediante à definição de regras que governem

a instituição, estabelecer que determinados aspectos da realidade bruta *count as* fatos institucionais.

A ideia básica com relação ao poder institucionalizado é que, em um sistema normativo, existem condições que determinam que o estado ou uma determinada ação de um agente x seja suficiente para que uma instituição y seja conduzida um novo estado normativo F . Para exemplificar, considere-se um sacerdote dirigindo uma cerimônia de casamento. Neste caso, o sacerdote é o agente x , que executa a ação que produzirá o estado normativo F “casado”. Este estado, por sua vez, é criado por uma instituição y (igreja, sociedade ou outra instituição). Nesse caso, o ato do sacerdote, em contexto adequado, *count as* um casamento.

Além das ações, o estado de um agente pode ter significado normativo. Uma pessoa com idade superior a um determinado número, por exemplo, pode ser socialmente considerada um idoso.

O modelo de [Jones and Sergot 1996] lançou algumas bases à inferência de fatos institucionais, definindo que a realidade percebida em nível institucional em um SMA pode ser vista como consequência da realidade verificada em nível bruto. Essa premissa serviu, direta ou indiretamente, de base para a elaboração dos modelos citados neste artigo.

3.2. O modelo de [Artikis et al. 2002]

Com base nos estudo descritos em [Jones and Sergot 1996], [Artikis et al. 2002] propõe um *framework* para especificação de sistemas computacionais compostos por sociedades de agentes. A proposta tem por base os conceitos de *social constraints*, *social roles* e *social states* e, além de utilizar o conceito de poder institucionalizado, utiliza também os conceitos de *posições normativas*, de *transições de estado*, utilizando o formalismo temporal do *Event Calculus* [Shanahan 1999].

As *social constraints* expressam a semântica das ações dos agentes, definindo quais estados estas podem iniciar ou terminar, além de especificar propriedades das ações, tais como validade, permissão e políticas para controle de execução. Na proposta, uma ação é válida se o agente que a executou tem poder institucional para tal. Um segundo nível de especificação nas *social constraints* refere-se à permissão, proibição ou obrigação de um agente na execução de uma ação. Esta definição é específica ao domínio de problema coberto pela aplicação. Pode-se especificar, por exemplo, que um agente tem permissão para executar uma ação se este tem poder institucional para tal. Além disso, a especificação de *social constraints* envolve a definição de políticas para sanção e *enforcement* na execução de ações. Esta definição também é específica conforme a aplicação.

Os *social roles* são conjuntos de condições necessárias para que agentes adotem papéis e conjuntos de regras que descrevem o comportamento que o agente deve ter enquanto cumprir um determinado papel. Estas condições são específicas da aplicação. Mas, na proposta, é salientada a existência de um protocolo de adoção de papéis, bem como a existência de um agente com poder de permitir ou não a adoção de papéis. As regras que regem o comportamento do agente durante o cumprimento de um determinado papel são expressas através de fórmulas que definem quais estados um agente pode iniciar ou terminar, além de definir os poderes e permissões adquiridos durante a adoção de um papel.

Os *social states* representam o estado do SMA, contendo informações sobre os poderes institucionais e estados normativos dos agentes, informações sobre o estado das instituições que compõem o SMA e informações sobre o estado do ambiente.

Com base nesse *framework* teórico, foi desenvolvido um *framework* computacional chamado *Society Visualiser*, que mantém o estado global de uma sociedade de agentes. Um módulo controlador captura mensagens dos agentes e eventos gerados no sistema, que são submetidos a um compilador de estados sociais. Este compilador armazena, além das informações recebidas do sistema, uma base de dados contendo as regras que determinam as mudanças de estado do sistema. Estas regras são representadas na forma de *Event Calculus* em linguagem *Prolog*.

3.3. AGRE, AGREEN e MASQ

O modelo AGRE, proposto em [Ferber et al. 2005], parte do princípio de que as interações entre os agentes acontecem através do ambiente. Este ambiente pode ser particionado em *spaces* que abrigam partições do ambiente, divididas em duas categorias: física (*physical space*) e social (*social space*). Como exemplo, podemos citar um cenário envolvendo uma partida de futebol. O *space* físico seria a parte do mundo que envolve elementos como campo de futebol, bola e uniformes das equipes. O *space* social seria o conjunto de regras necessárias à realização da partida, como, por exemplo, definição das equipes, esquemas táticos, regras da partida e fórmula do campeonato.

Os aspectos físicos e sociais do sistema, no modelo AGRE são colocados em um mesmo nível de abstração, ambos sendo considerados aspectos pertencentes ao ambiente, pois, segundo os autores, as características fundamentais destes elementos são semelhantes: o estado dos *spaces* independe da percepção dos agentes e modifica-se mediante à ação dos agentes. Além disso, os elementos que compõem o *spaces* tem natureza reativa e determinista.

Segundo a crítica exposta por [Báez-Barranco et al. 2007], o modelo AGRE é deficiente por não tratar, de forma apropriada, da criação de fatos institucionais e, em busca de suprir essa deficiência, foi proposto o modelo AGREEN. Neste modelo, o conceito de *space* é explorado e refinado, de forma a possibilitar a modelagem de influências entre diferentes *spaces* através de ligações (*links*).

Estas ligações podem ser: (i)Causais: definem a ocorrência de eventos locais em função de eventos ocorridos em outro *space*; (ii)Lógicas: definem a alteração de propriedades locais em função de propriedades verificadas em *spaces* externos; (iii)Sociais: relações entre *spaces* físicos e sociais, com significado semelhante às regras *count as*.

Os tipos de ligações que um *space* admite definem sua classificação em físico ou social: um *space* é considerado “social” se possuir ligações sociais com outros *spaces*. A definição destas ligações, no entanto, não concretizou, efetivamente, a implementação do operador *count as* no modelo AGREEN.

Os conceitos desenvolvidos no modelo AGREEN foram utilizados por [Stratulat et al. 2009] na proposta do modelo MASQ, que trata da realidade institucional através da conjunção de aspectos objetivos e subjetivos. Neste modelo, as interações com infraestruturas que modelam aspectos ambientais (tanto físicos quanto sociais), com caráter objetivo, são classificadas como interações externo-coletivas, e se concretizam

através dos *spaces*. Além deste tipo de interação, são definidas interações internocoletivas, de natureza subjetiva, que descrevem como um conjunto de agentes descreve e interpreta a parte objetiva do sistema. Essa interpretação é feita pelos agentes através daquilo que os autores do modelo chamaram de *cultura*. Cultura, nesse caso, pode ser explicada como sendo o conjunto de objetos utilizados pelos agentes de uma instituição para interpretar a realidade bruta. Ontologias e normas são exemplos de tais objetos. A realidade institucional, criada a partir da cultura em que o agente está inserido, foi chamada pelos autores do modelo MASQ de *cultural space*. Cada agente, segundo essa proposta, tem uma interpretação particular da realidade bruta pois diferentes agentes inseridos em uma mesma cultura podem ter percepções parciais e diferentes entre si dos fatos brutos.

Com este modelo, é possível aplicar *constitutive rules*, no formato $X \text{ count as } Y \text{ in } C$, em que X são fatos no *brute space* e Y são as interpretações feitas pelos agentes. Estas *constitutive rules* podem ser formais, possuindo alguma forma de representação no *brute space* e sendo acessíveis aos agentes, ou informais, sem qualquer representação externa, mas presente de forma consensual na mente dos agentes.

Conforme o modelo MASQ, mesmo havendo a possibilidade de haver *constitutive rules* formais, não há representação, externa à mente dos agentes, da realidade institucional. Ou seja, os fatos institucionais estão presentes apenas na mente dos agentes pertencentes a uma determinada cultura. Para exemplificar, pode-se utilizar um cenário com um agente *motorista* e um agente *policial*. Suponha-se que o motorista execute o fato bruto de passar um semáforo com sinal vermelho. No modelo MASQ, para tal situação, não existe um mecanismo que informe ao policial que houve uma infração. O que existe é a percepção, por parte do policial, de que houve o fato bruto de um motorista cruzar o sinal vermelho. Além disso, o policial tem acesso às regras que regem o *social space* em que os agentes estão inseridos. De posse dessas informações, o policial *pode* interpretar que houve uma infração.

Além disso, o modelo propõe a possibilidade de um mesmo agente estar inserido em diversas culturas. Como consequência, um mesmo fato bruto pode gerar diferentes interpretações nas diferentes culturas. Estas diferentes interpretações podem criar diferentes consequências deônticas que podem ser, inclusive, conflitantes. Eventuais conflitos na interpretação de fatos brutos devem ser resolvidos pelos agentes.

3.4. O modelo de [Piunti 2009]

Em [Piunti 2009] é exposto um modelo que trata de SMA utilizando três dimensões distintas: agentes, organizações e ambiente. Os agentes constituem as entidades computacionais autônomas, que têm capacidade, raciocinam e tomam decisões. A organização é a infraestrutura que armazena informações sobre normas, papéis, objetivos globais etc, com a finalidade de possibilitar que os agentes tenham comportamento condizente com o esperado pela organização. O ambiente, por sua vez, incorpora entidades que não são modeláveis sob a forma de agentes ou organizações, mas que fornecem suporte aos agentes para que estes busquem atingir seus objetivos. Estas três dimensões interagem entre si, seguindo os seguintes princípios básicos:

- Agentes interagem com ambiente e organização buscando atingir seus objetivos;
- Entidades organizacionais controlam o comportamento dos agentes e regulam a utilização de recursos, procurando manter o estado desejado do ambiente;

- Elementos do ambiente afetam agentes e organização.

A partir desta estrutura baseada em três dimensões distintas, surge a noção de *Embodied Organisation*, em que a organização é vista como uma parte do SMA cujo estado sofre e exerce influência do ambiente. A partir dessa interação, é possível implementar as *constitutive rules* e *regulative rules* propostas por Searle.

As infraestruturas de organização e ambiente seguem o modelo *Agents & Artifacts* (A&A) [Omicini et al. 2008], que tem por base entidades chamadas artefatos. Estes artefatos são entidades reativas que disponibilizam informações e funcionalidades aos agentes para que estes atinjam seus objetivos. Para tanto, são definidas duas infraestruturas distintas, ambas compostas por artefatos: uma infraestrutura organizacional (OMI) e uma infraestrutura ambiental (EMI). Assim, além de os agentes poderem interagir com a organização e com o ambiente, existe a possibilidade de serem definidas interações entre organização e ambiente.

Os artefatos da EMI e OMI estão inseridos em um *workspace*, que pode ser considerado uma partição do ambiente em que encontram-se os recursos, tanto físicos quanto organizacionais, necessários à realização de um determinado conjunto de atividades dos agentes. O conceito de *embodied organisation* surge dessa infraestrutura: uma organização pode estar incorporada em uma estrutura mais elaborada em que os fatos que modificam o estado da organização são originados no ambiente e alguns fatos verificados no ambiente são consequência de eventos ocorridos na organização.

Essa ligação entre as dimensões ambiental e organizacional concretiza aspectos da construção da realidade social proposta por Searle: eventos ocorridos no *workspace* transformam-se em fatos institucionais através de *constitutive rules*. De forma semelhante, eventos ocorridos na organização podem ter reflexos no ambiente, segundo o modelo, podem ser implementados através de *enact rules*.

A proposta apresentada pelos autores foi elaborada tomando por base a linguagem *Jason* para programação dos agentes [Bordini et al. 2007], *Moise* para modelagem e implementação da organização [Hübner et al. 2007] e *Cartago* para implementação do ambiente [Ricci et al. 2010]. As regras *count as* e *enact* foram descritas mas não foi descrito, até o momento, qualquer trabalho de implementação destas regras no *framework JaCaMo*.

3.5. Situated Eletronic Institutions

As *Eletronic Institutions* (EI), propostas em [Esteva 2003], são um *framework* que visa garantir o cumprimento de convenções sociais, fornecendo recursos para regulação das atividades dos agentes, buscando a manutenção do estado global desejado pela organização. Uma EI, na definição original, é autocontida, abrigando em si todos os elementos necessários a seu funcionamento.

Esta autocontenção, no entanto, conflita com a ideia de um sistema aberto, em que entidades não conhecidas de antemão (agentes, outras instituições etc) podem interagir com a instituição. Em um sistema com essa característica, a instituição precisa raciocinar a respeito do ambiente em que está inserida. A partir dessas premissas, em [Campos et al. 2009] é proposta uma extensão às EI, denominada *Situated Eletronic Institutions* (SEI). Para que essa interação entre a instituição e o ambiente seja possível, a

instituição deve ter uma representação interna do estado institucional dos agentes conhecidos, das propriedades dos elementos externos à instituição e das propriedades da própria instituição.

No modelo SEI, o ambiente é composto por tudo aquilo que não pertence à instituição e, por isso, para referir-se ao ambiente, é usada a palavra *mundo*. Para fazer a interface entre a instituição e o mundo, foram definidos elementos chamados *modelers*, pertencentes à instituição. Estes elementos obtêm informações sobre o estado de elementos externos e agem sobre estes elementos quando necessário, em função de fatos institucionais. Os fatos institucionais são gerados em função de *relevant actions*, que são ações verificadas no mundo e que, mediante à interpretação da instituição, podem ter algum significado institucional e consequências sobre o mundo. Esta interpretação das *relevant actions* pode ser considerada como o operador *count as* nas SEI.

No modelo de SEI, as instituições não possuem uma representação interna dos fatos institucionais (operador *count as*). Como consequência, a instituição não é conduzida, explicitamente, a um estado normativo. Em lugar disso, existe apenas a representação de ações a serem tomadas em função de ocorrências, no ambiente, de fatos que sejam relevantes à instituição. Pode-se utilizar, por exemplo, o fato de um automóvel cruzar um semáforo com sinal fechado. A SEI possui apenas a representação de que, diante de tal fato, um policial deve ser avisado da infração, não possuindo qualquer representação que indique que “cruzar o sinal fechado *count as* infração”.

3.6. O modelo de [Dastani et al. 2009]

Em [Dastani et al. 2009] é proposta uma abordagem baseada em normas para a geração de fatos institucionais. No modelo proposto, as ações executadas pelos agentes são observadas e seu efeito no ambiente é determinado por normas. Esse efeito pode ser diferente conforme o estado do sistema no momento da execução da ação. O SMA é composto, segundo a proposta, três conjuntos de informações: um conjunto de predicados representando os fatos brutos e, por consequência, o estado do ambiente; um conjunto de predicados representando os fatos institucionais e, por consequência, o estado institucional do sistema; o conjunto da configuração de todos os agentes que interagem com o sistema.

As ações executadas pelos agentes têm efeitos sobre os fatos brutos, modificando o estado do ambiente. A instituição pode, então, detectar efeitos normativos em função deste estado. Um exemplo de regra *count as*, citado pelos autores da proposta, tratando de um sistema de controle de submissão de artigos, é dado por

$$\text{received}(As) \text{ and } \text{member}((A, Id), As) \text{ and } \text{pages}(Id) > 15 \Rightarrow \text{viol_size}(A)$$

Conforme o exemplo, um artigo *Id* submetido pelo autor *A*, contendo mais de quinze páginas é considerado (*count as*) uma violação cometida pelo autor *A*.

3.7. O modelo de [Aldewereld et al. 2010]

Em [Aldewereld et al. 2010] é proposto um modelo que parte do princípio de que para fazer com que um fato bruto tenha significado normativo, é necessário que o operador *count as* seja utilizado pelo menos duas vezes: uma vez para obter um fato institucional a partir de um fato bruto, e uma segunda utilização para dar significado normativo ao fato institucional.

Conforme essa proposta, as regras *count as*, além de conferirem interpretações normativas a fatos brutos em determinados contextos, podem ser usadas para que os agentes façam inferências a respeito dos próprios contextos. Um exemplo, citado pelos autores da proposta, trata sobre o tratamento de situações de catástrofe: em eventos pouco críticos, ambulâncias podem ser usadas para atendimento às vítimas; em situações mais críticas, caminhões do exército também podem ser usados para tal. Dessa forma, se um agente, através de sua percepção de mundo detectar que caminhões do exército estão sendo usados para atendimento às vítimas, pode inferir que o contexto em que está inserido refere-se a um evento crítico.

Diante desse raciocínio, os agentes podem decidir quais ações adotar: em um contexto *X*, uma ação *A* pode ser considerada (*count as*) permitida; em um contexto *Y*, a mesma ação *A* pode ser considerada proibida. Dessa forma, se um agente for capaz de raciocinar sobre o contexto em que está inserido, pode ser capaz de elaborar melhores planos de ação.

A partir destas premissas, os autores da proposta citam a possibilidade de implementação de um *reasoner* para aplicação de regras *count as* implementado na máquina de regras *Drools* [JBoss], fornecendo exemplos de como é possível implementar as regras. Na implementação, os diversos contextos possíveis são definidos explicitamente; em seguida, são definidas as regras *count as* que ligam fatos brutos a fatos institucionais e, por fim, são definidas regras que ligam os fatos institucionais e fatos institucionais a normas. Apesar disso, no entanto, não é citada nenhuma aplicação concreta do *reasoner* em um SMA.

4. Análise dos modelos

Esta seção apresenta o resultado de uma análise preliminar, à luz da teoria de construção da realidade social de Searle, dos modelos descritos. Através da análise realizada, foi possível identificar uma série de aspectos, em nível conceitual, que são levados em conta na elaboração dos modelos estudados. Estes aspectos são resumidos na Tabela 1 e serão apresentados a seguir.

4.1. Dimensões

Algumas das propostas “separam” o mundo de forma a focar em partes específicas que são de domínio da aplicação. Os modelos AGRE/AGREEN/MASQ adotam o conceito de *space*, enquanto o modelo de [Piunti 2009] usa o conceito de *workspace*. Nestes modelos, as interações que podem gerar fatos institucionais requerem que os objetos envolvidos nessas interações estejam inseridos nessas partições do ambiente. O modelo SEI, por sua vez, divide o sistema, segundo o ponto de vista institucional, entre elementos que fazem parte da instituição e o restante do mundo. O particionamento do mundo tem relação direta com a forma como os diversos modelos interpretam as diferentes dimensões que compõem um SMA. Em [Artikis et al. 2002] e [Aldewereld et al. 2010], um SMA é dividido em duas dimensões: uma dimensão representando a instituição e outra representando a realidade bruta, que inclui os agentes, sem uma representação separada para os objetos que compõem o ambiente em que os agentes atuam. Os modelos de [Dastani et al. 2009], [Piunti 2009] e o MASQ dividem um sistema multiagente em três dimensões (agentes, ambiente e instituição) e, segundo esses modelos, os fatos institucionais são o produto, em nível institucional, da ação dos agentes sobre o ambiente.

Tabela 1. Comparação entre modelos

Modelo	Dimensões			Representação do Estado Institucional	Criação de F.I.		Fatos Brutos		Artefatos de Mediação
	Ag	Amb	Inst		Ev	Est	Ações dos agentes	Estado do sistema	
Artikis et al. 2002	✓			Inferência sobre a dimensão física	✓	✓	✓	✓	✓
MASQ	✓	✓	✓	<i>Cultural Spaces</i>	✓	✓		✓	✓
Piunti 2009	✓	✓	✓	Artefatos organizacionais	✓		✓		✓
SEI	✓	✓	✓	Não existe	✓		✓		
Dastani et al. 2004	✓	✓	✓	Predicados da dimensão institucional		✓	✓	✓	
Aldewereld et al. 2010	✓			Inferência sobre a dimensão física	✓		✓		

4.2. Representação do estado institucional

Existem diferenças na forma como os modelos podem fornecer um *snapshot* representando o estado de uma instituição em um determinado momento. Nos modelos de [Artikis et al. 2002] e de [Aldewereld et al. 2010], esse estado é dado pela aplicação de regras de inferência específicas da instituição, em *Event Calculus*, sobre os fatos ocorridos na dimensão física. Através dessa inferência, é possível saber quais são as obrigações, proibições, permissões e papéis atribuídos aos agentes. O modelo MASQ utiliza o conceito de *cultural space*, que é a representação, por parte dos agentes, da realidade institucional. Segundo o modelo, não existe realidade institucional fora da mente dos agentes e, como consequência, pode-se afirmar que a realidade institucional, nesse caso, seria a síntese das interpretações feitas por todos os agentes. No modelo de [Piunti 2009], o estado institucional pode ser verificado através das propriedades observáveis dos artefatos organizacionais. No modelo SEI não há representação do estado institucional, uma vez que as consequências dos fatos brutos, segundo tal proposta, são convertidas, pela instituição, em normas que são imediatamente repassadas aos agentes. O modelo de [Dastani et al. 2009] armazena um conjunto de predicados que descreve o estado da instituição.

4.3. Criação de fatos institucionais

Outro aspecto observado nos modelos foi o tipo de informação utilizada para a criação de fatos institucionais. No modelo de [Artikis et al. 2002], os fatos institucionais são gerados a partir de eventos que, sob o contexto do estado do sistema, criam novos estados normativos. Uma abordagem semelhante é utilizada no modelo MASQ, em que tanto

eventos quanto o estado do sistema disparam a criação de fatos institucionais. No modelo de [Piunti 2009] e no SEI, no entanto, a criação de fatos institucionais se dá somente através da ocorrência de eventos no ambiente. No modelo de [Dastani et al. 2009], os fatos são criados através da análise do estado do sistema, que é avaliado ciclicamente e evoluí através de ações dos agentes. E, por fim, no modelo de [Aldewereld et al. 2010], os fatos institucionais são criados a partir do monitoramento dos eventos ocorridos ao longo da execução do sistema.

4.4. Fatos brutos

Percebe-se, nos modelos apresentados, que os fatos brutos, geradores de fatos institucionais, podem ser divididos em três categorias:

- Ações dos agentes: neste caso, uma ação do agente pode gerar um fato institucional. Um sacerdote que oficia um casamento, por exemplo, através de sua fala, cria o fato institucional do casamento. Deve-se ressaltar que isso deve estar suportado pela aceitação coletiva da autoridade do agente, conforme Searle. Em alguns modelos, como o de [Piunti 2009], para que a ação de um agente possa gerar um fato institucional, é necessário que esta seja executada sobre um objeto existente no ambiente.
- Estados dos agentes: fatos institucionais podem ser decorrentes do estado de um agente. Cita-se, por exemplo, uma pessoa que tem setenta anos de idade e, em determinada instituição, é considerada idosa;
- Passagem do tempo: a passagem do tempo pode ser um fato bruto que determina a criação de fatos institucionais.

4.5. Artefatos de mediação

Algumas áreas de estudo da interação humana, como a *Teoria da Atividade* e a *Cognição Distribuída* deram origem ao conceito de “artefato de mediação”, que são objetos que intermediam a interação entre as pessoas [Omicini et al. 2008]. No contexto de SMA também pode-se utilizar artefatos que mediem as interações no sistema. O modelo de [Piunti 2009] tem por base a utilização de tais artefatos para mediar a ação dos agentes sobre o ambiente e transformar tais ações em fatos institucionais; no MASQ, os *social spaces* exercem o papel de artefatos de mediação de tal interação.

Os demais modelos, apesar de utilizarem a ideia de que fatos brutos geram fatos institucionais, não utilizam artefatos de interação para representar explicitamente os objetos que compõem o ambiente. Não existe, por exemplo, a representação de um objeto “sala de aula” que, ao sofrer uma ação “entrar”, confira ao agente o papel de aluno. O que existe é a representação de que um agente, executando a ação entrar em um objeto porta, passará a ter as obrigações, permissões e proibições de um aluno.

4.6. Implementação

Entre os modelos estudados, verificou-se que não há implementação concluída e disponível para utilização em SMA com interação entre organização e ambiente. O modelo de [Artikis et al. 2002] descreve uma aplicação específica utilizada para exemplificar os aspectos teóricos, porém sem referência à disponibilização para utilização e integração com outras ferramentas. O MASQ foi descrito como um modelo e os autores aplicaram o modelo ao jogo *Warboot*, que faz parte da plataforma *MadKit* [MadKit 2011]. O

modelo de [Piunti 2009], apesar de ser baseado em ferramentas já desenvolvidas, como *Jason*, *Moise* e *Cartago*, não possui implementação em que interações entre artefatos organizacionais e ambientais possam ser modeladas e implementadas com facilidade. O modelo de [Dastani et al. 2009] e o modelo SEI também não possuem implementação relacionada descrita. O modelo de [Aldewereld et al. 2010] apresenta o exemplo de uma implementação de um *reasoner* para ligar fatos brutos a fatos institucionais. Não é apresentado, no entanto, qualquer exemplo de integração do *reasoner* com agentes atuando em um SMA. Os autores da proposta citam tal integração, inclusive, como uma perspectiva de trabalho futuro.

5. Considerações Finais

Os modelos analisados neste artigo utilizam as ideias de Searle com relação à criação de fatos institucionais a partir de fatos brutos, aplicando ideias relacionadas ao estudo das sociedades humanas a sistemas multiagente.

Sistemas multiagentes exigem precisão, lógica e formalismo rigorosos na elaboração dos modelos a serem implementados. Esse maior rigor faz com que a realidade social de Searle reflita-se, nos modelos estudados, apenas como uma adaptação, com semelhanças e diferenças. Essa adaptação implica também, que os modelos estudados apresentem diferenças entre si.

Não pretendeu-se, neste artigo, qualificar as diferenças para eleger o melhor modelo. Buscou-se apenas traçar um paralelo entre as diferentes abordagens para a criação de fatos institucionais. Através desse paralelo, foi possível perceber diferentes aspectos, com diferentes abordagens, que implicam em diferentes maneiras de modelar e implementar a criação de fatos institucionais a partir de fatos brutos em sistemas multiagentes. Pode-se citar como exemplo questões como a utilização de uma representação explícita e separada dos objetos que compõem o ambiente, a criação de fatos institucionais através de eventos ou estado e a utilização de artefatos de mediação.

Estas diferenças, apresentadas através de considerações a respeito dos modelos e pode servir como base para a definição de pontos de projetos que podem ser usados por base para avaliação e definição de novos modelos relacionados ao assunto.

Referências

- Aldewereld, H., Álvarez-Napagao, S., Dignum, F., and Vázquez-Salceda, J. (2010). Making norms concrete. In van der Hoek, W., Kaminka, G. A., Lespérance, Y., Luck, M., and Sen, S., editors, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 807–814, Toronto, Canada. International Foundation for Autonomous Agents and Multiagent Systems.
- Artikis, A., Pitt, J., and Sergot, M. (2002). Animated specifications of computational societies. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 3 - AAMAS '02*, pages 1053–1062, New York, New York, USA. ACM Press.
- Báez-Barranco, J., Stratulat, T., and Ferber, J. (2007). A unified model for physical and social environments. In *Proceedings of the 3rd international conference on Environments for multi-agent systems III*, pages 41–50, Berlin, Heidelberg. Springer-Verlag.

- Bordini, R. H., Wooldridge, M., and Hübner, J. F. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason* (Wiley Series in Agent Technology). John Wiley & Sons.
- Campos, J., López-Sánchez, M., Rodríguez-Aguilar, J., and Esteva, M. (2009). Formalising situatedness and adaptation in electronic institutions. In Hübner, J., Matson, E., Boissier, O., and Dignum, V., editors, *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, volume 5428 of *Lecture Notes in Computer Science*, pages 126–139. Springer Berlin / Heidelberg.
- Dastani, M., Tinnemeier, N., and Meyer, J.-J. (2009). *A programming language for normative multi-agent systems*, chapter XVI, pages 397–417. Information Science Reference, Hershey, PA, USA.
- Esteva, M. (2003). *Electronic institutions. from specification to development*. PhD thesis, Universitat Politecnica de Catalunya.
- Ferber, J., Michel, F., and Baez, J. (2005). AGRE: Integrating environments with organizations. *Environments for Multi-agent Systems*, pages 48–56.
- Hübner, J., Sichman, J., 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):370–395.
- JBoss, C. Jboss drools business rules. <http://www.jboss.org/drools>.
- Jones, A. and Sergot, M. (1996). A formal characterisation of institutionalised power. *Logic Journal of IGPL*, 4(3):427.
- MadKit (2011). The madkit project (a multi-agent development kit).
- Omicini, A., Ricci, A., and Viroli, M. (2008). Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456.
- Piunti, M. (2009). *Situating agents and organisations in artifact-based work environments*. PhD thesis, Universit` di Bologna.
- Ricci, A., Piunti, M., and Viroli, M. (2010). Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*, pages 1–35.
- Searle, J. (1995). *The Construction of Social Reality*. Free Press.
- Shanahan, M. (1999). The event calculus explained. In Wooldridge, Michael J. and Veloso, M., editor, *Artificial intelligence today: Recent trends and developments*, pages 409–430, Berlin, Heidelberg. Springer-Verlag.
- Stratulat, T., Cedex, M., Ferber, J., and Tranier, J. (2009). MASQ : towards an integral approach to interaction. In *Proc. of 8th Conf. on Agents and Multi Agent Systems (AAMAS-09)*, volume 2, pages 813–820.

Protocolo de Controle Organizacional em Sistemas Multiagente

Mauricio Rocha¹, Fabiana Lorenzi¹, Stanley Loh^{1,2,3}

¹Universidade Luterana do Brasil (ULBRA)
A. Farroupilha, 8001 - Canoas – RS – Brasil

²Universidade Católica de Pelotas (UCPEL)
R. Félix da Cunha, 412 – CEP 96010-000, Centro, Pelotas – RS

³Faculdade de Tecnologia SENAC-RS
R. Gonçalves Chaves, 602 – CEP 96015-560, Centro, Pelotas - RS

mauriciorocha665@gmail.com, fabilorenzi@gmail.com, sloh@terra.com.br

Abstract. This paper presents the study of a technique for implementation of organization control systems. The proposed protocol is based on MASPA and is able to perform searches in the agent network, finding the agent capable of performing the task with a low cost. Experiments were conducted to validate the presented protocol.

Resumo. Este artigo apresenta o estudo de uma técnica para a implementação de controle organizacional em sistemas multiagente. O protocolo proposto é baseado em MASPA e é capaz de efetuar buscas dentro de sua rede, encontrando assim o agente mais capacitado de realizar uma determinada tarefa com um custo reduzido de mensagens na rede. Experimentos foram realizados para validar o protocolo apresentado.

1. Introdução

No dia-a-dia todos precisam se relacionar com outros seres humanos para exercer seu trabalho e conquistar seus objetivos. É de conhecimento de todos que ao relacionar-se com outro ser humano o indivíduo deve respeitar regras e apresentar determinado comportamento conforme o perfil da pessoa com quem está falando. A sociedade se baseia em regras de cooperação e competição entre diversos grupos que nela convivem, tendo cada grupo sua visão da sociedade.

Um dos principais desafios no desenvolvimento de sistemas multiagente é encontrar mecanismos de coordenação entre agentes com objetivos e visões diferentes. Gerar uma visão mais ampla de uma rede de agentes e passar esta informação a agentes que possuem apenas uma visão parcial da rede é um dos grandes desafios para gerar soluções eficientes dentro de sistemas distribuídos complexos. Pela dinâmica envolvida no sistema deve-se avaliar um método que permita técnicas de tentativa e erro para que os agentes consigam encontrar seu lugar e exercer suas atividades. Assim, cada agente pode ser encaixado em novos grupos e, após muitas avaliações negativas do seu

desempenho, deixar de ser usado para determinado objetivo. O agente é reavaliado e enviado para outro grupo onde seu perfil seja mais adequado, assim como na sociedade humana.

Indivíduos de maior sucesso tendem a guiar, gerenciar e supervisionar outros indivíduos. Conforme sobem na escala social estes supervisores desenvolvem uma visão cada vez mais ampla da sociedade à qual pertencem, auxiliando os indivíduos abaixo dele e indicando indivíduos quando alguém de nível superior (supervisor de supervisores) precisar de um indivíduo com um determinado conhecimento e habilidade.

O presente trabalho tem como principal objetivo apresentar uma proposta de desenvolvimento de uma técnica de controle organizacional e simulá-la para verificar sua eficácia na busca de informações em uma rede de agentes. Esta técnica de controle organizacional tem como objetivo ser uma solução a ser seguida por profissionais que necessitem deste tipo de controle em suas aplicações, sendo um padrão que envolve os agentes sem grande complexidade e necessidade de alterações no código do mesmo, apenas adição de código simples para funções específicas.

Este artigo está organizado da seguinte forma: a seção 2 apresenta os principais conceitos necessários para a compreensão do trabalho apresentado. A seção 3 apresenta a solução proposta baseada em MASPA, onde discute-se sobre o projeto, as metodologias e como funciona a técnica de controle organizacional. Os resultados obtidos são apresentados na seção 4 e por último são apresentadas as considerações finais.

2. Referencial Teórico

Esta seção apresenta os principais conceitos utilizados neste trabalho.

2.1 Aprendizado por reforço e Trust

Aprendizado por reforço é uma técnica de IA que visa a criação de um sistema que aprende através de respostas obtidas do ambiente em que ele executa uma ação (Sutton & Barto, 1998). Se esta ação gerar uma resposta negativa em relação ao seu objetivo ela será utilizada como uma experiência que demonstra que aquela ação não deve ser utilizada em determinado ambiente. Se a ação gerar resposta positiva esta será avaliada como uma ação interessante que pode ser usada novamente para obter bons resultados naquele ambiente. O sistema prossegue até encontrar uma ação que corresponda da melhor forma dentro do nível de precisão que sistema requer, podendo se adaptar a mudanças no ambiente (Sutton & Barto, 1998).

O aprendizado por reforço tem sido muito pesquisado em sua forma distribuída, sendo assim chamado de aprendizado por reforço multiagente. Os estudos de aprendizado por reforço distribuído geralmente possuem foco em como distribuir a experiência dos agentes com bom nível de treinamento em determinada área entre os agentes menos treinados. Como encontrar de forma eficaz os agentes treinados em sistemas sob arquitetura *peer-to-peer* (P2P) é também um dos grandes desafios dos sistemas de aprendizado por reforço multiagente (Sims *et al.*, 2004). Para isto têm sido pesquisadas técnicas de controle organizacional.

Trust é uma técnica de inteligência artificial que pode ser definida como uma expectativa ou uma chance em que um agente pode confiar em outro agente. Esta

técnica tem sido estudada por áreas que vão além da ciência da computação, como a psicologia, sociologia, ciências cognitivas e economia. (Ramchurn *et al.*, 2004)

Exemplos de aplicação de *Trust* são, principalmente, sistemas que focam nas relações de confiança entre agentes em um sistema multiagente onde as relações de confiança podem variar com o tempo ou outros fatores.

Trust propõe uma reavaliação da relação entre os agentes de um sistema multiagente a fim manter somente as relações que não prejudiquem os agentes, evitando assim retrabalho e troca de informações falsas. *Trust* é atualmente utilizado para verificar as relações de confiança entre agentes em sistemas de comércio eletrônico, sistemas de recomendação, sistemas P2P em geral, entre outros. (Walter *et al.*, 2008)

2.2 Controle organizacional

Controle organizacional é uma área de sistemas multiagente que visa organizá-los e estruturá-los de forma a atingir objetivos da forma mais eficiente possível. Este controle se baseia em uma estrutura de vários níveis na qual os objetivos organizacionais, papéis e responsabilidades dos agentes são desenvolvidos, distribuídas e mantidas dinamicamente. Desta forma os agentes podem, individualmente, tomar decisões respeitando regras da estrutura como um todo. (Zhang *et al.*, 2009)

Controle organizacional é um conceito que vem de ciências da área humana como administração e antropologia. Este conceito é utilizado em outras ciências para definir os estudos sobre organização de pessoas em sociedade de forma que todos trabalhem fazendo sua parte em um sistema maior. Também, focando na área administrativa trata-se no modo em que funcionários de uma empresa podem ser coordenados de forma a atingirem o melhor nível de eficiência em suas tarefas e, quando não forem capazes de realizar determinada tarefa, passar esta para o profissional ou departamento responsável. Esta coordenação se baseia na correta separação de departamentos e políticas empresariais, fazendo com que profissionais com diferentes níveis de conhecimento da empresa onde se encontra ou de suas atividades saibam cuidar de suas tarefas.

Técnicas de controle organizacional têm tido grande enfoque ultimamente para melhorar o desempenho em sistemas multiagente que utilizam arquitetura P2P. Através do controle organizacional os agentes possuem uma melhor visão do sistema como um todo tendo menor esforço em tarefas de busca na rede e negociação com outros agentes (Zhang *et al.*, 2009).

Exemplos de uso de técnicas de controle organizacional são mais facilmente encontrados em sistemas P2P devido a sua maior necessidade de organizar informações através de agentes inteligentes. Essas técnicas são bastante recentes e ainda não são usadas em larga escala para ambientes de produção. Porém no meio acadêmico existem muitos protótipos de sistemas que utilizam estas técnicas para sistemas de recomendação, redes de sensoriamento e interpretação adaptativa de rastreamento de veículos, busca de informações em sistemas P2P e roteamento de pacotes (Zhang *et al.*, 2009).

2.3 MASPA

MASPA (*Multi-Agent Automated Supervisory Policy Adaptation*) é uma especificação de *framework* para a formação de uma estrutura multinível de controle organizacional

baseada em agentes supervisores. O objetivo da especificação é aumentar a escalabilidade de sistemas multiagente que utilizam aprendizado por reforço. Este aumento da escalabilidade se dá devido a melhorias quanto ao tempo de convergência de algoritmos de aprendizado por reforço através de interação mais eficaz entre os agentes. (Zhang *et al.*, 2009)

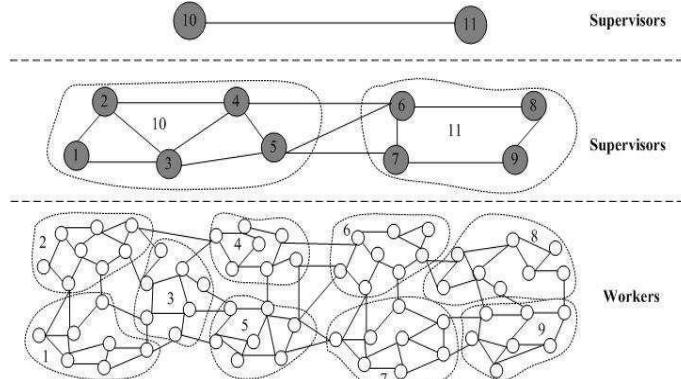


Figura 1: Arquitetura básica da rede (Zhang *et al.*, 2009)

A figura 1 é um exemplo de estrutura organizacional MASPA onde existem 3 camadas: uma camada do nível mais baixo onde a maioria dos agentes ficam, uma camada supervisora onde para cada grupo de agentes de baixo nível há um supervisor que receberá informações destes, e uma camada de supervisores onde há um supervisor para cada grupo de supervisores.

A especificação define uma estrutura organizacional multinível para supervisão automatizada e um protocolo de comunicação para trocar informações entre agentes de nível baixo e agentes supervisores. Os estados abstratos dos agentes de nível baixo são enviados para seus superiores para que estes gerem uma visão mais ampla da rede. Esta visão mais ampla será usada para criar informações de supervisão que serão enviadas aos níveis mais baixos, difundindo assim o conhecimento da rede. (Zhang *et al.*, 2009)

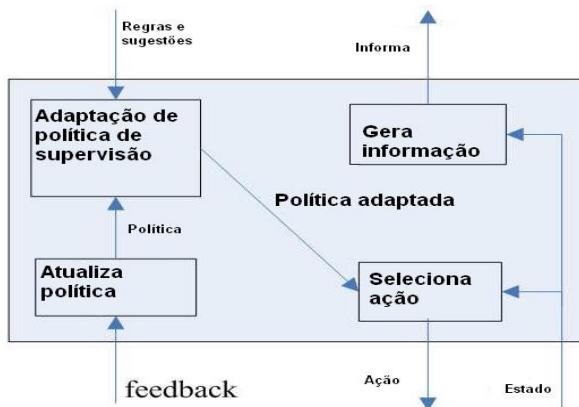


Figura 2: Fluxo de dados na rede MASPA (adaptada de Zhang *et al.*, 2009)

Para que haja esta troca de informações é definido um protocolo com 3 tipos de mensagens: regra, sugestão e informação. Regras restringem ações de comportamento dos subordinados de um determinado supervisor, não sendo flexíveis. Sugestões

indicam a preferência do supervisor para um determinado comportamento de seus subordinados. Informações são os dados do agente subordinado sobre suas atividades, enviados ao agente supervisor. Este conjunto de mensagens, embora simples pode ser implementado em diversos sistemas para formar um protocolo de troca de informações entre agentes subordinados e supervisores. A figura 2 ilustra o fluxo das informações baseadas nesse protocolo.

3. Desenvolvimento do protocolo MASPA para controle organizacional

Informações turísticas podem ser encontradas em diversos *websites* da internet. Diversos portais e *sites* de agências de turismo possuem muitas informações úteis, porém estas informações podem conter distorções de um *site* para outro. Em geral as fontes de informações mais completas são as próprias entidades responsáveis pelo turismo no local. Para isto cada entidade responsável costuma desenvolver um *site* próprio com informações voltadas ao turismo.

Embora o desenvolvimento de um site seja uma estratégia de *marketing* correta, é uma iniciativa que, em geral, é feita de forma muito individual, não havendo uma interligação com outras agências. O usuário que não sabe o que cada cidade pode oferecer teria que utilizar de um mecanismo de busca que, por não ser focado em turismo, pode não lhe prover informações confiáveis. Para buscar apenas informações confiáveis seria necessário um sistema de busca promovido pelos próprios órgãos competentes.

Este trabalho apresenta o desenvolvimento de um protocolo baseado em MASPA para controle organizacional sobre um sistema multiagente. Os agentes deste sistema possuem objetivos diferentes, porém todos recebem uma mesma informação de entrada: a busca do usuário. Devido aos custos e à complexidade de desenvolvimento e implantação de um protocolo completo de acordo com as regras definidas no MASPA a proposta deste trabalho é de prover uma versão simplificada da técnica de controle organizacional. Embora não aplique todas as regras, a proposta é de que o protocolo expansível de acordo com as necessidades e a complexidade do sistema que o utiliza. Desta forma podendo ser desenvolvidas novas capacidades de controle organizacional, tanto como módulos genéricos quanto soluções customizadas para redes específicas. As buscas se baseiam em palavras-chave enviadas à rede em formato texto. A resposta é retornada em formato texto.

O problema apresentado baseia-se em um ambiente onde existem diversos agentes inteligentes com diversas especialidades capazes de analisar uma busca do usuário e, a partir de suas próprias informações oferecem uma resposta. Através de um agente as informações sobre diferentes assuntos e interesses são solicitadas ("Que horas são na China?" ou "Onde fica a obra 'O triunfo da morte' de Bruegel?"). Obviamente cada pedido requer um agente específico que possui a informação necessária. Por exemplo, no caso da obra de Bruegel, o museu do Prado, na Espanha, poderia disponibilizar um agente que responde às perguntas através de informações sobre as obras nele situadas. Baseando-se no fato de que cada agente funciona de um determinado modo e foi desenvolvido por uma equipe diferente, encontrar estes agentes em um sistema empresarial de grande porte, ou na internet, seria bastante problemático e, provavelmente, impraticável sem o controle organizacional adequado.

3.1 O Protocolo

O protocolo desenvolvido baseia-se em um agente inicial conectado a uma rede de agentes podendo enviar requisições de busca dentro dela. Este agente pode lidar com questões como apresentar alternativas quando a rede enviar muitas respostas para a mesma pergunta, sendo possível utilizá-lo também para gerar uma *interface* na qual o usuário pode escolher o que ele acha que irá lhe interessar, e guardar um breve histórico de mensagens anteriores para manter estado conversacional simples.

A rede não possui regras quanto à entrada e saída de agentes, sendo que novos agentes podem entrar e sair na rede a qualquer momento. Aos agentes é delegado um protocolo básico de *trust* baseado em aprendizado por reforço. Este algoritmo de *trust* deve ser executado somente pelos agentes supervisores. Para estar de acordo com os padrões do protocolo a arquitetura da rede sempre possui no mínimo 3 camadas: agente inicial, níveis de supervisionamento (mínimo 1) e nível dos agentes.

O agente inicial localiza-se no topo da cadeia de supervisores da rede. As solicitações feitas ao agente de entrada são repassadas ao(s) supervisor(es) de mais alto nível e então chegam até os agentes seguindo a estrutura de supervisionamento. Conforme a figura 3, cada supervisor analisa se é possível que haja alguém no seu grupo capaz de trazer uma resposta adequada à entrada recebida. Caso exista, ele envia a mensagem a todos os agentes e estes agentes retornam um valor de estímulo.

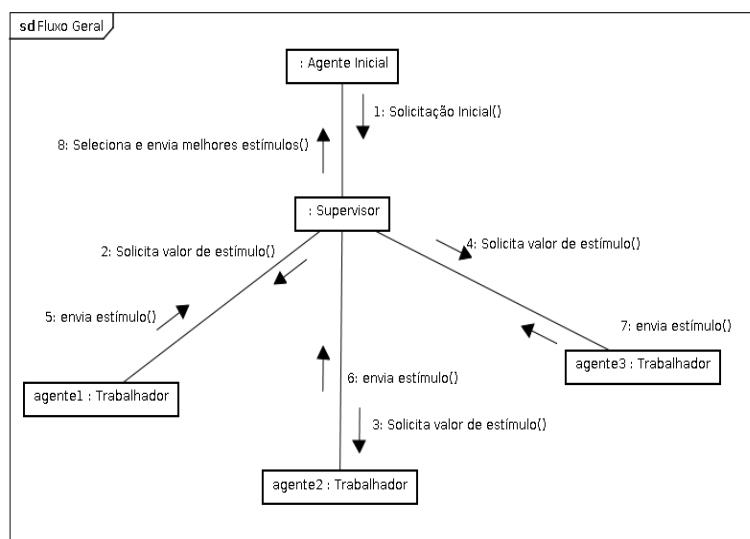


Figura 3: Processo de busca e seleção de agentes.

O valor de estímulo representa o quanto o agente se auto-avalia capaz de receber a resposta. Este valor é então repassado aos níveis superiores e é definido no supervisor de mais alto nível qual ou quais os agentes que responderão à pergunta. Após estes agentes serem chamados e a pergunta ser efetivamente respondida, a resposta é passada ao usuário final e este avalia se a resposta foi boa para sua necessidade, este processo é demonstrado na figura 4. Caso a avaliação seja positiva, o agente ou os agentes que responderam recebem pontos positivos dos supervisores indicando que são confiáveis. Caso contrário, eles recebem pontos negativos indicando que não são confiáveis e podem estar enviando estímulos errados para serem priorizados.

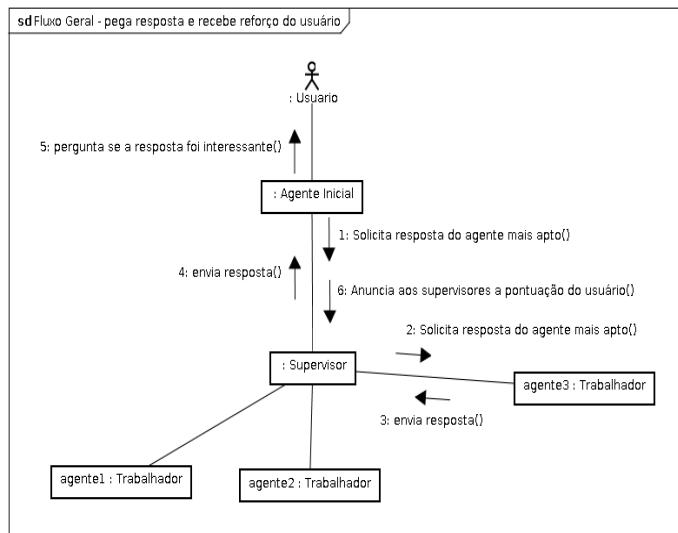


Figura 4: Processo de solicitação de resposta do agente.

O agente inicial tem o objetivo de centralizar a comunicação do usuário com a rede, não sendo necessário que ele participe ativamente da busca. Ao receber uma entrada, o agente repassa aos supervisores de nível inferior que ele considera capazes de atender à solicitação. Caso ele seja o supervisor de nível mais inferior ele envia uma solicitação de estímulo para cada agente trabalhador, coleta as respostas, processa as de maior significância e repassa estes valores para os supervisores de nível superior. Ao receber valores de estímulo de diversos supervisores de nível inferior o supervisor de nível superior efetua uma triagem e repassa aos seus supervisores. Este encadeamento faz com que apenas as respostas dos agentes mais significantes à entrada sejam enviadas ao agente inicial.

Após o agente inicial selecionar o agente mais capaz para responder à pergunta solicitada ele pode enviar a solicitação de resposta ao agente trabalhador, seguindo a hierarquia. Neste momento os supervisores apenas roteiam a mensagem pela rede, criando a melhor rota do ponto de vista hierárquico. A resposta é enviada de baixo para cima da mesma forma que foi enviada a pontuação de estímulo. Após todo o processo o supervisor recebe a informação sobre aquele agente específico que respondeu à entrada, armazenando então o valor de recompensa para ser usado posteriormente em uma verificação da confiança a ser depositada neste agente.

A chave do bom funcionamento desta rede de sistemas multiagente se baseia no fato que os supervisores constantemente recebem informações sobre os agentes trabalhadores de seus grupos e dos supervisores, dependendo da posição em que se encontram já que eles apenas se comunicam com suas duas camadas imediatas. Estas informações enviadas são guardadas para que sejam usadas como informações gerais do grupo em que estão.

Os agentes trabalhadores são os agentes que realmente possuem o conhecimento que pode ser usado para responder de forma correta à entrada do usuário. Apesar de estarem obrigatoriamente no nível mais baixo da hierarquia eles podem ser promovidos a supervisores ou até mesmo agentes iniciais dependendo das características da rede à qual pertencem. Embora possa haver essa promoção de um posto a outro o agente

trabalhador continua estando no nível inferior, apenas assumindo atividades em outro nível da hierarquia, possuindo então a característica de poder estar sendo referenciado em até 3 posições ao mesmo tempo. Assim como os outros tipos de agentes este não pode transgredir a hierarquia da rede, isto é, deve sempre exercer as atividades de forma isolada.

Frequentemente o agente trabalhador deve enviar informações sobre si para seu supervisor. Estas informações devem ser detalhadas e em formato compatível com o formato usado pelos outros agentes trabalhadores, podendo ser diferente em cada domínio de negócio que a aplicação se encaixa. Estas informações são utilizadas para descrever o agente ao seu supervisor. Para o envio destas informações recomenda-se fortemente o uso de mensagens de sugestão, regra e informação definidos no modelo MASPA, embora em muitas redes não será interessante utilizar os 3 tipos por não haver necessidade.

Ao receber uma solicitação de valor de estímulo o agente deve receber a entrada e, através de rotina própria, retornar um valor de estímulo compatível com os valores da rede. Esta resposta é enviada ao seu supervisor imediato e sobe pela hierarquia. No momento em que o agente recebe uma solicitação de resposta ele responde ao seu supervisor imediato baseado na entrada enviada. Um agente trabalhador nunca recebe uma resposta de recompensa em relação a uma resposta que tenha enviado, apenas supervisores podem ter esta visão. Se o agente é supervisor e trabalhador ele deve interpretar ambos os papéis independentemente, não sendo permitido que o valor de recompensa sirva para manipular futuras respostas.

5. Validação do protocolo

Para validar o protocolo proposto diversos experimentos foram realizados, com o objetivo de contar o número de mensagens enviadas e avaliar os custos computacionais envolvidos em cada busca dentro da rede de agentes.

As avaliações foram feitas inicialmente em uma rede com 15 agentes, sendo esta composta pelo agente inicial e 4 supervisores com 6 agentes trabalhadores (agentes folha) cada. Os testes foram realizados na rede com o protocolo desenvolvido e com o protocolo de *broadcast*, e os resultados obtidos em ambos foram comparados. Como será observado nos resultados da pesquisa, o *broadcast* pode dar uma visão errada de baixa quantidade de mensagens enviadas em redes pequenas.

Ambos os protocolos foram testados em redes fictícias geradas em um simulador. Este simulador foi desenvolvido especialmente para esta validação.

4.1 O Simulador

A validação do protocolo desenvolvido foi realizada através de um simulador de uma rede seguindo o padrão MASPA de um modo simplificado. A rede possui agentes com as seguintes características:

- não mantém estado conversacional;
- todos os agentes são compatíveis com o protocolo definido pela rede;
- respeita as regras e protocolos específicos da rede.

Os valores de entrada são sempre texto devido à característica da aplicação de busca. O valor de estímulo é definido em formato de porcentagem, podendo utilizar até 2 casas decimais após a vírgula (ex: 95,85) e ir do valor 0 (zero) ao valor 100 (cem). O agente que se considera apto para responder corretamente à requisição tende ao valor mais alto, isto é, 100% e o agente que não se considera apto para responder à requisição tende ao valor mais baixo, isto é, 0%. Em caso de empate de valores de estímulo o agente inicial utiliza como critério de desempate a ordem do objeto de resposta na lista do agente inicial (ordem de chegada).

A estrutura da rede é estática, não podendo um tipo de agente ser dinamicamente promovido a outro posto que não seja o seu inicial. Não há plano de ação para caso de falha em um agente supervisor ou agente inicial. Inicialmente, os grupos de agentes estão separados manualmente e, após iniciada, a rede não sofre alterações. A quantidade de agentes que a rede suporta é indefinida. Na figura 5 pode-se ver a interface do simulador executando em uma requisição passo-a-passo, onde está sendo validada uma rede com 2 supervisores, 1 agente inicial e 12 agentes trabalhadores.

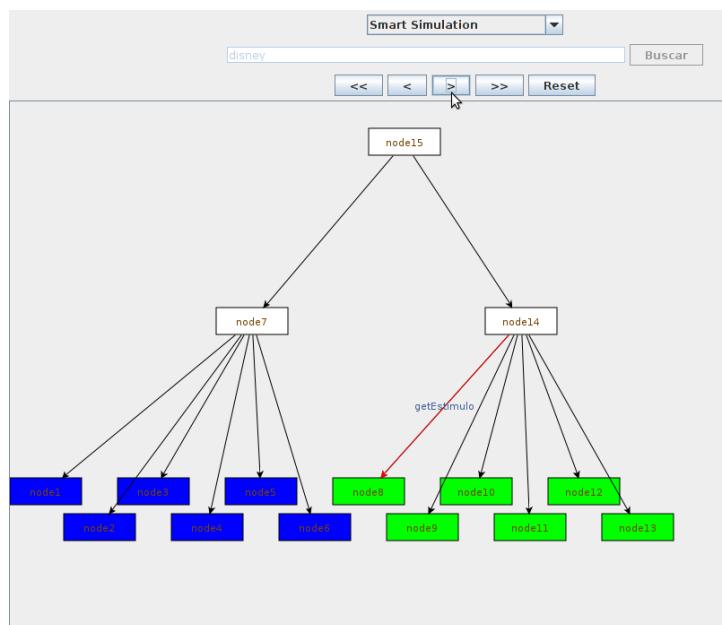


Figura 5: Screenshot do simulador

4.2 Resultados

Através do simulador desenvolvido foi possível contar o número de mensagens enviadas em diferentes situações em uma rede fictícia, assim como a verificação do fluxo das mensagens. A contagem de mensagens enviadas é importante, pois quanto menos mensagens, menor o custo para a camada rede. A simulação do protocolo em uma rede virtual permitiu a análise da ordem de envio dos dados na rede e verificação de possíveis melhorias relativas ao fluxo das chamadas entre agentes.

A quantidade de mensagens enviadas pelo protocolo foi comparada à quantidade de mensagens enviadas por uma busca em *broadcast*. É importante ressaltar que no modo *broadcast* não está sendo utilizada nenhuma técnica de *trust*, de modo a evitar o envio de mensagens não relacionadas diretamente à busca da informação na rede.

Diversas buscas foram simuladas, visando obter respostas de agentes que se localizam em diferentes regiões da rede. Por exemplo, o usuário digitou “Pateta” e a resposta foi gerada pelo agente “Disney World”.

Foram feitas 10 consultas utilizando-se palavras-chave que seriam encontradas apenas em um agente específico da rede. A média obtida no protocolo MASPA simplificado foi 28, contra 15 do *broadcast*. Devido à simplicidade do algoritmo de *broadcast*, seu resultado foi melhor, pois menos mensagens foram enviadas na busca por *broadcast*. Esta diferença de quase o dobro de mensagens na busca pelo modelo de MASPA simplificado possui duas razões: o tamanho da rede e os mecanismos adicionais necessários no modelo que utiliza MASPA.

Novos testes foram realizados, desta vez utilizando uma rede maior, e observou-se que o *broadcast* acaba enviando uma mensagem para cada agente da rede. Isto indica que a quantidade de mensagens em uma rede que utiliza *broadcast* aumenta linearmente com o tamanho da rede. No protocolo MASPA simplificado, entretanto, a quantidade de mensagens enviadas depende da correta distribuição de supervisores na rede.

Portanto ao utilizar *broadcast* em uma rede com 2.000 agentes haverão 2.000 mensagens enviadas, enquanto no MASPA simplificado a quantidade de mensagens varia de acordo com a busca e as decisões tomadas pelos supervisores da rede. Dependendo da topologia da rede e informações dos supervisores, o método inteligente obtém valores menores que o modo *broadcast*. A figura 6 demonstra graficamente a tendência de envio de mensagens conforme o tamanho da rede.

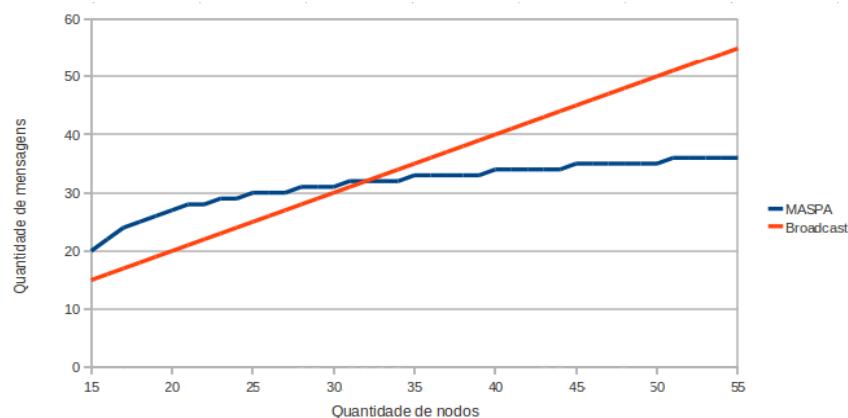


Figura 6: Tendência de envio de mensagens de acordo com o tamanho da rede

No modo *broadcast* todas as mensagens enviadas ao agente inicial estão em forma bruta, cabendo a este agente identificar o que lhe é útil. Utilizando o protocolo MASPA simplificado há uma maior filtragem dos dados e melhor distribuição de processamento dos dados pelos agentes supervisores.

Após algumas avaliações verificou-se que há casos bastante específicos onde este roteamento e filtragem das mensagens pode se tornar inconveniente. Esta inconveniência se dá quando a aplicação é simples demais e não dispõe de dados suficientes a serem enviados entre agentes a ponto de haver necessidade de filtragem. O

delay de rede entre os agentes pode tornar o modelo desenvolvido ineficaz já que ele requer inicialmente uma solicitação de estímulo, após uma solicitação de envio de resposta e, finalmente, um envio de *feedback* ao supervisor. Se as mensagens são pequenas e houver um *delay* relativamente grande na rede o MASPA implementado se mostrará 3 vezes mais lento que um simples *broadcast*, podendo inviabilizar um sistema que requer respostas extremamente rápidas.

Foram realizadas também simulações onde havia um agente mal intencionado na rede. Este agente, após enviar estímulos errados, recebeu valores baixos de *trust* e foi marginalizado na rede. Utilizando-se de *broadcast* não há como controlar este aspecto da segurança da rede, já que os supervisores apenas repassam as informações para seus subordinados sem fazer qualquer avaliação prévia. Ao utilizar-se do MASPA simplificado proposto neste artigo o agente malicioso foi ignorado, evitando trocas desnecessárias de mensagens na rede e garantindo maior qualidade das respostas.

Os resultados obtidos através do simulador e de observação do protocolo mostra que apenas em casos muito específicos de sistemas multiagente simples e com baixo volume de troca de dados a técnica de controle organizacional não obteve resultados positivos. Conforme o crescimento de uma rede de agentes pode-se observar o aumento da necessidade de uma técnica de controle organizacional para agilizar a comunicação entre os agentes. Desta forma a troca de mensagens se passa de forma mais eficaz, agregando qualidade ao sistema.

6. Considerações Finais

Após a verificação dos resultados obtidos é possível avaliar a importância da utilização de técnicas de controle organizacional em um sistema multiagente para otimização de envio de mensagens e melhor interação entre os agentes. Em sistemas multiagente de grande porte e distribuídos geograficamente a diminuição de requisições entre agentes é chave para a diminuição do tempo de resposta do sistema.

Também foi possível verificar que dar poder aos agentes inferiores para reportarem suas atividades é uma boa prática, pois permite uma maior taxa de atualização dos supervisores.

As vantagens do uso de controle organizacional não são apenas ganhos em desempenho, pois há filtragem para que apenas os agentes mais capazes sejam selecionados e enviem suas respostas. Esta filtragem gerada pelas camadas de supervisão gera maior qualidade do serviço, evitando até mesmo que haja envio excessivo de resposta, que poderia gerar um excesso de informação a ser analisada pelo agente inicial.

Com as vantagens já citadas pode-se analisar que começa a ser possível direcionar o foco do desenvolvimento de sistemas multiagente a plataformas mais complexas. Estas plataformas podendo ser compostas por agentes especializados em diversas áreas, desenvolvidos por diferentes grupos e que mesmo assim podem cooperar e complementar-se de forma transparente. Com o desenvolvimento de redes cada vez mais complexas pode ser possível até mesmo a integração entre redes de agentes. Um exemplo de integração poderia ser uma rede de informações turísticas que se integra

com uma rede de informações de hotelaria e uma rede de informações de transporte público.

Este trabalho deixa questões em aberto que apontam para trabalhos futuros, como por exemplo:

- Implementação de um sistema que utilize as técnicas aqui apresentadas e verifique os desafios durante o desenvolvimento da aplicação;
- Mecanismos mais avançados de *Trust* e tolerância a falhas;
- Separação de agentes em grupos de forma automática no contexto desse tipo de rede.

Referências

Ramchurn, S. D., Huynh, T. D. and Jennings, N. R. (2004) "Trust in Multiagent Systems". *The Knowledge Engineering Review*, 19 (1). pp. 1-25.

Sims, M., Corkill, D., Lesser, V. (2004) "Separating Domain and Coordination Knowledge in Multi-Agent Organizational Design and Instantiation". In: Proceedings of the AAAI-04 Workshop on Agent Organizations: Theory and Practice, pp. 1 - 7 , San Jose, California.

Sutton, R. & Barto, A. (1998) Reinforcement Learning: An Introduction. Cambridge, MIT Press.

Walter, F. E. et al. (2008) "A model of a trust-based recommendation system on a social network". In: Journal Autonomous Agents and Multi-Agent Systems, Amsterdam, p.57-74.

Zhang, C., Abdallah, S., Lesser, V. (2009) "Integrating Organizational Control into Multi-Agent Learning". In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, Vol: 2, pp. 757 - 764 , Budapest, Hungary.

Parte III

Resumos Expandidos

Agentes Computacionais na Simulação da Dengue

Rogério Luís Rizzi¹, Rodolfo Lorbieski¹, Carlos Henrique França¹, Leonardo Pereira Merlin¹, Claudia Brandelero Rizzi¹, André Luiz Brun¹, Reginaldo Zara¹

¹Centro de Ciências Exatas e Tecnológicas – Universidade Estadual do Oeste do Paraná (UNIOESTE) - Caixa Postal 711 – 85819-110 – Cascavel – PR – Brasil

rogeriorizzi@hotmail.com, rdlorbieski@gmail.com,
c.henrique.info@gmail.com, leoviper@gmail.com,
claudia_rizzi@hotmail.com, andrebrun@hotmail.com, razara@unioeste.br

Abstract. This paper summarizes the three main modules of a computing system whose goal is to automate specific actions aiming to provide combat and control dengue. The module I is an information system on Dengue. Modules II and III perform simulations on dengue using computational agents.

Resumo. Este trabalho resume os três principais módulos de um sistema computacional cujo objetivo é automatizar ações pontuais visando contribuir no controle e combate a dengue. O módulo I é um sistema de informações em Dengue. Os módulos II e III realizam simulação da dengue usando agentes computacionais.

1. Introdução

A epidemiologia é o estudo da frequência, da distribuição e dos determinantes dos estados ou eventos relacionados à saúde em específicas populações e aplicação desses estudos no controle dos problemas de saúde. A epidemiologia visa, dentre outros objetivos, identificar e entender o agente causal e fatores relacionados aos agravos da saúde, identificar e explicar os padrões de distribuição geográfica das doenças, estabelecer metas e estratégias de controle, estabelecer medidas preventivas, subsidiar os processos de planejamento, gestão e avaliação de políticas públicas em saúde [Last, 1995]. Neste contexto, estudos, pesquisas e ações em epidemiologia, a exemplo do presente trabalho, se justificam.

Dengue é uma doença infecciosa causada por vírus com curso benigno ou grave, dependendo da forma que se apresente: infecção inaparente, dengue clássica, febre hemorrágica da dengue, síndrome do choque da dengue. O agente transmissor, no Brasil, é o mosquito *Aedes aegypti*. A vida média de uma fêmea adulta é de aproximadamente 45 dias e uma vez infectada pelo vírus, ela permanecerá assim até o fim de sua vida. A transmissão pode ocorrer do indivíduo infectado para o mosquito sadio, que se torna infectado, ou do mosquito infectado para o indivíduo suscetível (todos os indivíduos são suscetíveis).

A transmissão e a propagação da dengue possivelmente são influenciadas diretamente pelas redes de contato dos indivíduos (*small worlds*) [Filippis, 2011], ou seja, os locais que esses indivíduos freqüentam, o ambiente característico desses locais, as condições climáticas, dentre outros. Estes elementos precisam ser considerados quando o que se deseja é modelar e realizar simulações sobre dengue.

Uma modelagem compartmental baseia-se na divisão da população de hospedeiros em categorias, entre as quais os hospedeiros (humanos e mosquitos) fluem com taxas que dependem das características próprias da doença e dos vetores de transmissão, entre outros fatores. A população hospedeira é subdividida em classes, sendo que as mais comuns são suscetíveis (S), latentes (E), infectados (I) e removidos (R). Nesse caso, e para populações fechadas, considera-se que a quantidade de indivíduos da população, N , é tal que $N = S+E+I+R$, onde S = quantidade de indivíduos suscetíveis, I = quantidade de indivíduos infectados e R = quantidade de indivíduos recuperados [Vynnycky, 2010].

Levando em conta essas considerações, este trabalho consiste na apresentação da proposta de elaboração de um sistema computacional voltado à dengue, composto essencialmente por três módulos que interagem entre si: um módulo que constitui um sistema de informação para gerenciamento de dados no controle à dengue (SIGDengue) e outros dois módulos nos quais serão utilizados agentes computacionais para fins de simulação, um tendo foco no agente mosquito e outro no agente humano. Esta proposta está inserida no projeto PRONEX-Dengue (Programa de Apoio a Núcleos de Excelência), que tem ações desenvolvidas em âmbito nacional, com financiamento do CNPq, cujo principal objetivo é desenvolver modelos matemáticos para aplicação no controle da Dengue. O objetivo tanto do sistema de informações quanto das simulações apresentadas neste trabalho é contribuir para a tomada de decisões no âmbito do controle e combate à dengue e terá como alvo e estudo de caso a cidade de Cascavel/PR.

2. Módulo I: O Sistema de Informações SIGDengue

O SIGDengue é um sistema de informações georreferenciado que possui dois objetivos principais. O primeiro é integrar os dados e informações sobre dengue, atualmente disponíveis de maneira dispersa e desordenada bem como sobre outros dados importantes no trato da doença a exemplo de informações meteorológicas, densidade populacional, distribuição geográfica do município, entre outros. O segundo é viabilizar a rápida obtenção dos dados armazenados em forma de diversos tipos de relatórios e estatísticas sobre dengue, bem como a visualização em mapa dos casos (suspeitos, notificados, confirmados) de dengue e dos resultados das simulações efetuadas pelos modelos definidos e projetados empregando agentes computacionais (módulos II e III). Para manipulação dos mapas, optou-se pela utilização da linguagem de marcação KML (*Keyhole Markup Language*) considerando não apenas questões relativas a esta tecnologia, mas também porque este é o formato atualmente utilizado na cidade onde o trabalho está sendo realizado. Além dos mapas KML, o protótipo do SIGDengue utiliza a linguagem JAVA e o sistema gerenciador de banco de dados *PostgreSQL*.

A figura 1 apresentada a seguir ilustra (a) um mapa do município, (b) um mapa do município identificando as distribuições geográficas da dengue, (c) uma tela do protótipo na qual são digitadas informações sobre o trabalho realizado pelo setor de endemias do município (SISFAD) e (d) uma tela com o tipo de informações atualmente disponíveis; cabe dizer que foram coletados e digitados determinados dados que compreendem o período de 2007 a 2010.



Figura 01 (a) Bairros do município

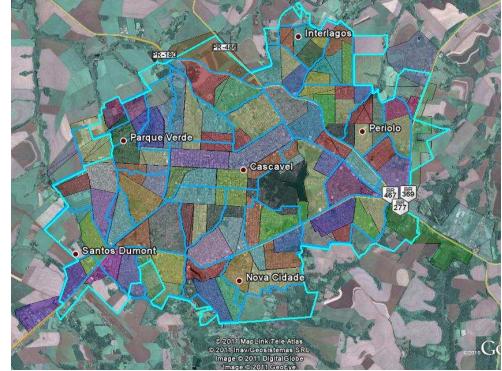


Figura 01 (b) Localidades KML

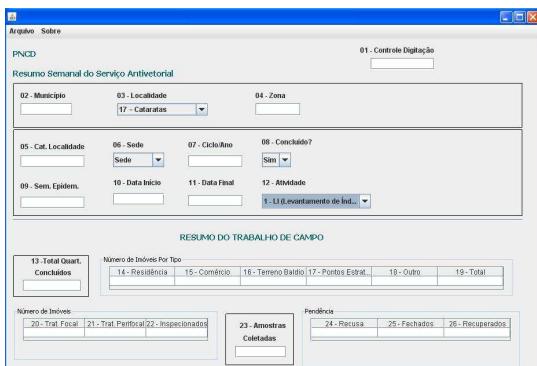


Figura 01 (c) Protótipo – dados

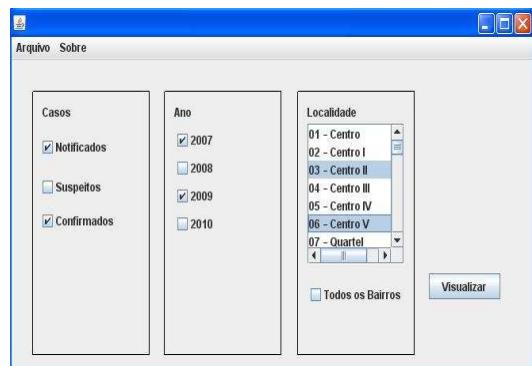


Figura 01 (d) Protótipo - relatórios

3. Simulações com Agentes Computacionais

A modelagem computacional das simulações realizadas neste trabalho emprega a abordagem dos modelos compartimentais ($S+E+I+R$), das redes de contato e leva em conta a dinâmica espaço-temporal das interações entre os indivíduos. Considera também aspectos espaciais e ambientais como focos de procriação, alimentação e a locomoção dos hospedeiros. A individualização dos hospedeiros está baseada em Sistemas de Agentes que, para o caso da dengue, apresenta versatilidade na precisão dos parâmetros, no acompanhamento individualizado do agente e nas implicações globais de seus relacionamentos. Os agentes computacionais possuem, basicamente, sensores, uma arquitetura mental e atuadores. Sensor pode ser entendido como constituído por todas as mensagens que chegam ao agente e são compreendidas por ele. A arquitetura de estado mental é uma simplificação da estratégia BDI (*Beliefs, Desire, Intention*). Atuadores estão sendo modelados como as mensagens resultantes do conjunto de ações da intenção. Está sendo utilizado o *framework* JADE (*Java Agente DEvelopment Framework*) [JADE, 2011]. Com isto, a comunicação dará de forma distribuída com troca de mensagens como na topologia *peer-to-peer*.

3.1. Módulo II: Simulações com Agentes Mosquitos

O módulo de simulação com os mosquitos *Aedes aegypti* descreve, ao longo do tempo, as interações desses vetores em um ambiente georreferenciado, facilitando análises comportamentais da infestação e estimativa de áreas de risco no município, a exemplo do feito por [Almeida, 2010]. Essas interações são o resultado da relação de um conjunto de sensores refletidos em uma arquitetura de estado mental de um agente em

certo tempo. Dentre os principais sensores, destaca-se o visual, para a percepção de entidades no ambiente e os sensores de temperatura e odor, influenciando nas condições de sobrevivência. O foco deste módulo é contemplar, ao máximo, as possíveis interações entre um certo agente mosquito com os demais agentes (mosquitos ou não).

3.2. Módulo III: Simulações com Agentes Humanos

O módulo de simulação de humanos descreve o comportamento humano ao longo do tempo, simulando a propagação da dengue tendo como principal referência o trabalho de [Liu et al., 2009]. As possíveis interações entre os agentes estão sendo determinadas pela conectividade entre os mesmos, sendo modelada através de uma rede de mundo pequeno, que visa descrever a realidade das conexões humanas na sociedade e em particular, as de deslocamentos físicos (a exemplo do deslocamento um agente humano de seu local de trabalho para seu local de estudo). Os agentes possuem baixo nível de cognição devido ao escopo do problema, porém a comunicação e relacionamento entre os mesmos é um aspecto essencial e passível de refinamento na modelagem; pretende-se incluir aspectos que demonstrem as relações sociais referentes a trabalho, estudo e lazer dos agentes humanos e analisar a implicação desses contatos na dinâmica da dengue.

4. Comentários e discussões

Atualmente já existe uma versão do SIGDengue (módulo I) que permite o cadastro de informações em dengue do município de Cascavel/PR e a visualização em mapa dessas informações. Sua conclusão está prevista para outubro de 2011. As simulações (módulo II e III) estão em fase de modelagem e definição de parâmetros. A interação entre esses três módulos se dará através de um ambiente de 3 camadas contendo uma malha que viabiliza o contato entre o meio ambiente (1^a camada), os agentes humanos (2^a camada) os agentes mosquitos (3^a camada). As implementações ocorrerão até setembro de 2011. Os testes, calibração e validação serão realizadas na sequência e para os quais serão utilizados dados reais referentes aos anos de 2007 a 2010.

5. Referências Bibliográficas

- Almeida, S.J. et all. Multi-agent modeling and simulation of an *Aedes aegypti* mosquito population. Environmental Modelling & Software. v 25, i 12, dez 2010.
- Last, J.M. Um Dicionário de Epidemiologia. Lisboa. Departamento de Estudos e Planejamento em Saúde, 1995.
- Liu Tao; LI Xia; LIU XiaoPing Integration of small world networks with multi-agent systems for simulating epidemic spatiotemporal transmission. Guangzhou, Ago 2009.
- Filippis, A.M.B. Epidemiologia da Dengue. Apresentação na I Oficina de Modelagem de Rede de Dengue. Rio de Janeiro, 2011.
- JADE - Java Agent Development Framework. <http://jade.tilab.com/>, mar 2011.
- Vynnycky, E.; White, R.G. Na Introduction to Infectious Disease Modelling. Oxford University Press, 2010.

Um sistema de apoio à decisão baseado em agentes para simulação de impactos gerados pela instalação de indústrias

Gerson L. Nunes¹, Camila D. Thomasi¹, Márcio M. Jogueiro¹, Diana F. Adamatti¹

¹Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Caixa Postal 474 – 96.201-900 – Rio Grande – RS – Brasil

{dfrabu, camilathomasi, marcio.juguero, dianaada}@gmail.com

Abstract. This paper presents a Decision Support System model based on Multi-Agent-Based Simulation, which was adapted to pollutants' dispersion simulator. It is expected that the decision-making occurs in an environmentally responsible manner.

Resumo. Este artigo apresenta um modelo de um Sistema de Apoio à Decisão baseado em Sistemas Multiagentes, o qual foi adaptado a um simulador de dispersão de poluentes. A partir deste sistema, espera-se que a tomada de decisão ocorra de forma ambientalmente responsável.

1. Introdução

Com o crescente aumento populacional e o advento de novas tecnologias surge a necessidade de criar meios para preservação dos recursos naturais existentes. Outro problema encontrado relaciona-se ao rápido crescimento industrial e tecnológico que causa uma excessiva emissão de poluentes na atmosfera, provocando sérios danos ambientais e a saúde da população.

A simulação computacional é utilizada para prever consequências futuras que auxiliem na tomada de decisão. A vantagem dessa ferramenta consiste em avaliar os riscos que serão causados sem necessidade de causar perturbação na região que está sendo avaliada, Freitas Filho (2001).

Este trabalho propõe-se a apresentar o primeiro modelo um sistema de apoio à decisão (SAD) baseado em Simulação baseada em Multiagentes (Multi-Agent-Based Simulation - MABS) adaptado a um simulador de dispersão de poluentes na atmosfera que utilize modelos mais próximos da realidade.

2. Técnicas utilizadas

Os agentes são entidades computacionais que se caracterizam por terem autonomia de decisão e execução, além de um planejador de ações que concretize suas decisões para obtenção de suas metas.

Quando os agentes apresentam-se em comunidades executando ações de modo coordenado e cooperativo numa sociedade, baseando-se em um comportamento social estamos diante de um **Sistema Multiagentes**, Rezende (2002).

Já, de acordo com Adamatti (2007), a **Simulação Baseada em Multiagentes** (MABS) deriva da união da simulação com os sistemas multiagentes, tendo como finalidade simular uma situação real. Os agentes produzem ações locais resultantes da interação com o ambiente e com os demais agentes.

Para Thompson et al (2004), um **Sistema de Apoio a Decisão em Grupos** (SAD-G) pode ser descrito como um sistema computacional interativo que visa resolver determinados problemas através de um grupo responsável pela tomada de decisão, que pode ser formado por pessoas e/ou máquinas.

A importância de um SAD-G consiste em que cada participante processe as informações resultando na escolha de uma ação, possibilitando uma diversidade de contribuições e para que elas ocorram são de fundamental importância a comunicação, coordenação e cooperação entre componentes pertencentes ao grupo.

3. Simulador desenvolvido

O principal objetivo do simulador é auxiliar os envolvidos na escolha do local para a implantação de novas indústrias através de ferramentas computacionais de simulação que fazem uma previsão dos possíveis riscos na região de interesse.

O simulador foi desenvolvido em MATLAB que disponibiliza conjuntos de funções específicas chamadas de *Toolboxes*. Neste trabalho foi utilizado o *Mapping Toolbox* que fornece ferramentas e utilitários para análise e criação de mapas, MathWorks (2011).

O GIS (Geographic Information System) foi utilizado para acelerar o processamento de objetos espaciais em um mapa digital e para visualização, análise, criação e gestão de dados geo-referenciados, Fuller et al.(2007).

O simulador foi subdividido em quatro módulos: manipulação de mapas, inserção de novos objetos, propagação da poluição e SAD.

A manipulação de mapas tem como funções: manipular e apresentar arquivos vetoriais. A inserção de novos objetos permite escolher a posição geográfica das construções e indústrias e seus parâmetros iniciais de simulação.

A propagação da poluição foi implementada inicialmente utilizando um modelo hipotético, que está sendo substituído por modelos gaussianos de propagação de poluição na atmosfera, visando obter resultados mais reais para as simulações.

4. Modelo do SAD proposto

Nesta seção é descrita toda a estruturação e o funcionamento do SAD e a contribuição de cada agente para obtenção da decisão final, objetivando atender os interesses de todos os agentes envolvidos. Este sistema ainda está em fase de pesquisa e desenvolvimento, sendo apresentado aqui seu primeiro modelo.

4.1. Agentes e seus critérios de avaliação

O SAD possui os seguintes agentes: coordenador, político, secretaria do meio ambiente, representante da população e empresário. O agente coordenador é o responsável pela comunicação entre usuário e tomadores de decisão. Os dados da simulação são repassados pelo coordenador aos demais agentes que efetuam cálculos levando em consideração seus próprios interesses decidindo qual o local a ser inserida a nova indústria.

Os pesos de cada critério variam de acordo com o perfil de cada agente. Para o cálculo de avaliação são levados em consideração os critérios mostrados na figura 1.

Político	Empresário	Representante da população	Secretária do Meio Ambiente
<ul style="list-style-type: none"> - Custo : 0,1 - Tempo de construção: 0,8 - Impacto ambiental: 0,6 - Opinião pública: 1 - Geração de empregos: 1 - Lucros: 0,5 - Impostos : 1 	<ul style="list-style-type: none"> - Custo : 1 - Tempo de construção: 1 - Impacto ambiental: 0,3 - Opinião pública: 0,2 - Geração de empregos: 0,5 - Lucros: 1 - Impostos : 1 	<ul style="list-style-type: none"> - Custo : 0,5 - Tempo de construção: 1 - Impacto ambiental: 0,7 - Opinião pública: 1 - Geração de empregos: 1 - Lucros: 0,1 - Impostos : 0,7 	<ul style="list-style-type: none"> - Custo : 0,7 - Tempo de construção: 1 - Impacto ambiental: 1 - Opinião pública: 0,5 - Geração de empregos: 0,5 - Lucros: 0,6 - Impostos : 0,7

Figura 1. Critérios de Avaliação e seus respectivos pesos.

Para decisão do local a ser inserida a nova indústria, os agentes baseiam-se em um cálculo de avaliação representado pela figura 2. Nesta fórmula é feito um somatório ponderado do critério multiplicado pelo peso atribuído ao mesmo. Ao final deste cálculo a localização da indústria que possui a maior avaliação é escolhida pelo agente como o melhor local.

A decisão de cada agente é passada ao agente coordenador, que utiliza um sistema de votação para verificar qual localização beneficia o maior número de agentes, ou seja, qual local recebeu mais votos. Em caso de empate, o agente coordenador efetua um novo cálculo de avaliação com o objetivo de desempate visando assim atender aos interesses de todos.

$$\text{AVALIAÇÃO} = ((\text{Custo} \times \text{Tx_Custo}) + (\text{Tempo de Construção} \times \text{Tx_TempoConst}) + (\text{Impacto ambiental} \times \text{Tx_ImpAmbiental}) + (\text{Opinião Pública} \times \text{Tx_opiniãoPública}) + (\text{Geração de empregos} \times \text{Tx_geraçãoEmpregos}) + (\text{Lucros} \times \text{Tx_lucros}) + (\text{Impostos} \times \text{Tx_impostos}))$$

Figura 2. Cálculo de avaliação

O mecanismo de votação foi o escolhido objetivando que cada agente efetue o cálculo de forma distribuída, baseando-se na ideia de inteligência artificial distribuída (IAD), ficando o coordenador responsável apenas pela contabilização do resultado.

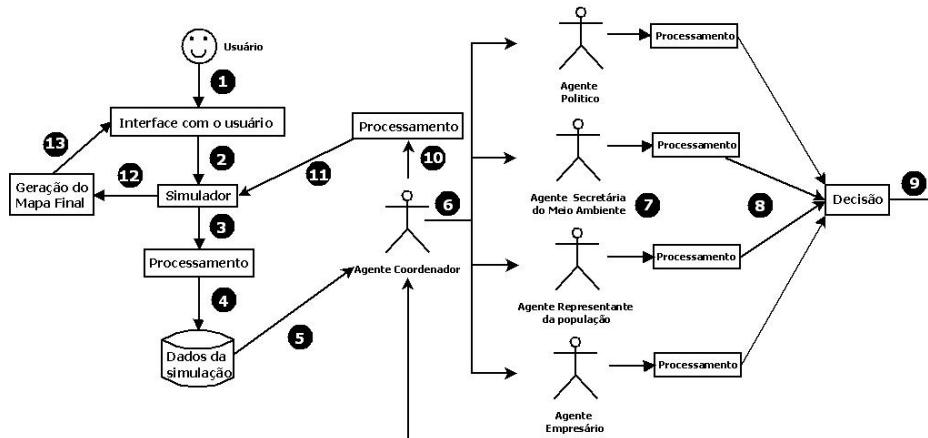
4.2. Funcionamento do SAD

A figura 3 ilustra o funcionamento do SAD. O sistema segue os seguintes passos:

- ① Escolha dos possíveis locais para inserção da indústria;
- ② Definição dos parâmetros da indústria que são passados ao simulador;
- ③ Início do processamento do simulador;
- ④ Armazenamento dos dados que serão acessados pelos agentes;
- ⑤ O coordenador acessa os dados armazenados;
- ⑥ O coordenador repassa as informações para os demais agentes participantes;
- ⑦ Cada agente processa sua avaliação individual;
- ⑧ O agente decide o local a ser inserida a indústria de acordo com seu perfil;
- ⑨ Cada agente informa sua decisão ao coordenador;
- ⑩ O coordenador contabiliza o número de votos;
- ⑪ O coordenador repassa a decisão final para o simulador;
- ⑫ O simulador gera um mapa final com a localização da nova indústria;
- ⑬ O mapa é apresentado ao usuário.

5. Conclusões e trabalhos futuros

O simulador proposto visa escolher o melhor local a ser inserida a nova indústria utilizando sistemas multiagentes, por assemelharem-se ao comportamento humano. Como estes agentes alimentam o sistema de apoio à decisão, teremos comportamentos e decisões similares as que serão tomadas no mundo real. O simulador, mesmo que não tome uma decisão autônoma, auxiliará os respectivos órgãos responsáveis a fazer a melhor escolha.

**Figura 3. Funcionamento do SAD**

Como trabalhos futuros, temos a inserção de modelos mais reais de dispersão de poluentes bem como os ajustes das regras e cálculos do agente coordenador. Pretende-se ainda estudar a viabilização do uso de outras técnicas como estratégias de decisão coletiva. Outra meta seria estender o trabalho desenvolvido não apenas para o cálculo de impactos gerados pela instalação de indústrias como também para outros tipos de construções. Assim mesmo que não tome uma decisão autônoma, o simulador auxiliará os respectivos órgãos responsáveis a fazer a melhor escolha.

Agradecimentos

Agradecemos à FAPERGS (Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul) pelo financiamento e pelas bolsas de pesquisa e desenvolvimento, projeto número 008/2009.

Referências

- Adamatti, D. F. (2007) "Inserção de Jogadores Virtuais em Jogos de Papéis para uso em sistemas de apoio a decisão em Grupos: um Experimento no Domínio da Gestão dos Recursos Naturais". Tese de Doutorado. Escola Politécnica da Universidade de São Paulo, p.196.
- Fuller, M. M, Wang, D., Gross, L. J., Berry, M. W (2007), "Computational Science for Natural Resource Management, Computing in Science & Engineering", v.9, n.4, p.40-48.
- Freitas Filho, P. J. (2001), "Introdução à Modelagem e Simulação de Sistemas - com Aplicações em Arena", ed. Visual Books, Florianópolis, SC, Brasil . v. 1. 322 p.
- Mathworks- MATLAB and Simulink for Technical Computing, MATLAB Introduction and Key Features. Disponível em: <http://www.mathworks.com/products/matlab>
Acesso em: 21 fev. 2011.
- Rezende, S. O. (2002), "Sistemas Inteligentes - Fundamentos e Aplicações". 1. ed. MANOLE: São Paulo, SP, Brasil, p. 550.
- Thompson C. E. M., Gouveia Neto N. C., Da Silva A. P. F., De Pinho M. A. B., Dornelas J. S. (2004), "Satisfação no Uso de Sistema de Apoio À Decisão em Grupo: Evidências e Construções", Simpósio de Excelência em Gestão e Tecnologia (SEGeT), Resende, RJ, Brasil.

Modelando a Organização Social de um Experimento de Gestão Social de Hortas Urbanas

I.A.S. Santos¹, T. F. Rodrigues¹, G. Dimuro², A.C.R. Costa¹, G.P. Dimuro¹

¹C3 – FURG, CEP 96.201-900 – Rio Grande – RS – Brasil

²Departamento de Arquitetura - Universidade de Sevilha - Espanha

{iverton, trodrigues02, glenda.dimuro, ac.rocha.costa, gracaliz}@gmail.com

Abstract. This paper presents the initial version of an agent-based model of the social organization of a particular experiment of social management of urban gardens in Seville, Spain. More specifically, the paper presented details of the initial version of the model, and prospects tools for its proper implementation.

Resumo. Este trabalho apresenta a versão inicial de um modelo baseado em agentes para a organização social de um experimento de gestão social de hortas urbanas, realizado na cidade de Sevilha, Espanha. Mais especificamente, são apresentados detalhes da versão inicial do modelo, bem como são prospectadas ferramentas para a implementação apropriada do modelo.

1. Introdução

Através da produção e gestão social do habitat podemos aumentar a participação cidadã no processo de (re)construção e transformação urbana aumentando a relação entre seus diversos membros: pessoas, comunidades, organizações, natureza, que, unidos por redes, possibilitam o intercâmbio de bens e informação e aumentam a diversidade potencial de comportamentos [Omicini, Ricci e Viroli 2008].

Entretanto, diante da atual configuração social e urbana, quais transformações do território urbano seriam capazes de respeitar as diversidades e diferenças e ao mesmo tempo ajustar as cidades para um melhor convívio com a natureza?

Através da simulação multiagente de organizações sociais tal questão poderá ser colocada e talvez respondida, pois podemos considerar cada organização e instituição urbana como um sistema multiagente e, assim, simular as relações possíveis entre cada agente em tal organização social. Neste sentido, este trabalho visa a implementação de um modelo de simulação baseado em agentes de um ecossistema urbano e sua respectiva gestão social, de modo que a validação e a análise de possíveis interações sejam passíveis de observação, bem como sua comparação com a realidade.

Motivados pela atual tendência de desenvolvimento da agricultura urbana e periurbana, ou seja, da (re)criação de hortas dentro das cidades, serão utilizadas como estudo de caso as hortas urbanas da cidade de Sevilha (Espanha).

2. Gestão social de hortas urbanas na Cidade de Sevilha

Na cidade de Sevilha (Espanha) temos alguns exemplos de organizações cidadãs que,

por meio de processos participativos e algumas vezes autogeridos – realizados através da aliança entre distintos atores: cidadãos, técnicos e governo - tentam produzir e melhorar as condições do seu próprio habitat através do projeto de hortas urbanas [Rodrigues, Dimuro, et al 2010].

Essas hortas podem ser chamadas de “hortas de ócio” ou até de “hortas de lazer”, pois nelas se criam relações sociais entre os usuários e programas de lazer, muitas vezes mantidos por ONGs.

Dentre as diversas hortas existentes em Sevilha, foi escolhida para estudo de caso a Horta de São Jerônimo, localizada num parque homônimo. Administrada pela ONG *Ecologistas en Acción*, possui legislação que regulamenta o comportamento e participação no projeto da horta em si. Dessa forma, aproveitamos a legislação existente para iniciarmos a modelagem e posterior simulação desse sistema social.

3. Ferramentas para a Modelagem Organizacional

Para implementação do modelo de simulação será utilizada a plataforma JaCaMo [BORDINI; HUBNER, 2010], a qual prove um suporte em alto nível para desenvolver agentes, ambientes, e organizações de forma sinérgica. A Plataforma é baseada em três plataformas/linguagens independentes, Jason [Bordini, Hubner e Wooldridge 2007], CartAgO [Ricci, Santi, Piunti et al 2010] e MOISE+ [Hübner, Sichman e Boissier 2002].

Acredita-se que a abrangência de recursos oferecidos pela plataforma JaCaMo supre as necessidades para implementação do modelo de simulação da organização do processo de gestão social das hortas de Sevilha. Por limitação de espaço, porém, omitimos aqui uma descrição mais detalhada dessa plataforma.

4. Modelagem Inicial

Alguns avanços no que se refere a modelagem da proposta foram atingidos, assim descreve-se nesta seção detalhes de uma modelagem inicial.

Tal sistema pode claramente ser reproduzido usando-se agentes, visto que organizações sociais podem ser simuladas usando-se agentes cognitivos. Para tal, foi decidido fazer a modelagem do sistema proposto subdividindo-o em três grandes áreas: modelagem organizacional (onde se determinam os papéis que os agentes podem assumir), modelagem funcional (onde se delineiam os processos que o constituem e que os agentes podem executar) e a modelagem ambiental (onde se descreve o ambiente onde os agentes viverão).

Na horta, existe uma hierarquia de papéis, ficando a ONG em seu topo. Ela regulamenta e fiscaliza a utilização da horta, bem como o relacionamento entre os que a usam através de legislação própria. Também escolhe quem poderá participar do projeto de cultivo, mediante análise do candidato previamente inscrito em lista de espera.

Dentre os outros papéis identificados, existem os que cultivam sua parte da horta (hortelões), que podem ser subdivididos em titulares e auxiliares. Os primeiros respondem pela administração da sua parcela e os últimos são ajudantes dos primeiros, escolhidos por eles mesmos. Além desses papéis, há os técnicos especializados e os universitários. Ambos fornecem auxílio técnico para os hortelões, relativo à forma de cultivo e locais específicos para tal. Submetidos à ONG ficam todos esses papéis, e ela própria submetida ao governo de Sevilha, que concede ajuda financeira para a

manutenção do projeto.

O sistema em análise é constituído por normas, identificadas a partir do regimento determinado pela ONG. Elas podem ser classificados em constitutivas (quando apenas levam o sistema de um estado a outro) ou regulativas (quando mudam o estado do sistema para algum mais desejável, ou seja, o direcionam a uma adequação à legislação vigente) as quais se dão por meio de punições ou recompensas.

O ambiente é constituído da horta de cultivo, subdividida em pequenas células chamadas “parcelas”, onde cada hortelão pode realizar seu plantio. O lugar possui ferramentas a sua disposição e adubo natural. O uso desses utensílios também é normatizado pela ONG, bem como a irrigação das parcelas.

Identificados os papéis, a estrutura organizacional que os inter-relaciona, os processos inerentes ao sistema e o ambiente onde o mesmo se desenvolve, podemos iniciar o processo de modelagem computacional. A Figura 1 mostra a hierarquia básica da Horta de São Jerônimo, com os papéis até então identificados.



Figura 1. Hierarquia da Horta São Jerônimo.

Tipo de norma	Situação em que se aplica	Ação normatizada					Verificador da Ação	Sanções (Punições, Recompensas)
		Pré-condição	Normatização	Id da ação	Resultados	Papel realizador		
Constitutiva	Durante o uso da horta	Requerimento de auxílio junto à organização	Permissão	Construção de um depósito para guardar utensílios	Constrói o depósito; Não constrói o depósito	Hortelão	Organização da horta	-
Regulativa	Durante o uso da horta, mensalmente	Possuir uma horta de cultivo	Obrigação	Pagar mensalidade	Ganha auxílio para o cultivo E Continua na horta; Não ganha o auxílio de cultivo E Sai da horta (depende do número de faltas graves)	Hortelão	Organização da horta	Punição: Falta grave (cumulativa)
Constitutiva	Durante o uso da horta	Informar à organização da horta	Permissão	Designar um ajudante	Ajudante designado OUX Negação do pedido	Hortelão	Organização da horta	-
Regulativa	Durante o uso da horta	Possuir uma parcela	Proibição	Modificação do desenho da horta	Continuar na horta; Sair da horta (depende do número de faltas graves)	Hortelão	Organização da horta	Punição: Falta grave (cumulativa)

Figura 2. Fragmento da tabela de regras.

A Figura 2 mostra um fragmento da tabela de normas já produzida. Nela se relaciona o tipo de norma, a situação em que se aplica, a ação normatizada (subdividida em pré-condição, tipo de normatização, identificação da ação, seus resultados e papel realizador), seu verificador e, caso haja, sanções aplicáveis. Nesta perspectiva, entende-

se a primeira linha da tabela como: uma norma constitutiva, aplicável durante o uso da horta pelo hortelão, a qual será verificada (neste caso, deferida ou não) pela organização. A ação normatizada se resume a, mediante requerimento prévio à organização, o hortelão possuir ou não a permissão de construir um depósito para guardar seus utensílios de cultivo.

5. Conclusão

Pretendemos realizar simulações multiagentes que nos permitam colher resultados capazes de responder as perguntas colocadas nesse artigo, bem como observar como a relação entre os diversos atores envolvidos na produção e gestão social do ecossistema urbano influem no seu próprio processo de desenvolvimento. Também os componentes essenciais para o desenvolvimento de uma comunidade de prática poderão ser simulados e comparados com a realidade.

No que se refere a plataforma JaCaMo (escolhida para a implementação do modelo), a mesma envolve uma série de recursos para o desenvolvimento de sistemas multiagentes, principalmente no que se tange a modelagem de uma forma sinérgica dos agentes, seu ambiente e a organização dos mesmos, indicando assim positivamente para a continuidade de uso da mesma.

Por fim, pode-se destacar que uma identificação dos papéis, da estrutura organizacional que os inter-relaciona, e dos processos inerentes ao sistema e o ambiente onde o mesmo se desenvolve, já foi realizada, sendo assim os próximos passos se dão sobre modelagem computacional com o *framework* MOISE+.

6. Referências bibliográficas

- Bordini, R. H.; Hubner, J. F.; Wooldridge, M. (2007) Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley, University of Liverpoll.
- Bordini, R. H.; Hbner, J. F. JaCaMo Project. (2010) <http://jacamo.sourceforge.net/>, Fevereiro.
- Hübner, J. F.; Sichman, J. S.; Boissier, O. (2002) “MOISE+: towards a structural, functional, and deontic model for MAS organization”, In: Proc. Of the First Intl. Joint Conf. on Autonomous Agents and Multiagent Systems, pages 501–502. ACM, New York.
- 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, Dezembro.
- Ricci, A.; Santi, A.; Piunti, M et al. (2010) CArtAgO (Common ARTifact infrastructure for AGents Open environments). <http://cartago.sourceforge.net/>, Fevereiro.
- Rodrigues, T. F. ; Dimuro, G. ; Von Grol, C. ; Pinheiro, T. V. ; Dimuro, G. P.; Costa, A. C. R. (2010) "Simulação Baseada em Agentes de Processos de Gestão Social de Ecossistemas Urbanos: o caso das hortas urbanas da cidade de Sevilla." In: Anais do 4th MCSUL, Rio Grande, p. 257-260.

Uma plataforma para desenvolvimento de sistemas multiagente BDI na Web

Diogo de Campos¹, Maurício Oliveira Haensch¹, Ricardo Azambuja Silveira¹

¹Departamento de Informática e Estatística - Centro Tecnológico

Universidade Federal de Santa Catarina (UFSC)

Caixa Postal 476 – 88040-900 – Florianópolis – SC – Brasil

{xxdigoxx, mauricio, silveira}@inf.ufsc.br

Abstract. *The Artificial Intelligence in Web applications is increasingly present, however, the use of multiagent systems does not increase in the same proportion. One of the causes of this problem is the difficulty found in creating such systems with the current developing platforms. The objective of this work is to create a new tool to assist in implementing, in a simple but powerful way, all sorts of BDI multiagent systems in the Web.*

Resumo. *A Inteligência Artificial em aplicações na Web é cada vez mais presente, porém, o uso de sistemas multiagente não cresce na mesma proporção. Uma das causas deste problema é a dificuldade de se criar sistemas desse tipo com as plataformas de desenvolvimento atuais. O objetivo desse trabalho é criar uma nova opção de ferramenta para implementar de maneira simples porém poderosa os mais diferentes tipos de sistemas multiagente BDI na Web.*

1. Introdução

A utilização de Inteligência Artificial é cada vez maior em serviços diversos disponibilizados em páginas na Web. Podemos, por exemplo, encontrá-la em sites de consultas e pesquisas, sendo usada em personalização de páginas e aplicada diretamente nos mais variados jogos online. Além de muitos tipos diferentes de aplicação, existem também várias maneiras de se usar Inteligência Artificial para implementar estes serviços, como redes neurais, lógica difusa e agentes inteligentes.

Dentre as maneiras citadas, nota-se que a aplicação de agentes inteligentes na Web ainda não é tão grande, e acreditamos que parte dessa carência seja pela dificuldade de desenvolver sistemas multiagente com as poucas ferramentas que existem atualmente, e que são, por vezes, difíceis de serem utilizadas.

Ao analisar diversas ferramentas e linguagens de agentes disponíveis [Bordini et al. 2005], concluímos que seria possível desenvolver uma nova plataforma de sistemas multiagente que possibilitasse criar aplicações de maneira simples, e de modo com que estas sejam executadas no lado do cliente, em oposição ao modelo de execução no servidor, através da execução via *browser* sendo utilizado para acessar a aplicação.

2. Metodologia

Para desenvolver a plataforma proposta, a linguagem escolhida para a descrição dos agentes foi a 3APL-M (Artificial Autonomous Agent Programming Language - Mobile) [Koch 2005]. Além da linguagem de descrição de agentes, foi analisada também

uma plataforma, responsável pela execução dos agentes, desenvolvida em Java que está disponível na página do projeto 3APL-M. A escolha foi feita devido às características presentes na linguagem e nessa plataforma, como simplicidade, tamanho reduzido e rápido processamento, restrições comuns de dispositivos móveis, alvos da linguagem 3APL-M, e que são propriedades interessantes também para a nossa plataforma, que visa permitir que aplicações rodem em *browsers*.

Após decidirmos utilizar a linguagem 3APL-M, pesquisamos maneiras de se executar as aplicações no lado do cliente via *browser*. Para tornar isso possível, escolhemos a tecnologia *Google Native Client (Google NaCl)* [Google 2010]. Esta ferramenta é do tipo *sand-boxing* e permite que código desenvolvido em C++ seja executado de maneira nativa na máquina do cliente, com as devidas restrições de segurança. Desse modo, a combinação de uma plataforma em C++ que execute agentes escritos na linguagem de descrição de agentes 3APL-M com a tecnologia *Google NaCl* permite que agentes sejam criados e executados diretamente na máquina do cliente, sem exigir um esforço proibitivo nos servidores.

A adição de funções mais complexas e de acordo com as necessidades do usuário da plataforma é feita através de código na linguagem JavaScript, que é uma linguagem de *script* já muito reconhecida, utilizada e documentada, que possui a característica de ser executada em *browsers*. Com a tecnologia *Google NaCl* é possível fazer a troca de mensagens entre funções JavaScript com o código C++ da plataforma de agentes. A divisão entre a interface visível pelo desenvolvedor e da plataforma em si pode ser vista em 1.

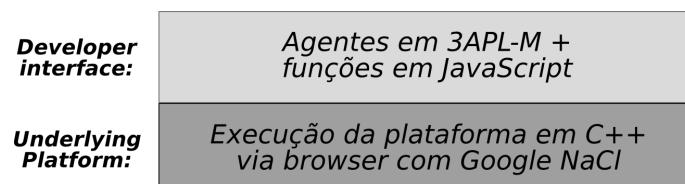


Figure 1. Organização da plataforma

O objetivo final é de que a plataforma permita que um desenvolvedor Web consiga criar um sistema multiagente sem depender de muito conhecimento novo, pois basta que este se familiarize com a linguagem 3APL-M para descrever os agentes. Toda a comunicação necessária entre os agentes e o *browser* é feita através de JavaScript, que já é conhecida pela grande maioria destes desenvolvedores. Os agentes descritos em 3APL-M poderão utilizar funções escritas em JavaScript para realizar suas intenções. A plataforma em si, desenvolvida em C++, fica num nível completamente abstruído do desenvolvedor e executa o processamento da deliberação dos agentes nativamente na máquina do cliente. Em 2 pode ser observado o fluxo de execução de uma aplicação desenvolvida na plataforma.

3. 3APL-M e Google NaCl

A linguagem 3APL-M é uma variação para dispositivos móveis da linguagem 3APL (*Artificial Autonomous Agent Programming Language*) [Dastani 2006] e se preocupa em ter um tamanho reduzido para possibilitar seu uso nesses dispositivos, em geral muito mais

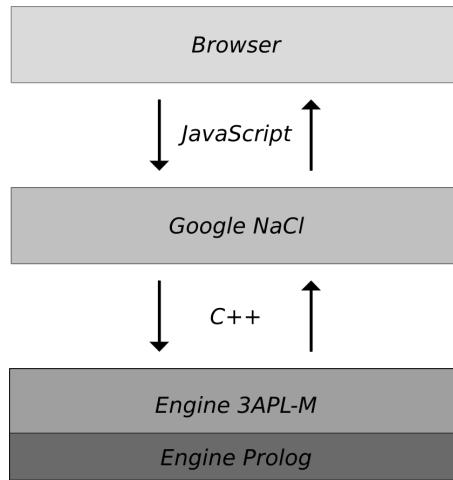


Figure 2. Fluxo de execução da uma aplicação na plataforma.

limitados em termos de memória e capacidade de processamento que computadores pessoais.

Tanto a linguagem original quanto a sua variação *Mobile* possuem uma estrutura de agente baseada no conceito de raciocínio prático de Michael Bratman de crenças, desejos e intenções [Bratman 1999], ou *BDI* (*Belief, Desire, Intention*). O modelo de agente BDI simples utilizado pela linguagem 3APL-M faz com que a descrição de agentes seja relativamente fácil, o que influenciou na escolha da linguagem para a nossa plataforma. Além disso, a implementação da plataforma de execução da linguagem é simples, ajudando a diminuir a carga de processamento envolvida na execução dos agentes, melhorando os resultados na execução via *browser*.

O *Google Native Client* é uma plataforma de código-aberto de *sandboxing* para execução segura de código nativo não confiável através de um *browser*. Essa tecnologia permite que aplicações Web com uma grande demanda de processamento ou de interatividade, como jogos, aplicações multimídia e análise e visualização de dados, sejam executadas na máquina do cliente com acesso restrito, formando um ambiente seguro. Mais detalhes sobre o funcionamento da tecnologia pode ser encontrados na página do projeto, assim como o kit desenvolvimento disponibilizado.

É importante ressaltar que essa tecnologia se preocupa em manter a neutralidade de *browsers*, o que também conta positivamente para atingirmos um dos objetivos do nosso trabalho, que é permitir uma expansão do uso de agentes em aplicações, e apesar de ser um projeto recente, o *Google Native Client* já pode ser utilizado em máquinas x86 de 32 e 64 bits com os sistemas operacionais Windows, Mac OS X e Linux, o que amplia ainda mais o público-alvo da nossa plataforma.

4. Conclusão

O projeto encontra-se atualmente no estágio de desenvolvimento da plataforma de execução de agentes na linguagem C++. O código-fonte da plataforma 3APL-M desenvolvido em Java, disponível na página da linguagem, está sendo utilizado para auxiliar na implementação e em testes durante a codificação.

Após o término desta etapa, será realizada a integração com a tecnologia *Google NaCl* e, por fim, serão adicionados mecanismos para possibilitar a chamada de funções em JavaScript diretamente na descrição das ações do agente em 3APL-M, que serão usadas para fazer a comunicação dos agentes com a interface Web.

Com a plataforma proposta, esperamos contribuir com uma nova opção para o desenvolvimento de aplicações com sistemas multiagente na Web, tendo como vantagem uma maior facilidade de implementação de sistemas, pois necessita de pouco conhecimento novo por parte do desenvolvedor.

References

- Bordini, R. H., Dastani, M., Dix, J., and Fallah-Seghrouchni, A. E., editors (2005). *Multi-Agent Programming: Languages, Platforms and Applications*, volume 15 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*. Springer.
- Bratman, M. E. (1999). *Intention, Plans, and Practical Reason*. Cambridge University Press.
- Dastani, M. (2006). 3apl platform user guide - www.cs.uu.nl/3apl/download/java/userguide.pdf.
- Google (2010). Google native client, <http://code.google.com/p/nativeclient/> (acessado em novembro de 2010).
- Koch, F. (2005). 3apl-m: Platform for lightweight deliberative agents www.cs.uu.nl/3apl-m/.

NetLogo e o modelo Climate Change como ferramenta para a Simulação do Efeito Estufa

Antônio Carlos da Rocha Costa¹, Diogo Veber Lima¹, Nisia Krusche¹

¹Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Caixa Postal 474 – 5553 3233 6623 – Rio Grande – RS – Brasil
{dvlima,nkrusche}@furg.br, ac.rocha.costa@gmail.com

Abstract. *This paper explores the use of modeling tool of multiagent systems called NetLogo to simulate the phenomenon of the Greenhouse Effect. Modifying an existing model called Climate Change, the paper presents the result of this adaptation of the model and the result obtained with the use of NetLogo.*

Resumo. *Este artigo explora o uso da ferramenta de modelagem de sistemas multiagentes denominada NetLogo para a simulação do fenômeno do Efeito Estufa. Modificando um modelo já existente denominado Climate Change, o artigo apresenta o resultado dessa adaptação do modelo e o resultado obtido com o uso do NetLogo.*

1. Introdução

O uso da abordagem de sistemas multiagentes na simulação de modelos climáticos e matemáticos tem aumentado recentemente. Isto ocorre devido aos atributos desta abordagem e das técnicas orientadas a agentes como um todo.

Dentre aos atributos essenciais a agentes está a percepção de “autonomia dos agentes”. Então como a ferramenta NetLogo trabalha com sistemas multiagentes e com essa percepção, ela foi escolhida para modelar o fenômeno do Efeito Estufa.

Devido à constante necessidade de gerar ferramentas para o ensino acadêmico através de inovações tecnológicas, novos meios são criados, estudados e implementados. Polato [1] define que “da soma entre tecnologia e conteúdos, nascem oportunidades de ensino”. Então, conciliando essa necessidade surgiu a idéia de se modelar computacionalmente o fenômeno do Efeito Estufa para ser utilizado no meio acadêmico.

2. O ambiente de modelagem NetLogo

O NetLogo é um ambiente de modelagem programável, no qual pode-se criar modelos, ou seja, simulações do mundo real, de uma forma dinâmica relativamente simples e agilizada. A linguagem que ele utiliza é baseada na linguagem de programação Logo. Outra vantagem considerável em usar o Ambiente NetLogo é o fato dele trabalhar com sistemas multiagentes. No modelo usado para simular o fenômeno do efeito estufa foram empregados vários agentes.

Além disso ele é adaptado à simulação de fenômenos naturais e/ou sociais com as seguintes vantagens:

- a) Proporciona à modelagem de milhares de indivíduos(pessoas, insetos, organizações, etc.) que interagem com o ambiente em sua volta(da simulação) e entre si;
- b) Proporciona à modelagem de sistemas complexos que evoluem conforme o tempo;
- c) Proporciona explorar as interações entre os indivíduos e os padrões dessas interações.

Como o modelo a ser simulado era o fenômeno do efeito estufa, precisava-se de uma ferramenta no qual teria que trabalhar com cada indivíduo (agente) separadamente, ou seja, usar sistemas multiagentes para depois fazer cada um interagir com os outros de forma independente. Então foi escolhido o NetLogo para construir o modelo desejado.

“Sistemas Multiagentes (SMA) são sistemas compostos por múltiplos elementos computacionais interativos denominados agentes. Agentes são entidades computacionais com duas habilidades fundamentais: (1) de decidir por si próprios o que devem fazer para satisfazer seus objetivos de projeto e (2) interagir com outros agentes de forma social ”[2].

Embora o ambiente NetLogo seja uma ferramenta fácil de programar, uma das características que dificultam a adaptação do modelo é o fato dele trabalhar atualmente somente com simulações em 2D. Como o objetivo principal é simular o fenômeno do efeito estufa, ou seja, um modelo que é em 3D, temos que modificar e adaptar algumas fórmulas para que a simulação chegue perto dos resultados esperados.

3. O modelo Climate Change

O Ambiente NetLogo vem acompanhado com alguns modelos como exemplo. Um desses modelos é o modelo Climate Change feito em 2007 [3]. Foi desenvolvido um estudo sobre o modelo Climate Change e averiguou-se que ele era suficientemente estável para começar a adaptação para modelar o fenômeno do efeito estufa.

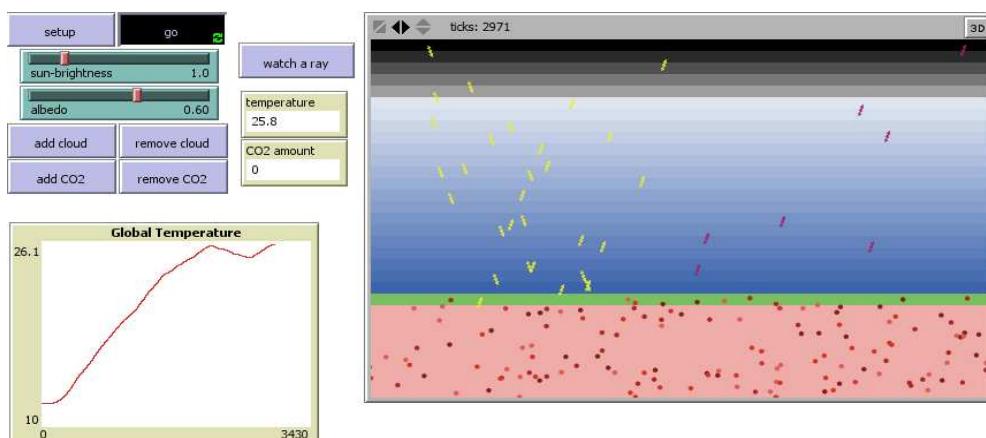


Figure 1. O modelo Climate Change[3].

O modelo possui as seguintes características:

- a) botão de setup do modelo, ou seja, para restaurar as configurações anteriores;
- b) botão de inicialização do modelo;
- c) controle do brilho do sol;
- d) controle do nível de albedo, ou seja, é a medida da refletividade da superfície do solo;
- e) botão para adicionar as nuvens e também de removê-las;
- f) botão para adicionar as moléculas de dióxido de carbono e também de removê-las;
- g) botão para assistir um raio específico do Sol que chega até o solo da terra;
- h) monitor da temperatura;
- i) monitor da quantidade das moléculas de dióxido de carbono;
- j) gráfico da temperatura global;
- k) janela para acompanhar a animação do modelo.

Os raios solares chegam até o solo da terra, alguns aquecem o solo e outros são refletidos. Os que são absorvidos são denominados “heats” (ondas de calor), e depois de um certo tempo são emitidos na forma de raios infravermelhos. As nuvens e o dióxido de carbono absorbem assim aumentando a temperatura e às vezes originando um efeito estufa.

4. O modelo Climate Change adaptado

A partir do modelo Climate Change, foram feitos algumas modificações no código do modelo. Nesse modelo modificado foi adicionado uma camada atmosférica(faixa em azul) que absorve os raios infravermelhos que “saem” do solo da terra onde 50% deles irão refletir de volta(raios azuis) para o solo e os outros 50% irão ser emitidos para cima(raios pretos). Além disso foram criados e adicionados os “heats” azuis que ficam na camada atmosférica.

Nessa nova versão para o modelo se aproximar do Efeito Estufa foi adicionado também à simulação o fator temperatura em ambos “heats”. Nessa versão modificada cada agente “heat” tem uma temperatura inicial igual a zero e cor branca. Conforme os raios solares para os “heats” vermelhos e os raios infravermelhos para os “heats” azuis eles vão ganhando cor e temperatura, ou seja, os raios transferem uma unidade de temperatura a cada vez que chegam nos “heats”. Quando cada agente “heat” chega a

uma temperatura específica são criados raios infravermelhos, ou seja, essa energia(temperatura) acumulada é emitida e o agente “heat” volta a ter sua cor branca inicial e também a sua temperatura zero inicial.

Com esse novo modelo conseguimos aperfeiçoar o modelo original para alcançar o objetivo principal, que é modelar o fenômeno do Efeito Estufa. Essa modificação pode ser analisada e vista na Figura 2 abaixo:

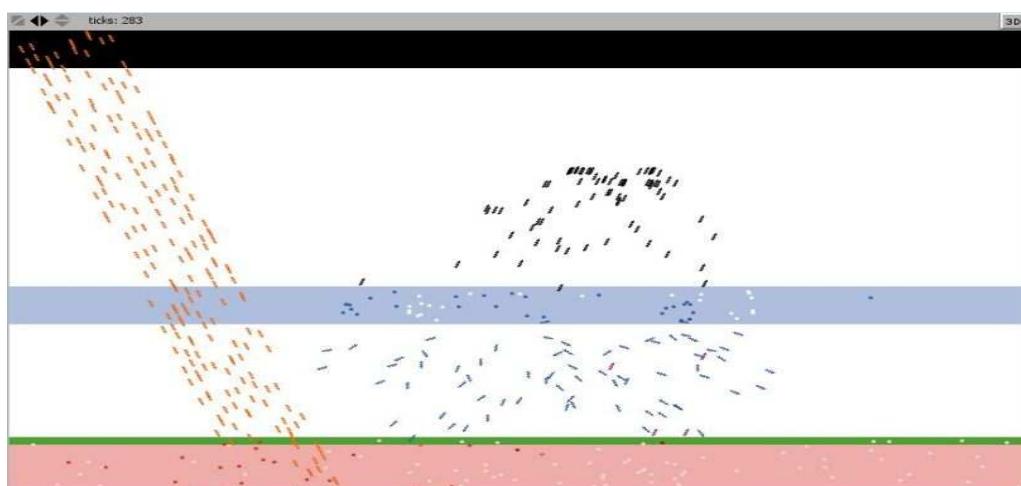


Figura 1. O modelo Climate Change adaptado.

5. Conclusão

Este artigo apresenta as vantagens em se trabalhar com o ambiente NetLogo para modelar o fenômeno do efeito estufa. Embora o projeto ainda não esteja em sua fase de desenvolvimento final, ele já apresenta um considerável desenvolvimento e nos mostra a capacidade dos sistemas multiagentes para simulação de modelos climáticos.

Referências

- [1] Polato, A., 2010, http://revistaescola.abril.com.br/avulsas/223_materiacapa_abre.shtml.
- [2] Wooldridge, Michael J., 2009, An Introduction to MultiAgent Systems, John Wiley and Sons Ltd., Inglaterra.
- [3] Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/> e <http://ccl.northwestern.edu/netlogo/models/ClimateChange>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

DIMI 3D- AGENTE COMPANHEIRO PARA UM AMBIENTE VIRTUAL DE APRENDIZAGEM

Michael Fernando Künzel, Andréa Konzen da Silva, Rejane Frozza, Daniela Bagatini, Beatriz Lux

Universidade de Santa Cruz do Sul - UNISC

Departamento de Informática

Av. Independência, 2293- CEP 96640-000 - Santa Cruz do Sul – RS

(51) 3717.7393

maikel.kunzel@gmail.com; {andrea;frozza;bagatini;lux}@unisc.br

RESUMO

Estudos estão direcionados para uma das áreas da inteligência artificial, a computação afetiva, buscando melhorar a interação homem-computador. Tendo em vista que a aprendizagem é uma das atividades mais ricas desenvolvidas pelo homem, a implantação de novas tecnologias neste processo é promissora. Na computação afetiva, encontram-se os conhecimentos da psicologia e da ciência cognitiva, sendo possível estudar como os sistemas computacionais podem detectar, classificar e responder às emoções humanas, tornando os computadores aptos a reconhecer e transmitir emoções. O reconhecimento das emoções pode ocorrer através da voz, do comportamento observável, das expressões faciais e dos sinais fisiológicos. Neste contexto, encontram-se os agentes pedagógicos (quando inseridos em ambientes de aprendizagem), que desempenham funções de auxílio à aprendizagem, podendo ser animados ou não e classificados como tutor ou companheiro. O agente companheiro é um personagem afetivo que busca acompanhar a aprendizagem do estudante e motivá-lo frente às dificuldades encontradas. Com base nisto, foi desenvolvido um agente pedagógico companheiro com emoções integrado a um sistema tutor inteligente.

Palavras-chave: Ambientes de aprendizagem. Computação afetiva. Emoções. Agentes pedagógicos tutor e companheiro.

ABSTRACT

Studies are being directed to one of the areas of artificial intelligence, affective computing, aiming to improve human-computer interaction. Learning process is an important activity for the humans. In affective computing, the areas as psychology and cognitive science are closed. It is possible to study how computer systems can detect, classify and respond to human emotions, making computers able to capture, recognize and transmit emotions. The recognition of emotions can occur through the voice, observable behavior, facial expressions and physiological signals. In this context, there are pedagogical agents (in learning environments), that perform functions to aid learning. These agents can be animated or not, and classified as a tutor or companion. The companion agent is a character with emotions that guide the students in the virtual environment. This work presents the development of a companion agent with emotions.

Keywords: Learning environments. Affective computing. Emotions. Tutor and companion pedagogical agents.

1 INTRODUÇÃO

De acordo com (AZEVEDO; TAVARES, 2001), a aprendizagem é uma das atividades mais ricas desenvolvida pelo homem, devido à assimilação de novos conhecimentos. Geralmente, o local onde este aprendizado ocorre é na sala de aula, verificando assim, a importância do papel do professor. Na visão de (SILVA; HECK, 2008) experiências agradáveis, em relação à utilização de recursos tecnológicos nas atividades de ensino, têm sido relatadas por estudantes e professores.

Um dos recursos utilizados para que isto ocorra são os ambientes virtuais de aprendizagem e, entre eles, mais especificamente os Sistemas Tutores Inteligentes (STI), que são *softwares* utilizados para dar suporte às atividades de aprendizado dos estudantes, com propósitos educacionais que incorporam técnicas da Inteligência Artificial, como o uso de agentes pedagógicos (GAVIDIA; ANDRADE, 2003).

Desta forma, STI e agentes pedagógicos podem ser utilizados como um recurso de auxílio ao professor no processo de aprendizagem dos estudantes, complementando conteúdos e/ou apresentando novos conhecimentos.

Além do agente tutor, que realiza funções para guiar o aprendizado do estudante em um STI, também existe o agente companheiro, que realiza o papel de “colega virtual” do estudante no ambiente. Esse agente pode ser considerado um companheiro virtual, que é capaz de interagir com o estudante na solução de problemas, auxiliando o aprendizado (PETRY, 2005).

De acordo com o exposto, este trabalho teve como objetivo principal pesquisar o uso de agentes pedagógicos companheiros em STI, a fim de modelar e desenvolver um agente para integrar um ambiente virtual de aprendizagem.

Conforme (JAQUES; VICARI, 2005), psicólogos e pedagogos apontam a emoção como um fator importante na aprendizagem, pois quando o estudante apresenta uma emoção negativa, este pode apresentar resultados ruins. Desta forma, o agente pedagógico companheiro desenvolvido expressa alguns comportamentos emocionais, com a finalidade de integrar o ambiente virtual de aprendizagem do projeto de pesquisa “O uso de estilos cognitivos e de agentes pedagógicos no processo de ensino-aprendizagem em sistemas tutores inteligentes” (FROZZA, 2009) e atuar juntamente com o agente tutor Dóris (SANTOS, 2001).

2 AGENTE COMPANHEIRO DIMI

O agente companheiro foi desenvolvido com base no agente DIMI, modelado inicialmente por Luis Felipe da Rosa e Silva em seu trabalho de conclusão de curso (SILVA, 2002).

A arquitetura do agente DIMI é formada pelos módulos cognitivo (age através do raciocínio), perceptivo (age por percepção) e reativo (age por impulso). O agente original contava apenas com o módulo cognitivo, que era responsável por receber as informações coletadas pelo agente tutor Dóris e tomar a decisão sobre que estratégias de ensino que deveriam ser utilizadas pelo ambiente para um determinado estudante. Na nova arquitetura desenvolvida, o agente possui uma base de conhecimento.

O agente companheiro DIMI expressa as seguintes emoções: felicidade, tristeza e companheirismo. A tristeza é expressa quando o estudante tem dificuldades, ao errar um exercício ou quando este não está gostando da interação. Na figura 1 é apresentada a sequência do comportamento do agente frente à emoção de tristeza.



Figura 1: Expressão de tristeza do DIMI no ambiente virtual de aprendizagem

Fonte: Elaborado pelos autores.

A figura 2 mostra como é expressa a felicidade, no momento em que o estudante está gostando da interação e não apresenta dificuldades, acertando os exercícios; e companheirismo que é o estado padrão do agente durante a interação estudante/ambiente.



Figura 3: Expressão de felicidade do DIMI no ambiente virtual de aprendizagem

Fonte: Elaborado pelos autores.

A forma de expressão das emoções pelo agente companheiro ocorre através do comportamento observável, tais como: se o estudante pular uma página; se as respostas do estudante estão corretas quando estiver desenvolvendo os exercícios; se o estudante necessita de ajuda.

O agente companheiro DIMI assume os papéis: colaborativo e causador de problemas, que também pode ser uma característica típica de um colega em sala de aula. Está integrado no sistema tutor inteligente, o qual é um ambiente educacional. Este ambiente educacional faz parte das pesquisas realizadas por um grupo, e pode ser verificado em (FROZZA, 2009).

O agente DIMI e suas expressões foram desenvolvidos na ferramenta *Blender*, mantida pela *Blender Foundation* (<http://www.blender.org>). Suas regras de comportamento foram implementadas no *Netbeans*.

Durante a interação do estudante com o ambiente virtual, o agente DIMI monitora quando este passa para a próxima página, questionando sobre um exercício do conteúdo em questão, conforme a figura 3. Quando o estudante responde corretamente a questão, o agente expressa a emoção de felicidade, apresentando uma mensagem de incentivo e realizando um novo questionamento ao estudante por meio de outro exercício. Já, ao errar a questão, o agente emite uma mensagem de notificação e o incentiva à leitura de um material complementar, expressando tristeza.



Figura 3: Reação do agente DIMI quando o estudante avança para a página seguinte

Fonte: Elaborado pelos autores.

3 CONCLUSÃO

Através das emoções, é possível estimular os seres humanos, por isto é importante que o computador as reconheça, a fim de motivar o estudante a realizar as tarefas propostas, mesmo frente a dificuldades. Neste contexto, estão inseridos os agentes pedagógicos, que podem atuar como tutor ou companheiro.

O agente companheiro possui um papel importante no ambiente de aprendizagem, pois é ele que motiva e acompanha o estudante durante a realização das tarefas, seja através de dicas ou mensagens de apoio para que este prossiga ou causando problemas, característica que também pode ser apresentada por um colega “real” em sala de aula.

O aprimoramento do agente DIMI caracterizou-se com sua arquitetura formada pelos módulos cognitivo, perceptivo e reativo, sendo que as emoções do agente companheiro são expressas através do comportamento observável na interação do estudante com o ambiente.

4 REFERÊNCIAS

- AZEVEDO, Breno F. T.; TAVARES, Orivaldo de L. Um Ambiente Inteligente para Aprendizagem Colaborativa. 2001.<<http://www.br-ie.org/pub/index.php/sbie/article/viewFile/144/130>>
- FROZZA, R.; et. al. Dóris 3 D: agente pedagógico baseado em emoções. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 20., 2009, Florianópolis. Anais... Florianópolis: SBIE, 2009.
- GAVIDIA, Jorge Juan Zavaleta; ANDRADE, Leila Cristina Vasconcelos de. Sistemas Tutores Inteligentes. Jun 2003. <<http://www.cos.ufrj.br/~ines/courses/cos740/leila/cos740/STImono.pdf>>.
- JAQUES, P. A.; VICARI, R. M. Estado da arte em ambientes inteligentes de aprendizagem que consideram a afetividade do aluno. Revista informática na educação: teoria & prática, v. 8, n. 1, 2005. p. 15-38.
- PETRY; Patrícia Gerent . Um sistema para o ensino e aprendizagem de algoritmos utilizando um companheiro de aprendizagem colaborativo. 2005. <<http://www.tede.ufsc.br/teses/PGCC0717.pdf>>
- SANTOS, C. T. dos; et.al. Dóris – um agente pedagógico em sistemas tutores inteligentes. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 12., 2001, Vitória. Anais... Vitória: SBIE, 2001.
- SILVA, J. M. C. da; HECK G. Utilizando o questionário MSLQ para identificação de estados afetivos em um ambiente virtual de aprendizagem. Itajaí, 2008. Disponível em: <<http://www.inf.ufsc.br/~julia/pub/28.pdf>>. Acesso em: 23 abr 2010.
- SILVA, Luis Felipe da Rosa. DIMI – Um Agente Selecionador de Estratégias de Ensino para Sistemas Tutores Inteligentes. Santa Cruz do Sul: UNISC, 2002. (Trabalho de Conclusão).

SimDeCS: Arquitetura de um Sistema Multiagente para Simulação de Tomadas de Decisão em Cuidados de Saúde

Vinícius Maroni¹, João Batista Moosmann², Paulo Ricardo M. Barros¹, João Marcelo Fonseca¹, Eduardo Zanatta¹, Elton Erhardt¹, Alessandra Dahmer¹, Sílvio Cazella¹, Cecília D. Flores¹

¹Universidade Federal de Ciências da Saúde de Porto Alegre (UFCSPA)
Sarmento Leite, 245 – 90050-170 – Porto Alegre – RS – Brazil

²Universidade Feevale (FEEVALE)
RS 239, 2755 – 93352-000 – Novo Hamburgo – RS – Brazil

viniciusmaroni@gmail.com, moosmann@gmail.com, PBarros1979@gmail.com,
joaomarcelofonseca@gmail.com, eduzanatta@gmail.com,
eerhardt@terra.com.br, adahmer@ufcspa.edu.br, silvioc@ufcspa.edu.br,
dflores@ufcspa.edu.br

Abstract. This paper presents the architecture of the multiagent system SimDeCS (Intelligent Simulators for Decision Making in Health Care), describing the interaction between the agents that compose it and its practical application in the field of health education.

Resumo. Este artigo apresenta a arquitetura do sistema multiagente SimDeCS (Simuladores Inteligentes para a Tomada de Decisão em Cuidados de Saúde), descrevendo a interação entre os agentes que o compõem e sua aplicação prática na área de educação em saúde.

1. Introdução

Este trabalho tem como objetivo central apresentar a arquitetura de um framework para construção de simuladores virtuais inteligentes para a área da saúde, utilizando uma arquitetura de sistemas multiagentes e redes probabilísticas. Este trabalho busca a construção de simuladores alinhados pedagogicamente com a metodologia de ensino baseada em problemas e que possam ser aplicados em ambientes de educação à distância.

2. O Funcionamento do SimDeCS

O SimDeCS é um ambiente de aprendizado multiagente na área da saúde. Sua utilização parte da formulação de casos clínicos pelo docente através de ambiente web, utilizando uma linguagem de domínio específico (DSL) de alto nível: a VR-MED.

A DSL VR-MED (MOSSMANN, 2010) foi concebida para que programadores, apoiados por uma notação própria e simples, especifiquem características do caso de estudo em questão. Essa é uma notação gráfica que procura representar as características presentes no domínio nos casos clínicos da área da saúde e prover suporte para a execução destes, tal como um jogo de computador (Serious Game).

Subjacente à linguagem, as situações clínicas genéricas são expressas na forma unitária de redes bayesianas múltiplas seccionadas (MSBN). Durante o processo de modelagem da situação de estudo por parte do docente, a linguagem VR-MED provê a

interface necessária para a escolha das redes bayesianas (BN) existentes no repositório, a vinculação dessas redes com os personagens do jogo e a vinculação entre diferentes BN. Motores de aleatoriedade, dispersos conforme a pertinência de sua presença nas diferentes redes permitem variações nos sucessivos percursos dos casos.

A geração final de relatórios de desempenho, baseados nos eixos de correção de condutas investigativas (e não somente acerto diagnóstico), tempo despendido e custo estimado, permitem ao docente avaliar nesses eixos o desempenho do aluno.

3. A Arquitetura do SimDeCS

A arquitetura do SimDeCS é composta por um sistema multiagente composto por três agentes: *o Agente de DOMÍNIO, o Agente APRENDIZ, o Agente MEDIADOR*.

A interação entre estes três agentes é a essência de funcionamento do SimDeCS (Figura 1). Os casos clínicos são modelados no Agente de DOMÍNIO por um especialista de domínio, representado pelo professor, criando um ambiente virtual representado na forma de um jogo computacional (Serious Game) para ser utilizado pelo aluno, através do Agente APRENDIZ. As ações do aluno sobre o domínio modelado são avaliadas pelo Agente MEDIADOR.

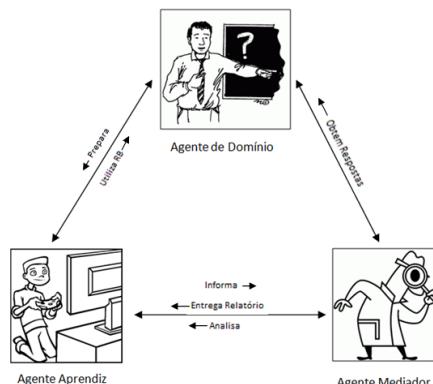


Figura 1 - Agentes do SimDeCS

4. O Agente de Domínio

Neste agente é realizada a modelagem do problema (caso clínico) por um especialista de domínio (médico, cirurgião dentista, etc.), através da VR-MED, utilizando um diagrama onde é possível especificar os detalhes do caso clínico em questão, assim como os personagens (pacientes, médicos, familiares, etc.) que participam do caso (Figura 2).

A partir deste diagrama originam-se Redes Bayesianas Múltiplas Seccionadas as quais constituem um repositório a ser utilizado pelos demais agentes. Este repositório poderá originar uma ou mais BN para um ou mais personagens do mesmo exercício. Uma rede modelada previamente servirá para balizar as consultas do Agente MEDIADOR durante a execução do exercício pelo aluno e, baseado nessa comparação, disparar as estratégias pedagógicas.

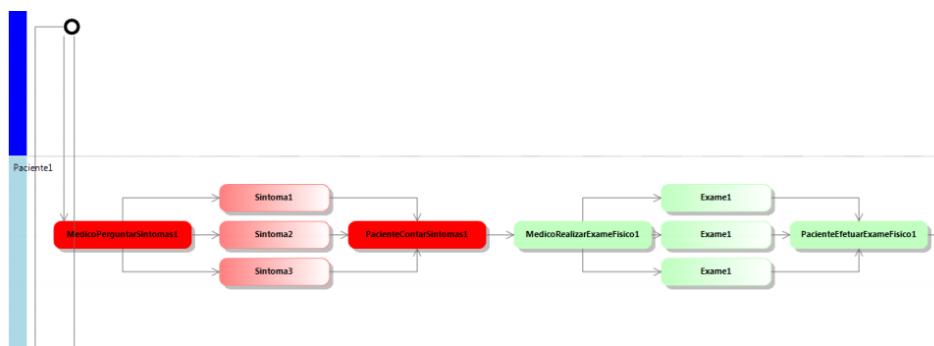


Figura 2 - Diagrama VR-MED utilizado pelo agente de domínio

5. O Agente Aprendiz

No SimDeCS o Agente APRENDIZ é representado através de um jogo computacional, do tipo Serious Game, em que, na perspectiva do aluno, há um caso representado no ambiente virtual sob o qual ele possui liberdade de ação dentro das possibilidades de atuação sobre os diferentes personagens modelados previamente pelo tutor (Figura 3).

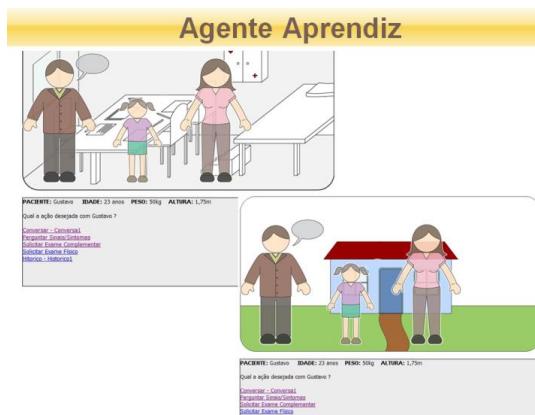


Figura 3 - Serious Game do Agente Aprendiz

Os Serious Games consistem em jogos computacionais aplicados ao ensino. Sendo sua principal característica ensinar conteúdos específicos de disciplinas ou treinar habilidades tanto operacionais como comportamentais (MORAIS, 2010).

No jogo gerado pelo SimDeCS, a escolha das ações a serem tomadas (anamnese, exame físico, exames complementares, atuações de outras naturezas) pode ou não disparar diferentes estratégias pedagógicas na interface do Agente APRENDIZ originadas por parte do Agente MEDIADOR.

6. O Agente Mediador

Ao Agente MEDIADOR, cabe a negociação do processo de decisão por parte do jogador com o conhecimento representado na rede bayesiana pelo especialista através do Agente de DOMÍNIO. Utilizando as probabilidades condicionais da rede, o incremento de tempo e custo para cada processo decisório, cabe ao Agente MEDIADOR fazer a escolha do momento e da natureza da estratégia pedagógica a ser

disparada quando o aluno apresenta comportamento aleatório (tentativa e erro), acumula escolhas de dispêndio excessivo de tempo ou custo ou direciona-se para a insolvência do jogo. Disparos de reforço ou incentivo também podem ser decididos pelo Agente MEDIADOR, notadamente quando em momentos de tomada de decisão crítica (informados pelo Agente de DOMÍNIO) como parte da boa prática pedagógica.

7. Conclusão

É notório que a utilização da educação a distância e de ambientes virtuais de aprendizagem no ensino é uma realidade que se reforça a cada dia. Na área da saúde, lançar mão destes recursos torna-se indispensável nos dias atuais a fim de reduzir custos, capacitar profissionais em territórios vastos como o de nosso país e reduzir riscos de práticas educativas clínicas.

8. Agradecimentos

Os autores agradecem o apoio financeiro do Ministério da Saúde, da Organização Pan-Americana da Saúde (OPAS) e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), através dos projetos UnA-SUS (Universidade Aberta do Sistema Único de Saúde) e Pró-ensino na Saúde.

9. Referências

- Fendel G., Spronk J., Multiple Criteria Decision Methods and Applications, Springer-Verlag, New York, 1983.
- Flores, C. D. Negociação Pedagógica Aplicada a um Ambiente Multiagente de Aprendizagem Colaborativa. 2005. Tese (Doutorado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Conselho Nacional de Desenvolvimento Científico e Tecnológico. Orientadores: Vicari, R. M.; Coelho, H.
- Mossmann, J. B., Maroni, V.; et. al. VR-MED: Linguagem de Domínio Específico para Ambientes Virtuais Aplicados à Educação Médica. In: XII Congresso Brasileiro de Informática em Saúde 2010. Porto de Galinhas/PE. Anais. 2010. CD-ROM.
- Morais, Alana Marques, et. al. Serious Games na Odontologia: Aplicações, Características e Possibilidades. In: XII Congresso Brasileiro de Informática em Saúde 2010. Porto de Galinhas/PE. Anais. 2010. CD-ROM.
- Scalese, R.J; Obeso, V.T.; S; Issenberg, B. Simulation Technology for Skills Training and Competency Assessment in Medical Education. *Journal of General Internal Medicine*. January; 23(Suppl 1) 2008, pp. 46–49.
- Seixas, L. J.; Flores, C. D.; Gluz J. C.; Vicari. R. M. Acompanhamento do processo de construção do conhecimento por meio de um agente probabilístico [online]. Via WWW. URL: http://saudecoletiva.ufcspa.edu.br/dflores/Publicacoes/AMPLIA_Sbie2004.PDF. Arquivo capturado em 27 de outubro de 2010.
- Vicari R.M., Gluz J., Flores C.D., Seixas L., Coelho H. AMPLIA: A Probabilistic Learning Environment, *International Journal of Artificial Intelligence in Education*, 18 (4), pp. 347-373, 2008.

Uma Abordagem Baseada em Leilões Combinatórios para Resolver o Problema de Transporte de Derivados de Petróleo

Kely P. Vieira, Cesar A. Tacla

Universidade Tecnológica Federal do Paraná (UTFPR) – Curitiba – PR – Brazil

kelyzincotec@gmail.com, tacla@utfpr.edu.br

Abstract. Recent researches show that approaches based on auctions and multi-agent systems are efficient on the resolution of planning problems in supply chains. This paper proposes a multi-agent system that uses combinatorial auctions to solve the oil derivative transport planning problem in multimodal networks in order to find a more efficient solution than with simultaneous auction techniques.

1. Introdução

O transporte de derivados de petróleo realizado através de redes multimodais pode ser visto como um problema de planejamento em cadeias de suprimento (CS). Sistemas Multiagentes (SMA) têm sido utilizados no planejamento de CS, visto que gastam menos tempo que os solvers de modelos matemáticos para encontrar soluções factíveis. Estas soluções são frequentemente próximas e por vezes iguais às soluções ótimas dos modelos matemáticos (Brito et al., 2010).

Problemas de planejamento em CS são difíceis de resolver de modo ótimo devido à complexidade computacional, quase sempre exponencial, pois o número de elementos e estados possíveis destes sistemas é elevado. Porém, soluções factíveis podem ser alcançadas através de mecanismos de mercado tais como: leilões.

Um leilão é um método de alocação de recursos e uma aplicação prática da teoria de jogos, visto que pode ser considerado um jogo cooperativo entre agentes (Ferriche, 2009). Os quatro principais mecanismos de leilão são: inglês, holandês, primeiro preço e segundo preço (Conitzer, 2010). Outras abordagens mais recentes estão sendo bastante utilizadas, entre elas, Leilões Simultâneos (LS) e Leilões Combinatórios (LC).

Em LS, diversos leilões, sendo um para cada item, são executados simultaneamente, e é permitido aos licitantes realizar uma oferta de compra para cada item. Em LC é possível realizar o leilão de duas maneiras, na primeira, diversos itens são colocados à venda em um único leilão, na segunda, são realizados vários leilões simultaneamente, sendo um para cada item, em ambas as maneiras, é permitido ao licitante realizar uma oferta de compra para um único item ou para um pacote de itens (Cramton; Shoham; Steinberg, 2006; Shoham; Leyton-Brown, 2009).

LC apresentam benefícios e problemas quando comparados a outros mecanismos de leilão. Dentre os benefícios, LC podem resultar em alocações mais eficientes se comparados aos mecanismos de leilões tradicionais, visto que eliminam o problema de exposição, ou seja, o licitante ao precisar de vários itens realiza ofertas individuais com valores altos, ficando exposto ao risco de não ser satisfeito com todas as ofertas e desta forma perder dinheiro, visto que LC permitem aos licitantes expressar completamente suas preferências em uma única oferta (Adomavicius; Gupta, 2005). Além disso, estudos mostram que LC são mais eficientes do que LS com relação às receitas obtidas pelos leiloeiros e a alocação de recursos para os licitantes (Wilenius, 2009).

Dentre os problemas, pode-se citar o de custo computacional para alocação e análise das ofertas para definição do vencedor, ou seja, o problema de determinação do vencedor, que ocorre devido à necessidade do leiloeiro varrer toda a lista de ofertas e então escolher a oferta que maximize seu lucro (Sandholm; Suri; Gilpin; Levine, 2002). Segundo Conitzer (2010) diversas abordagens têm sido propostas para resolver este problema, e, dependendo de como problema é modelado, a complexidade do problema pode ser polinomial ao invés de exponencial.

Neste trabalho, é proposta uma abordagem baseada em LC para resolver o problema de planejamento de transporte de derivados de petróleo na rede multimodal brasileira. Na abordagem, LC são utilizados como mecanismos de negociação entre os agentes, onde diversos leilões podem ocorrer simultaneamente por período de planejamento e os licitantes podem realizar ofertas para itens individuais ou pacotes de itens.

O objetivo é propor uma abordagem baseada em LC para que possa ser comparada à abordagem utilizada por Banaszewski et al. (2010), visto que LC podem encontrar soluções mais eficazes do que os mecanismos de leilão clássicos quando o custo é considerado no problema. Na seção 2 apresentamos uma visão conceitual do problema que será abordado. Na seção 3 apresentamos a proposta baseada em LC e na seção 4 apresentamos as conclusões prévias sobre o trabalho.

2. Visão conceitual do problema

O problema de transporte de derivados de petróleo consiste no planejamento do transporte entre os diversos modais que formam a cadeia petrolífera, de modo a garantir que os derivados sejam escoados entre bases produtoras e consumidoras e que não falte ou sobre produtos em nenhuma delas. O planejamento é complexo devido à quantidade de elementos que compõem o sistema e as restrições de estocagem e movimentação.

No cenário petrolífero brasileiro, bases consumidoras e produtoras são interligadas através de rotas. Uma rota pode ser composta por diversos tipos de segmentos (dutos, navios) e cada segmento possui uma capacidade por período de planejamento que deve ser respeitada. Na modelagem do problema, um período não está relacionado a uma quantidade fixa de dias, assim um período pode ser, por exemplo, de 7 ou 30 dias.

Com o planejamento deve ser garantido que ao final do período, todas as bases estejam com seus balanços próximos a zero ou a meta de cada base. Banaszewski et al. (2010) propôs um SMA baseado em LS para resolver o problema de planejamento de transporte de derivados de petróleo na rede multimodal brasileira, porém a abordagem proposta possui alguns problemas, os quais pretende-se resolver através de LC, dentre eles o risco de cair no problema de exposição, anteriormente explicado e o sequenciamento dos leilões, tornando a proposta não otimizada por parte do leiloeiro.

3. Uma abordagem baseada em leilões combinatoriais

A figura 1 apresenta uma cadeia de suprimentos composta por produtores (P1, P2 e P3), consumidores (C1, C2, C3, C4 e C5) e terminais de armazenamento (T1, T2, T3), onde cada entidade é chamada de base. Cada base deve buscar o balanço zero, ou seja, bases produtoras (estoque positivo) podem escoar produtos para bases consumidoras (estoque negativo) a fim de manter a cadeia balanceada no final de um período. Os segmentos (S1 a S14) interligam as bases, cada segmento possui valores mínimo e máximo em m^3 para o transporte de produtos em um período.

No cenário apresentado na figura 1, produtores poderiam realizar leilões simultâneos de seus produtos com estoque positivo no período e consumidores poderiam realizar ofertas de compra para itens individuais ou combinados.

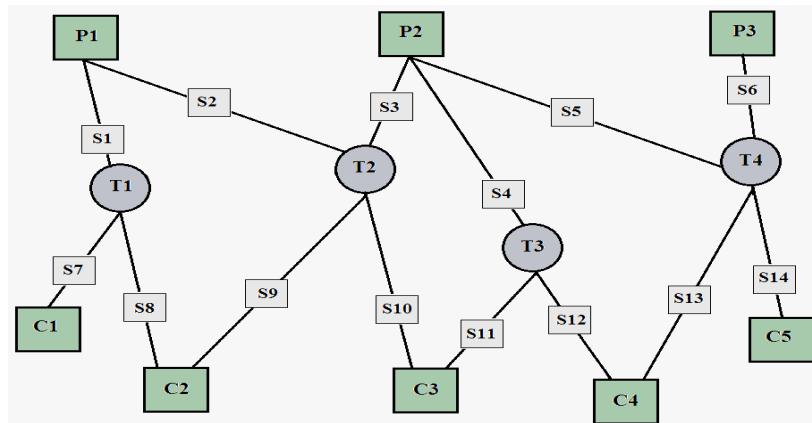


Figura 1. Rede de transporte de derivados

3.1. Método

Para avaliar a abordagem proposta um SMA será desenvolvido. O modelo proposto utiliza LC para realizar a negociação entre os agentes, que são cooperativos e auto incentivados, além disso, possuem informação completa e idêntica do problema.

O SMA será composto pelos seguintes agentes: Base (responsável por representar entidades produtoras e consumidoras), Segmento (responsável por realizar o controle de alocação e quantidade transportada em cada segmento de uma rota) e Gerente (responsável por controlar a execução dos leilões em cada período, levando em consideração a urgência que as bases têm para receber o produto). No cenário proposto, um produtor pode realizar um leilão para cada produto com estoque positivo, para isto deve solicitar ao gerente a permissão para realizá-lo, somente após esta permissão o produtor envia uma proposta para cada consumidor ligado a ele.

O consumidor aguarda até receber as propostas de cada leiloeiro (produtor), então verifica os produtos que precisa e monta uma oferta de compra para um ou vários itens. A oferta montada pelo consumidor deve conter o grau de urgência para receber o produto, calculado com base na quantidade de produtores ao qual está ligado e ao nível de estoque que possui. Para garantir a integridade do leilão, uma oferta combinatorial só pode ser realizada para produtos ofertados por um mesmo produtor.

Quando um leiloeiro recebe as ofertas de todos os consumidores, ele avalia qual é a melhor oferta e então define o ganhador para um ou vários leilões. Na abordagem proposta, não existe a obrigatoriedade de um agente consumidor enviar uma oferta para cada proposta de leilão recebida, visto que o agente pode fazer uma oferta combinatorial.

Quando os leilões de um produtor são finalizados, o gerente avalia qual ganhador possui maior urgência e então avalia a melhor rota para alocar para o transporte do(s) produto(s) para o consumidor. Portanto, é o gerente quem garante que consumidores com maior urgência recebam os produtos de que necessitam antes dos demais consumidores. Quando todos os leilões para o período são finalizados, um novo período pode ser iniciado.

4. Conclusões

Analizando a abordagem proposta por Banaszewski et al. (2010) é possível identificar alguns pontos que podem ser melhorados com a abordagem baseada em LC. Um deles é o

sequencialmento dos leilões na fase de aviso dos ganhadores e alocação de rotas para o transporte dos produtos. Em cenários com muitos produtos e bases a solução final pode demorar muito, visto que o sequenciamento entre os leilões demanda o tempo de espera de finalização de todos os leilões para então ser definida a ordem que os leilões irão finalizar.

Outro problema é o risco dos agentes ficarem preocupados com a satisfação de todas as suas necessidades, no caso precisarem de mais de um produto, e começarem a realizar ofertas muito além do que deveriam, correndo o risco de não ter todas as suas necessidades satisfeitas e desta forma terem um custo elevado (Problema da exposição).

Sendo assim, acredita-se que a abordagem baseada em LC, pode trazer resultados mais eficientes do que já foi desenvolvido, pois não existe o risco do problema da exposição, visto que os agentes montam ofertas combinatoriais expressando todas as suas necessidades, e estas possuem um peso maior do que ofertas de itens individuais. Além disso, não ocorrerá o sequenciamento na fase de alocação das rotas para o transporte dos produtos, visto que o agente gerente, ao término de cada leilão, irá fazer as validações necessárias de alocação da rota entre o produtor e o consumidor e então finalizar o leilão.

Portanto, apesar de existir o problema de custo computacional em LC, acredita-se que o modelo proposto, poderá trazer uma solução mais eficaz para o problema de transporte de derivados, visto que a alocação de recursos será otimizada. Devido ao fato de LC ser utilizado como um mecanismo de negociação entre os agentes, não será implementado nenhum mecanismo de projeto para evitar agentes mal intencionados, pois os agentes do SMA proposto serão colaborativos e auto incentivados.

Referências

- Adomavicius, G.; Gupta, A. (2005), “Toward Comprehensive Real-Time Bidder Support in Iterative Combinatorial Auctions”. *Information Systems Research*, v. 16, pp. 169–185.
- Banaszewski, R. F.; Pereira, F.; Tacla, C.; Simão, J. M.; Arruda, L. V.; Ribas, P. C. (2010), “Simultaneous Auctions in Transport Planning of Multiple Derivatives of Oil in Multi-Modals Networks”. 42º Simpósio Brasileiro de Pesquisa Operacional, Bento Gonçalves, RS.
- Brito, R. C., Tacla, C. A., Arruda, L. V. R. (2010), “A multiagent simulator for supporting logistic decisions of unloading petroleum ships in harbors”. *Pesquisa Operacional*, v.30, pp.729-750.
- Cramton, P.; Shoham, Y.; Steinberg, R. (2006). “Introduction to Combinatorial Auctions”. *Combinatorial Auctions*, MIT Press, Cambridge, MA, 2006, pp. 01-13.
- Conitzer, V. (2010), “Comparing Multiagent Systems Research in Combinatorial Auctions and Voting”. *Annals of Mathematics and Artificial Intelligence*, v. 58, pp. 239-259.
- Ferriche, R. C. (2009), “Teoria de Leilões com Aplicação ao Mercado de Petróleo Brasileiro”. Fundação Getúlio Vargas, Rio de Janeiro, 2009.
- Sandholm, T.; Suri, S.; Gilpin, A.; Levine, D. (2002), “Winner Determination in Combinatorial Auction Generalizations”. *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 69-76.
- Shoham, Y.; Leyton-Brown, K. (2009), “Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations”. 2009
- Wilenius, J. (2009), “Combinatorial and Simultaneous Auction: A Pragmatic Approach to Tighter Bounds on Expected Revenue”. Technical Report, Uppsala University, Maio 2009.

Simulação social baseada em dados de redes mediadas por blogs

Aline Macohin¹, Cion Ayres do Nascimento¹, Gustavo Alberto Giménez Lugo¹

¹Departamento Acadêmico de Informática – Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba – PR – Brasil

{aline.macohin,cion.ayres}@gmail.com, gustavo@dainf.ct.utfpr.edu.br

Abstract. *With the growing popularization of blogs, the development of tools that can extract and process data to present it in a clear and meaningful way becomes necessary so this obtained data can help the studies about the behavior of these networks. Under this aspect, in this work, blogs mediated networks were mapped using data available on the Internet to be used in the creation of computational models that can demonstrate the structure of these networks and allow to simulate situations based on obtained data.*

Resumo. *Com a crescente popularização dos blogs se faz necessária a criação de ferramentas que processem os dados destes sites para apresentá-los de forma clara e significativa e assim facilitar o estudo sobre o comportamento destes tipos de redes. Sob este aspecto, neste trabalho foi mapeada uma rede social com base nos traços de interações de blogs a partir de dados disponíveis na Internet, que serve de base para a criação de modelos computacionais que demonstrem a estrutura destas redes e assim permitam simular situações baseadas nos dados obtidos da rede.*

1. Introdução

Por muitos anos, psicólogos, antropólogos e pesquisadores do comportamento humano estudaram as capacidades sociais das pessoas e chegaram à conclusão que elas gostam de fazer parte de grupos sociais, e por isto formam comunidades e grupos *online*, para compartilhar interesses similares e realizar discussões com outro membro ou diversos membros (AGARWAL; LIU, 2008).

Estas redes sociais são formadas por simples componentes, como por exemplo, os usuários que fazem parte dela, e estes se auto-organizam ao adquirirem uma estrutura funcional, espacial e/ou temporal sem interferência dos outros (LORETO; CATTUTO; BALDASSARRI, 2007).

Dentre os vários tipos de serviços existentes que propiciam verificar traços das interações entre pessoas como o Flickr, YouTube, Wikis, del.icio.us, foram escolhidos os *blogs*, por serem um dos mais populares meios de comunicação e interação de massas, pois afetaram as interações sociais entre pessoas e comunidades, além de terem uma grande quantidade de conhecimento, e muito a ser descoberto e explorado (AGARWAL; LIU, 2008). Algumas das características úteis dos *blogs* no estudo de caso da disseminação de informações são a presença de conceitos inter-relacionados e publicações datadas associadas a indivíduos e identificadas por palavras-chaves.

Dentro do contexto destas redes, neste trabalho foi desenvolvida uma ferramenta capaz de extrair informações de *blogs* de forma automática com o objetivo de demonstrar a partir das informações obtidas, a estrutura destas redes de acordo com a citação de *links* entre *blogs*. Este trabalho servirá de base para estudos de como se disseminam as informações, como por exemplo, prováveis cenários de disseminação de notícias, recuperação de informações personalizadas, detecção de indivíduos e organizações com potencial interesse em temas específicos, simulações baseadas em eventos passados, entre outros.

Na seção 2 serão apresentados os estudos relacionados ao tema. Já na seção 3, serão abordados os resultados obtidos e por fim na seção 4, é citada a conclusão e os trabalhos futuros.

2. Estudos Relacionados

A extração de informações específicas de uma rede social, como por exemplo, a detecção de pessoas influentes que emitam opiniões sobre um determinado produto, são de grande valia para muitas empresas. Por isto, alguns estudos tem sido publicados para detectar indivíduos com posições sociais mais elevadas em redes mediadas por *blogs*, com o objetivo de identificar suas opiniões e formar estratégias de *marketing* (LI; TIMON, 2010).

Informações como as de mudanças que ocorrem na estrutura da rede, são fundamentais para a definição de um modelo computacional que replique a estrutura das redes mediadas por *blogs*. Contudo, para isto, diversos conceitos teóricos devem ser analisados dentro de uma rede social como os grupos sociais existentes, o isolamento, a popularidade, as conexões existentes, o prestígio, o equilíbrio, a influência, os subgrupos, a coesão social, a posição social, o papel social, a reciprocidade, a conformidade, entre outros (WASSERMAN; FAUST, 1994).

A partir de diversos atributos dos membros de uma rede social e de estatísticas que quantifiquem, por exemplo, a porcentagem de rotatividade dos membros mais influentes de um determinado grupo, pode-se realizar uma simulação de eventos e avaliar os resultados. Uma das abordagens que pode se utilizar para o levantamento de estatísticas da rede é a *blockmodelling*, desenvolvida por Snijders e Nowicki (1997) e esta é aplicada no trabalho de Lazega, Sapulete e Mounier (2009).

3. Discussão

Ao fim do processo de desenvolvimento, foi obtido um sistema extrator capaz de obter informações ininterruptamente do serviço Blogger¹ e um indexador de dados. Para a implementação do sistema foi utilizado a linguagem de programação Java que permitiu extrair informações relevantes do XML (Extensible Markup Language) de cada *blog* e armazená-las em um banco de dados. A partir destas informações, foi possível fazer correlações entre alguns dados e buscas por tempo, como por exemplo os *links* citados por um *blog* em um determinado intervalo de tempo, através do conteúdo e data de publicação dos *posts*.

¹ <http://www.blogger.com>

O sistema extrator de dados desenvolvido foi executado entre os dias 23/06/2010 e 27/08/2010, totalizando 3.810 *blogs*, 2.824.851 *posts*, 4.516.937 comentários, 2.406.272 *links* de *posts*, 4.308 autores donos de *blogs* e 2.152.802 utilizações destas *tags*, em que 238.323 são *tags* únicas.

Com a grande quantidade de informações obtidas, foi possível gerar grafos, através do *software* Gephi (<http://www.gephi.org>) por intervalo de tempo que ilustram a citação de *links* entre *blogs*. A partir destes grafos, possibilitou a identificação de comunidades², ou seja, conjunto de *blogs* que se citam entre si, e a identificação do valor de modularidade da rede, que demonstra o quanto fortemente estruturadas³ as comunidades estão (GIRVAN; NEWMAN, 2003), além de ser um valor que pode ser utilizado na simulação da rede social de *blogs*⁴. O cálculo do valor da modularidade é abordado no artigo de Girvan e Newman (2003), e tem como objetivo avaliar a qualidade da rede através de cálculos que levam em conta as conexões entre os nós e o número de comunidades existentes.

Na figura 1 pode ser verificado a alternância do valor da modularidade da rede de *blogs* entre 2001 e 2009, que apesar de ter variado, a rede atingiu um valor próximo de 0,7. Vale ressaltar ainda que a modularidade do ano de 2001 é zero devido a somente um blog extraído ter publicado notícias neste ano.

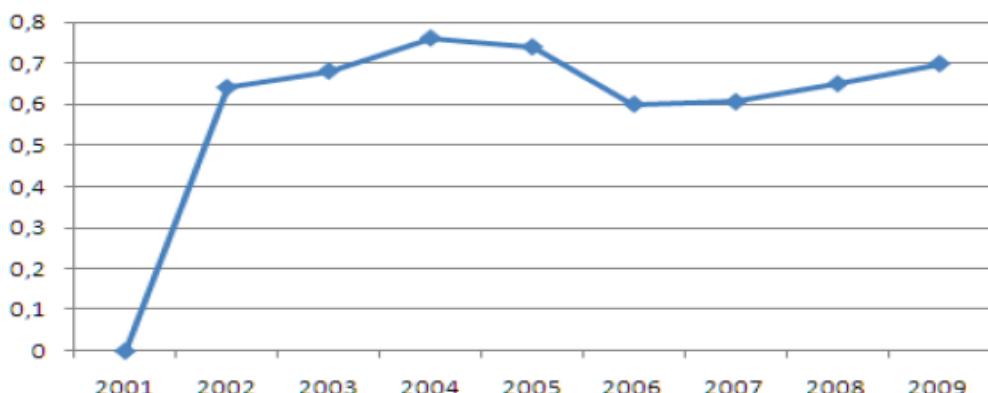


Figura 1 - Alterações sofridas na modularidade da rede entre 2001 e 2009.
Fonte: Própria.

4. Conclusão

Ao fim da execução do projeto foi possível obter uma codificação completa da extração de dados em linguagem Java que, em conjunto com o *software* Gephi para geração de

² Para a formação das comunidades e a identificação da modularidade da rede foi utilizado o algoritmo de Blondel, Guillaumé, Lambiotte e Lefebvre (2008).

³ Quando o valor da modularidade for mais perto de 0, a rede é considerada aleatória, e quando este valor se aproxima de 1, as comunidades possuem estrutura forte. Contudo, na prática, segundo Girvan e Newman (2003), os valores encontram-se entre 0,3 e 0,7.

⁴ Um exemplo de simulação que poderia ser realizada é utilizando a ferramenta NetLogo (<http://ccl.northwestern.edu/netlogo/>), em que os agentes seriam os *blogs* e a forma de se conectarem a outros agentes é através de citações de *links* de outros *blogs*. A partir do valor da modularidade de uma determinada rede daquele ano, pode-se determinar um valor de probabilidade de um *blog* pertencente a uma comunidade, citar um *blog* de outra comunidade.

grafos e detecção de comunidades, tornaram simples a visualização de alguns tipos de dados da rede social mediada por *blogs*.

A visualização da rede por meio de conexões entre *links* e *tags* utilizadas, possibilitou uma análise superficial de alterações no tempo envolvendo os conceitos manipulados (especialização, mudanças de tópico), temas mais populares e *blogs* mais citados no tempo.

Ainda podem-se levantar hipóteses para o aumento de *links* entre os nós ao longo dos anos, entre elas vale citar o aumento do número de usuários, a popularização do Blogger, a troca de *links* entre *blogs*, com o intuito de ambos se promoverem, e, com o aumento de *blogs*, aumenta também o número de *blogs* que possam falar do mesmo tema, ou seja, uma maior probabilidade de se conectar.

Vale citar também a relevância do trabalho já realizado, que serve como base para a detecção e rastreamento de fluxos de informações na *Web*, pois os dados de *blogs* alastram-se nas redes sociais com características similares às de contágio de doenças, com restrições específicas (tipos de grupos sociais, empresas, vizinhança geográfica, entre outros.). E é devido a estas características, que as informações contidas neles são de fundamental importância para detectar por exemplo, uma possível propagação de certos tipos de conteúdos através de simulações que utilizem como base valores estatísticos dos dados extraídos.

Referências

- Agarwal, Nitin; Liu, Hung. (2008) Blogosphere: Research Issues, Tools, and Applications.
- Blondel, Vincent D.; Guillaume, Jean-Loup; Lambiotte, Renaud; Lefebvre, Etinne. (2008) Fast unfolding of communities in large networks.
- Girvan, M; Newman, M. E. J. (2003) Finding and evaluating community structure in networks.
- Lazega, Emmanuel; Sapulete, Saraï; Mounier, Lisa. (2009) Structural stability regardless of membership turnover? The added value of blockmodelling in the analysis of network evolution.
- Li, Feng; Du, Timon C. (2010) Who is talking? An ontology-based opinion leader identification framework for word-of-mouth marketing in online social blogs.
- Loreto, V.; Cattuto, C.; Baldassarri, A. (2007) Information dynamics in webbased social systems.
- Snijders, T. A. B.; Nowicki, K. (1997) Estimation and prediction for stochastic blockmodels for graphs with latent block structure.
- Wasserman, Stanley; Faust, Katherine (1994) Social Network Analysis: Methods and Applications.

Desenvolvimento de um modelo de Simulação Social da cidade de Mandirituba – Brasil – Paraná

Leonardo A. Alcantara¹, Fernando J. Esmaniotto¹, Gustavo A. Giménez Lugo¹

¹Universidade Tecnológica Federal do Paraná – Curitiba – Paraná – Brasil

alcantaraleo@gmail.com, fernandoesmaniotto@gmail.com,
gustavo@dainf.ct.utfpr.edu.br

Abstract. Social simulation is a powerful tool in assisting city, state and federal management oversee populational growth. This paper discusses the creation a social simulation model prototype based on software agents, taking under account the Kohl model for urban segregation, and built on NetLogo.

Resumo. Simulação social é uma ferramenta poderosa no auxílio à gerência municipal, estadual e federal para análise de crescimento populacional. Este artigo discute a criação de um protótipo de modelo de simulação social baseado em agentes de software, considerando o modelo de Kohl para segregação urbana, e construída no NetLogo.

Introdução

O desenvolvimento urbano é um processo complexo que leva em consideração fatores sociais, políticos e econômicos. A utilização de GeoProcessamento, de acordo com [Godoy, 2004], é baseada numa visão estatística do mundo e elaboradas sobre suposições pré-estabelecidas. Uma abordagem dinâmica para modelos de crescimento urbano permite flexibilidade e possibilita novas representações dos dados e cenários.

A adoção de agentes de software no desenvolvimento de modelos de simulação social proporciona dinamismo ao mesmo. O uso dessa abordagem permite interatividade entre as entidades e com o ambiente onde elas se encontram.

Este artigo trata da criação de um protótipo de modelo de simulação social utilizando agentes para representar a população da cidade de Mandirituba no Paraná. O protótipo apresentado constitui o estudo inicial do desenvolvimento posterior de um modelo robusto e parametrizável.

O protótipo utiliza o modelo de Kohl, [Maretto et al, 2010] criado para prever o crescimento de sociedades pré-industriais. Este modelo foi utilizado devido ao fato de Mandirituba ser predominantemente rural e possuir 22.235 habitantes. Características similares às das cidades pré-industriais, contempladas no modelo de Kohl.

Descrição do modelo desenvolvido na ferramenta NetLogo

O protótipo foi implementado na ferramenta NetLogo. A ferramenta disponibiliza uma linguagem natural de programação permitindo aos desenvolvedores abstrair detalhes da criação e execução dos agentes e do mundo em si.

Dois conjuntos de dados simulados, do saneamento básico do município, serão utilizados. A utilização de dados de saneamento se deve ao fato de que apesar de pessoas residirem em locais sem luz elétrica o mesmo não se aplica a água.

O primeiro conjunto contém uma distribuição normal do saneamento a partir do centro urbano do município. Assim regiões afastadas possuem menos atratividade para os habitantes. O segundo conjunto contém dados similares, mas difere do primeiro por conter duas áreas de foco. A primeira o centro urbano e a segunda a região localizada entre o centro e a cidade de Curitiba. O intuito é simular duas áreas de interesse.

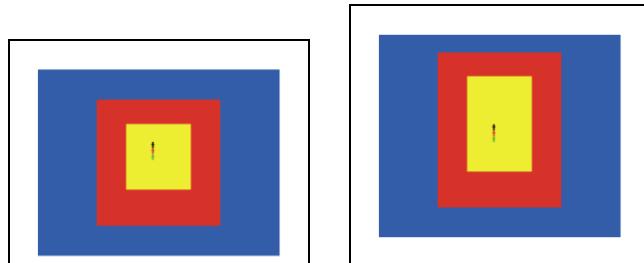


Ilustração 1 – Apresentação visual da distribuição do saneamento básico

Considerando a representação acima, a área em branco contém 50% de saneamento, a área em azul contém 75%, a área em vermelho contém 85% e a área em amarelo contém 90% de saneamento.

O modelo prevê a criação de três classes distintas de agentes sendo estas:

- Upper class (representado graficamente como )
- Middle class (representado graficamente como )
- Lower class (representado graficamente como )

Estas classes foram criadas para simplificar a classificação brasileira de classes A, B, C e outras. O modelo trabalha com habitantes com diferentes graus de poder aquisitivo. A utilização de três classes também remete ao modelo de Kohl, que prevê três classes, de acordo com [Maretto et al, 2010].

A renda per capita dos habitantes foi outro fator considerado na criação das classes dado que os trabalhadores recebem em média 2,2 salários mínimos. Considerando isto foi realizada a distribuição de classes em 5% de agentes Upper class, 10% de agentes Middle class e 85% de agentes Lower class.

Este protótipo não utilizará parametrizações. O modelo posterior contemplará o uso de parâmetros e os autores irão utilizar dados históricos obtidos como consumo de água, energia elétrica, população e outros.

Metodologia

Agentes procurando uma residência levarão em consideração a porcentagem de saneamento básico disponível na área. Agentes Upper class alocaram áreas com 90% ou mais de saneamento básico; agentes Middle class alocaram áreas com 60% ou mais e agentes Lower class alocaram áreas com 30% ou mais de saneamento.

Os agentes tentarão alocar as áreas de interesse primeiro, mas terão que obedecer a ordem de precedência. Ao procurar uma residência um agente poderá residir em áreas não alocadas ou que contém outro agente. Caso a área esteja alocada o primeiro poderá ocupá-la caso tenha precedência sobre o atual residente.

A precedência é determinada pelas classes dos agentes. Agentes Upper class tem prioridade sobre as outras classes, mas não sobre a mesma classe. Agentes Middle class tem precedência sobre agentes Lower class e estes não possuem precedência.

Caso um agente não tenha precedência ele irá realizar uma busca na vizinhança de Moore da área atual em busca de outra residência. Este comportamento será observado até o agente alocar uma área. O mesmo ocorre com agentes deslocados.

O NetLogo possui uma unidade temporal chamada tick que representa a execução de um conjunto de tarefas dos agentes. Foi definido que cada dia real corresponde a quatorze ticks. Uma execução simulará dez anos tornando possível obter uma visão temporal da ocupação do município. A cada dois anos uma fotografia do estado do modelo será extraída e apresentada.

Execução do modelo

As seguintes fotografias foram tiradas durante a execução do cenário.

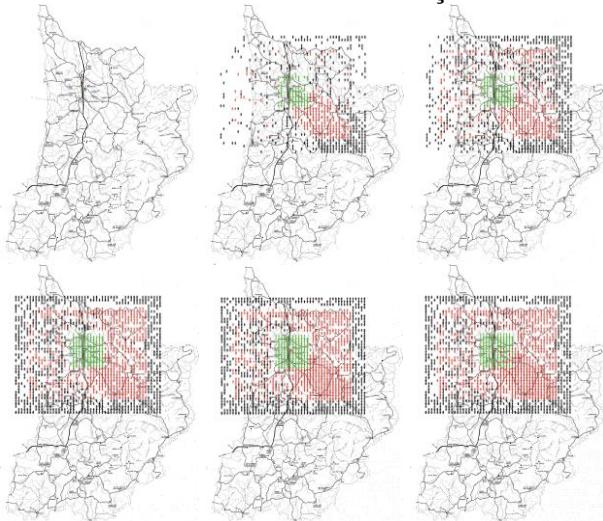


Ilustração 2 - Execução do primeiro conjunto de dados

As seguintes fotografias foram tiradas durante a execução do segundo cenário.

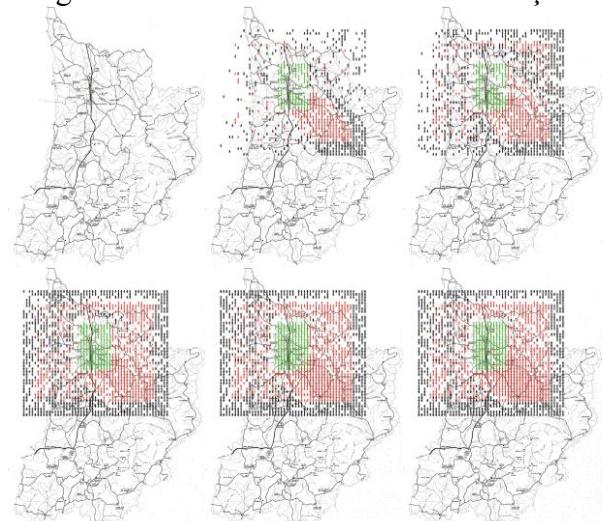


Ilustração 3 - Execução do segundo conjunto de dados

Conclusão

A criação de um modelo de crescimento urbano baseado em agentes possibilitou a execução de dois cenários distintos. O uso de agentes provedu flexibilidade ao modelo sendo que a única diferenciação entre as execuções foram os conjunto de dados.

As ilustrações 2 e 3 mostram a evolução da execução do modelo em um corte temporal permitindo-se verificar diferentes estágios do desenvolvimento. A partir do sexto ano o padrão final de alocação dos agentes é perceptível de maneira que os agentes Upper class se encontram principalmente no centro; os agentes Middle class se dispõem em um quadrado ao seu redor e os agentes Lower class se dispõem de maneira quadrangular a volta daqueles. Este comportamento se assemelha com o descrito pelo modelo de Kohl, [Maretto et al, 2010].

Ao longo da execução do modelo, em ambos os cenários, existe um movimento de disputa entre os agentes. Conforme o modelo é executado foi notado que agentes Lower Class e Middle Class, por se encontrarem em maiores quantidades, conseguiram alocar áreas privilegiadas. Posteriormente eles são deslocados por agentes Upper Class.

Conforme mencionado anteriormente a criação deste protótipo é a etapa inicial para a criação de um modelo que trabalhe com dados reais, e parametrizáveis, para produzir resultados sobre os mesmos. O estudo inicial feito a partir do modelo demonstra que a utilização desta ferramenta pode ser de extrema valia para gestores públicos.

Referências Bibliográficas

- Davidsson, Paul. (2000), “Multi Agent Based Simulation: Beyond Social Simulation”, em Multi Agent Based Simulation (LNCS Vol. 1979), Springer Verlag.
- Godoy, Marcela M. G. (2004), “Modelagem Dinâmica de Ocupação do Solo no Bairro Savassi, Belo Horizonte, Brasil”, Dissertação de Mestrado, Universidade Federal de Minas Gerais.
- Lana, Raquel M. (2009), “Modelos Dinâmicos Acoplados para Simulação da Ecologia do vetor do Aedes aegypti”, Dissertação de Mestrado, Universidade Federal de Ouro Preto.
- Mareto, Raian V. et al (2010), “Simulating Urban Growth and Residential Segregation through Agent-Based Modeling”, Brazilian Workshop on Social Simulation.
- Sjober, G.(1960), The Pre Industrial City: past and present American journal of sociology
- Svetsuk, Andres (2010), Path and place: a study of urban geometry and retail activity in Cambridge and Somerville, MA, Massachusetts Institute of Technology
- Teixeira, Gustavo G. e Soares-Filho, Britaldo S. (2009), “Simulação da tendência do desmatamento nas Cabaceiras do Rio Xingu, Mato Grosso – Brasil”, Anais XIV Simpósio Brasileiro de Sensoriamento Remoto, p. 5483-5490.
- Umbelino, Glauco. e Barbieri, Alisson. (2010), “Uso de Autômatos Celulares em estudos de população, espaço e ambiente”, Associação Brasileira de Estudos Populacionais.

Sobre Agentes e Controladores Preditivos

Marcelo Lopes de Lima^{1,2}, Jomi Fred Hübner¹, Eduardo Camponogara¹

¹Universidade Federal de Santa Catarina (UFSC)

Departamento de Automação e Sistemas
Florianópolis, SC

²Centro de Pesquisas da Petrobras (CENPES)
Rio de Janeiro, RJ

{mllima, jomi, camponog}@das.ufsc.br

Resumo. Neste trabalho estabelecemos uma relação entre agentes inteligentes BDI e controladores multivariáveis conhecidos como Controladores Preditivos baseado em Modelo (MPC). Os MPC são os controladores de maior sucesso na indústria de processos. Dessa relação propomos o que se poderia chamar de agentes MPC que representariam ganhos significativos na teoria e prática dos controladores atuais.

1. Introdução

Em geral, para o correto funcionamento dos processos industriais é necessário um estreito controle das variáveis mais importantes. Em uma refinaria de petróleo, por exemplo, a correta especificação da gasolina que utilizamos em nossos veículos depende de uma combinação de vazões, temperaturas e pressões que mantenham o processo de destilação do petróleo em um estado de equilíbrio adequado. Esse equilíbrio deve ser tal que garanta não só a especificação do produto mas também a segurança do processo considerando-se os limites físicos e legais das variáveis.

Para isso, são projetados controladores que devem tratar, de forma automática e robusta, as diversas interações entre as variáveis, as perturbações externas e os limites das variáveis impostos como restrições ao controlador. Como exemplo de perturbações externas ao controlador podemos citar: variações no tipo de petróleo, variações climáticas, desligamentos e mudanças no próprio processo por envelhecimento ou falhas. Os sistemas de controle estão, portanto, inseridos em um ambiente de variáveis *contínuas e discretas*, com restrições, interações e mudanças.

Uma das técnicas de controle mais empregadas na indústria de processos é o chamado Controlador Preditivo baseado em Modelo (também conhecido pela sigla em inglês, MPC - Model Predictive Controller) [Camacho and Bordons 2004]. Nesse tipo de controle, o processo é modelado através de equações que passam a constituir o conhecimento do controlador sobre o comportamento dinâmico do processo. O MPC têm como vantagens principais a possibilidade de incluir as interações e as restrições de processo diretamente na formulação do problema. Apesar de ser aplicado com grande sucesso, o MPC ainda se apresenta como um grande campo para pesquisas, principalmente no sentido de torná-lo mais robusto às não linearidades e aos erros de modelo, e mais adaptável às variações de processo.

Será mostrado que os controladores MPC apresentam semelhanças com a definição de *agentes inteligentes* feita por Wooldridge & Jennings em

[Jennings and Wooldridge 1998] e que têm uma arquitetura semelhante à arquitetura *Believe, Desire, Intentions* (BDI) [Bratman et al. 1988]. Os controladores poderiam ser vistos como *agentes* que atuam em um ambiente dinâmico com o propósito de manter as variáveis controladas em um ponto de equilíbrio. Muitas vezes esse ponto de equilíbrio é definido em um contexto maior de programação da produção.

O objetivo deste trabalho é o de propor uma classe de *agentes MPC* que incentivem a comunidade de Sistemas Multiagentes a interagir com a comunidade de Controle no sentido de desenvolver controladores (*agentes*) com maior robustez, escalabilidade e adaptabilidade do que os controladores MPC tradicionais. De igual importância é a relação entre os agentes e os operadores humanos e, também, a integração dos agentes ao objetivos corporativos, assunto pouco estudado pela comunidade de Controle.

2. Controlador Preditivo

O controlador preditivo é um programa composto por três blocos principais: um modelo do processo, uma função objetivo e um processo de decisão [Camacho and Bordons 2004]:

- *Modelo do processo*: permite fazer previsões do comportamento do sistema dado um plano de ações.
- *Função objetivo* (ou utilidade): serve como critério de decisão sobre a adequação do comportamento futuro previsto.
- *Processo de decisão*: permite a escolha do melhor plano de ações dado o critério de decisão. Em geral, o processo de decisão é um problema de otimização.

A Figura 1 mostra o funcionamento do MPC. De posse do modelo do processo, o controlador é capaz de calcular os efeitos futuros nas variáveis de saída \tilde{x} causados por um possível plano de ações \tilde{u} . O plano de ação escolhido será aquele que maximiza a utilidade do controlador, por exemplo, atingir rapidamente a referência com o mínimo de gasto de energia. Uma vez calculado o plano de ações ótimo, apenas a primeira ação do plano é aplicada ao processo (ambiente). Um novo ciclo se inicia com a leitura do novo estado atual e de um novo cálculo do plano de ações do qual a primeira ação será aplicada.

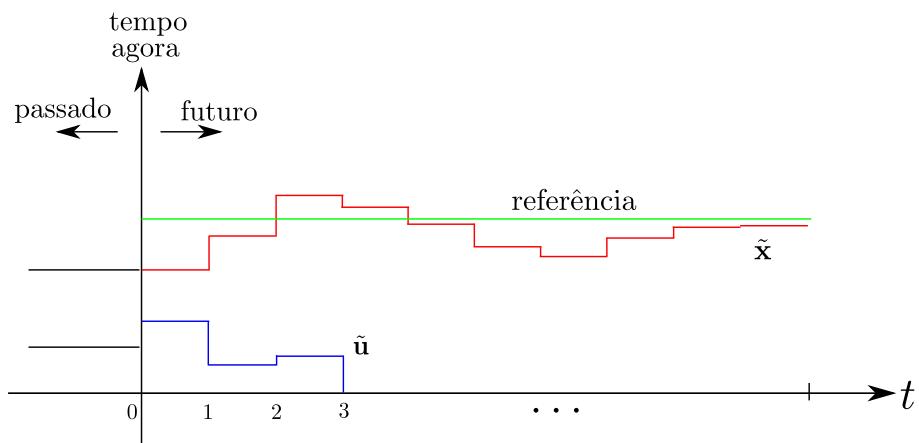


Figura 1. Visão Geral de um MPC

Poderão existir vários controladores MPC em um processo industrial, cada um responsável por um conjunto de variáveis e, nesse caso, deve-se cuidar da interação entre eles.

Podemos dizer, então, que o MPC é um sistema computacional, que interage com um ambiente e com os demais controladores, capaz de tomar ações de controle autônomas, baseadas em um modelo do ambiente, buscando atingir um objetivo representado por sua função utilidade.

Ao operador humano resta ligar ou desligar o controlador e ajustar os valores de referência para as variáveis de saída.

3. Agente Preditivo

Segundo [Jennings and Wooldridge 1998, Wooldridge 1999], um agente inteligente é um sistema computacional situado em um meio ambiente e capaz de tomar ações *flexíveis* e autônomas para atingir seus objetivos. Por flexibilidade entende-se: reatividade, proatividade e habilidade social, que resumiremos, de forma um tanto arbitrária, como adaptação e habilidade social. Um agente adaptável reage e se modifica de maneira adequada às mudanças do ambiente.

Um *Agente MPC* seria um agente com características tanto de um controlador MPC quanto de um agente inteligente. Observe na Tabela 1 que quanto mais flexível for o controlador MPC, mais próximo ele estará da definição de agentes inteligentes. Um agente MPC seria, portanto, um controlador MPC que incorpora a flexibilidade própria de agentes inteligentes.

Tabela 1. Comparação MPC e Agente MPC

Controlador MPC Distribuído	Agente Inteligente
<ul style="list-style-type: none"> • um sistema computacional • que interage com um ambiente • capaz de tomar ações (de controle) <ul style="list-style-type: none"> – autônomas – baseadas em modelo • para atingir um objetivo 	<ul style="list-style-type: none"> • um sistema computacional • que interage com um ambiente • capaz de tomar ações <ul style="list-style-type: none"> – autônomas – <i>flexíveis</i>: fruto de uma habilidade social e de adaptação • para atingir um objetivo

Também existe uma semelhança destacável entre os controladores MPC e os agentes BDI. Observe que é possível estabelecer um relacionamento direto entre os três blocos funcionais de um agente MPC, explicado na seção anterior, e os três blocos funcionais da arquitetura BDI. Esse relacionamento é:

1. Crenças \iff Modelo
2. Desejos \iff Função Objetivo
3. Intenções \iff Decisão

Dado as semelhanças entre os controladores MPC e os agentes BDI (inteligentes), a evolução dos controladores MPC tradicionais poderia se dar na direção da fusão desses conceitos e técnicas. Essa fusão resultaria no que se poderia chamar de agentes MPC.

Os agentes MPC poderiam resultar em inovações e em novos paradigmas teóricos aplicáveis à prática da engenharia de controle provendo os controladores, por exemplo, de maior capacidade de adaptação às falhas, às mudanças no ambiente, às ações dos outros e aos erros de modelagem (robustez) e não linearidades. É cada vez mais desejável, ainda, uma maior integração dos controladores MPC com os operadores humanos, evitando desligamentos, oferecendo diagnósticos, isolando falhas, e uma maior integração com os objetivos corporativos cujo escopo é muito mais abrangente que o escopo dos controladores.

4. Iniciativas na UFSC

Acreditamos que a busca por incorporar aos controladores MPC aspectos que o aproximariam mais daquilo que se poderia chamar de um *agente MPC* poderia ser de grande benefício na coordenação desses agentes, na integração com os sistemas corporativos e na interação com o operador humano.

Nesse sentido, podemos citar duas iniciativas em desenvolvimento no Departamento de Automação e Sistemas (DAS) da UFSC. Uma delas é focada no projeto de controladores distribuídos que sejam “altruístas” no sentido de incorporar em seus objetivos os objetivos dos agentes vizinhos, não de forma categórica, mas sim na medida certa à situação. Uma segunda iniciativa busca dotar esses agentes de um ambiente computacional efetivamente distribuído, em uma arquitetura BDI, proporcionada pela linguagem Jason [Bordini et al. 2007].

5. Conclusão

As semelhanças entre controladores MPC e agentes inteligentes sugere uma maior interação entre as comunidades de Controle e de Sistemas Multiagentes. Dessa interação poderiam surgir soluções de infra-estrutura de software, de interação com o operador humano, de sistemas de diagnósticos e isolamento de falhas, de coordenação e comunicação em aplicações distribuídas, integradas à programação da produção.

Referências

- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems In Agentspeak Using Jason*. John Wiley & Sons.
- Bratman, M. E., Israel, D. J., and Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355.
- Camacho, E. F. and Bordons, C. (2004). *Model Predictive Control*. Springer-Verlag, 2nd edition.
- Jennings, N. and Wooldridge, M. (1998). *Agent Technology: Foundations, Applications, and Markets*, chapter Applications of Intelligent Agents, pages 3–28. Springer-Verlag.
- Wooldridge, M. (1999). *Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence*, chapter 1 - Intelligent Agents, pages 27–78. The MIT Press.

Arquitetura de agentes credíveis em entretenimento eletrônico com processamento emocional multi-nível

Diego Prates de Andrade, Jomi Fred Hübner

¹Departamento de Automação e Sistemas – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

{diegopa, jomi}@das.ufsc.br

Resumo. Com a crescente demanda de jogos eletrônicos mais convincentes, a busca por técnicas que visem maior credibilidade torna-se eminente. Dentre as técnicas em voga, destaca-se o uso de emoções, e aqui propõe-se uma arquitetura de processamento emocional primário e secundário (multi-nível), visando obter um ganho em relação ao estado da arte em agentes credíveis para jogos eletrônicos.

1. Introdução

A indústria de jogos eletrônicos cresce a cada ano, adquirindo cada vez mais usuários de diferentes faixas etárias. Com este crescimento, a exigência feita pelos consumidores também cresce, e a busca por desafios mais realísticos torna-se uma demanda inevitável. Para permitir que tais desafios sejam superados, produtoras de jogos investem cada vez mais em recursos de inteligência artificial (IA), visando garantir um comportamento próximo ao que seres humanos teriam, enfrentando situações semelhantes às apresentadas nos cenários dos jogos.

Porém, ainda em muitos casos, o comportamento de alguns personagens ainda não passa a sensação de realismo ao jogador, apresentando normalmente ações apenas reativas, deterministas, e consequentemente, previsíveis. Este tipo de comportamento faz com que o jogador tenha dúvidas quanto a credibilidade dos NPCs (*Non-Player Character*, ou Personagem não-jogador). Esta dúvida pode comprometer drasticamente a sensação de imersão que o usuário gostaria de ter ao interagir com este tipo de entretenimento eletrônico. Fazendo com que este tenha descrença quanto a ilusão de vida do agente, e consequentemente, deixe de se identificar com o mesmo.

De acordo com Bates, o termo *believable agents*, ou agentes credíveis, como é comumente traduzido, designa não apenas agentes com uma maior autonomia, ou mesmo uma maior capacidade de resolução de problemas, mas sim a capacidade de “suspensão da descrença”, fazendo com que o espectador deixe de vê-los como seres totalmente artificiais e creia em seus comportamentos e atitudes [Bates 1994].

A procura por características comumente relacionadas ao conceito de inteligência, deve-se ao fato de que pesquisadores de IA normalmente focam seus estudos apenas em aspectos funcionais, deixando de lado características importantes como as emoções, que são essenciais para a credibilidade dos agentes [Bates et al. 1991].

Damásio divide as emoções em duas categorias distintas: emoções primárias e emoções secundárias. Sendo que entende-se por emoções primárias, os mecanismos fisiológicos responsáveis pela regulação do estado interno do corpo, tais como atividade

hormonal [Canamero 1997], sistema límbico, etc. Estes mecanismos podem realizar ações de tomada de decisão simples, como definir se uma determinada situação oferece perigo ou não, além de englobar outras características importantes como reflexos ou instintos, possibilitando uma reação mais rápida e independente do restante do processo deliberativo. Em contrapartida, as emoções secundárias estão situadas em um nível mais elaborado do processo de tomada de decisão, sendo normalmente atribuídas ao conceito mais comum de emoção, como o amor, a raiva, o medo, etc. Assim como as emoções primárias, as emoções secundárias também mantêm um caráter volátil, mas normalmente seus efeitos são mais duradouros, podendo influenciar futuras decisões [Damásio 1994].

Dadas algumas das necessidades levantadas na área de jogos eletrônicos, este documento descreve uma proposta preliminar de uma arquitetura de agentes credíveis, visando melhorar a credibilidade dos mesmos no setor de entretenimento eletrônico. Esta arquitetura busca primeiramente levar em conta as emoções primárias e secundárias, acrescentando ao processo deliberativo o processamento emocional multi-nível, estratégia pouco explorada na área. Além disto, espera-se utilizar fatores como a personalidade e o humor, visando incrementar a ilusão de vida proporcionada pelos agentes.

2. Trabalhos relacionados

Um dos trabalhos pioneiros na área, a arquitetura Tok, originada do projeto Oz, surgiu através da junção de outros subsistemas do mesmo projeto, somando processamento cognitivo, emocional e sensorial. Tendo como cerne o conceito de agentes com amplas capacidades, tais como reconhecimento de linguagem natural, sensoriamento, raciocínio, emoções, relações sociais, etc., ou simplesmente *broad agents* [Bates et al. 1991]. Porém, percebeu-se que não era necessário grande aprofundamento destas capacidades, que apenas uma abordagem superficial já atendia a demanda de credibilidade do projeto, possibilitando que os recursos fossem destinados a uma melhor integração destas capacidades. Este tipo de abordagem ficou conhecida como “*broad but shallow agent*”, ou agente amplo porém superficial, e é uma das bases para o funcionamento da arquitetura Tok [Bates et al. 1994].

Outro trabalho interessante é o *software Creatures* [Grand et al. 1997], baseado na arquitetura *Cyberlife*, e que tem como objetivo a simulação de vida artificial. Assim como outros trabalhos em agentes credíveis, esta arquitetura é composta por um mundo virtual, os agentes que o habitam e de objetos que podem ser manipulados pelo usuário ou por outros agentes. Ao contrário da maior parte dos trabalhos da área, o uso de redes neurais artificiais ocupa o lugar da clássica inteligência artificial simbólica. Para a regulação dos limiares de saída dos neurônios, foram utilizados algoritmos genéticos, fazendo com que o comportamento dos agentes fosse evolutivo. Em conjunto com as tecnologias citadas, foi utilizada uma estrutura bioquímica artificial, visando criar uma segunda camada de processamento, responsável pelas emoções primárias, assim como o trabalho de [Canamero 1997].

Vários outros trabalhos visam suprir a demanda de agentes cada vez mais credíveis, tomando diferentes abordagens, mas mantendo algumas características gerais, e objetivos que desejam alcançar, tais como tratamento de emoções, comportamento dirigido a objetivo, personalidades ricas, ações coerentes, personificação e comportamento evolutivo (aprendizado).

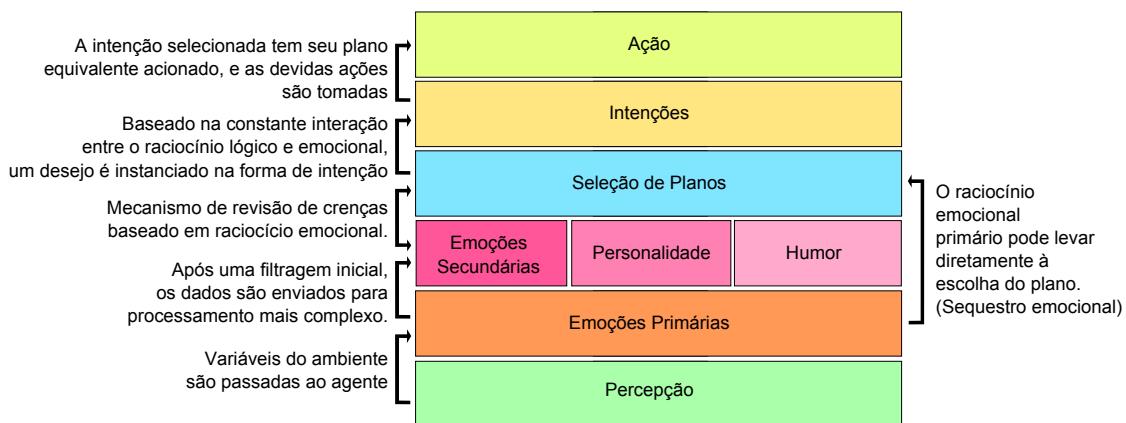


Figura 1. Diagrama de funcionamento da arquitetura proposta.

3. Proposta do modelo de processamento emocional multi-nível

Como visto em alguns dos trabalhos na área, o uso de emoções é quase que unânime, indicando que este é um bom caminho a ser tomado para o desenvolvimento de agentes credíveis. Também parece promissora a integração de processamento emocional primário e secundário, além da capacidade de demonstrar personalidades ricas, permitindo que a escolha de comportamento possa ser fortemente dirigida a personalidade do agente. Dito isto, propõe-se a criação de uma nova arquitetura para agentes credíveis, juntamente com um *middleware* que permita sua integração com jogos eletrônicos e dramas interativos.

Como ilustrado na Figura 1, o ciclo de funcionamento da arquitetura aqui proposta, inicia-se com o tratamento inicial da percepção do agente, através do processamento das emoções primárias. Nesta primeira etapa, algumas atitudes que se observadas em um ser biológico, poderiam ser consideradas reflexos, ou mesmo instintos, dão uma maior capacidade de reatividade ao agente. Já que em algumas circunstâncias, tais como no medo ou na raiva, as emoções primárias podem sobrepurar o restante do modelo cognitivo, levando o agente diretamente a uma ação, fato denominado como “sequestro emocional”[Goleman 2007]. Logo após este primeiro processamento, a constante interação entre o processamento de emoções secundárias, personalidade e humor, influenciam na tomada de decisão. A divisão entre emoções primárias e secundárias permite que o agente tenha a capacidade de reagir emocionalmente de maneira mais convincente, tal como pode ocorrer em acessos de raiva (sequestro emocional), expressar emoções complexas como o amor, etc.

Verificando que a arquitetura demanda de características ao mesmo tempo reativas e deliberativas, optou-se utilizar a arquitetura BDI (*Beliefs, Desires and Intentions*), que apresenta traços semelhantes [Bratman 1987]. Para permitir o uso desta arquitetura, foi escolhida a plataforma *Jason* [Bordini et al. 2007], baseada na linguagem *AgentSpeak* [Rao 1996], visto que esta já é de conhecimento dos autores.

Um dos grandes desafios para a execução deste projeto, será somar à arquitetura BDI existente, o conceito de *Broad-Shallow Agents*. Integrando várias habilidades como processamento emocional multi-nível (emoções primárias e secundárias) [Damásio 1994], comportamento dirigido à personalidade, personificação complexa, tratamento de humor, relações interagentes, etc [Loyall 1997].

4. Considerações finais e trabalhos futuros

Na continuação deste projeto, pretende-se concluir a implementação da arquitetura[Andrade and Hübner], possibilitando a criação de agentes com diferentes capacidades e variados graus de complexidade, possibilitando a experimentação em diferentes cenários. Além de promover um método de avaliação de credibilidade consistente, através da utilização de uma mescla de dados subjetivos, tais como uma versão adaptada do teste de Turing, com questionários e outros métodos a serem desenvolvidos [Livingstone 2006, Tencé and Buche 2008].

Com o intuito de simular o ambiente e permitir uma personificação completa dos agentes, será utilizado um motor 3D, um jogo eletrônico já existente, ou um gerenciador de drama interativo. A tecnologia à ser utilizada ainda será definida, mas deve abranger alguns fatores essenciais, tais como facilidade de integração com a plataforma Jason, e consequentemente com a plataforma Java, capacidade de demonstrar expressões faciais, modo de caminhar, alterações na fala (quando existir processamento de linguagem natural), etc.

Referências

- Andrade, D. P. and Hübner, J. F. Project BAAL. <http://baal.sourceforge.net>.
- Bates, J. (1994). The role of emotion in believable agents.
- Bates, J., Loyall, A., and Reilly, W. (1994). An architecture for action, emotion, and social behavior. *Artificial social systems*, pages 55–68.
- Bates, J., Loyall, B., and Reilly, W. S. (1991). Broad agents. *ACM SIGART Bulletin*, pages 38–40.
- Bordini, R., Hübner, J., and Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*. Wiley-Interscience.
- Bratman, M. (1987). Intention, Plans, and Practical Reason.
- Canamero, D. (1997). A hormonal model of emotions for behavior control. *VUB AI-Lab Memo*.
- Damásio, A. (1994). O erro de Descartes: emoção, razão e o cérebro humano.
- Goleman, D. (2007). Inteligência emocional: a teoria revolucionária que redefine o que é ser inteligente.
- Grand, S., Cliff, D., and Malhotra, A. (1997). Creatures: Artificial life autonomous software agents for home entertainment. In *Proceedings of the first international conference on Autonomous agents*, pages 22–29. ACM.
- Livingstone, D. (2006). Turing’s test and believable AI in games. *Computers in Entertainment (CIE)*, page 6.
- Loyall, A. (1997). *Believable agents: building interactive personalities*. PhD thesis.
- Rao, A. (1996). AgentSpeak (L): BDI agents speak out in a logical computable language. *Agents Breaking Away*, pages 42–55.
- Tencé, F. and Buche, C. (2008). Automatable evaluation method oriented toward behaviour believability for video games. In *International Conference on Intelligent Games and Simulation*, pages 39–43.

Índice de Autores

- Adamatti, D., 159
Alcantara, L. A., 191
Andrade, D. P., 199
Arruda, L. V., 39

Bagatini, D., 175
Banaszewski, R. F., 39
Barros, P. R., 97, 179
Bez, M., 97
Brito, M., 129
Brun, A. L., 155

Camponogara, E., 195
Campos, D., 167
Cazella, S. C., 97, 179
Choren, R., 63

Dahmer, A., 179
Dimuro, G., 51, 163

Emmendorfer, L., 51
Erhardt, E., 97, 179
Esmaniotto, F. J., 191

Flores, C. D., 97, 179
Fonseca, J. M., 179
França, C. H., 155
Frozza, R., 175

Hübner, J. F., 129, 195, 199
Haensch, M. O., 167

Jugueiro, M., 159
Künzel, M. F., 175
Krusche, N., 171

Lima, D. V., 171
Lima, M. L., 195
Loh, S., 141
Lorbieski, R., 155
Lorenzi, F., 141
Lucena, C., 63

Lugo, G. A. G., 121, 187, 191
Lux, B., 175

Macohin, A., 187
Maroni, V., 179
Merlin, L. P., 155
Mossmann, J. B., 179

Nardin, L. G., 75
Nascimento, C. A., 121, 187
Nogueira, K. E., 39
Nunes, G., 159

Pereira, A. H., 75

Rabelo, R., 109
Rizzi, C. B., 155
Rizzi, L. R., 155
Rocha Costa, A. C., 3, 51, 163, 171
Rocha, M., 141
Rodrigues, L. M., 51
Rodrigues, T., 163

Santos, I., 163
Sichman, J. S., 75
Silva, A. K., 175
Silva, J. L. T., 85
Silva, M. V. G., 121
Silveira, R. A., 167
Simão, J. M., 39

Tacla, C. A., 39, 183
Taranti, P., 63
Thomasi, C., 159

Uez, D., 85

Vieira, K. P., 183

Zambiasi, S. P., 109
Zanatta, E., 179
Zara, R., 155