

MECANISMO BASEADO EM LÓGICA DEÔNICA PARA O TRATAMENTO RACIONAL DE NORMAS DE OBRIGAÇÃO – Um estudo em Agentes Reativos Simples

Pedro I. F. Aragão, Gustavo A. L. de Campos, Mariela I. Cortés,
Francisco I. S. Cruz

Curso de Ciências da Computação – Universidade Estadual do Ceará (UECE)
Av. Dr. Silas Munguba, 1700 – 60740-000 – Fortaleza – CE – Brasil

{gustavo, mariela}@larces.uece.br, {aragao.pedro.ivo,
israel.santos.cr}@gmail.com

Abstract: When dealing with simple reactive agents in the presence of norms, we may incur in obligation norms that make the agent perform unnecessary and irrational actions in their task environment. Considering environments with limited resources, improve the agent performance is critical. This article proposes an approach to the appropriate treatment of obligation norms in order to avoid the performing of unnecessary and irrational actions for the normative reflex agent.

Resumo: Quando tratamos de organizações de agentes reativos simples na presença de normas, podemos incorrer em normas que obriguem o agente a executar ações desnecessárias e irracionais em seu ambiente de tarefa. Considerando ambientes com recursos escassos, é importante que o desempenho de cada agente seja otimizado. O presente artigo propõe uma abordagem para o tratamento adequado de normas de obrigação visando inibir a execução de ações desnecessárias e irracionais para o caso de agentes reativos simples.

1. INTRODUÇÃO

Um agente é uma entidade capaz de perceber o ambiente através de sensores e agir nesse ambiente por meio de atuadores [Russell e Norvig, 2004]. Agentes são entidades (i) autônomas e não passivas; e (ii) capazes de interagir através de troca de mensagens e não invocação explícita de uma tarefa, como no caso de objetos [Wagner 2003]. Os agentes podem ser classificados a partir das características de sua arquitetura interna como, por exemplo, a capacidade de armazenar o histórico de ações e de adquirir conhecimento.

Um conjunto de agentes cooperando ou disputando entre si, inseridos em um mesmo ambiente, consiste em um sistema multiagente (SMA). As normas fornecem um meio para regular o comportamento dos agentes [Figueiredo, 2011] em tais sistemas. Os sistemas multiagente normativos (SMAN), cujas normas possuem a função de regular o comportamento dos agentes, caracterizam-se pela inserção de proposições normativas

envolvendo os conceitos deônticos de obrigação, proibição e permissão [Figueiredo e Silva, 2010].

Em sistemas multiagentes normativos abertos, caracterizados pela variação populacional de agentes, as normas restringem e orientam o comportamento dos agentes visando a realização de objetivos não triviais da organização. Em ambientes de tarefas complexos, as organizações são responsáveis por regular o comportamento autônomo dos agentes participantes, definido conjuntos de normas que devem ser cumpridos pelos agentes. Essas normas são transmitidas aos agentes através dos papéis que são exercidos por eles na organização. As organizações agrupam os agentes de um SMA [Shoham, 1993].

O problema da implementação de um SMAN aberto consiste em fazer com que os agentes entrem em conformidade com as normas especificadas pelo projetista do sistema [Grossi et al., 2009]. Em geral, os agentes autônomos recebem as normas definidas pela organização e decidem por cumprir umas e violar outras, conforme seus objetivos no ambiente, que podem ser ou não os mesmos da organização ao qual pertencem [Modgil et al., 2009]. Visando alcançar objetivos estratégicos, uma organização de agentes racionais deve ser capaz de conceber mecanismos de regulação que sejam adequados a penalizar/recompensar aqueles agentes que violarem/cumprirem as normas, de acordo com condições no ambiente de tarefas.

No nível de agente individual, foram propostas adaptações nos processos decisórios dos agentes racionais visando adequar os mesmos para a realização de tarefas na presença de normas. Campos et al. (2012) desenvolveram uma abordagem para o caso dos agentes reativos simples baseados em regras condição-ação. A abordagem consiste em um refinamento da estrutura do programa de agente reativo simples [Russell e Norvig, 2004], adaptando-o a perceber e raciocinar com normas enviadas pela organização em tempo de execução de tarefas em ambientes dinâmicos.

Os desempenhos obtidos nos testes realizados utilizando o programa refinado foram satisfatórios conforme uma medida de avaliação pré-estabelecida. Entretanto, percebeu-se que a pontuação que marca o desempenho dos agentes é influenciada pela definição de um conjunto de normas adequadas, principalmente quando obrigados a executar ações sobre determinadas condições ambientais. Tais normas podem levar os agentes a executar ações desnecessárias e irracionais, consumindo recursos disponíveis sem alterar de forma significativa o ambiente e prejudicando o desempenho do agente na organização.

O artigo descreve um novo refinamento na estrutura do programa de agente reativo simples visando capacitá-lo à seleção de ações racionais em ambientes de tarefas dinâmicos regidos por normas. A abordagem proposta envolve a definição de um esquema automático de transformação entre os conceitos deônticos, fundamentado em um conjunto de equivalências e implicações lógicas entre proposições. Os primeiros resultados de um estudo comparativo, envolvendo agentes programados de acordo com o primeiro e o segundo refinamento mostraram que a abordagem é factível e melhora o desempenho individual dos agentes.

O artigo foi estruturado em seis seções. A Seção 2 apresenta o agente reativo normativo simples. A Seção 3 aborda o problema das ações obrigatórias desnecessárias. Na Seção 4 é apresentada a solução proposta para o problema. A Seção 5 apresenta os algoritmos responsáveis por automatizar as transformações entre normas. A Seção 6 apresenta, através de um caso de estudo simples, como a solução proposta neste artigo pode ser utilizada. Finalmente, a Seção 7 apresenta as conclusões e trabalhos futuros.

2. O AGENTE REATIVO SIMPLES NORMATIVO

O agente reativo simples não possui nenhum componente específico no compartimento intermediário, pois não possui uma base de conhecimento e nenhum objetivo explícito em tempo de modelagem. As propriedades comportamentais desta arquitetura contêm apenas as percepções e ações guiadas por regras condição-ação [Gonçalves et al., 2009]. Em ambientes observáveis, o desempenho do agente reativo simples baseado em regras condição-ação costuma ser satisfatório, dependendo do conjunto de regras condição-ação ser suficiente para a seleção de ações racionais. Em ambientes dinâmicos, estes agentes costumam responder mais rapidamente às mudanças nas condições do ambiente de tarefa que outros tipos de agente com arquiteturas mais complexas.

Segundo Figueiredo (2011) as normas providenciam um meio para regular o comportamento de um agente descrevendo as ações que podem ser executadas (permissão), ações que devem ser executadas (obrigação) e ações que não podem ser executadas (proibição). Influenciado pelo trabalho de Meneguzzi e Luck (2009) que propôs adaptações na arquitetura BDI, habilitando o agente a reconhecer normas, Campos et al. (2012) propuseram um refinamento na estrutura do programa de agente reativo simples descrito por Russell e Norvig (2004), visando capacitar o programa a reconhecer, em tempo de execução de tarefas em um ambiente dinâmico, normas definidas por uma organização, bem como selecionar ações permitidas, ou seja, que cumpram com tais normas.

Para evitar que o agente reativo simples normativo viole as normas definidas pela organização, Campos et al. (2012) propuseram a adição de subconjuntos de regras condição-ação no conjunto de regras condição-ação que definem as ações que são possíveis para um agente reativo. Desta forma foram definidos três diferentes grupos de regras condição-ação associados aos conceitos deônticos:

- Grupo de Obrigação: composto de regras relacionadas com as ações que devem ser executadas pelo agente.
- Grupo de Proibição: composto de regras relacionadas com as ações que não podem ser executadas pelo agente.
- Grupo de Permissão: especifica as regras relacionadas com as ações possíveis que podem ser executadas pelo agente.

Esta abordagem considera que se uma ação é obrigada, então o agente deve executá-la somente se não for proibida. Caso a ação seja proibida, então o agente irá inibir a execução da ação e escolherá outra ação para executar. Caso a ação seja

obrigatória e não proibida, o agente irá executá-la imediatamente. Caso não existam ações obrigatórias ou proibidas, então o agente deve executar alguma ação que seja possível e racional.

Assim, mais especificamente, a função ação do programa de agente reativo simples normativo foi especificada de maneira a evitar conflitos entre regras de diferentes grupos. Esta função foi concebida considerando fundamentalmente uma sequência de quatro passos principais:

(P1) primeiramente, faz-se uma busca por regras que pertencem ao grupo de regras obrigatórias, para achar aquelas ações que devem ser executadas e não são proibidas;

(P2) se existir alguma ação proibida, então a função inibe a regra de obrigação e busca por regras de proibição para achar aquelas ações que não são proibidas e podem ser realizadas de acordo com as condições do ambiente;

(P3) se não existe uma regra proibida, a função seleciona a ação que deve ser executada como indicado pela norma de obrigação;

(P4) finalmente, no caso onde não existe uma regra de obrigação (nem de proibição), a função busca por regras no grupo de permissão que corresponda a alguma ação que pode ser executada de acordo com o estado do ambiente.

A estratégia definida para selecionar ações é norteada pelo princípio de que o comportamento racional é alcançado quando o agente maximiza a sua recompensa, a partir de (a) selecionar uma ação obrigada que não seja proibida, (b) não selecionar uma ação proibida e (c) selecionar uma ação permitida e adequada com o estado do ambiente, que maximize o desempenho do agente. A medida de avaliação recompensa o agente reativo simples pelo cumprimento das normas que são definidas pela organização.

No trabalho de Cruz (2013), o agente proposto acima foi implementado utilizando o *framework* Jamder 2.0 [Rocha Jr. et al., 2013]. A Figura 1 apresenta o algoritmo da função agente reativo simples normativo.

Algoritmo 1: Função agente reativo simples normativo	
Entrada: estado do mundo Saída: ação para os atuadores	
Início	<p>se existem regras condição-ação com normas de proibição ativa então inibir ações inferida pelas regras;</p> <p>se existem regras condição-ação com normas de obrigação ativas então retornar ações inferidas pelas regras;</p> <p>verificar regras condição-ação retornar ação;</p>
Fim	

Figura 1: Função agente reativo simples normativo

Caso exista mais de uma ação retornada pelo módulo de obrigação, o agente irá executar a primeira ação retornada. A função agente da figura 1 executa o que foi proposto na Seção 2, sempre dando prioridade às normas de proibição. Caso uma ação não possua nenhuma especificação normativa, esta ação será tratada como permitida.

3. EFICIÊNCIA DO AGENTE REATIVO SIMPLES NORMATIVO

Na arquitetura normativa apresentada na seção anterior, o comportamento do agente é governado por normas [Figueiredo, 2011] através de permissões, proibições e obrigações. O ato de sancionar está frequentemente relacionado com o custo a ser pago pelo agente envolvido [Hollander e Wu, 2011] e pode ser associado com a quantidade de recursos ganhos ou gastos em uma interação. Assim, no caso específico em que os agentes normativos têm recursos limitados, ele necessita raciocinar sobre os recursos disponíveis na tomada de decisão. De forma simplificada, o custo pode ser diretamente mapeado para perda de valor de utilidade (medida de avaliação de desempenho) no acaso de agentes racionais. Em casos mais complexos, o custo pode impactar em termos de perda de confiabilidade ou amizade entre os agentes.

Agentes reativos simples não possuem mecanismos na sua tomada de decisão que lhes permitem distinguir entre ações que melhoram ou pioram seu desempenho, por exemplo, recursos que podem estar sendo consumidos desnecessariamente com a execução de uma ação obrigatória. Em particular, normas de obrigação podem levar o agente a executar ações (não proibidas) desnecessariamente e/ou irracionais, que leva ao desperdício de recursos do sistema e, conseqüentemente prejudica o desempenho do agente.

Considere os ambientes de tarefa dinâmicos semelhantes aos dos agentes aspiradores de pó onde periodicamente a organização define novas normas visando regular o comportamento dos seus agentes. Considere ainda que a medida de avaliação de desempenho estabelecida pela organização recompensa os agentes pelo cumprimento das normas durante a execução de ações no ambiente de tarefas. Especificamente, no caso do aspirador de pó, a organização recompensa o agente quando ele executa a ação de “aspirar” em um local que contém sujeira.

Nos casos em que o recurso energia disponível para a execução de tarefas é escasso, a medida de avaliação pode penalizar proporcionalmente as ações que são possíveis para o agente de acordo com o consumo decorrente da execução de cada ação. Entretanto, pode ocorrer de um agente desperdiçar energia se alguma norma definida de forma inadequada ou incompleta estiver obrigando o agente a executar uma ação cujo impacto no ambiente é irrelevante, por exemplo, quando o agente aspirador de pó é obrigado a aspirar durante um período de tempo uma determinada sala que está limpa [Campos et al., 2012].

Considerando que o cumprimento das normas da organização é recompensado pela medida de avaliação, seria irracional o agente descumpri-las. Assim, surge um conflito que não é tratado na implementação do programa de agente reativo simples refinado, uma vez que os agentes são impedidos de violar as normas da organização.

Um agente pode ser levado a executar ações desnecessárias e irracionais devido à presença de normas de obrigação definidas de forma inadequada ou incompleta quando:

- (1) O cumprimento da norma conduz o agente a executar ações que não alteram o ambiente de tarefa, porém consomem recursos finitos do ambiente, prejudicando a utilidade (desempenho individual) do agente;
- (2) O descumprimento da norma conduz o agente ao comportamento irracional, em função do agente deixar de receber a recompensa pelo cumprimento da norma e receber uma punição pelo descumprimento da mesma.

Neste trabalho, as ações resultantes de normas de obrigação mal definidas foram denominadas desnecessárias e irracionais, pois, em geral, a não execução destas ações é melhor em termos da medida de avaliação de desempenho estabelecida para o agente pela organização em que participa.

4. A ABORDAGEM PARA O TRATAMENTO RACIONAL DE NORMAS DE OBRIGAÇÃO

Segundo Meyer e Wieringa (1991), a lógica deôntica em um sistema multiagente é usada para restringir o comportamento dos agentes na forma de obrigação (o que o agente deve fazer), na forma de permissão (o que o agente pode fazer) e na forma de proibição (o que o agente não pode fazer). Seja O um operador de obrigação empregado para declarar regras cujos consequentes são ações obrigadas, P um operador de permissão para declarar regras cujos consequentes são ações permitidas e F um operador de proibição para declarar regras cujos consequentes são ações proibidas. Por exemplo, se existir uma regra condição-ação p cujo consequente seja a ação a , então as declarações simbólicas $O p$, $P p$ e $F p$ significam, respectivamente, que o agente é obrigado a executar a ação a , é permitido executar a ação a e é proibido executar a ação a .

De acordo com G. H. von Wright (1951), $P p \leftrightarrow \neg O \neg p$. Logo, $\neg P p \leftrightarrow O \neg p$. Mas, $\neg P p \leftrightarrow F p$. Assim, $F p \leftrightarrow O \neg p$ e, conseqüentemente, obtemos a seguinte equivalência entre proposições normativas envolvendo os conceitos deônticos obrigação e proibição $O p \leftrightarrow F \neg p$. Logo, no contexto do agente reativo simples normativo, o agente é obrigado a executar uma ação a se e somente se o agente for proibido de não executar a ação a . Como no contexto do agente baseado em regras condição-ação não existe a regra $\neg p$, então consideramos a não proibição de uma ação, isto é, não é proibido fazer a ação p (remetemos a ideia de permissão), como a proibição de todas as ações, excetuando a ação que da regra não proibida, ou seja, o agente está proibido de executar qualquer ação diferente de a , pois $\forall x, x \neq y : F \neg y \rightarrow (F x \wedge \neg F y)$. Por exemplo, no contexto do agente reativo simples normativo, obrigar o agente a executar a ação “aspirar” em um local durante algum período de tempo é equivalente a proibi-lo a “não aspirar” o local no período determinado, ou seja, proibi-lo de executar qualquer outra ação diferente da ação “aspirar” no local e no período indicado e não proibi-lo de executar a ação “aspirar”.

Assim, utilizando como base estas equivalências lógicas apresentadas, foi concebida uma abordagem para o tratamento das normas obrigatórias. A responsabilidade pelo tratamento de normas de obrigação é feita pela organização. Considerando as relações entre os conceitos deônticos de obrigação e de proibição, a

abordagem considera que a organização deve definir apenas normas em termos de proibições e não proibições, como no caso acima, em que proibir o agente de executar a ação “não aspirar” é equivalente a proibir as execuções de ações diferentes de “aspirar”. Como consequência, se a ação não proibida “aspirar” for racional, isto é, se o local do ambiente contiver sujeira, o programa refinado selecionará esta ação; caso contrário, ou seja, o local estiver limpo, simplesmente não selecionará ações (não agirá), pois as outras ações possíveis ao aspirador estarão proibidas.

No contexto do programa de agente, as regras condição-ação do agente reativo simples normativo as regras foram reagrupadas em apenas dois grupos, ou seja:

- Grupo de Proibição: especifica as regras relacionadas com as ações que não podem ser executadas pelo agente.
- Grupo de Não Proibição: especifica as regras relacionadas com as ações que podem ser executadas pelo agente.

A Figura 2 apresenta o diagrama esquematizando a nova abordagem do agente reativo simples refinado para receber normas definidas por uma organização. Esta nova abordagem considera que o novo agente reativo simples normativo executa apenas aquelas ações que não são proibidas, mas que são racionais, da mesma maneira que um agente racional o faria.

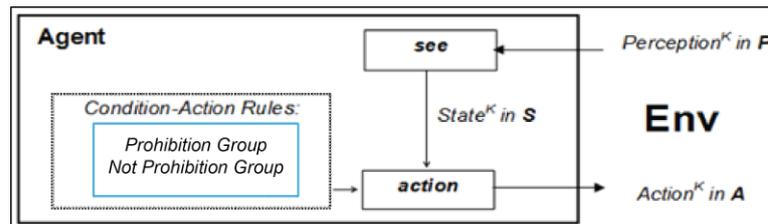


Figura 2. Diagrama esquematizado da nova abordagem

A nova função ação executa apenas dois passos, adaptados da sequência anterior composta de quatro passos:

(P1) primeiramente, a função ação busca por regras que pertencem ao grupo de regras proibidas, visando encontrar aquelas ações que não podem ser executadas; e

(P2) posteriormente, a função busca por regras no grupo de não proibidas visando encontrar alguma ação possível para o agente conforme as condições do ambiente de tarefa.

A Figura 3 apresenta o novo algoritmo da função agente reativo simples refinado em relação ao algoritmo descrito na Figura 1, que fundamenta a nova abordagem.

Algoritmo 2: Função agente reativo normativo da nova abordagem	
Entrada:	estado do mundo
Saída:	ação para os atuadores
Início	se existem regras condição-ação com normas de proibição ativa então inibir ações inferida pela regra;
	verificar a regra condição-ação retornar ação;
Fim	

Figura 3: Função agente reativo normativo da nova abordagem

Nesta nova abordagem, o agente não faz distinção entre regras de obrigação e regras de permissão, inserindo-as no grupo de regras não proibidas, evitando as ações desnecessárias. Desta forma, o agente reativo simples não sofrerá do problema das ações desnecessárias e irracionais ocasionadas pelas normas de obrigação. Porém as normas de obrigação devem ser todas transformação, pois a nova função de agente não possui o módulo de obrigação.

5. INCORPORANDO AS TRANSFORMAÇÕES NORMATIVAS NO PAPEL DO AGENTE

Usando as normas como leis sociais que são passadas por uma autoridade, nesta parte é apresentado o mecanismo de transmissão de normas para os agentes. De acordo com TAO+ [Freire et al., 2012], ilustrado na Figura 6, os agentes de um sistema multiagente normativo recebem as normas da organização em que estão vinculados através do papel de agente.

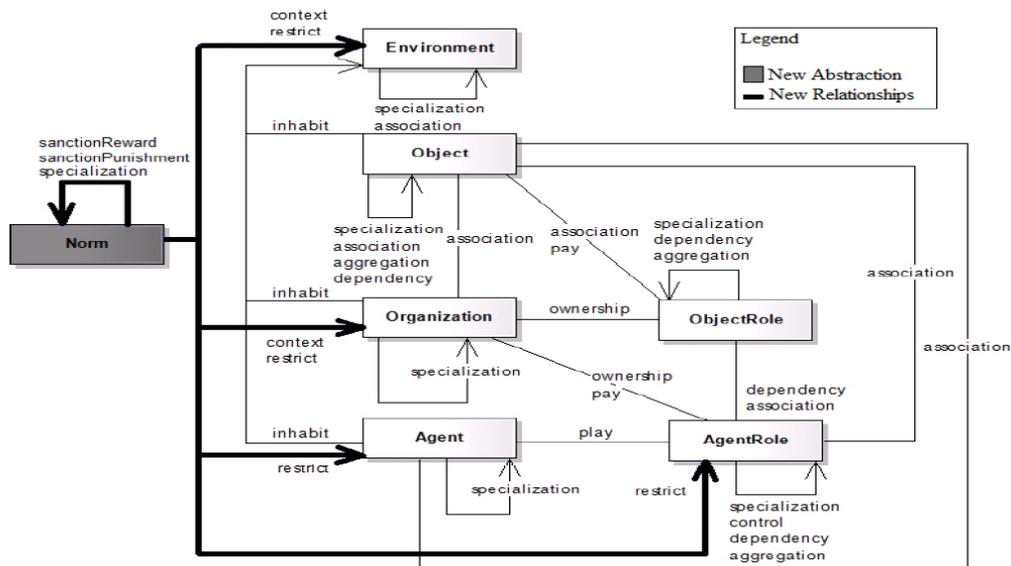


Figura 4. Relacionamento papel do agente normativo agente normativo no TAO+

Usando uma interface na organização como ponto de entrada para as normas em tempo de execução, foi concebido um mecanismo de automatização normativa que executa as transformações sobre as normas de obrigação. A Figura 5 descreve o

algoritmo que deve ser incorporado no papel do agente visando automatizar o processo de conversão de normas de obrigação em suas equivalentes proibitivas.

Algoritmo 3: Conversão das normas de obrigação em normas de proibição
Entrada: Lista de normas que restringem o agente Saída: Lista de normas contendo normas de permissão e proibição
<p>Início</p> <p>posição ← posição inicial da lista; enquanto a lista não for totalmente percorrida faça norma ← lista[posição]; se norma é uma norma de obrigação então transforma a norma em sua equivalente proibitiva; lista de saída ← normas equivalentes; senão lista de saída ← norma posição ← próxima posição;</p> <p>Fim</p>

Figura 5: Conversão das normas de obrigação em normas de proibição

O papel da organização é escolher qual dos grupos normativos apresentados na seção 2 e 4 que serão transmitidos para os agentes. A abordagem proposta considera que o processo de transformação é executado através de uma chamada de método inserido no papel do agente, cuja organização é responsável por escolher se o papel irá transmitir normas de obrigação ou apenas normas de proibição e permissão.

Colocando o algoritmo de transformação no papel de agente, podemos escolher qual agente terá as normas de obrigação transformadas em um conjunto de normas de proibição e permissão, de acordo com parâmetros que são enviados pela organização. Isto é, esta abordagem possibilita o usuário do sistema determinar quais agentes irão executar ações sobre normas de obrigação utilizando o processo descrito no Algoritmo 3, e deixando íntegra a arquitetura do agente, possibilitando uma liberdade de escolha para o usuário.

6. ESTUDO DE CASO

Esta seção apresenta algumas simulações realizadas considerando um ambiente de tarefa semelhante a uma das versões do mundo do aspirador de pó simplificado [Russell e Norvig, 2004]. As simulações visaram comparar o desempenho dos programas de agentes descrito nos algoritmos das figuras 2 e 4, concretizados usando a linguagem de programação Java. A Figura 5 ilustra o estado inicial do ambiente nas simulações.

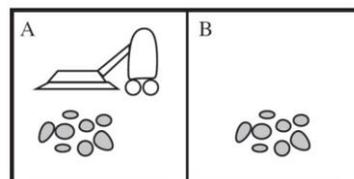


Figura 5. Situação inicial da simulação

Neste mundo simplificado existem apenas duas salas A (*roomA*) e B (*roomB*), que podem conter sujeira (Sujo) ou não (Limpo). Os agentes têm percepção local, ou seja, percebem a sala em que está e o estado da sala. Os agentes não conhecem a

configuração inicial do ambiente. Em qualquer momento, os agentes podem executar uma de três ações possíveis: aspirar (*suck*), mover para a esquerda (*left*), mover para a direita (*right*). Os agentes podem decidir por não executar qualquer ação entre aquelas que são possíveis (*no_op*). Conforme o estado inicial na figura acima indica, as duas salas estão inicialmente sujas e os agentes começam na sala A.

A medida de avaliação de desempenho estabelecida para os agentes: (1) recompensa com um ponto a mais quando o agente limpa uma sala; (2) penaliza com um ponto a menos quando o agente movimenta-se entre salas; (3) recompensa com três pontos a mais quando o agente cumpre uma norma; (4) penaliza com um ponto a menos quando o agente executa uma ação desnecessária e irracional. Três normas foram estabelecidas para os agentes: (N1) “Obrigado aspirar sala A das 10:00 às 12:00”; (N2) “Proibido mover-se para a esquerda das 13:00 às 15:00”; e (N3) “Proibido mover-se para a direita das 13:00 às 15:00”.

A Tabela 1 apresenta dez episódios registrados na história do agente aspirador de pó no ambiente de tarefa, programado de acordo com o Algoritmo 1, ou seja, considerando o primeiro refinamento.

Tabela 1. Desempenho do Algoritmo 1

	Estado			Ação	Pontos	Acumulado
	Onde está?	Estado roomA	Estado roomB			
1	<i>roomA</i>	Sujo	Sujo	<i>suck</i>	1	1
2	<i>roomA</i>	Limpo	Sujo	<i>right</i>	-1	0
3	<i>roomB</i>	Limpo	Sujo	<i>suck</i>	1	1
4	<i>roomB</i>	Limpo	Limpo	<i>left</i>	-1	0
5	<i>roomA</i>	Limpo	Limpo	<i>suck</i>	3 - 1	2
6	<i>roomA</i>	Limpo	Limpo	<i>suck</i>	3 - 1	4
7	<i>roomA</i>	Limpo	Limpo	<i>right</i>	-1	3
8	<i>roomB</i>	Limpo	Limpo	<i>no_op</i>	3	6
9	<i>roomB</i>	Limpo	Limpo	<i>no_op</i>	3	9
10	<i>roomB</i>	Limpo	Limpo	<i>left</i>	-1	8

Do primeiro ao quarto episódio, o Algoritmo 1 comportou-se racionalmente, obtendo o valor máximo possível de desempenho. Entretanto, o mesmo não acontece no quinto e no sexto episódios. Nestes episódios o agente foi recompensado com três pontos pelo cumprimento da norma de obrigação N1, mas também foi penalizado com um ponto a menos por ter aspirado em uma sala que estava limpa, consumindo desnecessariamente a energia disponível. Por sua vez, com a ativação das normas de proibição N2 e N3 nos últimos episódios, novamente o Algoritmo 1 apresentou um comportamento racional, obtendo o valor máximo de pontos possíveis por ter cumprido com as proibições descritas em N2 e N3.

Quanto ao registro do aspirador de pó programado de acordo com o Algoritmo 2 no mesmo ambiente de tarefa, o segundo refinamento propõe que a organização evite a declaração de normas envolvendo o conceito deôntico de obrigação, transformando-as em normas envolvendo os conceitos deônticos de proibição e permissão (não proibição). Assim, a norma de obrigação N1 deve ser transformada pela organização em novas normas equivalentes envolvendo o conceito deôntico de proibição. A

transformação da norma N1, que envolve o conceito deôntico de obrigação e, conseqüentemente, deve ser transformada pela organização em uma ou mais normas equivalentes envolvendo o conceito deôntico de proibição, pode ser melhor compreendida ressaltando-se duas etapas principais do processo de transformação:

(E1) considerando N1, empregando-se a equivalência $O p \leftrightarrow F \neg p$, obter como equivalente lógica a norma (N1') "Proibido não aspirar sala A das 10:00 às 12:00"; e

(E2) considerando N1', empregando-se a implicação $\forall x, x \neq y : F \neg y \rightarrow (F x \wedge \neg F y)$, obter como conseqüente lógico as normas: (N1.1) "Proibido ir para esquerda da sala A das 10:00 às 12:00", (N1.2) "Proibido ir para direita da sala A das 10:00 às 12:00", e (N1.3) "Proibido não operar na sala A das 10:00 às 12:00".

A Tabela 2 apresenta os dez episódios registrados na história do agente com o Algoritmo 2. Vale ressaltar, além do grupo de regras de proibição relacionadas às normas N2 e N3, esta etapa das simulações considerou também as outras regras de proibição, relacionadas com as três normas de proibição obtidas com a transformação da norma N1 nas normas N1.1, N1.2 e N1.3.

Tabela 2. Desempenho do Algoritmo 2

	Estado			Ação	Pontos	Acumulado
	Onde está?	Estado roomA	Estado roomB			
1	roomA	Sujo	Sujo	suck	1	1
2	roomA	Limpo	Sujo	right	-1	0
3	roomB	Limpo	Sujo	suck	1	1
4	roomB	Limpo	Limpo	left	-1	0
5	roomA	Limpo	Limpo	no_op	3	3
6	roomA	Limpo	Limpo	no_op	3	6
7	roomA	Limpo	Limpo	right	-1	5
8	roomB	Limpo	Limpo	no_op	3	8
9	roomB	Limpo	Limpo	no_op	3	11
10	roomB	Limpo	Limpo	left	-1	10

Semelhante ao que aconteceu com o agente que incorporou o Algoritmo 1, o agente que incorporou o Algoritmo 2 comportou-se racionalmente do primeiro ao quinto episódio, obtendo o valor de desempenho igual ao obtido pelo primeiro agente. Da mesma maneira, com a ativação das normas N2 e N3, no oitavo e nono episódios, o segundo agente obteve a mesma pontuação que o primeiro. O mesmo aconteceu no último episódio da história. Entretanto, o mesmo padrão de comportamento não se manteve no quinto e no sexto episódio. Nestes episódios o agente foi recompensado com três pontos pelo cumprimento da norma N1. Note que ao não executar ações nestes episódios (no_op), evitou duas penalizações sofridas pelo primeiro agente, que estava obrigado pela norma N1 a aspirar uma sala limpa. Assim nos episódios 5 e 6 todas as ações do agente estão proibidas, com exceção da ação de aspirar que está permitida.

Vale ressaltar que as normas N1.1, N1.2 e N1.3 estavam ativas durante a ocorrência destes episódios e o agente foi projetado para não operar, pois a única ação *suck* era desnecessária e irracional, visto que a sala A já estava limpa. Além de economizar energia, é importante notar que ao final da simulação, no décimo episódio, o programa agente que empregou o Algoritmo 2, obteve melhor desempenho que o agente que empregou o Algoritmo 1, mas isso se deve ao processo de transformação

empregado na norma N1 empregado pela organização antes de passar as normas para o agente. Isto se deve por causa da norma de obrigação N1 estar mal definida, pois caso a norma N1 fosse “Obrigado aspirar a sala A das 10:00 às 12:00, caso ela esteja suja” o agente da primeira arquitetura não executaria as ações desnecessárias e irracionais.

7. CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho propõe uma solução para evitar a execução de ações desnecessárias e irracionais geradas por normas mal definidas. Para isso, é feito o tratamento automatizado das normas de obrigação em suas equivalentes lógicas de proibição pelo papel do agente na organização à qual os agentes reativos simples normativos pertencem. Com a automatização e encapsulamento das transformações das normas de obrigação nas suas equivalentes proibitivas, a conversão normativa torna-se mais uma opção para se trabalhar com agentes normativos, podendo ou não ser executada dependendo da quantidade de recursos disponíveis no ambiente de trabalho e dos objetivos da organização. Notou-se também que usando apenas os conjuntos normativos de proibição e não proibição, ambos os agentes alcançaram as mesmas pontuações.

Como trabalho futuro, faz-se necessário uma forma de automatizar o mecanismo regulador para perceber quando algumas normas podem estar mal definidas, pois como o mecanismo regulador irá atuar em relação a tais normas, os critérios para sancionar usados pelo regulador podem prejudicar o desempenho de agentes de um sistema multiagente normativo.

REFERÊNCIAS

Campos, G. A.; Freire, E. S. S.; Cortés, M. I., **Norm-based behavior modification in reflex agents**. Em: 14th International Conference on Artificial Intelligence (ICAI), Las Vegas, Nevada, USA, Proceedings of the 14th International Conference on artificial Intelligence, 2012.

Christopher D. Hollander, Annie S. Wu, **Using the process of norm emergence to model consensus formation**. Em: 5th International Conference on Self-Adaptive and Self-Organizing Systems, Ann Arbor, MI, 2011.

Cruz, F. I. S, **Modificação do comportamento da arquitetura de agentes reativo simples baseado em normas – Uma implementação em JAMDER 2.0**. Trabalho de Conclusão de Curso (TCC), Universidade Estadual do Ceará, Ceará, Fortaleza, 2013.

Emmanuel S. S. Freire, Mariela I. Cortés, Enyo J. T. Gonçalves, Yrleyjânder S. Lopes, **Extending the Framework TAO with Norms for MultiAgent Systems**. Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações (WESSAC), Florianópolis, 2012.

Figueiredo, K.; Silva, V. T., **NormML: A Modeling Language to Model Norms**. Em: 1st Workshop on Autonomous Software Systems. Salvador, Brasil, 2010.

Figueiredo, K., **Modeling and Validation Norms in Multi-Agents Systems**. Dissertação (Mestrado em Computação). Universidade Federal Fluminense, Rio de Janeiro, Niterói, 2011.

Gonçalves, E. J. T. **Modelagem de arquiteturas internas de agentes de software utilizando a linguagem MAS-ML 2.0**. Dissertação de Mestrado. Universidade Estadual do Ceará. Centro de Ciência e Tecnologia. Fortaleza, 2009.

Grossi, D.; Gabbay, D.; Torre, L. van der, **The Norm Implementation Problem in Normative Multi-Agent Systems**. Em: Specification and Verification of Multi-agent Systems, p. 195-224, 2010.

Meneguzzi, F.; Luck, M., **Norm-based behaviour modification in BDI agents**. Em: 8th International Foundation for Autonomous Agents and Multiagent Systems, Budapeste, Hungria, Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, Volume 1, p. 177–184, 2009.

Meyer, J. J.; Wieringa, R. J., **Deontic logic in computer science: normative system specification**, John Wiley and Sons, Vrije Universiteit, Amsterdam, The Netherlands, 1991.

Modgil, S.; Faci, N.; Meneguzzi, F.; Oren, N.; Miles, S.; Luck, M., **A framework for monitoring agent-based normative systems**. Em: 8th International Conference on Autonomous Agents and Multiagent Systems, Budapeste, Hungria, Proceedings of the 8th International Foundation for Autonomous Agents and Multiagent Systems. Volume 1, p. 153–160, 2009.

Rocha Jr., R. M.; Freire, E. S. S.; Cortés, M. I., **Estendendo o Framework JAMDER para Suporte à Implementação de Sistemas Multi-Agente Normativos**. Em: IX Simpósio Brasileiro de Sistemas de Informação (SBSI), João Pessoa, Brasil. Anais do IX Simpósio Brasileiro de Sistemas de Informação (SBSI). Volume 1, p. 839-850, 2013.

Russell, S.; Norvig, P., **Inteligência Artificial: uma abordagem moderna**. 2ª Ed. São Paulo: Prentice-Hall, 2004.

Shoham, Y., **Agent-Oriented Programming**. Em: Artificial Intelligence, Volume 60, p. 51–92, 1993.

Wagner, G., **The Agent-Object-Relationship Meta-Model: Towards a Unified View of State and Behavior**. Information Systems, v. 28, n.5, p. 475–504, 2003.

Wright, G. H. von., **Deontic Logic System**. Em: Mind: New Series, Volume 60, Número 237, p. 1–15, 1951.