

IVERTON ADÃO DA SILVA DOS SANTOS

Um modelo e um framework para a Simulação de
Políticas Públicas por meio de Sistemas Multiagentes,
Jason e CArtAgO

FURG

Rio Grande

2013

UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM
COMPUTACIONAL

Um modelo e um framework para a Simulação de Políticas Públicas por meio de Sistemas Multiagentes, Jason e CArTAgO

Por

Iverton Adão da Silva dos Santos

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Modelagem
Computacional da Universidade Federal do
Rio Grande – FURG.

Orientador: Prof. Dr. Antônio Carlos da
Rocha Costa

Rio Grande, 2013.

Catálogo Internacional na Publicação (CIP)

S237m Santos, Iverton Adão da Silva dos.
Um modelo e um framework para a simulação de políticas públicas por meio de sistemas multiagentes, Jason e Cartago / Iverton Adão da Silva dos Santos. – 2013.
103 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande/FURG, Programa de Pós-Graduação em Modelagem Computacional.

Orientador: Dr. Antônio Carlos da Rocha Costa.

1. Modelagem e simulação baseada em agentes. 2. Políticas públicas. 3. Jason. 4. Cartago I. Costa, Antônio da Rocha . II. Título.

CDU:004.891

Catálogo na fonte: Bibliotecária Alessandra de Lemos CRB10/1530

IVERTON ADÃO DA SILVA DOS SANTOS

"UM MODELO E UM FRAMEWORK PARA A SIMULAÇÃO DE POLÍTICAS PÚBLICAS POR MEIO DE SISTEMAS MULTIAGENTES, JASON E CARTAGO"

Dissertação apresentada ao Programa de Pós Graduação em Modelagem Computacional da Universidade Federal do Rio Grande -FURG, como requisito parcial para obtenção do Grau de Mestre. Área concentração: Modelagem Computacional.

Aprovada em

BANCA EXAMINADORA



Prof. Dr. Antônio Carlos da Rocha Costa
Orientador - FURG



Prof.ª Dra. Graçaliz Pereira Dimuro
FURG



Prof.ª Dra. Raquel de Miranda Barbosa
FURG



Prof. Dr. Michael da Costa Móra
PUCRS

Rio Grande -RS
2013

*Dedico este trabalho à minha esposa Priscila Holz,
pelo amor e apoio que sempre me deu.*

AGRADECIMENTOS

A Deus, por nunca deixar de prover suas bênçãos sobre minha vida.

A minha esposa Priscila Holz, por ter ido comigo para Rio Grande para que eu realizasse o sonho de fazer o mestrado. Também por ter me incentivado em cada momento de tempo que dediquei para realizar este trabalho.

Aos meus pais, Ivo e Gecibel, pelo exemplo, apoio e carinho.

Ao meu orientador, Professor Antonio Carlos da Rocha Costa, pela atenção, paciência, motivação e compartilhamento de uma fração de seus conhecimentos.

Ao corpo docente do PPGMC pelas importantes aulas ministradas.

Aos meus colegas Flávia, Fernanda, Thiago, Maria Clara, pelos momentos que compartilhamos juntos.

Aos meus grandes amigos André Mendes da Rosa e Carlos Quadros, por serem meus companheiros a toda hora em Rio Grande.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro.

Obrigado a todos, são os responsáveis pelo alcance deste sonho. A partir deste momento a minha vida toma um novo rumo!

SUMÁRIO

RESUMO	9
ABSTRACT	10
1 INTRODUÇÃO	
1.1 Contextualização do trabalho e justificativa	11
1.2 Objetivos	12
1.3 Metodologia	13
1.4 Estrutura do Texto	14
2 POLÍTICAS PÚBLICAS	
2.1 Conceitos gerais sobre Políticas Públicas	15
2.2 Modelo Sistêmico	16
2.3 Modelo de Ciclo de Política	19
2.4 Os Instrumentos do governo	20
3 MODELAGEM E SIMULAÇÃO BASEADA EM AGENTES	
3.1 Agentes, Sistemas Multiagentes e seu uso em Modelagem e Simulação	24
3.3 Arquitetura BDI e AgentSpeak(L)	27
3.3 Jason	31
3.4 CArTAgO	32

4 UM MODELO BASEADO EM AGENTES PARA A SIMULAÇÃO DE POLÍTICAS PÚBLICAS

4.1 Uma abordagem sobre o <i>modelo sistêmico</i> e nos <i>instrumentos do governo</i>	34
4.2 Os componentes e a organização do modelo	35
4.2.1 Normas e Planos	36
4.2.2 Agentes Sociais e Recursos Comuns	36
4.2.3 Agente Governo e Agentes Detectores e Atuadores	36

5 UM FRAMEWORK BASEADO EM AGENTES PARA A SIMULAÇÃO DE POLÍTICAS PÚBLICAS

5.1 Uma implementação sobre as ferramentas Jason e CArtaGO	38
5.2 Os componentes, o funcionamento e a organização do framework: Uma compreensão em três cenários	38
5.2.1 Cenário 1: Normas de Obrigação	40
5.2.2 Cenário 2: Normas de Proibição	43
5.2.3 Cenário 3: Planos de ação	46
5.4 Considerações sobre o framework	48
5.5 Códigos Java e AgentSpeak(L) do framework: exemplo e comentários	50

6 ESTUDO DE CASO

6.1 Piracema e a tragédia dos comuns	69
6.2.1 O cenário descritivo utilizado	70
6.2.2 A modelagem e Simulação utilizando o modelo e framework elaborados	72
6.2.3 Discussão sobre os resultados obtidos na simulação	78

7 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	79
REFERÊNCIAS	81
ANEXO 1: Um exemplo básico em AgentSpeak(L)	85
ANEXO 2: Log de execução de um exemplo hipotético de simulação utilizando o framework	89
ANEXO 3: Log de execução da simulação do estudo de caso	98
ANEXO 4: Publicações do Trabalho	103

RESUMO

Este trabalho está inserido na área de Sistemas Multiagentes, mais especificamente Modelagem e Simulação Baseada em Agentes. Sua contribuição está relacionada aos aspectos instrumentais do processo de simulação de Políticas Públicas. Para isto, arquiteta, com base na literatura do processo de elaboração Políticas Públicas, um modelo compatível com o paradigma multiagente, bem como um *framework* com suporte sobre as ferramentas Jason e CArtAgO, que permite implementar de forma sistemática e em alto nível tal modelo. Por fim, adota exemplos e um estudo de caso na área de Políticas Públicas para gestão de recursos comuns, de forma a demonstrar a utilização daquele modelo e *framework*.

Palavras-chave: Modelagem e Simulação Baseada em Agentes, Políticas Públicas, Jason, CArtAgO.

ABSTRACT

This work is placed in the area of multiagent systems, more specifically Modeling and Agent-Based Simulations. Your contribution is related to the instrumental aspects of the simulation process for Public Policy. For this, an architect, based on the literature of public policy making process, a model compatible with the multi-agent paradigm and a framework with support on the Jason and CarTAgO tools, which allows you to implement a systematic and high-level such model. Finally, examples, and adopts a case study in the area of Public Policy for the management of common resources, in order to demonstrate the use of that model and framework.

Keywords: Agent-based models, Public Policy, Jason, CArtAgO.

1 INTRODUÇÃO

1.1 Contextualização do trabalho e justificativa

De acordo com (SILVA et al, 2011), formular políticas públicas tornou-se um grande desafio aos gestores públicos, uma vez que suas ações têm consequências diretas na vida dos cidadãos, seja promovendo benfeitorias, o que representa um ótimo retorno de investimento público – tanto do ponto de vista dos recursos como tempo, dinheiro e energia, despendidos pelo Estado, quanto do ponto de vista das melhorias nas vidas dos beneficiados direta ou indiretamente pela política; seja representando um mau negócio, que acarreta desperdício de recursos – materiais e humanos – e o consequente fracasso no alcance das metas estabelecidas para atender às demandas cidadãs.

Nesta perspectiva, como é descrito por (DESOUZA; LIN, 2011), desenhar políticas públicas exige o emprego de esforços na busca de soluções que consigam diminuir as possibilidades de erros em um cenário tão complexo, envolto por inúmeras condições e fatores que mudam a cada momento, como é o *sistema* de políticas públicas.

Especificamente no que se refere às pesquisas na área de Modelagem Computacional, pode-se dizer que a técnica de simulação de sistemas multiagentes é uma oportuna alternativa de contribuição ao desenho de políticas públicas, especialmente pelo seu potencial para modelagem, simulação e análise de sistemas complexos (GILBERT, 2008; GILBERT; TROITZSCH, 2005).

Segundo (RUSSELL; NORVIG 2004), um agente de software é uma entidade capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores. Os agentes inteligentes podem ser classificados de acordo com a maneira que eles coletam informações e agem no

ambiente. No caso de vários agentes cooperando ou disputando entre si, inseridos em um mesmo ambiente e trocando informações, chamamos esse sistema de multiagente (SMA).

Neste sentido, a simulação com SMA se baseia na ideia que é possível representar de maneira computadorizada o comportamento de entidades que são ativas no mundo, e através disto representar um fenômeno como o produto de interações de um conjunto de agentes que possuem autonomia operacional (FERBER, 1999).

É nesta direção que transcorre esta dissertação, de modo que sua contribuição se refere à concepção de um modelo e um *framework* que possam servir de suporte à modelagem e simulação de políticas públicas por meio do paradigma de sistemas multiagentes.

1.2 Objetivos

O objetivo geral deste trabalho é contribuir com os aspectos instrumentais do processo de simulação de Políticas Públicas. Disto, decorrem os seguintes objetivos específicos:

- Elaborar um modelo de simulação de Políticas Públicas compatível com o paradigma de Sistemas Multiagentes.
- Conceber um *framework*, com suporte na infraestrutura das ferramentas Jason e CArTAgO, no qual conste um conjunto de construções de código em alto nível de abstração (na forma de artefatos e programas AgentSpeak), o qual possa ser utilizado para a programação de simulações com base naquele modelo elaborado.
- Demonstrar com exemplos e estudos de caso a utilização do *framework* para implementar e executar simulações baseadas no modelo elaborado.

1.3 Metodologia

No que se refere ao aspecto metodológico de concepção deste trabalho, pode-se destacar dois pontos. O primeiro está relacionado à concepção do modelo, o qual teve como base de estudos de análise de política as abordagens de: Modelo Sistêmico, Modelo Cíclico e Instrumentos do Governo.

Como dito em (RODRIGUES, 2010) o conceito de políticas públicas não é único, pois varia de acordo com o enfoque teórico adotado e o contexto social ao qual ele se aplica. Ainda, o autor afirma que, de pronto, a resposta à pergunta “o que é política pública?” não é simples.

Contudo, segundo pesquisadores da área (HAM; HILL, 1993) (RUA, 2009), tais abordagens são predominantes na literatura, bem como lidam com esta tarefa de forma a simplificar, abstrair e sistematizar sua complexidade. Deste modo, considerou-se que as mesmas são adequadas e suficientes de um ponto de vista direcionado à modelagem computacional.

O segundo está relacionado à construção do *framework* para programar e executar o modelo proposto. Para este, optou-se pelo uso das abstrações e funcionalidades inerentes à infraestrutura das ferramentas Jason e CartAgO. O Jason pelo seu suporte expressivo na programação de agentes cognitivos. Já o CartAgO pela sua metáfora de abstração em alto nível de ambientes virtuais de agentes.

1.4 Estrutura do texto

O texto desta dissertação está estruturado em sete capítulos, com os seguintes conteúdos:

- **Capítulo 1** - A contextualização e justificativa do trabalho, assim como os objetivos e a metodologia;
- **Capítulo 2** - Conceitos e modelos teóricos fundamentais sobre Políticas Públicas;
- **Capítulo 3** – Uma síntese sobre agentes e sistemas multiagentes e sua aplicação em modelagem e simulação. Também são vistos uma arquitetura, linguagem e ferramentas para concepção de sistemas multiagentes;
- **Capítulo 4** - Um modelo para simulação de políticas públicas, utilizando a modelagem baseada em agentes;
- **Capítulo 5** – Um framework com as construções das ferramentas Jason e CArTAgO para a simulação de políticas públicas;
- **Capítulo 6** - Um estudo de caso de aplicação do modelo e framework elaborados (*Políticas Públicas de defeso, Piracema e o problema da tragédia dos comuns*);
- **Capítulo 7** – Considerações finais sobre os resultados alcançados por esta proposta;

2 POLÍTICAS PÚBLICAS

2.1 Conceitos gerais sobre Políticas Públicas

Segundo (LOPES, B.; J, AMARAL.; CALDAS, R, 2008), para atingir resultados em diversas áreas e promover o bem-estar da sociedade, os governos se utilizam das Políticas Públicas, que os autores apontam que podem ser definidas da seguinte forma: “(...) Políticas Públicas são um conjunto de ações e decisões do governo, voltadas para a solução (ou não) de problemas da sociedade (...).”

Segundo (RODRIGUES, 2010), uma das principais, se não a principal, característica das políticas públicas é que elas são ações e decisões que se revestem da autoridade soberana do poder público. Isso significa, de certa forma, que elas possuem um caráter impositivo, ou seja, força para se fazer valer. As políticas públicas estão ligadas, igualmente, à ideia de governo. Governo, segundo a autora, deve ser entendido como aquele conjunto de pessoas que, exercendo o poder político, orientam os rumos que a sociedade tomará, pois estão na cúpula do Estado (esferas federais, estaduais e municipais). As pessoas que fazem parte do Governo participam das decisões que resultarão em políticas públicas. Cabe a elas a escolha entre as diversas possibilidades e alternativas.

Ainda, de acordo com (SILVEIRA, 2007), Políticas Públicas podem ser entendidas como um conjunto de normas que orientam práticas e respaldam os direitos dos indivíduos em todos os níveis e setores da sociedade. Elas devem ter como base os princípios da igualdade e da equidade, disseminando o sentido de justiça social. Por meio delas, os bens e serviços sociais são distribuídos, redistribuídos, de maneira a garantir o direito coletivo e atender às demandas da sociedade.

2.2 Modelo Sistêmico

Em sua obra (WILDAVSKY, 1979) lembra de que “política é um processo e também um produto. O termo é usado para se referir ao processo de tomada de decisões e igualmente ao produto deste processo”. Segundo (HAM; HILL, 1993) ao procurar compreender as complexidades do processo de tomada de decisões, autores têm posto em evidência uma variedade de modelos dentre os quais o enfoque sistêmico esboçado por (EASTON, 1965) tem recebido considerável proeminência. De acordo com (RUA, 2009) o modelo sistêmico é o quadro analítico para o estudo de políticas públicas que predomina nas ciências sociais e que permite uma compreensão mais substantiva sobre como elas surgem, estruturam-se e tornam-se diretivas de governo em situações específicas.

Segundo (EASTON, 1965) os sistemas políticos são como sistemas biológicos e existem em um ambiente que contém uma variedade de outros sistemas, incluindo sistemas sociais e ecológicos. Para o autor, o que distingue as interações políticas de todos os outros tipos de interações sociais é que estão predominantemente orientadas para a alocação autoritária de valores em uma sociedade.

Em resumo, as alocações autoritárias distribuem de uma forma ou mais, entre três formas possíveis, coisas valorizadas entre as pessoas e os grupos. Uma alocação pode privar uma pessoa de algo valorizado que possua; pode impedir a consecução de valores que de outra forma teriam sido obtidos; ou pode permitir o acesso de pessoas a valores e impedir o de outras.

Uma alocação é autoritária quando as pessoas que a sofrem consideram-na como obrigatória. Existem várias razões para que membros de um sistema possam considerar-se obrigados. Mas, desprezando os aspectos particulares, é o fato de considerar as alocações como coercitivas que distingue as alocações políticas, segundo a conceituação que o autor usa, de outros tipos de alocações.

No modelo do autor, estes outros sistemas ainda são classificados em ambiente *intra-social* e ambiente *extra-social*, os quais juntos compõem o ambiente total do sistema político.

O ambiente *intra-social*: é aquela parte do meio ambiente físico-social que está fora do sistema político e, no entanto, dentro da mesma sociedade; Uma depressão na economia, uma mudança de valores e aspirações na cultura ou uma alteração na estrutura de classes podem ter consequências para o sistema político; no entanto, ocorrem na mesma sociedade que contém o sistema político.

O ambiente *extra-social*: é aquela parte do meio ambiente físico-social que está fora do sistema político e, no entanto, não está na mesma sociedade que o contém; Sob o ponto de vista dos Estados Unidos, a França é uma sociedade que contém um sistema político cujas ações terão consequências que poderão ultrapassar o limite do sistema político americano e concorrer para a formação de seu destino.

Para explicar os intercâmbios e transações entre o sistema político e seu meio ambiente, o autor desenvolveu um *modelo circular de sistema político*, que, resumidamente, consiste no seguinte:

- Um dado sistema político recebe *insumos*¹ de seu meio-ambiente;
- Por intermédio de estruturas e processos internos – numa operação denominada de conversão -, o sistema atua sobre essas entradas, transformando-as em *produtos*²;
- Os *produtos* retornam ao meio-ambiente;
- Isto transcorre num ciclo contínuo de “*feedback*” - dando-lhe condições de reajustar seu comportamento.

¹ Na literatura o termo varia entre *input* e *insumo*

² Na literatuta o termo varia entre *output* e *produtos*

Conforme (BRASILEIRO, 1973), os insumos, em seu sentido amplo, consistem em qualquer evento que altera, modifica ou afeta o sistema de qualquer maneira. Entretanto, em (EASTON, 1965) os mesmos são classificados em duas grandes categorias: *demandas*³ e *apoio*⁴. As demandas abrangeriam as reivindicações, sugestões, informações, ordens que entrariam no sistema político. O apoio seria a base com a qual o sistema processa as demandas: recursos financeiros, humanos, sentimentos de lealdade, obediência às leis etc.

Segundo (BRASILEIRO, 1973), os produtos, em seu sentido amplo, poderiam também ser descritos como todos os acontecimentos que têm consequências para o ambiente. Porém (EASTON, 1965), emprega-os apenas para designar tipos de ocorrência que envolvam alocação autoritária de valores: decisões coercitivas e as ações relacionadas que as completam. Por exemplo, normas, leis, regulamentos, decisões administrativas, planos de ação, favores e benefícios concedidos.

Segundo (HAM; HILL, 1993), um dos méritos do modelo sistêmico é que ele chama atenção para o relacionamento entre sistemas políticos e outros sistemas. Estes outros sistemas são mencionados como sendo simplesmente o meio-ambiente do sistema político.

Como comenta (MINOGUE, 1983), o que governos fazem envolve o todo da vida social, econômica e política, seja prática ou potencialmente. Políticas públicas são, auto-evidentemente, não um campo estreito de investigação, embora analistas de políticas possam bem se concentrar apenas em áreas estreitas de todo o campo. Políticas públicas fazem coisas a economias e sociedades, de forma que, em última análise, qualquer teoria explicativa satisfatória de políticas públicas deve também explicar as inter-relações entre Estado, política, economia e sociedade.

³ Na literatura o termo varia entre *demanda* e *exigência*

⁴ Na literatura o termo varia entre *apoio* e *suporte*

2.3 Modelo de ciclo de política

Segundo (RUA, 2009) uma forma de lidar com essa complexidade envolvida nas políticas públicas, sem descartar a dinâmica sistêmica, é associar o modelo sistêmico com o modelo do ciclo de política (*policy cycle*), que aborda as políticas públicas mediante a sua divisão em etapas sequenciais.

- Identificação e formulação da questão a ser tratada, no contexto socioeconômico em foco;
- Formulação e análise de soluções alternativas, isto é, das possíveis políticas para o tratamento da questão;
- Escolha de uma solução para ser implementada, isto é, escolha da política a ser implementada;
- Implementação da política;
- Monitoração e Avaliação dos efeitos da política e consequente revisão/reformulação da mesma.

Assim, como conclui (RODRIGUES, 2010): Políticas públicas são ações de Governo, portanto, são revestidas da autoridade soberana do poder público. Dispõem sobre ‘o que fazer’ (ações), ‘aonde chegar’ (metas ou objetivos relacionados ao estado de coisas que se pretende alterar) e ‘como fazer’ (estratégias de ação).

Portanto, pode-se dizer que tanto no modelo sistêmico quanto no modelo ciclo de política, a política é concebida como um sistema de entradas e saída (*input-output, ação/retroação*) o qual é formado por interações com seu meio e que responde adaptando-se melhor ou pior a ele.

Segundo (EASTON, 1965), a capacidade de um sistema responder derivará de dois de seus processos internos: A informação sobre o estado do sistema e seu

ambiente poderem ser comunicada as autoridades; O sistema ser capaz de agir para tentar mudar ou manter alguma condição dada do sistema.

Neste sentido, ressalta-se a necessidade de *ferramentas* que forneçam suporte para captar informações sobre o contexto em questão, bem como para alterá-lo. Estes são respectivamente denominados segundo (HOOD; MARGARATTS, 2007) como *detectores* e *efetadores*, os quais são detalhados na próxima seção.

2.4 Os Instrumentos do Governo

De acordo com a abordagem de (HOOD; MARGARATTS, 2007), podemos imaginar o governo possuindo um conjunto de ferramentas administrativas - como ferramentas para carpintaria ou jardinagem.

Deste modo, o autor distingue em dois conjuntos simples, o que chama de instrumentos do governo. Primeiro, distingue entre instrumentos de governo para "detecção" denominados detectores e seus instrumentos para "efetivação" denominados efetadores. Detectores são tudo que o governo utiliza para absorver informação. Efetadores são tudo que o governo pode usar para tentar fazer impacto no mundo exterior.

Segundo o autor, para o governo, o qual é - ou pretende ser - pré-eminentemente uma maneira de controlar a sociedade, esses recursos são fundamentais para a sua existência e, certamente, a sua eficácia.

Ainda, afirma que o governo baseia seus detectores e efetadores sobre um esquema denominado 'NATO'. Que é apenas uma sigla conveniente que resume quatro recursos básicos que os governos tendem a possuir em virtude de serem governos. Estes quatro recursos básicos são "*nodalidade*", "*autoridade*", "*tesouro*" e "*organização*".

Cada um destes tipos de ferramenta pode ser implantado tanto para a coleta de informações, ou “detecção”, quanto para moldar o comportamento, ou “efetivação”. Ou seja, cada uma deles proporciona uma capacidade de governo diferente, pode ser “gasto” de uma forma diferente, e está sujeita a um limite diferente, conforme a seguir:

- **Nodalidade** - denota a propriedade de estar no meio de uma rede de informação ou social. Dá ao governo a capacidade de tráfego receptivo de informações. Também equipa o governo com uma posição estratégica para distribuir informações. Nodalidade é usada através de mensagens, tais como notificações, anúncios públicos ou de propaganda, bem como por meio de coleta de informações.
- **Autoridade** - denota a posse do poder legal ou oficial, poder de exigir oficialmente, garantir, julgar. É a capacidade distintiva do governo para exigir dos cidadãos, associações e empresas para fazer as coisas, apoiado pela possibilidade de sanções legais. Por definição, aos governos é concedido o poder ou o direito de dar ordens ou tomar decisões sobre os outros. Autoridade é utilizada através de símbolos, tais como certificados, leis e sanções.
- **Tesouro** - indica a posse de um estoque de “dinheiros” ou “bens móveis fungíveis”. Isso significa não só (ou necessariamente) dinheiro no senso comum cotidiano de notas ou moedas, mas qualquer coisa que tem a propriedade do dinheiro, ou seja, de “fungibilidade” - Isto é, a capacidade de serem livremente trocados. Governos na maioria dos casos possuem pelo menos algumas unidades de “tesouro” nesse sentido. O governo pode usar o seu tesouro, como forma de tentar influenciar de fora ou como uma forma de comprar recursos de vários tipos, ou comprar informações. Os governos podem gastar seus recursos financeiros para atingir objetivos políticos. O Tesouro é gasto em salários, recompensas, materiais e outros equipamentos.

- **Organização** - denota a posse de um estoque de pessoas ou recursos que provem habilidades. Podem ser soldados, trabalhadores, burocratas, terrenos, edifícios, materiais, computadores e equipamentos, de alguma forma organizada. Os governos, na maioria dos casos, possuem um mínimo de "organização". Dá ao governo a capacidade física de agir diretamente, usando suas próprias forças. Ou seja, o uso pelo governo de seus recursos diretamente controlados, no cumprimento de tarefas públicas.

Assim, o governo pode simplesmente obter informação em virtude da sua nodalidade (ou por fazer-se nodal), por comprá-la, por exigir oficialmente, ou extraindo-a por meio de algum dispositivo físico. De mesma forma, ele pode tentar influenciar o mundo externo, através do envio de mensagens com base em sua nodalidade, pela autoridade, pelo tesouro e pela organização.

A nodalidade pode permitir ao governo receber informações, da mesma forma como o governo pode dar razão para ser ouvido. As pessoas podem dar informações ao governo simplesmente por causa de sua centralidade social e visibilidade – “porque ele está lá”. Informações dessa natureza são de certa forma livre para o governo e os detectores que as buscam são chamados *receptores nodais*.

O governo pode usar sua autoridade legal para obter informações sobre a demanda. Este tipo de *detector* é chamado de *requisição*, significando a informação que é recolhida na forma de uma demanda oficial (caracteristicamente acompanhada de sanções ameaçadas por incumprimento).

O tesouro pode ser usado para comprar informações. Este tipo de detector é chamado de *recompensa*, denotando informações que o governo recebe, em troca de algo.

Por fim, o governo pode usar sua organização para obter informações, na forma de aparelhos físicos ou mecânicos, que em grande parte ignoraram a

necessidade de uma motivação humana, os quais são chamados detectores de *ergonomia*.

Segundo (HOOD; MARGARATTS, 2007), as agências governamentais utilizam a maioria dessas ferramentas periodicamente - que compreendem as formas básicas em fazer as coisas.

3 MODELAGEM E SIMULAÇÃO

BASEADA EM AGENTES

3.1 Agentes, Sistemas Multiagentes e seu uso em Modelagem e Simulação

Segundo (WOOLDRIDGE, 2002) agentes são sistemas computacionais situados em algum ambiente onde são capazes de realizar atividades de maneira autônoma para atingir seus objetivos, ou seja, eles devem existir dentro de um ambiente, agir de maneira independente e tomar suas próprias decisões. Ainda segundo o autor, para que um agente seja considerado inteligente. Além das características definidas anteriormente, ele deve ser reativo, pró-ativo e possuir habilidade social.

Ele é reativo quando percebe, processa e responde às alterações do ambiente, é pró-ativo se exhibe um comportamento motivado por seus objetivos e finalmente, é social quando interage com outros agentes do ambiente.

No trabalho de (BORDINI; VIEIRA, 2003) são apresentados alguns aspectos importantes para a compreensão do que é um agente. Para os autores, um agente é um sistema computacional que é capaz de:

- **percepção**: o agente é capaz de perceber alterações no ambiente
- **ação**: as alterações no ambiente são provenientes das ações que os agentes realizam constantemente no ambiente; um agente age sempre com o intuito de atingir seus objetivos (motivação), ou seja, com o intuito de transformar o ambiente de seu estado atual em um outro estado desejado pelo agente (veja motivação abaixo);

- **comunicação:** umas das ações possíveis de um agente é comunicar-se com outros agents da sociedade (i.e., que compartilham o mesmo ambiente). Como os agentes precisam coordenar suas ações, a comunicação entre eles é essencial
- **representação:** o agente possui uma representação simbólica explícita daquilo que acredita ser verdade em relação ao ambiente e aos outros agentes que compartilham aquele ambiente;
- **motivação:** como em SMA os agentes são (ou podem ser) autônomos, é essencial que exista não só uma representação do conhecimento do agente, mas também uma representação dos desejos ou objetivos (i.e., aspectos motivacionais) daquele agente; em termos práticos, isto significa ter uma representação de estados do ambiente que o agente almeja alcançar; como consequência, o agente age sobre o ambiente por iniciativa própria para satisfazer esses objetivos;
- **deliberação:** dada uma motivação e uma representação do estado atual do ambiente em que se encontra o agente, esse tem que ser capaz de decidir, dentre os estados de ambiente possíveis de ocorrerem no futuro, quais de fato serão os objetivos a serem seguidos por ele;
- **raciocínio e aprendizagem:** técnicas de inteligência artificial clássica para, por exemplo, raciocínio e aprendizagem podem ser estendidas para múltiplos agentes, aumentando significativamente o desempenho desses. Por exemplo no aspecto de deliberação, note que nem sempre se esperam essas características de raciocínio e aprendizagem de quaisquer agentes.

Por sua vez, na área de Sistemas Multiagentes (SMA) estão envolvidos estudos de tópicos em relação à interação desses agentes, tais: como competição, comunicação, cooperação, coordenação, negociação e organização social (WOOLDRIDGE; JENNINGS, 1995).

A simulação com SMA se baseia na ideia que é possível representar de maneira computadorizada o comportamento de entidades que são ativas no mundo, e através disto representar um fenômeno como o produto de interações de um conjunto de agentes que possuem autonomia operacional (FERBER, 1999).

Por exemplo, através da modelagem baseada em agentes é possível observar como os agentes individuais modelados (pessoas, bactérias, insetos, nações, ou organizações) interagem entre si e com seu ambiente. A simulação no computador é então usada para descobrir propriedades do modelo e, assim, ganhar entendimento dentro de um processo dinâmico, o que seria muito difícil de modelar com técnicas matemáticas padrões.

De acordo com (HÜBNER, 2003) a área de Sistemas Multiagentes é influenciada pela sociologia e, portanto, tem possibilitado uma concepção de sistemas com propriedades que até então somente sociedades possuíam. Neste sentido, SMA também são utilizados para modelar e simular sistemas complexos, como as diversas aplicações em simulação social (GILBERT, 2008; GILBERT; TROITZSCH, 2005).

Em sua obra (FERBER, 1999) afirma que os SMA trouxeram uma nova solução ao conceito de modelagem e simulação de ciências ambientais, oferecendo a possibilidade de representar diretamente indivíduos, seus comportamentos e suas interações.

Para (TISUE; WILENSKY, 2004) as perspectivas de trabalhar com sistemas complexos mediante o uso de simulações multiagente estão ganhando importância para os educadores em todos níveis de ensino, que estão voltando sua atenção para essas novas ferramentas, que têm influenciado mudanças nas salas de aula.

Por fim, no que se refere aos aspectos interdisciplinares da área, o autor (MACAL; NORTH, 2009) cita que os modelos de simulação baseados em Sistemas Multiagentes envolvem conceitos provenientes de diversas áreas do conhecimento, como, por exemplo, Biologia, Antropologia, Biomedicina,

Epidemiologia, Engenharia, Biologia, Economia, Teoria dos Jogos e Ciências Sociais.

3.4 Arquitetura BDI e AgentSpeak(L)

De acordo com (BORDINI; VIEIRA, 2003) as mais importantes arquiteturas de agentes deliberativos são baseadas em um modelo de cognição fundamentado em três principais atitudes mentais que são as crenças, os desejos, e as intenções (abreviadas por BDI, *beliefs, desires e intentions*)(RAO, 1996).

Resumidamente, esta arquitetura de agente está estruturada da seguinte forma. As *crenças* representam aquilo que o agente sabe sobre o estado do ambiente e dos agentes naquele ambiente (inclusive sobre si mesmo). Os *desejos* representam estados do mundo que o agente quer atingir (dito de outra forma, são representações daquilo que ele quer que passe a ser verdadeiro no ambiente). Em tese, desejos podem ser contraditórios, ou seja, pode-se desejar coisas que são mutuamente exclusivas do ponto de vista de ação prática. Normalmente se refere a *objetivos* como um subconjunto dos desejos que são todos compatíveis entre si. As *intenções* representam seqüências de ações específicas que um agente se compromete a executar para atingir determinados objetivos.

Uma das mais adotadas abordagens para implementar o modelo BDI é a forma de sistemas de planejamento reativos (GEORGEFF; LANSKY, 1987). Sistemas de planejamento reativos são sistemas que estão permanentemente em execução, reagindo a eventos que acontecem no ambiente em que estão situados através da execução de planos que se encontram em uma biblioteca de planos parcialmente instanciados.

A linguagem AgentSpeak(L) (RAO, 1996) foi projetada para a programação de agentes BDI na forma de sistema de planejamento reativos.

O comportamento de um agente definido em AgentSpeak(L), é descrito através da especificação de um conjunto de crenças iniciais e um conjunto de planos. As definições a seguir introduzem as noções necessárias para especificação de tais conjuntos. Existem várias similaridades com a sintaxe do Prolog, incluindo a convenção de uso de iniciais maiúsculas para os nomes de variáveis, e aspectos relacionados à forma dos predicados e unificação.

Um *átomo de crença* é simplesmente um predicado de primeira ordem na notação lógica usual, enquanto *átomos de crenças* ou suas negações são *literais de crença*. O conjunto inicial de crenças é apenas uma coleção de literais de crença que não dependem de variáveis.

No AgentSpeak(L) é possível definir dois tipos de objetivos (*goals*): objetivos de realização (*achievement goals*) e objetivos de teste (*test goals*). Objetivos de realização são predicados, assim como as crenças, precedidos com o operador ‘!’ enquanto os objetivos de teste são precedidos pelo operador ‘?’. Objetivos de realização declaram que o agente quer alcançar um estado do mundo onde o predicado associado é verdadeiro. Objetivos de teste declaram que o agente quer testar se o predicado associado é verdadeiro, isto é, se este pode ser unificado com a base de crença do agente.

Os eventos disparadores (*triggering events*) definem quais eventos podem iniciar a execução de planos. No AgentSpeak(L) são definidos dois tipos de eventos: aqueles relacionados à adição, precedidos pelo operador ‘+’, e aqueles relacionados à remoção, precedidos pelo operador ‘-’, de atitudes mentais (crenças ou objetivos).

Para realizar ações no ambiente, os planos devem fazer referências à *ações básicas* que o agente é capaz de realizar no ambiente onde está inserido. Estas ações são referenciadas como predicados simples, que são símbolos predicativos especiais utilizados para este propósito.

Um plano AgentSpeak(L) possui um cabeçalho que é formado por um evento disparador que denota o propósito do plano e uma conjunção de literais de

crença que forma um contexto que deve ser satisfeito para a execução do plano (o contexto deve ser consequência lógica da base de crenças do agente). Um plano também possui um corpo, que é formado por uma sequência de ações básicas, objetivos e sub-objetivos de realização ou teste.

A especificação de um agente ag em AgentSpeak(L) deve ser feita de acordo com a gramática apresentada na Figura 1. Como já citado, em AgentSpeak(L), um agente é especificado por um conjunto de crenças bs (*beliefs*) correspondendo à base de crenças inicial do agente, e um conjunto de planos ps que forma a biblioteca de planos do agente.

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

Figura 1: Sintaxe AgentSpeak(L) (HÜBNER; BORDINI; VIEIRA, 2004)

As fórmulas atômicas at da linguagem são predicados onde P é um símbolo predicativo e t_1, \dots, t_n são termos padrão da lógica de primeira ordem. Chama-se de *crença* uma fórmula atômica at sem variáveis e b é meta-variável para crenças. O conjunto inicial de crenças de um programa AgentSpeak(L) é uma sequência de crenças bs .

Um plano em AgentSpeak(L) é dado por p acima, onde te (*triggering event*) é o evento ativador, ct é o contexto do plano (uma conjunção de literais de crença) e h é uma sequência de ações, objetivos ou atualizações de crenças. A construção $te : ct$ é dita a *cabeça* do plano, e h o *corpo* do plano. O conjunto de planos de um agente é dado por ps como uma lista de planos.

Um evento ativador *te* corresponde à adição/remoção de crenças da base de crenças do agente (+*at* e -*at*, respectivamente), ou à adição/remoção de objetivos (+*g* e -*g*, respectivamente).

O agente possui um conjunto de *ações básicas* que utiliza para atuar sobre o ambiente. Ações são referidas por predicados usuais com a exceção de que um símbolo de ação *A* é usado no lugar do símbolo predicativo. Objetivos *g* podem ser objetivos de realização (!*at*) ou de teste (?*at*). Finalmente, +*at* e -*at* (no corpo de um plano) representam operações de atualização (*update*) da base de crença *u*, através da adição ou remoção de crenças respectivamente.

Note que uma fórmula !*g* no corpo de um plano gera um evento cujo evento ativador é +!*g*. Portanto, planos escritos pelo programador que tenha um evento ativador que possa ser unificado com +!*g* representam alternativas de planos que devem ser considerados no tratamento de tal evento.

Planos com evento ativador do tipo +*at* e -*at* são utilizados no tratamento de eventos que são gerados quando crenças são adicionadas ou removidas (tanto como consequência da percepção do ambiente, como devido a alterações de crenças explicitamente requisitadas no corpo de um plano). Eventos ativadores do tipo -!*g* são usados para o tratamento de falhas de planos, e eventos ativadores do tipo +?*g* e -?*g* não são utilizados na implementação atual da linguagem.

No ANEXO 1, apresenta-se um exemplo hipotético que é útil para a compreensão da utilização da linguagem AgentSpeak(L) utilizada neste trabalho.

3.5 Jason

Jason (BORDINI; HUBNER; WOOLDRIDGE, 2007) permite a implementação de agentes cognitivos baseados na arquitetura BDI (*Belief, Desires and Intentions*), sendo a programação realizada na linguagem AgentSpeak(L) (RAO, 1996).

Conforme (BORDINI; HUBNER; WOOLDRIDGE, 2007), além de interpretar a linguagem AgentSpeak(L) original, Jason possui os seguintes recursos:

- Inclui comunicação entre agentes baseada na teoria de atos de fala (AUSTIN, 1962);
- Possibilita que um sistema multiagente possa ser executado de maneira distribuída em uma rede por meio da plataforma JADE (KRZYSZTOF; MACIEJ, 2005);
- É multi-plataforma (característica herdada de sua implementação em Java);
- Está disponível sob licença GNU LGPL;
- Negação forte (*strong negation*), portanto tanto sistemas que consideram mundo fechado (*closed-world*) quanto mundo-aberto (*open-world*) são possíveis;
- Possui um mecanismo de tratamento de falha de planos, que é útil devido à natureza dinâmica típica de ambientes multiagentes;
- Viabiliza o uso de anotações em crenças, uma anotação padrão disponível é a origem de cada crença;

- Possibilita a utilização de rótulos em planos, os quais podem ser utilizados pelo programador e interpretador para definir prioridades no caso de vários planos forem aplicáveis para um dado contexto.

3.6 CArtAgO

CArtAgO (*Common ARTifact infrastructure for AGents Open environments*) (RICCI; SANTI; PIUNTI; et al, 2010) é um *framework* que possibilita a implementação de ambientes virtuais para sistemas de agentes.

Este *framework* permite a implementação do ambiente como uma camada computacional que encapsula as funcionalidades e os serviços de *artefatos* (objetos não-autônomos) que os agentes podem explorar em tempo de execução.

Conceitualmente, o **CArtAgO** é baseado no meta-modelo *Agents & Artefacts* (A&A)(OMICINI; RICCI; VIROLI, 2008) para modelar sistemas multiagentes.

Este modelo introduz uma metáfora de alto nível retirada da ideia de que humanos trabalham de forma cooperativa com o seu ambiente: *agentes* são como entidades computacionais que realizam algum tipo de tarefa orientada para alcançar um objetivo (em analogia aos trabalhadores humanos), e *artefatos* são como recursos e ferramentas dinamicamente construídas, manipuladas e compartilhadas pelos agentes para dar suporte e realizarem suas atividades individuais e coletivas (como artefatos no contexto humano).

Assim, programadores de agentes podem desenvolver artefatos os quais são instanciados no ambiente e podem prover serviços para os agentes, podendo inclusive realizar uma comunicação com serviços externos do tipo *web-services*.

Os principais elementos utilizados no CArtAgO são as *propriedades observáveis*, as *operações* e os *sinais*.

As *operações* são implementadas de forma a prover uma interface de utilização de um recurso (por exemplo, sendo um motor um recurso do ambiente, poder-se-ia ter operações ligar, acelerar e desligar). Do ponto de vista dos agentes, as operações são *ações* que podem ser executadas sobre o ambiente.

As *propriedades observáveis* servem para representar *atributos e o estado* de recursos do ambiente (por exemplo, cor, altura, largura, ligado ou desligado). Do ponto de vista dos agentes, as *propriedades observáveis* são mapeadas como *crenças*.

Por sua vez, os *sinais* possibilitam programar que os recursos avisem aos agentes sobre uma situação, os quais podem utilizar tais sinais para tomada de decisão (por exemplo, enviar um sinal caso a temperatura ultrapasse um limite).

Como princípio básico de funcionamento, um agente executa a operação nativa denominada *focus*, de modo que todas as alterações que ocorrem nas propriedades observáveis são informadas ao mesmo (por meio de um mapeamento em sua base de crenças). Neste sentido, por meio do uso das *operações*, outros agentes poderiam alterar os valores destas propriedades de maneira que estas modificações são percebidas por aquele.

Os artefatos devem ser criados pelos agentes em tempo de execução. Três ações básicas existem para este propósito: *makeArtifact*, *disposeArtifact* e *lookupArtifac*. A primeira ação instancia um novo artefato, a segunda permite remover um artefato e a última recupera um identificador único de uma instância de artefato, dado o nome lógico deste.

4 UM MODELO BASEADO EM AGENTES PARA A SIMULAÇÃO DE POLÍTICAS PÚBLICAS

4.1 Uma abordagem sobre o modelo sistêmico e nos instrumentos do governo

Para elaborar o modelo baseado em agentes para a simulação de políticas públicas proposto neste trabalho, tomou-se como base aspectos fundamentais do modelo sistêmico e da abordagem dos instrumentos do governo.

Do modelo sistêmico, os seguintes aspectos:

- Há um sistema político que interage (alocando *valores*) com outros sistemas, os quais são denominados de meio ambiente do sistema político;
- O sistema político recebe *insumos* de seu meio-ambiente;
- Por intermédio de estruturas e processos internos – numa operação denominada de conversão -, o sistema atua sobre essas entradas, transformando-as em *produtos*;
- Os *produtos* retornam ao meio-ambiente;
- Isto transcorre num ciclo contínuo de “*feedback*” - dando-lhe condições de reajustar seu comportamento.

Da abordagem dos instrumentos do governo, os seguintes aspectos:

- Há Detectores que o governo utiliza para absorver informação.
- Há Efetadores que o governo pode usar para tentar fazer impacto no mundo exterior.

4.2 Os componentes e a organização do modelo

Dada a fundamentação adotada para o modelo (seção 4.1), elaborou-se um modelo de simulação baseado em agentes para a simulação de políticas públicas, que tem componentes e organização conforme a Figura 2.

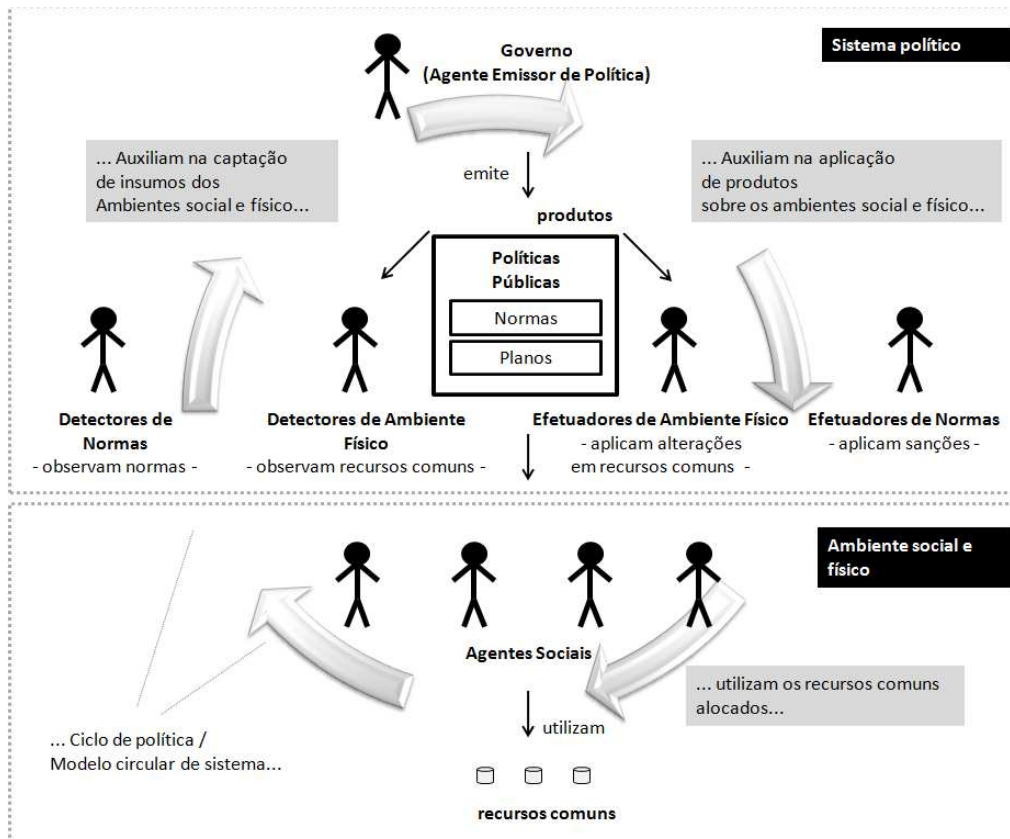


Figura 2: Arquitetura de um modelo baseado em agentes para o processo de simulação de políticas públicas.

4.2.1 Normas e Planos

Neste modelo, os *produtos* do sistema político (*políticas públicas*) considerados são *normas* (*essencialmente proibições e obrigações*) e *planos de ação*.

4.2.2 Agentes Sociais e Recursos Comuns

Por sua vez, os *valores alocados* são vistos unicamente como os *recursos comuns* do ambiente físico.

Já o *meio ambiente social* é composto por *agentes sociais*, que para atingir seus objetivos pessoais usufruem dos *recursos comuns* alocados pelas *políticas públicas*, de modo que podem ser conflituosos com tais alocações.

4.2.3 Agente Governo e Agentes Detectores e Atuadores

No que se refere aos agentes envolvidos no *sistema político*, tem-se:

agente governo: o qual é capaz de realizar ações como criar, modificar e remover as *políticas públicas* (concentrado em um único agente);

agentes governamentais detectores: que desempenham a função auxiliar de extração de informações de *insumo* (*demanda e apoio*) dos *ambientes social e físico*. Sendo de dois tipos fundamentais:

- *detectores de normas*, os quais observando as *normas* e o comportamento dos *agentes sociais*, captam informações referentes ao cumprimento/infração;
- *detectores de ambiente físico*, os quais pela execução de *planos de ação*, captam informações sobre estado corrente dos *recursos comuns* do ambiente físico.

agentes governamentais efetadores: que desempenham a função auxiliar de aplicação de *produtos* nos *ambientes social e físico*. Também, sendo de dois tipos fundamentais:

- *efetadores de normas*, que realizam a efetiva aplicação de *sanções/penalidades* prescritas nas *normas* aos *agentes sociais* que as descumpriram;
- *efetadores de ambiente físico*, que, por meio da execução de *planos de ação* alteram *propriedades* estruturais dos *recursos comuns* do *ambiente físico*.

5 UM FRAMEWORK BASEADO EM AGENTES PARA A SIMULAÇÃO DE POLÍTICAS PÚBLICAS

5.1 Uma implementação sobre as ferramentas Jason e CarTAgo

Para construção do framework proposto neste trabalho, optou-se pelo uso das abstrações e funcionalidades inerentes à infraestrutura das ferramentas Jason e CarTAgo. O Jason pelo seu suporte expressivo na programação de agentes cognitivos. Já o CarTAgo pela sua metáfora de abstração em alto nível de ambientes virtuais de agentes.

O framework busca diminuir o trabalho de implementação de simulações baseadas no modelo proposto no capítulo 4, fornecendo uma camada de abstrações de código intermediárias para aquele fim.

5.2 Os Componentes, o funcionamento e a organização do framework: uma compreensão em três cenários

Nesta seção, são utilizados cenários de forma a detalhar os elementos do *framework* que provê a implementação por programação sobre o uso das plataformas Jason e CarTAgo do modelo baseado em agentes para simulação de políticas públicas apresentado no capítulo 4.

O princípio essencial utilizado é a metáfora *Agents & Artefacts* – mencionada anteriormente no capítulo 3, seção 3.6 – o que direcionou ao conceito de *artefatos políticos*, que são *políticas públicas (normas e planos)* mapeadas em

artefatos CArtAgO, os quais são endereçados aos *agentes governamentais* e *agentes sociais*.

Estes *agentes*, por sua vez, possuem um conjunto pré-definido de comportamentos, concretizados em *programas AgentSpeak(L)*, e que permitem aos mesmos *interpretação* e *raciocínio* perante a *política pública*.

Por sua vez, os *recursos comuns* do *ambiente físico* que são alocados por aquelas *políticas públicas*, também são mapeados na forma de artefatos CArtAgO, de modo que suas características estruturais e funcionais concretizem-se em *propriedades observáveis* e *operações* disponíveis aos *agentes sociais* e *governamentais*.

Como já citado anteriormente, do ponto de vista dos agentes, operações são *ações* que podem ser executadas sobre o ambiente, já as *propriedades observáveis* representam *atributos* e o *estado do ambiente*, sendo mapeadas como *crenças*.

Assim, para definir uma simulação multiagente com base e suporte deste modelo e seu *framework*, se devem instanciar *políticas públicas* e *recursos comuns* por meio daqueles artefatos no ambiente a ser simulado, bem como definir agentes que devem ser programados fundamentalmente sobre a estrutura básica daquele conjunto de programas AgentSpeak(L).

A seguir, são exibidos cenários pelos quais são detalhadas as composições estruturais e funcionais de cada um dos artefatos e do conjunto de planos supracitados. Para uma melhor apresentação, são utilizados diagramas e fluxogramas para representar os diversos artefatos referentes às *normas*, *planos* bem como respectivos trechos de programação inerentes ao comportamento dos *agentes*.

5.2.1 Cenário 1: Normas de Obrigação

As *normas de obrigação* assumem o seguinte conteúdo:

- *id*: o identificador da norma;
- *destinatário*: especifica um papel social que os agentes sociais devem estar assumindo para que a norma se aplique;
- *ação*: especifica a ação a ser realizada pelo destinatário da norma;
- *condição*: especifica uma condição contextual necessária para aplicação da norma;
- *exceção*: especifica uma condição na qual a norma não é aplicada;
- *periodicidade*: especifica, para obrigações periódicas, um evento que deve ocorrer para verificar a condição;
- *sanção*: especifica a sanção a ser aplicada em caso da violação da norma.

A Figura 3 contém a representação estrutural e de funcionamento deste *artefato*, o qual denominou-se **NrmsObrg**.

As possíveis *operações* do mesmo são: *cria*, *modifica* e *remove*, as quais respectivamente servem para criar, modificar e remover o conteúdo da norma.

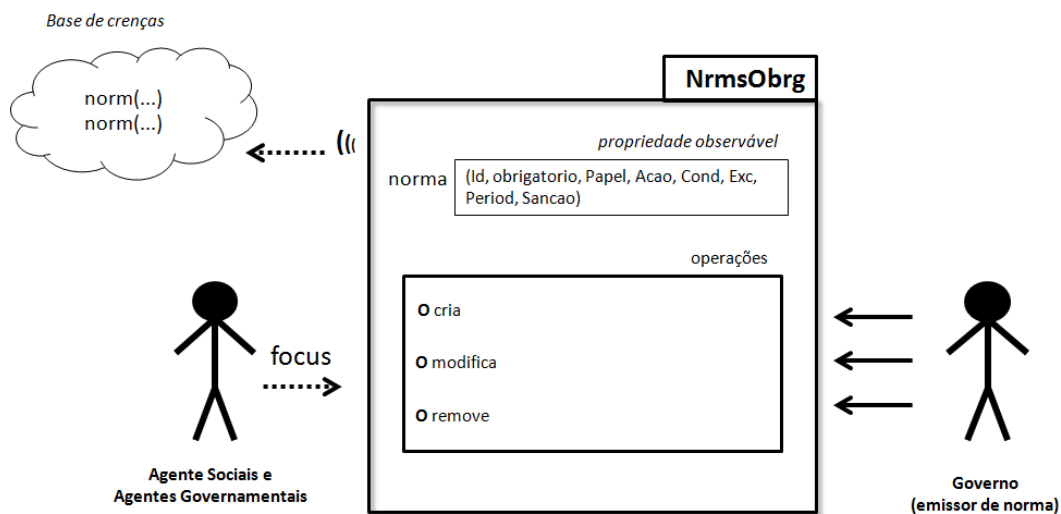


Figura 3: O uso do Artefato NrmsObrg.

Assim, tem-se que:

- Os *agentes sociais* e os *agentes governamentais* executam a operação *focus* sobre o artefato;
- O *agente governo* executa as operações *cria*, *modifica* e *remove* sobre o artefato;
- Os *agentes sociais* e os *agentes governamentais* são notificados, em sua base de crenças, das mudanças nas *propriedades observáveis* do artefato causadas pelas operações de criação, modificação e remoção;

O fluxograma da Figura 4 demonstra o comportamento dos *agentes sociais* perante os artefatos *NrmsObrg*.

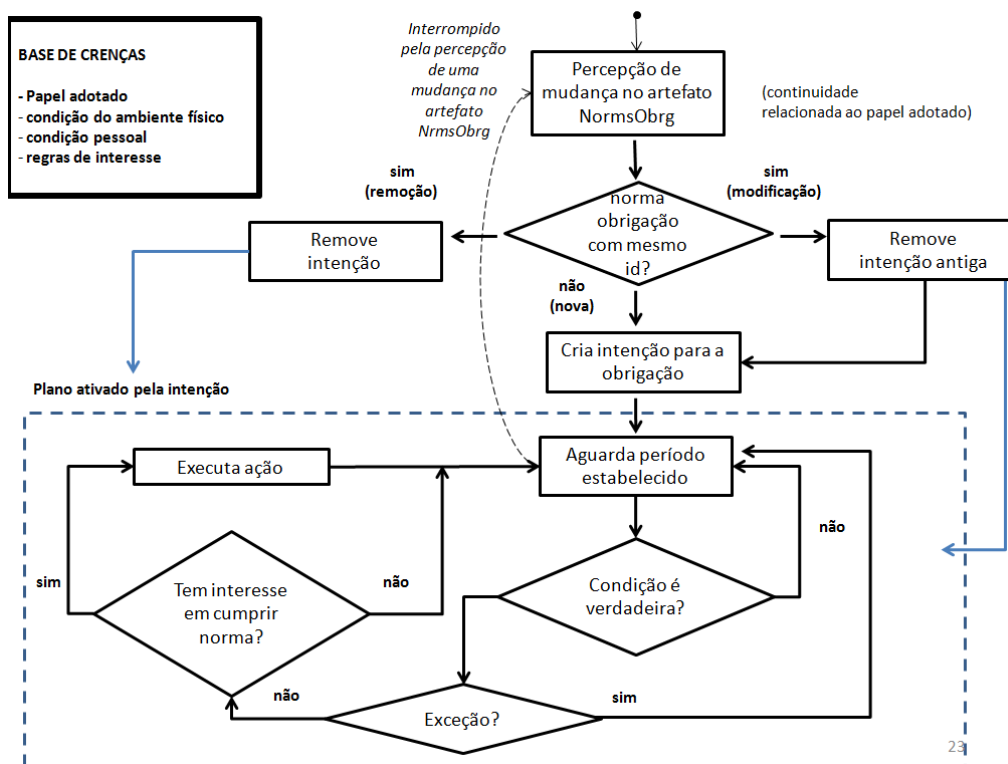


Figura 4: Fluxograma do comportamento dos *agentes sociais* para a manipulação de artefatos *NrmsObrg*.

Os *agentes detectores de norma* podem basear-se na *periodicidade* do artefato *NrmsObrg* para verificar o cumprimento pelos *destinatários* da *norma*. No caso da detecção do não cumprimento, o *detector de norma* pode informar o caso ao agente *governo* (que deverá ser responsável pela decisão sobre o caso) ou pode ele mesmo aplicar a sanção ao destinatário que descumpriu (neste caso o *detector de norma* é também um *efetuidor de norma*).

5.2.2 Cenário 2: Normas de Proibição

Normas de proibição e suas sanções associadas possibilitam que o *agente governo* proíba que certas ações sejam realizadas pelos *agentes sociais*. O artefato denominado **NrmsPrb** concretiza as *normas de proibição* e tem o seguinte conteúdo:

- *id*: o identificador da norma;
- *destinatário*: especifica um *papel social* que os agentes sociais devem estar assumindo para que a *norma* se aplique;
- *ação*: especifica a *ação* proibida pela norma;
- *condição*: especifica uma *condição contextual* necessária para aplicação da norma;
- *exceção*: especifica uma *condição contextual* sobre a qual a norma não se aplica.
- *sanção*: especifica a *sanção* a ser aplicada em caso da violação da norma;

A forma de uso do artefato NrmsPrb é similar ao exibido na Figura 3. O fluxograma que representa o comportamento dos agentes perante a NrmsPrb é exibido na Figura 5.

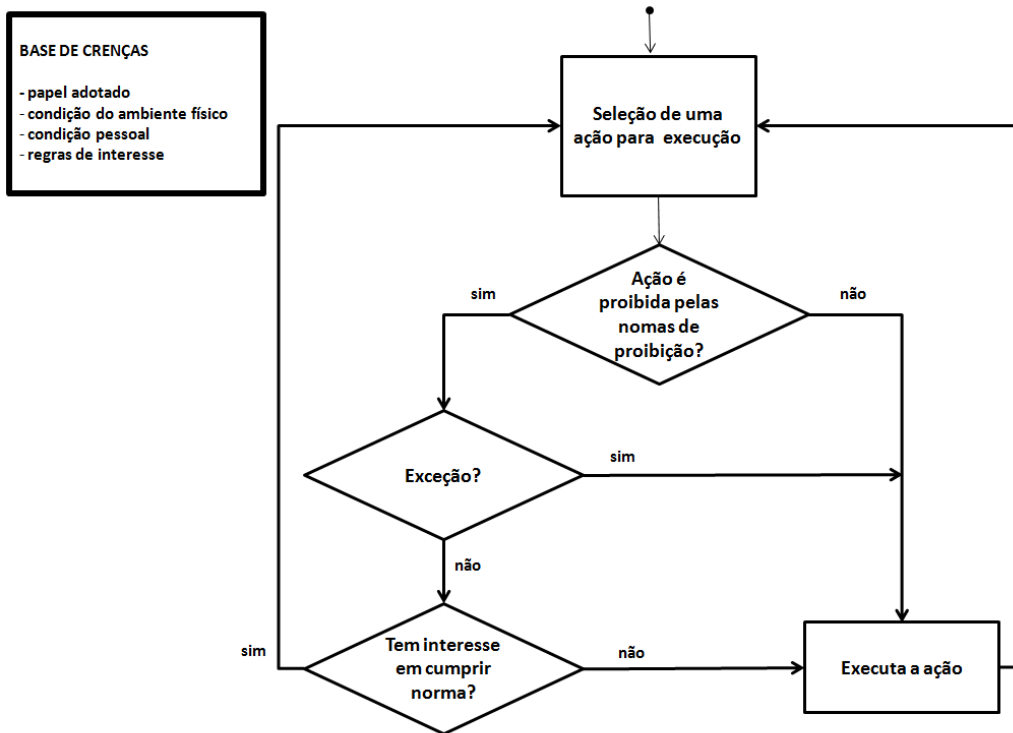


Figura 5: Fluxograma do comportamento dos *agentes sociais* para a manipulação de artefatos NrmsPrb.

Como citado anteriormente, os *recursos do ambiente físico* devem ser implementados por artefatos CArtAgO. Também devem gerar *signals* após cada operação ser executada, sendo este *signal* no formato $+acao(ACAO, ATOR, PAR)$. Onde *ACAO* é ação executada no *recurso*, *ATOR* é o *agente* que executou a ação, e *PAR* corresponde a qualquer informação complementar para caracterizar completamente a ação. Isto possibilita que *agentes* que realizem a operação *focus* sobre o mesmo, possam identificar ações, conforme Figura 6.

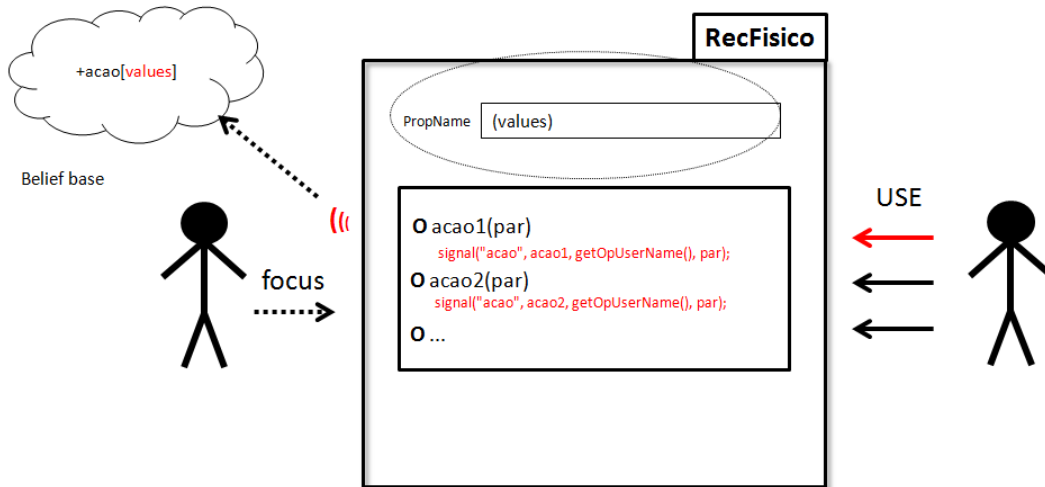


Figura 6: O uso de um Artefato RecFisico.

Assim, um *agente detector de norma* deve realizar a operação *focus* sobre o artefato *NrmsPrb* (de modo a perceber sobre o estado da norma), sobre o *recurso do ambiente físico* (para receber os *signals* que o mesmo emite quando uma *operação* é executada) e sobre a *infraestrutura organizacional* (para identificar o *papel* do *ator* que executou a ação). Desta forma, o *agente detector de norma* pode monitorar as ações executadas sobre o recurso físico e pode verificar por violações à norma de proibição, conforme Figura 7.

Os *agentes efetadores de norma* podem aplicar as sanções aos agentes violadores das normas, após serem informados da ocorrência das violações.

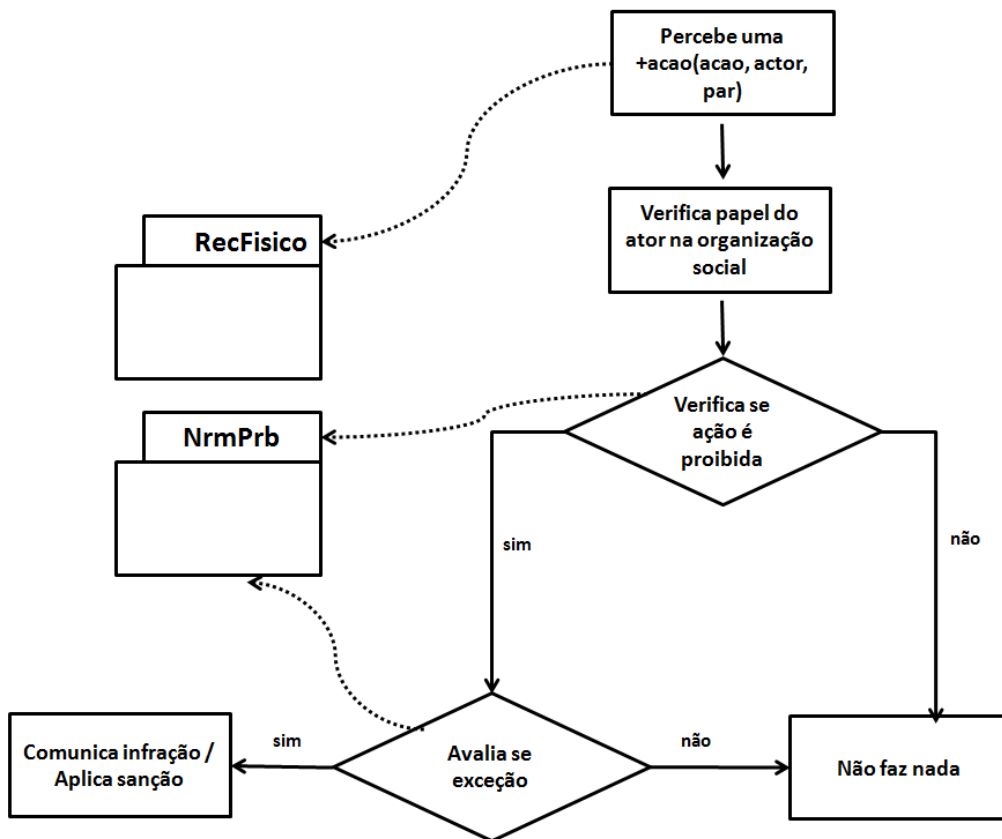


Figura 7: Esquema de comportamento para detecção de ações proibidas.

5.2.3 Cenário 3: Planos de ação

Como visto na seção 4.1, os *agentes governamentais detectores e efetadores de ambiente físico* executam *planos de ação* direcionados pelo *agente governo*. O artefato denominado **Plans** concretiza os *planos de ação* e tem o seguinte conteúdo:

- *id*: o identificador do plano;
- *condição*: especifica uma condição contextual necessária para aplicação do plano;

- **corpo**: especifica o conjunto de ações que deve ser executado;
- **periodicidade**: especifica um evento em que o plano deve ser executado (opcional).

A forma de uso do artefato Plans é apresentada na Figura 8.

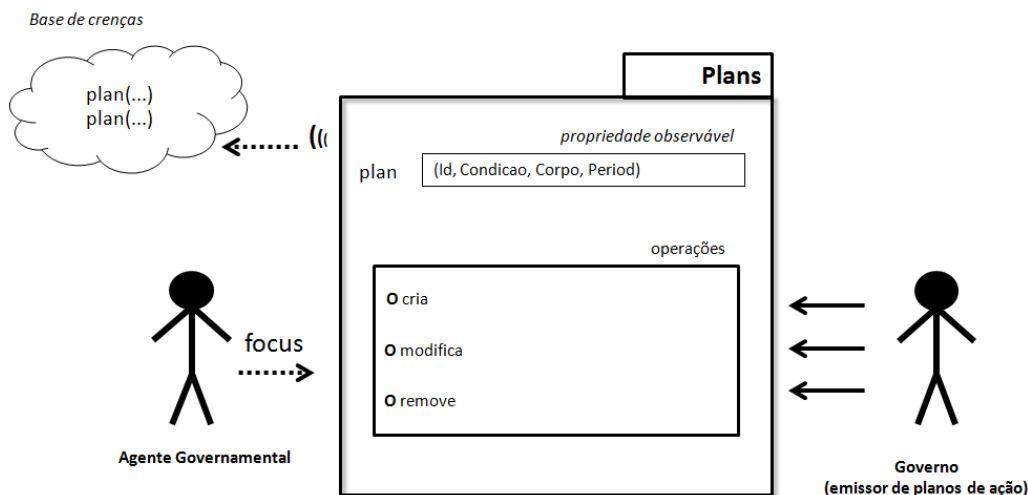


Figura 8: Forma de uso do artefato Plans.

Assim, o uso do artefato Plans corresponde:

- Os *agentes governamentais detectores e atuadores de ambiente físico* executam a operação *focus* sobre o artefato;
- O *agente governo* executa as operações *cria*, *modifica* e *remove* sobre o artefato;
- Os *agentes governamentais detectores e atuadores de ambiente físico* são notificados, em sua base de crenças, das mudanças nas *propriedades observáveis* do artefato causadas pelas operações de criação, modificação e remoção;

O fluxograma que representa o comportamento dos *agentes detectores e atuadores de ambiente físico* perante o artefato **Plans** é exibido na Figura 9.

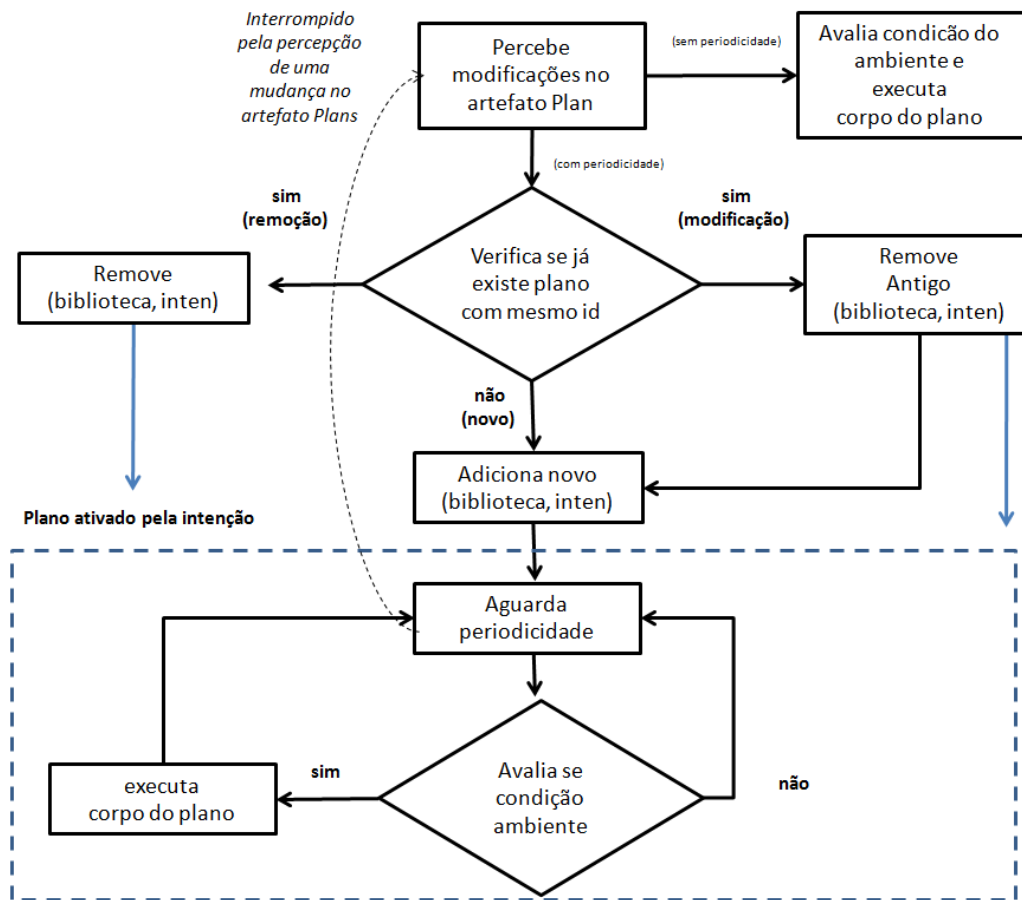


Figura 9: Forma de uso do artefato Plans.

5.3 Considerações sobre o framework

No que se refere a situações relacionadas em aspectos operacionais de implementação e de programação:

- Permite variação dinâmica em *tempo de execução* da *política pública*, por meio das operações (*cria, remove, modifica*) no decorrer da simulação.

- A infraestrutura organizacional é modelada por um *artefato* (uma espécie de cadastro da sociedade de agentes da questão modelada), deste modo o mapeamento agente-papel resulta em propriedades observáveis.
- A *periodicidade* é gerada por um “relógio global”, sob forma de um artefato CArtAgO, onde pode gerar *signals* indicando a passagem do tempo e períodos específicos. Por exemplo: hora, dia, semana, estação, período de chuva, etc... Assim, isso é externo, e sua correspondência temporal real, com as ações e acontecimentos possíveis, deve ser devidamente calibrada e especificada pelo programador.
- A *ação* corresponde a uma *operação* no artefato que modela o *recurso*.
- A *condição* e a *exceção* são *formulas lógicas* com as *propriedades dos artefatos*, com possibilidade do uso do recurso de *unificação*.
- As normas são limitadas ao conteúdo supracitado, entretanto por meio do uso de *sobrecarga do operador Java*, é possível prover as operações *cria*, *modifica* e *remove* para os mais diferentes tipos de conteúdo, ampliando o número de parâmetros na definição de normas (por exemplo, adicionar recompensa às normas). Por sua vez, o conjunto de planos *AgentSpeak* de manipulação e raciocínio dos *agentes sociais* e *governamentais* também pode ser estendido para lidar com tais ampliações.
- A utilização da operação *focus* implica que os *agentes sociais* e *governamentais* têm *informação totalmente observável* sobre o *ambiente físico* e das alterações na *política pública*. Contudo por meio da operação *stopfocus* pelos respectivos agentes, poder-se-ia simular o momento em que “param” de observar as políticas públicas e o ambiente físico, numa situação de *informação parcialmente observável* sobre o estado de ambos.
- Uma decisão de *política pública* pode gerar impacto indireto em outro recurso do ambiente físico (por exemplo: uma política que obriga a

queima de lixo para limpeza, implica na poluição do ar) isto poderia ser simulado por meio do recurso *linkArtifacts*, que conecta dois artefatos no *CARTAgO*.

5.5 Códigos Java e AgentSpeak(L) do Framework: exemplo e comentários

5.5.1 Artefato NormsObrg

Artefato que representa as *normas de obrigação*, com *operações* que são camadas intermediárias de código para *criar*, *remover* e *modificar*. Deste modo, deve ser incluído junto às bibliotecas do projeto Jason-CartAgO, para modelagem e simulação de políticas públicas com o framework proposto neste trabalho.

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import cartago.*;

public class NrmsObrg extends Artifact {

    List<ObsProperty> list_aux_norms_obrg = new ArrayList<ObsProperty>();

    @OPERATION
    public void cria(Object Id, Object Tipo, Object Papel, Object Acao,
                    String Condicao, String Exc, Object Freq, Object Sancao, Object
ParSancao) {
        defineObsProperty("norma", Id, "obrigatorio", Papel, Acao, Condicao,
Exc, Freq, Sancao, ParSancao);
        ObsProperty aux_obrg = getObsPropertyByTemplate("norma", Id,
"obrigatorio", Papel, Acao, Condicao, Exc,
Freq,Sancao,ParSancao);
        list_aux_norms_obrg.add(aux_obrg);
    }

    @OPERATION
    public void remove(Object Id, Object Tipo, Object Papel, Object Acao,
Object Condicao, Object Exc, Object Freq, Object Sancao,
Object ParSancao) {
        ObsProperty aux_obrg = getObsPropertyByTemplate("norma", Id,
"obrigatorio", Papel, Acao, Condicao, Exc,
Freq,Sancao,ParSancao);
```

```

        list_aux_norms_obrg.remove(aux_obrg);
        removeObsPropertyByTemplate("norma", Id, "obrigatorio", Papel, Acao,
            Condicao, Exc, Freq,Sancao,ParSancao);
    }

    @OPERATION
    public void modifica(int Id, Object Tipo, Object Papel, Object Acao,
        String Condicao, String Exc, Object Freq, Object Sancao, Object
ParSancao) {

        int indice = -1;
        Iterator<ObsProperty> i = list_aux_norms_obrg.iterator();
        while (i.hasNext()) {
            ObsProperty aux_obrg = i.next();
            if (aux_obrg.intValue(0) == Id) {
                indice++;
                aux_obrg.updateValues(Id, "obrigatorio", Papel, Acao,
Condicao,Exc, Freq,Sancao,ParSancao);
                list_aux_norms_obrg.add(indice, aux_obrg);
                System.out.println(aux_obrg.toString());
                break;
            }
        }
    }
}

```

5.5.2 Artefato NormsPrb

Artefato que representa as *normas de proibição*, com *operações* que são camadas intermediárias de código para *criar*, *remover* e *modificar*. Deste modo, deve ser incluído junto às bibliotecas do projeto Jason-CartAgO, para modelagem e simulação de políticas públicas com o framework proposto neste trabalho.

```

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import cartago.*;

public class NrmsPrb extends Artifact {

    List<ObsProperty> list_aux_norms_prb = new ArrayList<ObsProperty>();

    @OPERATION
    public void cria(int Id, Object Tipo, Object Papel, Object Acao,
        Object Condicao, Object Exc, Object Sancao, Object ParSancao)
    {

```

```

        defineObsProperty("norma", Id, "proibido", Papel, Acao, Condicao, Exc,
            Sancao, ParSancao);
        ObsProperty aux_prb = getObsPropertyByTemplate("norma", Id, "proibido",
            Papel, Acao, Condicao, Exc, Sancao, ParSancao);
        list_aux_norms_prb.add(aux_prb);
    }

    @OPERATION
    public void remove(int Id, Object Tipo, Object Papel, Object Acao,
        Object Condicao, Object Exc, Object Sancao, Object ParSancao)
    {

        ObsProperty aux_prb = getObsPropertyByTemplate("norma", Id, "proibido",
            Papel, Acao, Condicao, Exc, Sancao, ParSancao);
        list_aux_norms_prb.remove(aux_prb);
        removeObsPropertyByTemplate("norma", Id, "proibido", Papel, Acao,
            Condicao, Exc, Sancao, ParSancao);
    }

    @OPERATION
    public void modifica(int Id, Object Tipo, Object Papel, Object Acao,
        Object Condicao, Object Exc, Object Sancao, Object ParSancao)
    {

        int indice = -1;
        Iterator<ObsProperty> i = list_aux_norms_prb.iterator();
        while (i.hasNext()) {
            ObsProperty aux_prb = i.next();
            if (aux_prb.intValue(0) == Id) {
                indice++;
                aux_prb.updateValues(Id, "proibido", Papel, Acao, Condicao,
                    Exc, Sancao, ParSancao);
                list_aux_norms_prb.add(indice, aux_prb);
                System.out.println(aux_prb.toString());
                break;
            }
        }
    }
}

```

5.5.3 Artefato Plans

Artefato que representa os *planos de ação*, com *operações* que são camadas intermediárias de código para *criar*, *remover* e *modificar*. Deste modo, deve ser incluído junto às bibliotecas do projeto Jason-CartAgO, para modelagem e simulação de políticas públicas com o framework proposto neste trabalho.

```

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import cartago.*;

public class Plans extends Artifact {

    List<ObsProperty> list_aux_plans = new ArrayList<ObsProperty>();

    // operação de criação de plano sem repetição
    @OPERATION
    public void cria(int Id, String PlanEvent, String PlanContext,
                    String PlanBody) {
        defineObsProperty("plano", Id, PlanEvent, PlanContext, PlanBody);
    }

    @OPERATION
    public void cria(Object Id, String PlanContext, String PlanBody, Object Freq) {
        defineObsProperty("plano", Id, PlanContext, PlanBody, Freq);
        ObsProperty aux_plan = getObsPropertyByTemplate("plano", Id,
                                                       PlanContext, PlanBody, Freq);
        list_aux_plans.add(aux_plan);
    }

    @OPERATION
    public void remove(Object Id, String PlanContext, String PlanBody,
                      Object Freq) {
        ObsProperty aux_plan = getObsPropertyByTemplate("plano", Id,
                                                       PlanContext, PlanBody, Freq);
        list_aux_plans.remove(aux_plan);
        removeObsPropertyByTemplate("plano", Id, PlanContext, PlanBody, Freq);
    }

    @OPERATION
    void modifica(int Id, Object PlanContext, Object PlanBody, Object Freq) {
        int indice = -1;
        Iterator<ObsProperty> i = list_aux_plans.iterator();
        while (i.hasNext()) {
            ObsProperty aux_plan = i.next();
            if (aux_plan.intValue(0) == Id) {
                indice++;
                aux_plan.updateValues(Id, "plano", PlanContext, PlanBody,
Freq);

                list_aux_plans.add(indice, aux_plan);
                System.out.println(aux_plan.toString());
                break;
            }
        }
    }
}

```

5.5.4 O artefato Recurso Comum

O artefato *recurso comum* deve ter sua composição estrutural modelada como *propriedades observáveis*. Já as ações sobre o mesmo, são modeladas como *operações*. A seguir um exemplo muito simplificado de um artefato *recurso comum* que representa uma *floresta*.

```
import cartago.Artifact;
import cartago.OPERATION;
import cartago.ObsProperty;

public class Floresta extends Artifact {

    // situação inicial da Floresta

    int arvores_adultas = 500; // número de árvores adultas
    int arvores_desenvolv = 300; // número de árvores em desenvolvimento

    void init()
    {
        defineObsProperty("qtd_arvores_adultas", arvores_adultas);
        defineObsProperty("qtd_arvores_desenvolv", arvores_desenvolv);
    }

    @OPERATION public void reflorestar()
    {
        arvores_desenvolv +=50;

        ObsProperty qad = getObsProperty("qtd_arvores_desenvolv ");
        qad.updateValues(arvores_desenvolv);

        signal("acao", "reflorestar", getOpUserName(), 50);
    }
    @OPERATION public void cortar()
    {
        arvores_adultas -=1;

        ObsProperty qac = getObsProperty("qtd_arvores_adultas");
        qac.updateValues(arvores_adultas);
        signal("acao", "cortar", getOpUserName(), 1);
    }
}
```

5.5.5 Agente Governo

A utilização do framework proposto implica no uso do *agente governo* para programar as políticas públicas e direcioná-las sobre os *recursos alocados* da *sociedade* em questão.

Neste sentido, por meio deste, realiza-se a utilização dos artefatos *NrmsObrg*, *NrmsPrb* e *Plans*, assim como dos *agentes governamentais* apresentados anteriormente.

Assim, um exemplo simplificado de elaboração de *políticas públicas para desmatamento e sustentabilidade* para o caso *hipotético da floresta*, pode ocorrer de acordo com o seguinte formato.

```
1      !gerir_floresta.
2      +!gerir_floresta : true
3 <-
4      makeArtifact("nrmsobrg", "NrmsObrg", [], NrmsObrg);
5      makeArtifact("nrmsprb", "NrmsPrb", [], NrmsPrb);
6      makeArtifact("plansdetector", "Plans", [], PlansDet);
7      makeArtifact("plansefector", "Plans", [], PlansEfe);

8 .send(detect_prb_norm, tell, newpol("nrmsprb", "floresta", "sociedade"));
9 .send(efector_prb_norm, tell, newpol("nrmsprb", "sociedade"));
10     .send(detector_env, tell, newpol("plansdetector", " floresta "));
11     .send(efector_env, tell, newpol("plansefector", " floresta "));
12     .send(ag_ob_pb, tell, newpol("nrmsobrg", "nrmsprb"));

13     cria (1, obrigatorio, lenhador, reflorestar,
14     qtd_arvores_desenvolv(QAD) & QAD < 100,
15     qtd_arvores_adultas(QAC) & QAC > 500, +mes,
16     multa, 100)[artifact_name("nrmsobrg)];

17     cria (2, proibido, lenhador, cortar, qtd_arvores_adultas(QAC) & QAC < 200,
18     no_except, multa, 20)[artifact_name("nrmsprb)];

19     modifica (2, proibido, lenhador, cortar, qtd_arvores_adultas(QAC) & QAC < 400,
20     no_except, multa, 40)[artifact_name("nrmsprb)];

21     cria(1, qtd_arvores_desenvolv(QAD) & QAD < 20,
22     .send(governo, tell, qtd_arvores_desenvolv(QAD),
23     +mes)[artifact_id("plansdetector")].

23     + qtd_arvores_desenvolv(QAD): QAD < 20
24     <-      cria(2, true, reflorestar.)[artifact_name("plansefector")].
```


Onde:

- **Linhas 1 até 7:** O objetivo inicial é *!gerir_floresta* e o plano para este objetivo corresponde em *+!gerir_floresta* que consiste em criar os *artefatos de normas de obrigação, normas de proibição* e os *artefatos de planos*.
- **Linhas 8 até 12:** Posteriormente pode comunicar os agentes utilizando a operação interna *.send* do Jason.
- **Linhas 13 até 16:** Norma de obrigação criada por meio da *operação cria no artefato NrmsObrg*. Estabelece que é obrigatório o lenhador reflorestar caso o número de árvores em desenvolvimento seja menor que 100 unidades, sendo a exceção se o número de árvores adultas for maior que 500 unidades, e a periodicidade da ação é uma vez por semana e a multa é de 100 unidades monetárias.
- **Linhas 17 até 18:** Norma de proibição criada por meio da *operação cria no artefato NrmsPrb*. Estabelece que é proibido o lenhador cortar caso o número de árvores adultas seja menor que 200 unidades, sendo que não há exceção e a multa é de 20 unidades monetárias.
- **Linhas 19 até 20:** Um exemplo de uso de modificação de *norma de proibição*, estabelecendo um limiar maior para o corte, contudo com uma multa maior para infração. Isto poderia ser estabelecido, por exemplo, em uma situação que se percebe que a floresta está sendo sustentável.
- **Linhas 21 até 22:** Além das normas de proibição e obrigação, a política pública também estabelece um *plano de ação* para a floresta. Assim, por meio do agente governo cria-se um *plano de ação* aos *detectores de ambiente* para que verifiquem uma vez ao mês se a quantidade de árvores em desenvolvimento é menor que 20 unidades, e, neste caso, comuniquem à ele.
- **Linhas 23 até 24:** Caso onde, ao perceber a comunicação do detector de ambiente, opta-se por delegar um *plano de ação ao efetuator* para que

execute a ação reflorestar. O caso onde as ações de *detector* e *efetuator* são concentradas em um único agente, além da não necessidade que o *agente governo* faça uma intervenção pós-percepção de um *contexto*, pode ser escrito da seguinte forma:

```
cria(1,  
    qtd_arvores_desenvolv(QAD) & QAD < 20,  
    reflorestar.  
    , +mes)[artifact_name("plans_detector_efector")].
```

5.5.6 Agente Social

Através do framework proposto, um agente social que adota o papel de lenhador e utiliza a floresta alocada pelas políticas públicas anteriormente criadas, pode ser escrito conforme exemplo abaixo.

```
1      acao_rec (cortar,100).
2      acao_custo(reflorestar,8).

3      !observe_artefato("sociedade").
4      !adoptRole("lenhador").
5      !observe_artefato("floresta").
6      !observe_artefato("calendario1").

7      !cortar.

8      +!cortar
9      <-      .wait("+hora");
10     {
11           !executeAction(cortar);
12     }
13     !cortar.

14     include{bloco_raciocinio_obrg_prb.asl}

15     ob_crit_p_infr(ACAO)
16     :-      acao_custo(ACAO,C) & C > 10.

17     pb_crit_p_inf(ACAO)
18     :-      papel(Papel) &
19           .term2string(ACAO, SACAO) &
20           norma(Id, "proibido", Papel, SACAO, SCondicao, EXC, PEN, PP) &
21           acao_rec (ACAO,R) & R > PP.

22     +sanction(SANCTION, PSANCTION, ACAO) :
23     <- ?acao_rec(ACAO,R);
24     -+acao_rec_custo(ACAO,R-PSANCTION).
```

Onde:

Linhas 1 até 2: Indicam a recompensa da ação cortar e o custo da ação reflorestar.

Linhas 3 até 6: Objetivos de adotar um papel de lenhador, utilizar a floresta, e utilizar o calendário para controlar o tempo.

Linhas 7: Objetivo de cortar.

Linhas 8 até 13: Plano do agente para o objetivo de cortar. A cada hora que passa, executa a operação cortar sobre a floresta. Com a utilização do framework, as ações passam pelo *!executeAction(ACAO)* para serem avaliadas se são proibidas.

Linha 14: Incorporação do bloco de raciocínio sobre normas de proibição e obrigação (ver abaixo).

Linhas 15 até 16: Escreve-se então os critérios para infringir as obrigações, em formato de regras com cabeçalho *ob_crit_p_infr(ACAO)*, e que possuem tratamento no bloco de raciocínio. Isto preserva a autonomia de decisão do agente, permitindo-lhes escolher se irão ou não fazer o que a norma estabelece. Neste caso, no exemplo da floresta, estipulou-se que infringe a obrigação de qualquer ação caso o custo desta ação seja maior que 10 unidades monetárias. Regras para ações específicas podem ser escritas substituindo a variável *ACAO*.

Linhas 17 até 21: Escreve-se então os critérios para infringir as proibições, em formato de regras com cabeçalho *pb_crit_p_infr(ACAO)*. Isto preserva a autonomia de decisão do agente, permitindo escolher se irá ou não fazer o que a norma estabelece. Neste caso, no exemplo da floresta, estipulou-se que infringe uma proibição para uma ação qualquer se a recompensa de uma ação for maior que a penalidade.

Linhas 22 até 24: Escreve-se então o tratamento caso receba sanção por uma *ação*. Neste caso serve para toda e qualquer *ação*, diminuir sua *recompensa* com o valor da mesma menos o valor da *penalidade* que é recebida. Para tratamentos específicos, pode-se reescrever um *plano* com o comportamento perante a *sanção* individual de cada *ação*. A ação nativa do Jason - *+crenca(VALUES)*, atualiza uma crença.

Bloco de raciocínio sobre normas de proibição e obrigação, incorporado em um arquivo *bloco_raciocinio_obrg_prb.asl* para ser utilizado pela diretiva *include* na plataforma Jason.

```

1      +!observe_artefato(Art) : true
2      <-      lookupArtifact(Art,ArtId);
3              focus(ArtId).
4
5
6      -!observe_artefato(Art) : true
7      <-      .wait(10);
8              !observe_artefato(Art).
9
10     +!adoptRole(PAPEL)
11     <-      .my_name(NAME);
12             adotarpapel(NAME,PAPEL);
13             +papel(PAPEL);
14             .term2string(NAME,SNAME);
15             ?agente_papel(SNAME, SPAPEL);
16             .println("papel adotado: ", SPAPEL).
17
18     -!adoptRole(PAPEL)
19     <-      .wait(100);
20             !adoptRole(PAPEL).
21
22     +newpol(Nrmsobrg, Nrmsprb)
23     <-      !observe_artefato(Nrmsobrg);
24             !observe_artefato(Nrmsprb).
25
26     -norma(Id, "obrigatorio", Papel, Acao, Condicao, Exc, Freq):
27         papel(PAPEL)
28         & PAPEL == Papel
29     <-      .drop_intention(obj_manutencao(execute_norm(Id)));
30             .println("Norma: ", Id, "removida !").
31
32     +norma(Id, "obrigatorio", Papel, Acao, Condicao, Exc, Freq):
33         papel(PAPEL)
34         & PAPEL == Papel
35     <-      .drop_intention(obj_manutencao(execute_norma(Id)));
36             !!obj_manutencao(execute_norm(Id)).
37
38     +!obj_manutencao(execute_norm(Id)): true
39     <-      ?norma(Id,_,_,_,_,Per);
40             .wait(Freq);
41             !execute_norm(Id);
42             !!obj_manutencao(execute_norm(Id)).
43
44     +!execute_norm(Id):
45     norma(Id,"obrigatorio",Papel,Acao,Condicao,Exc,Per)
46     & condicao(Condicao)
47     & not excecao(Exc)
48     & not ob_crit_p_infr(Acao)
49     <-      .term2string(ACTION,Acao);

```

```

42         .println("Executando: ",
43             norma(Id,"obrigatorio",Papel,Acao,Condicao,Exc,Per));
44         ACTION. // Executa a ação

44         +!execute_norm(Id):
norma(Id,"obrigatorio",Papel,Acao,Condicao,Exc,Per)
45             & not condicao(Condicao) | ob_excecao(Exc) |
ob_crit_p_infr(Acao)
46         <-         .println("Esta norma nao abrange meu caso! "). // Não executa a
ação.

47         condicao(Condicao)
48 :-             .term2string(COND, Condicao) &
49             COND.

50         excecao(SExc)
51 :-             .term2string(TExc, SExc) &
52             TExc.

53         +!executeAction(ACTION) : not proibido(ACTION) | pb_crit_p_inf(ACTION)
54         <-         ACTION. // Executa a ação

55         +!executeAction(ACTION)
56         <-         .println("Não fiz porque é proibido, não me encaixo na excessão e não
tenho a 57             razão para infringir"). // Não executa a ação

58         proibido(ACAO)
59         :-         papel(Papel) &
60             .term2string(ACAO, SACAO) &
61             norma(Id, "proibido", Papel, SACAO, SCondicao, SExc, _, _) &
62             condicao(SCondicao) &
63             .println(ACAO, "é proibido!") &
64             not excecao(SExc) &
65             .println(ACAO, "não é caso de exceção!").

66         condicao(SCondicao)
67         :-         .term2string(COND, Condicao) &
68             COND.

69         excecao(SExc)
70         :-         .term2string(TExc, SExc) &
71             TExc.

```

Onde:

- **Linhas 1 até 8:** Bloco que corresponde à um *plano* para localizar um artefato específico e passar a perceber as propriedades observáveis do mesmo (por meio da operação *focus*).

- **Linhas 9 até 18:** Bloco que corresponde à um *plano* que adota um papel na *estrutura organizacional* e adiciona *crença* no formato padrão +papel(PAPEL) indicando o papel adotado.
- **Linhas 19 até 21:** Plano que corresponde ao tratamento de uma percepção de uma nova *política pública*. Neste caso irá chamar o plano para *observar as normas de obrigação e proibição* da mesma.
- **Linhas 22 até 26:** Se percebe a remoção de uma norma e esta norma se refere ao papel que o mesmo exerce, ele então remove a intenção de seguir a norma, pois isto não se faz mais necessário.
- **Linhas 27 até 31:** Percebe um evento relacionado à uma norma de obrigação. Sendo o Id já existente, caracteriza uma *modificação*. Sendo o *Id* ainda não existente, caracteriza uma *adição*. Em ambos os casos o plano só se aplica se referir-se ao papel executado pelo agente.
- **Linhas 32 até 36:** Plano que executa uma *norma* de acordo com a *periodicidade* estabelecida.
- **Linhas 37 até 52:** Bloco que corresponde à um *plano* verifica se a *ação* de uma *norma* de *obrigação* deve ser executada, ou seja, se a *condição*, *exceção* e *critérios* para infringir a norma são uma consequência lógica dos fatos do ambiente.
- **Linhas 53 até 71:** Bloco que corresponde a um *plano* que verifica se uma *ação é proibida* por uma *norma* de *proibição* deve ser executada, ou seja, se a *condição*, *exceção* e *critérios* para infringir a norma são uma consequência lógica dos fatos do ambiente.

5.5.7 Detectores e Efetadores de Ambiente

Os *planos de ação* de uma *política pública* são delegados aos agentes governamentais Detectores e Efetadores de Ambiente, os quais são estruturados sobre o código a seguir, permitindo-lhes a execução daqueles *planos de ação*.

```
1      !observe_artefato("calendario1").

2      +newpol(Plans, Recurso)
3      <-      !observe_artefato(Plans);
4              !observe_artefato(Recurso).

5      +!observe_artefato(Art) : true
6      <-      lookupArtifact(Art,ArtId);
7              focus(ArtId).

8      -!observe_artefato(Art) : true
9      <-      .wait(10);
10             !observe_artefato(Art).

11     +plano(Id, PlanContext, PlanBody):
12     <-      .concat("plan", Id,PlanEvent);
13             .concat("+!", PlanEvent, ":", PlanContext, "<- ", PlanBody, PlanFull);
14             .add_plan(PlanFull);
15             !!executePlanNoRep(Id,PlanEvent, PlanContext,PlanBody).

16     +plano(Id, PlanContext, PlanBody, Freq):
17     not existe_plano(Id)
18     <-      .concat("+!", "plan",Id, PlanEvent);
19             .concat(PlanEvent, ":", PlanContext, "<- ", PlanBody, PlanFull);
20             .add_plan(PlanFull);
21             .print("PLANO ", Id ," CRIADO");
22             .term2string(TPlanEvent,PlanEvent);
23             !!executePlanRep(Id, TPlanEvent, PlanContext, Freq).

24     +plano(Id, PlanContext, PlanBody, Freq):
25     existe_plano(Id)
26     <-      .drop_intention(executePlanRep(Id,_,_,_));
27             .concat("+!", "plan",Id, PlanEvent);
28             .relevant_plans(PlanEvent, LP, LL);
29             .nth(0,LL,PlanPast);
30             .remove_plan(PlanPast);
31             .concat(PlanEvent, ":", PlanContext, "<- ", PlanBody, PlanFull);
32             .add_plan(PlanFull);
33             .print("PLANO ", Id , " MODIFICADO");
34             !!executePlanRep(Id, PlanEvent, PlanContext, Freq).
35     -plano(Id, PlanContext, PlanBody, Freq):
36     <-      .drop_intention(executePlanRep(Id,_,_,_));
37             .concat("+!", "plan",Id, PlanEvent);
38             .relevant_plans(PlanEvent, LP, LL);
39             .nth(0,LL,PlanPast);
40
```



```

41         .remove_plan(PlanPast);
42         .print("PLANO ", Id , " REMOVIDO");

43     +!executePlanNoRep(Id,PlanEvent,PlanContext,PlanBody):
44         existe_contexto(PlanContext)
45     <-     !PlanEvent;
46         .print("Plano ", TPlanEvent , " Executado").

47     +!executePlanNoRep (Id,PlanEvent,PlanContext,PlanBody):
48         not existe_contexto(PlanContext)
49     <-     .println("Condição necessária para aplicação de plano não existe! ").

50     +!executePlanRep(Id, PlanEvent, PlanContext, Freq):
51     existe_contexto(PlanContext)
52     <-     !PlanEvent;
53         .wait(Freq);
54         .println("Frequência aguardada: ", Freq);
55         !!executePlanRep(Id,PPlanEvent, PlanContext, Freq).

56     +!executePlanRep(Id,PPlanEvent, PlanContext, Freq):
57     not existe_contexto(PlanContext)
58     <-     .println("Condição necessária para aplicação de plano não existe!");
59         .wait(Freq);
60         .println("Frequência aguardada: ", Freq);
61         !!executePlanRep(Id,PPlanEvent, PlanContext, Freq).

62     existe_contexto(PlanContext)
63     :-     .term2string(TermPlanContext, PlanContext) &
64         TermPlanContext.

65     existe_plano(Id)
66     :-     .concat("+!", "plan", Id, PlanEvent1) &
67         .relevant_plans(PlanEvent1, LP) &
68         .length(LP,X) &
69         X > 0.

```

Onde:

- **Linha 1:** objetivo que utiliza o artefato calendário concebido para modelagem da dimensão temporal da simulação , permitindo ao agente acompanhamento de eventos ao decorrer do tempo.
- **Linhas 2 até 4:** Plano que corresponde ao tratamento de uma percepção de uma nova *política pública*. Neste caso, irá chamar o plano para *observar os planos de ação e o recurso do ambiente físico* sobre o qual serão exercidos os *planos de ação* delegados pelo agente *governo*.

- **Linhas de 5 até 10:** Bloco que corresponde a um *plano* para localizar um artefato específico e passar a perceber as propriedades observáveis do mesmo (por meio da operação *focus*).
- **Linhas 11 até 15:** Bloco para tratar um *plano de ação* criado pelo agente *governo* e que deve ser executado uma única vez (criado sem o parâmetro *periodicidade*). Consiste inicialmente em convertê-lo para o formato de um plano *AgentSpeak(L)* e posteriormente adicioná-lo na biblioteca de planos. Então, `!!executePlanNoRep(Id,PlanEvent,PlanContext,PlanBody)` é um objetivo o qual pode ter um entre dois possíveis casos aplicáveis (ver linhas 43 até 46, e linhas 47 até 49, abaixo).
- **Linhas 16 até 23:** Bloco para tratar a criação de um novo *plano de ação* pelo agente *governo* (ou seja, indica que não foi encontrado um plano na biblioteca do agente com o mesmo identificador, *ver linhas 65 até 69*, abaixo) e que deve ser executado repetidamente (criado com o parâmetro *periodicidade*). Consiste em inicialmente convertê-lo para o formato de um plano *AgentSpeak(L)* e posteriormente adicioná-lo na biblioteca de planos. Por fim ocorre a chamada `!!executePlanRep(Id, PlanEvent, PlanContext, Freq)`, que tem no quarto parâmetro a *Periodicidade*, o qual é utilizado para realizar a repetição da chamada do respectivo plano, e que a cada repetição pode ter um entre dois possíveis casos aplicáveis (*ver linhas 53 até 58, e linhas 59 até 64, abaixo*).
- **Linhas 24 até 34:** Caso que trata a modificação de um plano de ação. Ou seja, indica que foi encontrado um plano na biblioteca do agente com o mesmo identificador passado (*ver linhas 68 até 72*, abaixo), e assim é necessário remover a estruturação antiga do plano da biblioteca, assim como removê-lo das intenções do agente. Após isso, então, ocorre a atualização de acordo com a modificação, que consiste na conversão dos novos elementos do plano de ação para o formato de um plano *AgentSpeak(L)* e posteriormente adição na biblioteca de planos. Dado isto

chama !!executePlanRep(Id,PlanEvent, PlanContext, Freq) para realizar as avaliações por periodicidade, como no primeiro caso da criação.

- **Linhas 35 até 42:** Caso que trata a remoção de um plano, que consiste em removê-lo da biblioteca de planos, assim como removê-lo das intenções do agente.
- **Linhas 43 até 46:** Caso onde o contexto para o qual foi delegado se aplica (ou seja, a *condição* é uma consequência lógica dos fatos do ambiente), assim, deve ser executado, portanto realiza a chamada do evento ativador do respectivo plano adicionado !PlanEvent.
- **Linhas 47 até 49:** Caso onde o contexto para o qual o *plano de ação* foi delegado não se aplica (ou seja, a *condição* não é uma consequência lógica dos fatos do ambiente).
- **Linhas 50 até 55:** Caso onde o contexto para o qual foi delegado se aplica (ou seja, a *condição* é uma consequência lógica dos fatos do ambiente), assim, deve ser executado, portanto realiza a chamada do evento ativador do respectivo plano adicionado !PlanEvent.
- **Linhas 56 até 61:** Caso onde a condição para o qual o *plano de ação* foi delegado não se aplica (ou seja, a *condição* não é uma consequência lógica dos fatos do ambiente).
- **Linhas 62 até 64:** avalia se a condição é uma consequência lógica dos fatos do ambiente.
- **Linhas 65 até 69:** verifica se há um plano da biblioteca que tenha na construção de seu evento ativador o identificador (id) passado.

5.5.8 Detectores e Efetadores de Norma

```
1      +newpol(Nrmsprb, Recurso, Sociedade)
2      <-      !observe_artefato(Nrmsprb);
3              !observe_artefato(Recurso); // ou outro plano específico com variação
de stopFocus()
4              !observe_artefato(Sociedade).

5      +!observe_artefato(Art) : true
6      <-      lookupArtifact(Art,ArtId);
7              focus(ArtId).

8      -!observe_artefato(Art) : true
9      <-      .wait(10);
10             !observe_artefato(Art).

11     +acao(ACAO, ACTOR, PAR):
12     proibido(Id,ACAO,ACTOR,PAR)
13     <-      !plan_detect_prb(Id,ACAO,ACTOR,PAR).

14     +acao(ACAO, ACTOR, PAR)
15     <-      .println("acao: ", acao, " actor: ", actor, " parametro: ", par, " não
proibido").

16     proibido(Id, ACAO, ACTOR, PAR)
17     :-      papel(ACTOR, PAPEL) &
18             norma(Id, "proibido", PAPEL, ACAO, SCondicao, SExc, _, _) &
19             .term2string(TCondicao, SCondicao) &
20             TCondicao &
21             .term2string(TExc, SExc) &
22             not TExc.

23     papel(ACTOR, PAPEL)
24     :-      agente_papel(ACTOR, PAPEL).

25     +!plan_detect_prb(Id,ACAO, ACTOR, PAR)
26     <-      .send(efector_prb_norm, tell, comunicao_infracao(Id,ACAO, ACTOR,
PAR)).
```

Onde:

- **Linhas 1 até 4:** Plano que corresponde ao tratamento de uma percepção de uma nova *política pública*. Neste caso irá chamar o plano para *observar as normas de proibição, o recurso do ambiente físico e o artefato* que representa *os papéis na sociedade*. A linha 3 de observação do *recurso do ambiente*, pode ser substituída por outras variações de planos,

possibilitando a percepção parcialmente observável do recurso. Pode-se para isso utilizar a operação *stopFocus()*, com períodos específicos de parada ou outros critérios.

- **Linhas de 5 até 10:** Bloco que corresponde a um *plano* para localizar um artefato específico e passar a perceber as propriedades observáveis do mesmo (por meio da operação *focus*).
- **Linhas 11 até 15:** Bloco que corresponde a um *plano* que verifica se uma *ação* executada por *agente social* é *proibida* por uma *norma de proibição*. Para isto chama avaliação da regra proibido(Id,ACAO, ACTOR, PAR). Se for proibida, chama o objetivo !plan_detect_prb(Id,ACAO,ACTOR,PAR) que selecionará o plano para o tratamento da infração, determinado pelo modelador da simulação (linhas 25 até 26).
- **Linhas 16 até 22:** Regra que considera o *papel do agente*, a *condição e a exceção* da norma para avaliar se são uma consequência lógica dos fatos do ambiente.
- **Linhas 23 até 24:** Regra para encontrar o papel de um agente na infraestrutura organizacional.
- **Linhas 25 até 26:** Escreve-se então os planos para tratar a infração no formato +!plan_detect_prb(Id,ACAO, ACTOR, PAR). Neste exemplo, comunica-se um agente efetuidor de norma de proibição que terá função exclusiva de tratar a infração, podendo conter critérios de acúmulo, ou outras medidas necessárias. Outro exemplo seria simplesmente acessar a norma infringida (através do Id) e então enviar diretamente ao infrator (através do ACTOR), neste caso o agente teria as funções acumuladas de detector e efetuidor de norma.

6 ESTUDO DE CASO

6.1 Piracema e a Tragédia dos Comuns

Um problema clássico socioeconômico estudado em Teoria dos Jogos é a Tragédia dos comuns (um caso particular do dilema dos prisioneiros com múltiplos jogadores), no qual agentes racionais utilizam um recurso comum até seu esgotamento. Este problema foi citado inicialmente por (HARDIN, 1968), e tem sido inspiração na elaboração de Políticas Públicas para gestão de recursos ambientais.

Em (TOCANTINS et. al., 2011) apresenta-se um cenário relacionado à pesca nas regiões tropicais do país, no qual o produto da pesca é propriedade comum e de livre acesso, tornando o manejo pesqueiro uma tarefa difícil. Trata-se de um assunto em que a natureza por si só não dá conta da resolução, dado o seu contexto político e socioeconômico. Neste sentido, ocorrem intervenções governamentais, as quais implicam na existência de regras ou códigos para definir a ação da comunidade, caso contrário os interesses individuais levariam o sistema a um colapso econômico-ecológico, pois os indivíduos, racionalmente, não iriam optar pela colaboração em sistemas de recursos comuns (LOPES, CAMILO, et. al., 2010).

Ainda no contexto das intervenções governamentais na gerência dos recursos pesqueiros, podem-se citar os esforços na elaboração de Políticas Públicas para o fenômeno denominado Piracema. Este ocorre quando os peixes sobem as cabeceiras para desova ficando mais expostos e facilmente capturáveis. No entanto, este período é primordial, pois se refere diretamente à reprodução e conseqüentemente disponibilidade de novos cardumes para pesca, de modo que se for interrompido irá ocasionar a extinção das espécies e conseqüentemente do recurso natural.

6.1.1 O Cenário descritivo utilizado

Mais especificamente a política pública elaborada pelo governo na tentativa de solucionar este problema concretizou-se por meio da Lei nº 7.679, de 23 de novembro de 1988 conhecida como a Lei do Defeso (LEI 7.679, 1988), a qual é estabelecida anualmente pelo IBAMA. A seguir, descreve-se um cenário real de aplicação da política pública de defeso, para o caso da Piracema na região dos rios goianos, de acordo com (REVISTA PESCA, 2012).

A Piracema já começou e até 28 de fevereiro de 2012 fica permitida somente a pesca para subsistência (cota de três quilos por pescador para consumo no local) e a científica.

Neste período prevalece à proibição de pesca em embarcações nos rios goianos. Durante a Piracema, a maioria das espécies nativas sobe os rios em cardumes para acasalamento e reprodução. Por isso, a Secretaria de Meio Ambiente e dos Recursos Hídricos (Semarh) fiscaliza e restringe a pesca. A proibição entrou em vigor no dia 1º de novembro. Para pescar neste período, é necessária (assim como em qualquer época do ano) a obtenção de licença de pesca. Além disso, é preciso ficar atento ao volume de pescado e ao fato de não poder transportá-lo.

A pesca só pode ser feita utilizando linha de mão ou vara, caniço simples, com molinete ou carretilha, com iscas naturais ou artificiais (a utilização dos anzóis múltiplos somente será permitida com iscas artificiais, na modalidade de arremesso e corrico). É terminantemente proibida a utilização de iscas naturais exóticas à bacia e a pesca nas lagoas marginais das bacias hidrográficas dos rios Araguaia/Tocantins, Paranaíba e São Francisco (entende-se por lagoas marginais, áreas de alagados, alagadiços, lagos, banhados,

canais ou poços naturais que recebem águas dos rios ou de outras lagoas em caráter permanente ou temporário).

Durante a Piracema fica proibida qualquer atividade de pesca além da supracitada, inclusive o uso de qualquer tipo de material predatório, limitadas as quantidades de peixes embarcados assim como obediência rigorosa ao tamanho mínimo de captura. Também fica proibido o transporte de pescado, exceto aquele proveniente de aqüicultura, ou seja, aqueles correlacionados as atividades de despesca, o transporte, a comercialização, o beneficiamento, a industrialização e o armazenamento do pescado das espécies provenientes de pisciculturas devidamente registradas na Semarh (Secretaria do Meio Ambiente e dos Recursos Hídricos) ou órgão ambiental competente, acompanhados da comprovação de origem do pescado.

*O descumprimento destas condições **sujeita o infrator à multa, apreensão do material de pesca, embarcação e processo administrativo.** A Semarh mantém a fiscalização nos rios e estradas de acesso. **Apesar do rigor da lei, muitos pescadores ainda não entenderam** a necessidade de respeitar a Piracema e continuam praticando a pesca ilegalmente, juntando-se aos demais poluidores, predadores e trazendo por consequência a crescente falta de peixes a cada novo ano. Para o pescador consciente e que respeita não apenas a lei dos homens, mas principalmente a lei da natureza, este período é uma excelente oportunidade para praticar o pesque-e-solte em nossos rios.*

6.1.2 A Modelagem e Simulação utilizando o modelo e framework elaborados

A partir do cenário e lei supracitados, formulou-se o modelo de simulação com os elementos e características conforme as tabelas a seguir:

Tabela 1. Recurso Comum Rio

<i>População de Peixes</i>	Recurso comum com característica finita compartilhado pelos agentes sociais.
peixes:43200	Representa uma população de peixes da mesma espécie, com peso de cada peixe igual a 1kg (população atual hipotética do rio no início do período da piracema).
periodo: piracema	Período de 120 dias.
unid_min: 7200	Unidades mínimas que necessitam permanecer no rio para que o mesmo continue sustentável e seja viável um novo ciclo após o término da Piracema. Neste caso, estabeleceu-se que é necessário remanescer no mínimo 20% da população de peixes.
ext:100	Área total de extensão do rio, correspondendo a 100 km.

Tabela 2. Agentes Sociais

Pescadores	São agentes racionais, ou seja, buscam a estratégia que lhes oferece maior recompensa, a qual advem das diferentes técnicas de pesca representadas aqui por técnica “a” e “b”. Neste caso específico, foram considerados 100 agentes pescadores.
tec_a: 1	A técnica de pesca “a” representa a pesca com anzol e outros métodos artesanais e subtraem um número menor de unidades do rio. Assim, estabeleceu-se que a mesma oferece 1 unidade de recompensa para o agente.
tec_b: 5	A técnica de pesca “b” representa a pesca com outros métodos mais agressivos e maior capacidade de captura (exemplo redes de arrasto), subtrai um número maior de unidades do rio. Assim, estabeleceu-se que a mesma implica em uma maior recompensa para o agente, neste caso 5 unidades, portanto, o agente tende a optar por esta técnica.
freq_pesc: 3	Número de vezes que um agente pesca por dia. Considerando que este é racional e a frequência máxima de pesca é 3 vezes por dia (manhã, tarde e noite) o mesmo irá optar pela maximização do período de exploração.
qtd_min_sub: 3	Quantidade mínima necessária para sobrevivência de cada agente

Tabela 3. Agentes Detectores e Efetuadores

Fiscal	Agente responsável por monitorar o cumprimento das normas da Política Pública. Para este caso, atribui-se a este agente a dupla função de detector e efetuator de norma, neste sentido ele tanto percebe infrações quanto aplica as penalidades.
trecho:10	Área do rio que irá cobrir, correspondente a 10 km.
freq: 3	Número de vezes que irá monitorar o ambiente (manhã, tarde, noite).
custo: 3	Valor de custo de cada fiscal, que corresponde à frequência com que irá efetuar a monitoração.

Foram então elaborados quatro cenários de simulação, os quais foram simulados no período de 120 dias, que é correspondente a duração da Piracema (1º novembro a 28 de fevereiro).

No primeiro cenário não foram estabelecidas normas nem utilizados agentes de fiscalização. Deste modo, não se teve custo político, os agentes não cooperaram (prevalecendo a escolha pela “técnica b” por todos os agentes e em todas as oportunidades de pesca), e ocorreu consumo diário de 1500 kg. Portanto, o recurso esgotou-se, perdurando apenas 28 dias dos 120 da Piracema, não sendo sustentável.

No segundo cenário, estabeleceram-se normas nas quais o descumprimento resulta em uma penalidade menor que a recompensa da “técnica b”, porém com uma quantidade alta de agentes fiscais, conforme Tabela 4:

Tabela 4. Elementos da Política do cenário 2

Normas	Normas que cada pescador deve cumprir;
<code>cria (n3, proibido, pescador, tec_b, periodo("piracema"), sem_excecao, multa, 3)[artifact_id("nrmsprb")];</code>	<p>Esta norma estabelece que seja proibido pescar com “técnica b” no período da Piracema, sem exceção e com 3 unidades de multa.</p>
<code>cria (n3, proibido, pescador, tec_a, periodo("piracema"), qtd_pesc(ACTOR, QP) & QP <= 3, multa, 1)[artifact_id("nrmsprb")];</code>	<p>Esta norma estabelece que seja proibido pescar com “técnica a” no período da Piracema, sendo que a exceção é um limite de 3 unidades diárias (correspondente a subsistência), e com 1 unidade de multa.</p>
<code>n_fiscais: 10</code>	<p>Número de fiscais</p>

Deste modo, o custo político foi alto, porém nenhum agente cooperou, prevalecendo ainda a escolha pela “técnica b” num consumo diário de 1500 kg. Portanto, o recurso esgotou-se, perdurando apenas 28 dias dos 120 da Piracema, configurando novamente a não sustentabilidade do ambiente.

No terceiro cenário estabeleceu-se normas nas quais o descumprimento resulta em uma penalidade que corresponde à recompensa das técnicas, porém com uma quantidade baixa de agentes fiscais, conforme Tabela 5:

Tabela 5. Elementos da Política do cenário 3

Normas	<i>Normas que cada pescador deve cumprir;</i>
<p>cria (n3, proibido, pescador, tec_b, periodo("piracema"), sem_excecao, multa, 5)[artifact_id("nrmsprb")];</p>	<p>Esta norma, estabelece que seja proibido pescar com “técnica b” no período da Piracema, sem exceção e com 5 unidades de multa.</p>
<p>Cria (n3, proibido, pescador, tec_a, periodo("piracema"), qtd_pesc(ACTOR, QP) & QP <= 3, multa, 1)[artifact_id("nrmsprb")];</p>	<p>Esta norma estabelece que seja proibido pescar com “técnica a” no período da Piracema, sendo que a exceção é um limite de 3 unidades diárias (correspondente a subsistência), e com 1 unidade de multa.</p>
<p>n_fiscais: 3</p>	<p>Número de fiscais</p>

Obteve-se assim um custo político baixo, poucos agentes cooperaram, ainda prevalecendo a escolha pela “técnica b” pela maioria dos agentes, num consumo diário de 1140 kg. Portanto, o recurso esgotou-se, perdurando apenas 37 dias dos 120 da Piracema, embora configurando uma melhora em relação ao cenário1, continuou não sendo sustentável.

No quarto cenário estabeleceram-se normas nas quais o descumprimento resulta em uma penalidade correspondente à recompensa das técnicas, e uma quantidade alta de agentes fiscais, conforme Tabela 6:

Tabela 6. Elementos da Política do cenário 4

Normas	Normas que cada pescador deve cumprir;
<code>cria (n3, proibido, pescador,tec_b, periodo("piracema"), sem_excecao, multa, 5)[artifact_id("nrmsprb")];</code>	Esta norma, estabelece que seja proibido pescar com “técnica b” no período da Piracema, sem exceção e com 5 unidades de multa.
<code>cria (n3, proibido, pescador, tec_a, periodo("piracema"), qtd_pesc(ACTOR, QP) & QP <= 3, multa, 1)[artifact_id("nrmsprb")];</code>	Esta norma, estabelece que seja proibido pescar com “técnica a” no período da Piracema, sendo que a exceção é um limite de 3 unidades diárias (correspondente a subsistência), e com 1 unidade de multa.
<code>n_fiscais: 10</code>	Número de fiscais

Obteve-se assim um custo político alto, todos os agentes cooperam, modificando a escolha preponderante para a “técnica a”, num consumo diário de 300 kg. Assim, o recurso e os agentes mantiveram-se durante todo o período da Piracema e ainda permaneceram 20% (7200) da população dos peixes, possibilitando a continuidade de um novo ciclo no rio, portanto demonstrando o impacto da estrutura política sobre a sustentabilidade.

No ANEXO 3, apresenta-se um *log* que representa um pequeno trecho da simulação apresentada nesta seção.

6.1.3 Discussão sobre os resultados obtidos na simulação

Em (FIANI, 2009) encontramos uma análise, na perspectiva da Teoria dos Jogos, para os resultados obtidos, bem como para os comportamentos de cenários simulados para este estudo de caso. A seguir, as análises do autor para cada um dos cenários:

- Cenário 1: no caso em que todos se comportam de forma não-cooperativa, o resultado para todos é o pior possível, de modo que as possibilidades de bem-estar em uma sociedade na qual todos se comportam com oportunismo de curto prazo descrito pelo jogo do dilema dos prisioneiros, são muito reduzidas.
- Cenário 2: nem sempre uma autoridade externa ao jogo possui informações suficientes para identificar o valor correto da punição a ser aplicada, e uma punição com valor insuficiente é inócua: os jogadores acham que “vale a pena” sofrer a punição diante dos ganhos líquidos que, ainda assim, podem obter.
- Cenário 3: estabelecer uma instituição que identifique e puna comportamentos não-cooperativos tem um custo, custo este que cresce com o aumento do número de jogadores que podem adotar comportamentos não cooperativos. Assim, este cenário demonstra que investimentos baixos em monitoramento políticos são realmente insuficientes.
- Cenário 4: o custo para que todos cooperem é alto, ainda o autor afirma que instituições que vigiem todos os agentes podem ser impraticáveis.

7 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho buscou prover, através da técnica de simulação de sistemas multiagentes, uma contribuição ao desenho de políticas públicas, especialmente para sua modelagem e simulação.

É possível citar dois resultados principais alcançados por este trabalho. O primeiro é o levantamento e definição dos principais elementos de uma política pública, no que se refere a uma perspectiva de simulação de políticas públicas em sistemas multiagentes: normas, planos, agentes sociais, agentes governamentais.

O segundo está no nível de implementação. Neste sentido, os recursos aqui apresentados podem servir como instrumentos iniciais para programação, utilizando Jason e CArTAgO, de um sistema multiagente que envolva aqueles aspectos básicos constituintes em uma política pública.

Portanto, por meio da utilização do modelo e *framework* resultantes deste trabalho, juntamente com a infraestrutura das ferramentas Jason e CArTAgO, um modelador poderia descrever, executar, experimentar e avaliar os efeitos decorrentes dos seguintes cenários de simulação:

- modelar estruturalmente e funcionalmente recursos comuns, em um ambiente;
- definir diferentes perfis de papéis de agentes sociais, além de objetivos próprios, e critérios intrínsecos que os fariam transgredir ou não as normas políticas;
- proibir ações e estabelecer diferentes contextos que estas proibições se aplicam;

- modificar as exceções das ações proibidas;
- elaborar diferentes sanções a serem aplicadas;
- remover proibições que haviam sido estabelecidas;
- obrigar ações e estabelecer diferentes contextos que estas obrigações se aplicam;
- modificar o contexto em que se aplicam as obrigações;
- alterar a frequência/período em que as ações obrigadas devem ser executadas;
- modificar as exceções das obrigações;
- remover obrigações que haviam sido estabelecidas;
- delegar diferentes planos de ação para os detectores de ambiente, indicando, por exemplo, diferentes propriedades a serem monitoradas;
- delegar diferentes planos de ação aos efetadores, de modo a modificar as propriedades dos recursos;

No que se refere à demonstração do modelo e framework para simulação de Políticas Públicas, utilizou-se um estudo de caso baseado no problema clássico Tragédia dos Comuns abordado na Teoria dos Jogos, o qual é voltado para a pesca predatória no período da Piracema. Embora com a utilização de cenários simplificados, obtiveram-se informações compatíveis com as conclusões da literatura.

Por fim, pode-se relacionar como trabalhos futuros a expansão dos mecanismos propostos para aplicá-los aos componentes dos diferentes níveis organizacionais de uma sociedade, além do nível básico de papéis sociais a que eles estão ligados, como os grupos sociais, organizações, etc.

REFERÊNCIAS

AUSTIN, J. How to Do Things with Words. 1962.

BRASILEIRO, A. M. O município como Sistema Político. Rio de Janeiro, FGV, 124 p. 1973

BORDINI, R. H.; HUBNER, J. F.; WOOLDRIDGE, M. Programming Multi-Agent Systems in AgentSpeak using Jason. University of Liverpool: Wiley, 2007.

BORDINI, R. H.; VIEIRA R. Linguagens de Programação Orientada a Agentes: uma introdução baseada em AgentSpeak(L). Revista de Informática Teórica e Aplicada – UFRGS, , 2003, V 10, N. 1, 32 p.

BORDINI, R. H.; HUBNER, J. F. et al. Jason: a java-based agentspeak interpreter used with SACI for multi-agent distribution over the net. Manual, first release. [S.l.], 2004.

BROWN. L.; HARDING. A.; Social Modelling and Public Policy: Application of Microsimulation Modelling in Australia *Journal of Artificial Societies and Social Simulation* vol. 5, no. 4 Disponível por WWW em <<http://jasss.soc.surrey.ac.uk/5/4/6.html>> , 2002.

BRYNARD, P. Policy implementation, in Cloete and Wissink. (eds.) Improving public policy. Pretoria. Van Schaik. 2000.

COSTA, A. C. R.; DIMURO, G. P.; et al.; Modelagem e Simulação de Políticas Públicas <http://mspp.c3.furg.br/> acesso em 03/2010.

DEZOUSA, K.C.; LIN, Y. Towards Evidence-Driven Policy Design: Complex Adaptative Systems and Computational Modeling. The Innovation Journal: The Public Sector Innovation Journal, vol.16, 2011.

EASTON, D.; A Framework for Political Analysis. Prentice-Hall, Englewood Cliffs, 1965.

FERBER, J. Multi-Agent Systems: an introduction to distributed artificial intelligence. Boston, MA, USA: Addison-Wesley Longman Publishing, 1999.

FIANI, R. Teoria dos Jogos: com aplicações em economia, administração e ciências sociais. Rio de Janeiro, Elsevier, 3ª edição, 2009.

Engineering the policy making life cycle (ePolicy) Disponível por WWW <http://cress.soc.surrey.ac.uk/web/projects/59-epolicy>, 2012.

GEORGEFF, M. P.; LANSKY, A. L. Reactive reasoning and planning. In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI'87), 13–17 July, 1987, Seattle, WA, pages 677–682, Manlo Park, CA. AAAI Press / MIT Press.

GILBERT, N. Agent-based Models. Los Angeles: SAGE, 2008.

GILBERT, N.; TROITZSCH, K. G. Simulation for the social scientist. Milton Keynes: Open University Press, 2005.

HAM, C.; HILL, M. The Policy Process in the Modern Capitalist State. London: Harvester, 1993.

HARDIN, G. The tragedy of the commons. Science 162, 1968.

HILL, M.: The Public Policy Process. Pearson Longman, London, 2009. (5th ed.).

HOOD, C., MARGETTS, H.: The Tools of Government in the Digital Age. Palgrave Macmillan, London, 2007.

HÜBNER, J. F. Um modelo de reorganização de sistemas multiagentes. 246 f. Tese (Doutorado em Engenharia Elétrica), Escola Politécnica de Universidade de São Paulo, São Paulo, 2003.

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

KRZYSZTOF, C.; MACIEJ, G.; PAWEL, K.; MICHAL, S.; MARCIN, P. Efficiency of JADE Agent Platform, Scientific Programming, 2005.

LEI 7.679 Disponível em:

<http://www.prpe.mpf.gov.br/internet/Legislacao/MeioAmbiente-e-Urbanismo/Leis/LEI-N1-7.679-DE-23-DE-NOVEMBRO-DE-1988>, acesso em julho/2012. 1988.

LOPES, B.; AMARAL, J.; CALDAS, R. Políticas Públicas: conceitos e práticas. Belo Horizonte : Sebrae/MG, 2008. 48 p.

LOPES, CAMILO A. S.; et al. Cerrado, gerais, Sertão comunidades tradicionais dos Sertões Roseanos. volume I, Universidade Estadual de Montes Claros, 2010.

LOPEZ, F.L., LUCK, M., DINVERNO, M.: A normative framework for agent based systems. Computational and Mathematical Organization Theory, 2005.

MACAL, C.; NORTH, M. Agent-based modeling and simulation. In Winter Simulation Conference (WSC), pages 86 –98, 2009.

MINOGUE, M., Theory and Practice in Public Policy and Administration, Policy and Politics, 1983.

OMICINI, A.; RICCI, A.; VIROLI, M. Artifacts in the A&A meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems, 17(3):432–456, Dec. 2008.

Open Collaboration for Policy Modeling Disponível por WWW <http://www.ocopomo.eu/>, 2012.

RAO, A. S.; GEORGEFF, M. P. An Abstract Architecture for Rational Agents. In: IN-INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING (KR'92), 3., 1992. Proceedings... Morgan Kaufmann, 1992. p.439–449.

Reportagem: GO reforça fiscalização durante Piracema. Revista Pesca. Disponível em <http://www.revistapesca.com.br/noticias/615-go-reforca-fiscalizacao-durante-piracema>. 2012.

RAO, ANAND. S. AgentSpeak(L): BDI agents speak out in a logical computable language. In: WORKSHOP ON MODELLING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD (MAAMAW'96), 7., 1996, Eindhoven, The Netherlands. Anais... London: Springer-Verlag, 1996. p. 42–55.

RICCI, A.; SANTI, A.; PIUNTI, M et al. CArtAgO (*Common ARTifact infrastructure for AGents Open environments*). Disponível por WWW em <http://cartago.sourceforge.net/> acesso em 03/2010.

RODRIGUES, M. M. A. Políticas públicas. São Paulo: Publifolha, 2010.

- RUA, M. G. Políticas públicas , Florianópolis : Departamento de Ciências da Administração / UFSC; [Brasília] : CAPES : UAB, 2009. 130p. : il.
- RUSSELL, S.; NORVIG, P. Inteligência Artificial: uma Abordagem Moderna, 2ª ed. Prentice-Hall: São Paulo, 2004.
- SILVA, V. C.; TROTTMAN, P.; COELHO, F. S.; SARTI, M. F.; A Abordagem de Sistemas Complexos em Administração (Pública): conceitos, agenda de pesquisa e uma aplicação na subárea de Administração/Políticas Pública(s). XIV SemeAd Seminários em Administração, 2011, ISSN 2177-3866
- SILVEIRA, A. Caderno de psicologia e políticas públicas. Curitiba : Gráfica e Editora Unificado, 2007. 50 p.
- THEODORE J. LOWI. Distribution, Regulation, Redistribution: The Functions of Government, 1964.
- TISUE, S., WILENSKY, U. NetLogo: Design and Implementation of a Multi-Agent Modeling Environment. Proceedings of Agent 2004, Chicago, 2004.
- TOCANTINS, N.; ROSSETTO, O.C.; BORGES, F.R. Abordagem socioeconômica dos pescadores afiliados à colônia z11: município de poconé, pantanal de mato grosso, brasil. Revista Geográfica da América Central 2 Disponível em <http://www.revistas.una.ac.cr/geografic>. 2011.
- WEISS, G. (Ed.). Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence. Cambridge, Massachusetts: The MIT Press, 1999.
- WILDAVSKY, A. No risk is the highest risk of all. *American scientist*. v.67, January-February, 1979.
- WOOLDRIDGE, M. W. An Introduction to MultiAgent Systems. John Wiley and Sons, Chichester, 2002.
- WOOLDRIDGE, M.; JENNINGS, N. Intelligent Agents: Theory and Practice. Manchester Metropolitan University, Department of Computing & Queen Mary & Westfield College, Department of Electronic Engineering, UK, 1995.

ANEXO 1: Um exemplo básico em AgentSpeak(L)

O cenário utilizado nesta seção foi apresentado em (BORDINI; VIEIRA, 2003). Dois robôs estão coletando lixo no planeta Marte. O robô r1 procura por lixos depositados no solo do planeta e quando algum lixo é encontrado, o robô coleta o lixo e o leva para o local onde r2 encontra-se, larga o lixo lá e retorna ao local onde o lixo foi encontrado para continuar sua busca a partir daquela posição anterior. O robô r2 está posicionado ao lado de um incinerador; todo o lixo levado por r1 é colocado no incinerador. Os pedaços de lixo são colocados randomicamente em uma grade que define o território do planeta¹³. Outra fonte de não-determinismo é a imprecisão do braço do robô em pegar os pedaços de lixo. A ação de pegar o lixo pode falhar, mas o mecanismo é bom o suficiente para não falhar mais que duas vezes seguidas; ou seja, no pior caso r1 irá tentar apanhar o lixo três vezes.

O código AgentSpeak(L) para r1 é dado abaixo. No código, cada plano é anotado com um *label* para que se possa referir ao plano no texto que segue. As ações que possuem um ponto (‘.’) em seu nome, denotam ações internas, uma noção introduzida em [3]. Essas ações são executadas internamente pelo agente, e não afetam o ambiente como as ações básicas que o agente executa.

Agent r1

Beliefs

```
pos(r2,2,2).
checking(slots).
```

Plans

```
+pos(r1,X1,Y1) : checking(slots) & not(garbage(r1)) (p1)
<- next(slot).

+garbage(r1) : checking(slots) (p2)
<- !stop(check);
!take(garb,r2);
!continue(check).

+!stop(check) : true (p3)
<- ?pos(r1,X1,Y1);
+pos(back,X1,Y1);
-checking(slots).
```

```

+!take(S,L) : true (p4)
<- !ensure_pick(S);
!go(L);
drop(S).

+!ensure_pick(S) : garbage(r1) (p5)
<- pick(garb);
!ensure_pick(S).

+!ensure_pick(S) : true <- true. (p6)

+!continue(check) : true (p7)
<- !go(back);
-pos(back,X1,Y1);
+checking(slots);
next(slot).
+!go(L) : pos(L,X1,Y1) & pos(r1,X1,Y1) (p8)
<- true.

+!go(L) : true (p9)
<- ?pos(L,X1,Y1);
moveTowards(X1,Y1);
!go(L).

```

A crença inicial do agente é sobre a posição do agente r2 na grade que define o território do planeta em que r1 deve procurar lixo, e que sua tarefa inicial será verificar os diversos pontos do território, procurando lixo. Todos os planos são explicados abaixo.

O plano p1 é usado quando o agente percebe que ele está numa nova posição e está procurando por lixo. Se não há lixo percebido naquele ponto, só o que deve ser feito é a ação básica `next(slot)` que move o robô para o próximo ponto na grade (com exceção do local do incinerador, pois o lixo nessa posição é tratado por r2). Note que essa é uma ação básica do ponto de vista do agente: assume-se que o robô possui os mecanismos físicos necessários para se mover a uma *próxima* posição no território, seguindo algum caminho predeterminado de tal forma que todas as posições sejam ordenadas e o mecanismo pode então detectar quando não há mais nenhuma posição seguinte a ser verificada.

O ambiente simulado provê a informação sobre a existência de lixo nas posições dos robôs r1 e r2, quando o agente realiza percepção do ambiente. Quando r1 percebe lixo em sua posição, a crença `garbage(r1)` é adicionada à base de crenças de tal maneira que o plano p2 pode ser então usado. A tarefa de lidar com um pedaço de lixo percebido é decomposta em três partes, envolvendo

subobjetivos de realização que garantem que: (i) o robô irá parar de procurar por lixo de maneira consistente (lembrando seu último ponto de procura para que a tarefa possa ser continuada daquele ponto); (ii) o lixo seja levado até a posição de r_2 ; e (iii) a tarefa de procurar por lixo será, então, retomada. Cada um desses objetivos são realizados respectivamente pela execução dos três planos que seguem.

Quando o agente pretende atingir o subobjetivo (i) acima, o plano p_3 é sua única opção, e ele é sempre aplicável (já que seu contexto é vazio). O agente recupera da sua base de crenças sua posição atual (a posição é inserida na base de crenças através da percepção do ambiente). O agente então toma nota da informação de para onde deve retornar para a continuação da sua busca por lixos. Isso é feito pela adição de uma crença na base de crenças: $+pos(back, X1, Y1)$. O agente remove da sua base de crenças a informação de que está procurando por lixo, e após isso o agente terá o objetivo de levar o lixo até a posição de r_2 e retornar.

O subobjetivo (ii) é tratado pelo plano p_4 , que diz que para o robô levar o lixo até a posição de r_2 , ele deve coletar o lixo e atingir o subobjetivo de ir para a posição de r_2 e, quando chegar lá, ele poderá finalmente largar o lixo. Note que $pick(garb)$ e $drop(garb)$ são ações básicas, ou seja, são coisas que o robô pode realizar fisicamente no ambiente por meio de seu hardware.

Os planos p_5 e p_6 juntos asseguram que o robô irá continuar tentando pegar um pedaço de lixo até que ele não possa mais perceber o lixo na grade (i.e., até que a ação de coleta seja realizada com sucesso). Lembre que o mecanismo de coleta é impreciso, e o robô pode ter que tentar algumas vezes até obter êxito na execução da ação.

O plano p_7 é usado para que o agente continue a tarefa de verificar a existência de lixo na grade que representa o espaço de território a ser limpo. O agente precisa atingir o subobjetivo de retornar para sua posição anterior ($!go(back)$), e uma vez lá ele poderá remover a nota que fez sobre aquela posição, lembrando que voltou ao estado de procura por lixo e, assim, procedendo para o próximo ponto na grade.

Os últimos dois planos são usados para atingir o objetivo de ir para uma posição específica na grade (representada pela constante com a qual a variável L está instanciada).

O plano p9 recupera a crença que o agente possui sobre a posição na grade em que fica a localidade L, move-se uma posição na grade em direção ao ponto referente às coordenadas em questão `moveTowards(X1,Y1)`, e volta a ter o objetivo de mover-se (continuar movendose) em direção a L; note que o plano é recursivo. O plano p8 estabelece o fim da recursão dizendo que não há mais o que fazer para atingir o objetivo de ir em direção a L se o agente já está naquela posição (o plano anterior não mais será aplicável neste caso).

O agente r2 é definido pelo código `AgentSpeak(L)` abaixo. Tudo o que ele faz é queimar os pedaços de lixo (`burn(garb)`) quando ele percebe que há lixo nessa posição (`+garbage(r2)`).

```
Agent r2  
+garbage(r2) : true  
<- burn(garb).
```

Crenças `garbage(r2)` são adicionadas à base de crenças do agente a partir da percepção do ambiente, sempre que existir lixo no ponto da grade onde r2 se encontra.

ANEXO 2: Log de execução de um exemplo hipotético de simulação utilizando o framework

Este exemplo, teve por objetivo utilizar o framework para gestão de uma horta comunitária.

- Dado que se o período for de seca as hortaliças morrem, foi criado a norma 1, a qual estabelece que é obrigatório o hortelão regar se o período for seca, sem exceção, uma vez ao dia. Conforme a seguir:

```
cria( 1,
      obrigatorio,
      hortelao,
      regar,
      periodo("seca"),
      sem_excecao,
      +hora
    )[artifact_name("nrmsobrg)];
```

- Hortaliças em desenvolvimento (ainda não maduras) não podem ser colhidas em qualquer com condição e sem exceção.

```
cria
  ( 3,
    proibido,
    hortelao,
    colher_desenvolv,
    sempre,
    sem_excecao,
    multa, 20
  )[artifact_name("nrmsprb)];
```

- Os hortelões realizam seu trabalho com as sementes fornecidas pelo governo. Assim, os detectores de ambiente verificam uma vez ao dia se as mesmas estão esgotadas, e caso sim, comunicam ao governo, que por sua vez delega um plano ao efetuidor para que forneça cinco sementes, conforme planos a seguir:

```
+num_sementes(S): S == 0
<- cria
  (1,
   sempre,
   fornece_sementes(5.))[artifact_name("plansefactor)].
```

Funcionamento básico:

As hortaliças amadurecem quando são regadas 5 vezes.

O Hortelão Pretende colher uma vez por hora se tiver hortaliça madura, se não tiver irá colher hortaliça em desenvolvimento, infringindo a norma 3. Também pretende plantar uma vez por hora se tiver sementes. Portanto a eficiência de seu trabalho depende dos planos de fornecimento do governo.

Log de simulação

LEGENDA:

NHM - número hortaliças maduras
NHD - número hortaliças em desenvolvimento
NS - número sementes

Calendário situação inicial

Estação: verão

Período: seca

Situação inicial da horta

NHM: 5 NHD: 5 NS: 5

CALENDÁRIO ----- HORA: 1

[ag_ob_pb] Novo objetivo: colher_madura

[ag_ob_pb] Novo objetivo: plantar

[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

[ag_ob_pb] colheu hortaliças!

Situação atual da horta

NHM: 4 NHD: 5 NS: 5

[detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

[ag_ob_pb] plantou!

Situação atual da horta

NHM: 4 NHD: 6 NS: 4

[detect_prb_norm] ACAO: plantar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

[ag_ob_pb] seguindo norma: n1

[ag_ob_pb] regou!

Situação atual da horta

NHM: 4 NHD: 6 NS: 4

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

[detector_env] CONDIÇÃO NECESSÁRIA PARA APLICAÇÃO DE PLANO NÃO EXISTE!

CALENDÁRIO ----- HORA: 2

[ag_ob_pb] Novo objetivo: colher_madura

[ag_ob_pb] Novo objetivo: plantar

[ag_ob_pb] seguindo norma: n1obrigatorioregarperíodo("seca")

[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

[ag_ob_pb] colheu hortaliças!

Situação atual da horta

NHM: 3 NHD: 6 NS: 4

[ag_ob_pb] regou!

Situação atual da horta

NHM: 3 NHD: 6 NS: 4

[detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

[ag_ob_pb] plantou!

Situação atual da horta

NHM: 3 NHD: 7 NS: 3

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[detect_prb_norm] ACAO: plantar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 3

[ag_ob_pb] Novo objetivo: colher_madura
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Novo objetivo: plantar
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
[ag_ob_pb] colheu hortaliças!

Situação atual da horta
NHM: 2 NHD: 7 NS: 3

[ag_ob_pb] regou!

Situação atual da horta
NHM: 2 NHD: 7 NS: 3

[ag_ob_pb] plantou!

Situação atual da horta
NHM: 2 NHD: 8 NS: 2

[detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[detect_prb_norm] ACAO: plantar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 4

[ag_ob_pb] Novo objetivo: colher_madura
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Novo objetivo: plantar
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
ag_ob_pbcolheu hortaliças!

Situação atual da horta
NHM: 1 NHD: 8 NS: 2
ag_ob_pb regou!

Situação atual da horta
NHM: 1 NHD: 8 NS: 2

ag_ob_pb plantou!
Situação atual da horta
NHM: 1 NHD: 9 NS: 1

[detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[detect_prb_norm] ACAO: plantar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 5

[ag_ob_pb] Novo objetivo: colher_madura
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Novo objetivo: plantar
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
ag_ob_pbcolheu hortaliças!

Situação atual da horta
 NHM: 0 NHD: 9 NS: 1
 [detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
 ag_ob_pb regou!
 ag_ob_pb regou 5 vezes!

Situação atual da horta
 NHM: 1 NHD: 8 NS: 1

ag_ob_pb plantou!
 Situação atual da horta
 NHM: 1 NHD: 9 NS: 0

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
CALENDÁRIO ----- HORA: 6

[detect_prb_norm] ACAO: plantar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
 [ag_ob_pb] Sem sementes!
 [ag_ob_pb] Novo objetivo: colher_madura
 [ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
 [ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
 ag_ob_pbcolheu hortaliças!

Situação atual da horta
 NHM: 0 NHD: 9 NS: 0
 ag_ob_pb regou!

Situação atual da horta
 NHM: 0 NHD: 9 NS: 0

[detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
CALENDÁRIO ----- HORA: 7

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
 [ag_ob_pb] Sem sementes!
 [ag_ob_pb] Sem hortaliças maduras!
 [ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
 [ag_ob_pb] Então vou colher uma em desenvolvimento!
 [ag_ob_pb] Novo objetivo: colher_desenvolv
 [ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
 [detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
 ag_ob_pb regou!

Situação atual da horta
 NHM: 0 NHD: 9 NS: 0

ag_ob_pbcolheu hortaliças desenvolv!

Situação atual da horta
 NHM: 0 NHD: 8 NS: 0

[detect_prb_norm] ACAO: colher_desenvolv ACTOR: ag_ob_pb PARAMETRO: 1 É PROIBIDO
 [detect_prb_norm] INFRAÇÃO COMUNICADA!
 [efector_prb_norm] Comunicação de infração recebida:
 [efector_prb_norm] ACAO: colher_desenvolvACTOR: ag_ob_pbPARAMETRO: 1
 [efector_prb_norm] PAPEL DESCOBERTO: hortelao
 [efector_prb_norm] SANSAO: multa20
 [efector_prb_norm] Sanção submetida ao agente: ag_ob_pb
 [ag_ob_pb] Fui sancionado com: multa20por: colher_desenvolv

CALENDÁRIO ----- HORA: 8

[ag_ob_pb] Sem sementes!
 [ag_ob_pb] Sem hortaliças maduras!
 [ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")

[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 8 NS: 0

CALENDÁRIO ----- HORA: 9

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 8 NS: 0

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 10

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
ag_ob_pb regou!
ag_ob_pb regou 5 vezes!
Situação atual da horta
NHM: 1 NHD: 7 NS: 0

CALENDÁRIO ----- HORA: 11

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Novo objetivo: colher_madura
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
ag_ob_pbcolheu hortaliças!

Situação atual da horta
NHM: 0 NHD: 7 NS: 0

ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 7 NS: 0

[detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 12

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv

[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 7 NS: 0

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 13

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 7 NS: 0

[ag_ob_pb] Novo objetivo: colher_desenvolv
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir

CALENDÁRIO ----- HORA: 14

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 7 NS: 0
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 15

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
ag_ob_pb regou!
ag_ob_pb regou 5 vezes!

Situação atual da horta
NHM: 1 NHD: 6 NS: 0

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 16

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Novo objetivo: colher_madura
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
ag_ob_pbcolheu hortaliças!

Situação atual da horta
NHM: 0 NHD: 6 NS: 0

ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 6 NS: 0

[detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 17

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 6 NS: 0

CALENDÁRIO ----- HORA: 18

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 6 NS: 0

CALENDÁRIO ----- HORA: 19

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 6 NS: 0

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 20

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
ag_ob_pb regou!
ag_ob_pb regou 5 vezes!

Situação atual da horta
NHM: 1 NHD: 5 NS: 0

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] Novo objetivo: colher_desenvolv

[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir

CALENDÁRIO ----- HORA: 21

[ag_ob_pb] Sem sementes!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Novo objetivo: colher_madura
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
ag_ob_pb regou!

Situação atual da horta
NHM: 1 NHD: 5 NS: 0

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
ag_ob_pbcolheu hortaliças!

Situação atual da horta
NHM: 1 NHD: 5 NS: 0

[detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 22

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 5 NS: 0

CALENDÁRIO ----- HORA: 23

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir
ag_ob_pb regou!

Situação atual da horta
NHM: 0 NHD: 5 NS: 0

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 24

[ag_ob_pb] Sem sementes!
[ag_ob_pb] Sem hortaliças maduras!
[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")
[ag_ob_pb] Então vou colher uma em desenvolvimento!
[ag_ob_pb] Novo objetivo: colher_desenvolv

CALENDÁRIO ----- DIA: 1

[detector_env] FREQUENCIA AGUARDADA: +dia
ag_ob_pb regou!

Situação atual da horta

NHM: 0 NHD: 5 NS: 0

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir

[gov_ob_pb_plan] RECIBI INFORMAÇÃO DO ENVIRONMENT DETECTOR
[gov_ob_pb_plan] ----- PLANO P2 CRIADO:
efector_env forneceu sementes!

Situação atual da horta

NHM: 0 NHD: 5 NS: 0

[efector_env] PLANO p2_esementes EXECUTADO

CALENDÁRIO ----- HORA: 1

[ag_ob_pb] Sem hortaliças maduras!

[ag_ob_pb] Novo objetivo: plantar

[ag_ob_pb] seguindo norma: n1obrigatorioregarperiodo("seca")

[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

[ag_ob_pb] Então vou colher uma em desenvolvimento!

[ag_ob_pb] Novo objetivo: colher_desenvolv

[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir

ag_ob_pb plantou!

Situação atual da horta

NHM: 0 NHD: 6 NS: 4

ag_ob_pb regou!

ag_ob_pb regou 5 vezes!

Situação atual da horta

NHM: 1 NHD: 5 NS: 4

[detect_prb_norm] ACAO: plantar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

[detect_prb_norm] ACAO: regar ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

CALENDÁRIO ----- HORA: 2

CALENDÁRIO ----- PERÍODO DE CHUVA

Calendário situação atual

Estação: verão

Período: chuva

[ag_ob_pb] Novo objetivo: colher_madura

[ag_ob_pb] Novo objetivo: plantar

[ag_ob_pb] Esta norma nao abrange meu caso!

[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

ag_ob_pbcolheu hortaliças!

Situação atual da horta

NHM: 0 NHD: 5 NS: 4

ag_ob_pb plantou!

Situação atual da horta

NHM: 0 NHD: 6 NS: 3

[detect_prb_norm] ACAO: colher_madura ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO

ANEXO 3: Log de execução da simulação do estudo de caso

Obs: log de dois dias de simulação com um agente pescador e um agente detector de norma.

[ag_ob_pb] Meu nome é: ag_ob_pb

[ag_ob_pb] papel adotado: pescador
Relatorio Estoque criado
Calendário situação inicial
Periodo: piracema
[efector_prb_norm] Governo criou nova política!
[detect_prb_norm1] Governo criou nova política!
[detect_prb_norm1] Governo criou nova política!
[gov_ob_pb_plan] ----- NORMA N1 OBRIGAÇÃO CRIADA:
[gov_ob_pb_plan] ----- NORMA N2 PROIBIÇÃO CRIADA:
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 0
declaração: ag_ob_pb0
[gov_ob_pb_plan] ----- NORMA N3 PROIBIÇÃO CRIADA:
[gov_ob_pb_plan] Política criada!

CALENDÁRIO ----- manhã!

[ag_ob_pb] turnomanha
[ag_ob_pb] agora tec_b vale mais a pena
[ag_ob_pb] tec_bé proibido!
[ag_ob_pb] tec_bnão é caso de exceção!
[ag_ob_pb] Novo objetivo: tec_b
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
Qtd_peixes: 4315
[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_b proibido!
[detect_prb_norm1] tec_bnão é caso de exceção!
[detect_prb_norm1] ACAO: tec_b ACTOR: ag_ob_pb PARAMETRO: 5 É PROIBIDO
[detect_prb_norm1] INFRAÇÃO COMUNICADA!
[efector_prb_norm] Comunicação de infração recebida:
[efector_prb_norm] ACAO: tec_bACTOR: ag_ob_pbPARAMETRO: 5
[efector_prb_norm] PAPEL DESCOBERTO: pescador
[efector_prb_norm] SANSAO: multa5
[efector_prb_norm] Sanção submetida ao agente: ag_ob_pb
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[ag_ob_pb] pescou recalculou: 5
[ag_ob_pb] Fui sancionado com: multa5por: tec_b
[ag_ob_pb] NEWREC: 0
declaração: ag_ob_pb0
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 0
declaração: ag_ob_pb0

CALENDÁRIO ----- tarde!

[ag_ob_pb] turnotarde
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_aé proibido!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
Qtd_peixes: 4314

[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_a proibido!
[detect_prb_norm1] ACAO: tec_a ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[ag_ob_pb] pescou recalculou: 1
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 1
declaração: ag_ob_pb1

CALENDÁRIO ----- noite!

[ag_ob_pb] turnonoite
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_aé proibido!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
Qtd_peixes: 4313
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_a proibido!
[detect_prb_norm1] ACAO: tec_a ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] pescou recalculou: 2
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 2
declaração: ag_ob_pb2

CALENDÁRIO ----- DIA: 1

[ag_ob_pb] novo dia!
[ag_ob_pb] zerando estoque...
declaração: ag_ob_pb0

CALENDÁRIO ----- manhã!

[ag_ob_pb] turnomanha
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_aé proibido!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
Qtd_peixes: 4312
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_a proibido!
[detect_prb_norm1] ACAO: tec_a ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] pescou recalculou: 1
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 1
declaração: ag_ob_pb1

CALENDÁRIO ----- tarde!

[ag_ob_pb] turnotarde
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_aé proibido!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
Qtd_peixes: 4311
[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_a proibido!
[detect_prb_norm1] ACAO: tec_a ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[ag_ob_pb] pescou recalculou: 2

[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 2
declaração: ag_ob_pb2

CALENDÁRIO ----- noite!

[ag_ob_pb] turno noite
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_a é proibido!
[ag_ob_pb] tec_anão é caso de exceção!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
Qtd_peixes: 4310
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_a proibido!
[detect_prb_norm1] tec_anão é caso de exceção!
[detect_prb_norm1] ACAO: tec_a ACTOR: ag_ob_pb PARAMETRO: 1 É PROIBIDO
[detect_prb_norm1] INFRAÇÃO COMUNICADA!
[efector_prb_norm] Comunicação de infração recebida:
[efector_prb_norm] ACAO: tec_aACTOR: ag_ob_pbPARAMETRO: 1
[efector_prb_norm] PAPEL DESCOBERTO: pescador
[ag_ob_pb] pescou recalculou: 3
[efector_prb_norm] SANSAO: multa1
[efector_prb_norm] Sanção submetida ao agente: ag_ob_pb
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] Fui sancionado com: multa1por: tec_a
[ag_ob_pb] NEWREC: 0
declaração: ag_ob_pb0
[ag_ob_pb] quantidade pescada verificada: 0
declaração: ag_ob_pb0

CALENDÁRIO ----- DIA: 2

[ag_ob_pb] novo dia!
[ag_ob_pb] zerando estoque...
declaração: ag_ob_pb0

CALENDÁRIO ----- manhã!

[ag_ob_pb] turno manhã
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_a é proibido!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir
Qtd_peixes: 4309
[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_a proibido!
[detect_prb_norm1] ACAO: tec_a ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[ag_ob_pb] pescou recalculou: 1
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 1
declaração: ag_ob_pb1

CALENDÁRIO ----- tarde!

[ag_ob_pb] turno tarde
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_a é proibido!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

Qtd_peixes: 4308
[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_a proibido!
[detect_prb_norm1] ACAO: tec_a ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[ag_ob_pb] pescou recalculou: 2
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 2
declaração: ag_ob_pb2

CALENDÁRIO ----- noite!

[ag_ob_pb] turnoite
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_aé proibido!
[ag_ob_pb] tec_anão é caso de exceção!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir

CALENDÁRIO ----- DIA: 3

[ag_ob_pb] novo dia!
[ag_ob_pb] zerando estoque...
declaração: ag_ob_pb0

CALENDÁRIO ----- manhã!

[ag_ob_pb] turnomanha
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_aé proibido!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

Qtd_peixes: 4307
[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_a proibido!
[detect_prb_norm1] ACAO: tec_a ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[ag_ob_pb] pescou recalculou: 1
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 1
declaração: ag_ob_pb1

CALENDÁRIO ----- tarde!

[ag_ob_pb] turnotarde
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_aé proibido!
[ag_ob_pb] Novo objetivo: tec_a
[ag_ob_pb] Fiz porque ou não é proibido, ou me encaixo na exceção, ou tenho razão para infringir

Qtd_peixes: 4306
[detect_prb_norm1] papel descoberto:pescador
[detect_prb_norm1] tec_a proibido!
[detect_prb_norm1] ACAO: tec_a ACTOR: ag_ob_pb PARAMETRO: 1 NÃO PROIBIDO
[ag_ob_pb] aguardou +acao(ACAO,ACTOR,PAR)
[ag_ob_pb] pescou recalculou: 2
[ag_ob_pb] seguindo norma: n1obrigatoriodeclarar_estoqueperiodo("piracema")
[ag_ob_pb] aconteceu nova ação!
[ag_ob_pb] quantidade pescada verificada: 2
declaração: ag_ob_pb2

CALENDÁRIO ----- noite!

[ag_ob_pb] turnoite
[ag_ob_pb] agora tec_a vale mais a pena
[ag_ob_pb] tec_aé proibido!

[ag_ob_pb] tec_anão é caso de exceção!
[ag_ob_pb] Novo objetivo: tec_a

[ag_ob_pb] Não fiz porque é proibido, não me encaixo na exceção e não tenho a razão para infringir

ANEXO 4: Publicações do trabalho

Artigos Completos

- RODRIGUES, H. D. ; SANTOS, I. A. S; DIMURO, G. ; DIMURO, G. P. ; ADAMATTI, D. F. ; Jerez, E. de M. . **A MAS for the Simulation of Normative Policies of the Urban Vegetable Garden of San Jerónimo, Seville, Spain.** In: VII Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações, 2013, São Paulo. Anais WESAAC 2013. São Paulo: USP, 2013. v. 1. p. 79-84.
- SANTOS, I. A. S.; Costa, A. C. R.; Dimuro. G. P.; Mota, F. P. **Um framework para simulação de Políticas Públicas aplicado ao caso da Piracema, sob o olhar da Teoria dos Jogos.** In: 9º Encontro Nacional de Inteligência Artificial (ENIA), Curitiba. -, 2012.
- SANTOS, I. A. S. ; Costa, A. C. R. **Toward a Framework for Simulating Agent-Based Models of Public Policy Processes on the Jason-CARtAgO Platform.** In: 20 th European Conference on Artificial Intelligence (ECAI), On: 2nd International Workshop on Agent-Based Modeling for Policy Engineering Montpellier. -, 2012.
- SANTOS, I. A. S. ; Costa, A. C. R. . **Simulando a Execução de Políticas Públicas através de Jason e CARtAgO.** In: WESAAC 2012 - VI Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações, 2012, Florianópolis. Anais do WESAAC 2012. Florianópolis : UFSC, 2012. p. 81-91.

Resumo

- SANTOS, I. A. S. ; Costa, A. C. R. . **Ferramentas para Modelagem e Simulação Social.** In: 10ª MOSTRA DE PRODUÇÃO UNIVERSITÁRIA - FURG, 2011, RIO GRANDE. 10ª MPU, 2011.