

Avaliando Agentes Reativos Simples em Ambiente de Tarefas Parcialmente Observáveis e Normativos

Levi Figueiredo, Gustavo Campos e Mariela Cortés

Abstract— Humanos necessitam de leis que regulem as suas ações. Da mesma forma, quando aplicamos normas em sociedades de agentes, percebemos que tende a aumentar o desempenho do sistema segundo critérios que o projetista quiser. A proposta é avaliar o efeito que um tipo de norma de proibição de ações pode ter no desempenho de agentes reativos simples e agentes baseados em modelos com regras condição-ação, em ambientes de tarefas parcialmente observáveis. Estes agentes têm como princípio inibir aquelas regras, em seus conjuntos de regras condição-ação, que contenham alguma ação proibida como consequente. Os agentes demonstraram capacidade de entender estas normas específicas e evitar a realização de ações proibidas.

Index Terms— Humanos, Leis, Normas, Sociedade, Agentes, Métrica, Reativos, Modelo e Proibição

I. INTRODUÇÃO

A longo da história humana, naturalmente, foi percebida a necessidade de orientar comportamentos humanos a fim de que o limite de um não ultrapasse o limite do outro. Tendo necessidade desta orientação de comportamento, foram criadas normas dentro de grupos humanos. A princípio estas normas apareceram como regras informais socialmente impostas. No entanto, como o passar do tempo, estas normas foram institucionalizadas em diversas organizações, em geral, determinado um comportamento esperado para cada uma de um conjunto de situações específicas [1].

Da mesma forma que as normas são atualmente aplicadas às organizações humanas, é possível aplicar esta noção em uma sociedade de agentes artificiais visando controlar o comportamento dos agentes e aumentar o desempenho do sistema, de acordo com os critérios que o projetista desejar. Realmente a noção de normas vem sendo empregado como meios eficazes para o alcance desta finalidade para normalizar seus comportamentos [2]. Sistemas multiagentes normativos podem ser definidos como um sistema onde contém vários agentes organizados por meio de mecanismos para representar, comunicar, distribuir, detectar, criar, modificar, impor e deliberar sobre normas, além de detectar violação de normas, assim como seu cumprimento [8]. Mais especificamente, um agente artificial contém um programa de computador, aqui denominado programa do agente, que têm a capacidade

de receber informações perceptivas do ambiente por meio dos sensores, selecionar e enviar uma ação para que sejam executadas pelos atuadores do agente no ambiente. Diz-se que o programa é racional se fizer a coisa certa, de acordo com a medida de avaliação de desempenho estabelecida pelo projetista [9]. Russell e Norvig [9] destacam as estruturas de quatro tipos de programas para auxiliar no projeto de agentes artificiais: Agentes reativos simples, agentes baseados em modelo, agentes orientados a objetivos e agentes orientado a utilidade.

Este trabalho foca em dois tipos de programas de agentes: os reativos simples e os reativos baseados em modelo. Em resumo, um programa de agente reativo simples seleciona suas ações baseando-se na informação perceptiva atual que chega de seus sensores e em um conjunto de regras condição-ação, ignorando o histórico de percepções prévias. Visando lidar com a observabilidade parcial do ambiente, um programa baseado em modelos mantém em memória uma descrição de estado de seu ambiente, o qual depende do histórico das percepções, e seleciona suas ações baseando-se nestas informações e em um conjunto de regras condição-ação.

Conceber agentes que sejam racionais para funcionar em ambientes parcialmente observáveis não é uma tarefa trivial. Existem várias propostas desenvolvidas para o problema em diversos ambientes de tarefas específicos [3,4,6]. A maioria das propostas foca no desenvolvimento de mecanismos de atualização de estados internos para, primeiro, reduzir o conjunto de estados possíveis compondo os estados de crença do ambiente de tarefas do agente e, posteriormente, selecionar as ações levando-se em consideração informações em regras, metas e/ou funções utilidades [9].

Este artigo propõe investigar o efeito que um tipo de norma de proibição de ações específico pode ter no desempenho, em ambientes parcialmente observáveis, dos programas de agentes reativos simples e baseados em modelos e regras condição-ação. Os programas avaliados são capazes de entender estas normas e evitar a realização de ações proibidas por normas ativadas no ambiente de tarefas. Estes programas incorporam um princípio muito semelhante ao desenvolvido em outros trabalhos que tratam do assunto, ou seja, inibem aquelas regras no conjunto de regras condição-ação que contenha alguma ação proibida em seus consequentes [3,4,6].

O artigo foi organizado em mais três seções. Primeiramente, a Seção 2 vem descrever o refinamento para o processamento de normas das arquiteturas focadas neste artigo. A Seção 3 apresenta a metodologia usada para avaliar o desempenho dos programas. A Seção 4 apresenta os resultados obtidos de quatro experimentos em quatro tipos de ambientes diferentes,

L. Figueiredo, M. Cortés e G. Campos pertencem ao laboratório de Padrões e Qualidade de Software, localizado na Universidade Estadual do Ceará.
24 de Julho, 2017

variando a visibilidade dos sensores dos agentes, arquitetura e a presença de normas no ambiente. A Seções 5 apresenta algumas conclusões e os próximos passos.

II. ARQUITETURA DOS AGENTES NORMATIVOS

A norma deste trabalho será descrita de acordo com o trabalho de Meneguzzi e Luck [7] e Oren et al. [11]. A formalização da norma é da seguinte maneira:

Norma: (Ativação,Expiração,Tipo de Norma)

A ativação representa a condição para que a norma ative, a expiração é a condição para a norma desativar e o tipo de norma podem ser quatro: Proibido agir, obrigado a agir, proibido alcançar um certo estado e obrigado a alcançar um certo estado.

Para que o agente seja capaz de perceber e raciocinar com normas enviadas pela organização em tempo de execução de tarefas, sua arquitetura interna foi refinada. Os agentes deste trabalho possuem as melhorias em seu programa de agente segundo trabalhos anteriores [3,4,6]. Os programas que foram refinados são: reativos simples e reativos baseados em modelo.

A. Normas em Agentes Reativos Simples

Russel e Norvig [9] formalizam o comportamento de agentes reativos simples sendo capazes de coletar um conjunto de percepções do ambiente, mapea-las em um conjunto de estados possíveis e, através de um conjunto de regras condição-ação, mapear em um conjunto de ações possíveis para que o atuador possa selecionar uma dentre estas.

Figueiredo e Silva [5,10] vem falar que os sistemas multiagentes normativos são caracterizados pela inserção de proposições normativas envolvendo os conceitos deonticos de obrigação, proibição e permissão. Dentro destes três tipos de normas, Campos et al. [4] divide as regras condição-ação nestes três grupos e propõe um raciocínio para selecionar estas regras.

Primeiramente, faz-se uma busca por regras que pertencem ao grupo de regras obrigatórias, para procurar aquelas ações que devem ser feitas e não são proibidas. Se existir ações proibidas, a função inibe a regra de obrigação e busca por regras de proibição para achar ações que não são proibidas e podem ser realizadas de acordo com as condições do ambiente. Se não existe uma regra proibida, a função seleciona a ação que deve ser realizada como mostrado pela norma de obrigação. Finalmente, caso não exista uma regra de obrigação (nem de proibição), a função procura por regras no grupo de permissão que corresponda a alguma ação que pode ser feita de acordo com o estado do ambiente.

Já no trabalho de Aragão et al. [3], foi percebido que os agentes podem executar ações irracionais e desnecessárias nos casos em que ações obrigatórias (não proibidas), visto que a lógica de tomada de decisão no programa de agentes simples anterior não distingue entre ações obrigatórias que melhoram ou que pioram o desempenho do agente, em função de algum recurso que poderá estar sendo consumido desnecessariamente com a execução da ação.

Para que este problema seja resolvido, Aragao et al. [3]

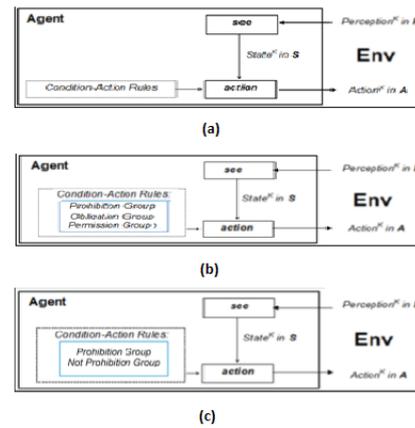


Figura 1. Evolução do programa de agente reativo simples para raciocinar com as normas.

refinou a arquitetura para que o agente raciocine entre somente dois grupos de normas: proibidas e não proibidas. A organização ficaria responsável de converter os três tipos de normas citados anteriormente para estes dois tipos e enviar para o agente processar.

Nestes dois trabalhos de refinamento para agentes reativos simples, a mudança ocorre somente nas regras de condição-ação com o objetivo de inibir ações indesejadas, como podemos ver na Figura 1. A Figura 1 (a) vem mostrar o programa dos agentes reativos simples sem nenhum refinamento. Na Figura 1(b), está refinado para suportar os três grupos da deontica, segundo Campos et al. [4] e na Figura 1(c), para os dois grupos mapeados da deontica, segundo Aragão et al. [3].

B. Normas em Agentes Reativos Baseados em Modelo

A formalização do comportamento do programa de agente, segundo Russel e Norvig [9], dos baseados em modelo é bastante semelhante aos sementes reativos. A diferença consiste no acréscimo da função próximo, no qual coleta o estado anterior, os efeitos das ações do agente no ambiente e a mudança do ambiente independente das ações. Primeiramente, os sensores coletam um conjunto de percepções do ambiente. Estas percepções são mapeadas em estados que representam a informação perceptível que não são acessíveis ao agente. Logo depois, a função próximo mapeia estas percepções em um novo conjunto de estados. Estes estados são mapeados em ações que irão ser selecionadas pelos atuadores. Por último, o estado interno é atualizado.

Para o refinamento dos agentes reativos simples, foram considerados somente dois dos quatro tipos de normas propostas por Meneguzzi e Luck [7]: Proibido agir e obrigado a agir. Para os agentes reativos baseados em modelo, é considerado todos os quatro tipos de normas, acrescentando proibido alcançar um certo estado e obrigado a alcançar um certo estado.

As mudanças no programa do agente baseado em modelo está na função próximo e nas regras condição-ação. A função próximo irá ser modificada para ser capaz de processar normas referentes a estados alcançados. Toda norma referente a estado

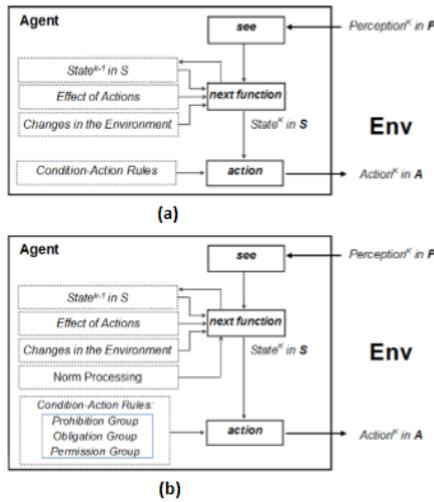


Figura 2. Evolução do programa de agente reativo baseado em modelo para raciocinar com as normas.

alcançado irá ser transformada em proibição ou obrigação de ações, de forma temporária, até o próximo ciclo de processamento. Por exemplo, tendo com base no ambiente do aspirador de pó, normas como "Coordenada (3,3) é obrigada a ficar limpa" seriam transformadas em normas temporárias como "Obrigado a limpar a Coordenada (3,3)".

Estas normas temporárias e as permanentes (seriam todas referentes a proibição e obrigação de ações que não passaram por transformações) irão agora selecionar as regras de condição-ação, segundo o refinamento de Campos et al. [4] para agentes reativos simples. A Figura 2 vem mostrar o programa de agentes reativos baseado em modelo sem (a) e com (b) o refinamento.

III. METODOLOGIA

O ambiente de tarefas proposto para realizar os experimentos consiste em uma representação de uma casa com cinco compartimentos, três superiores e dois inferiores, separados por cinco paredes. Em todos os experimentos, em média seis locais de cada compartimento foram marcados como contendo sujeira. O ambiente é estático, isto é, uma vez que o local foi limpo, não volta a conter sujeira. A Figura 3(a) mostra a geografia do ambiente 1. A Figura 3(b) apresenta o ambiente 2. A Figura 3(c) representa o ambiente 3 e a Figura 3(d) descreve o ambiente 4 de tarefas dos aspiradores de pó.

Os agentes aspiradores de pó foram representados por meio de círculos coloridos, nos cinco compartimentos da casa. Os sensores de cada agente permitem a percepção de informações de um ou dois locais nas adjacências do local onde o agente está. Os atuadores de cada agente permitem a realização de movimentos e de limpeza da sujeira do ambiente. Se um agente perceber um local contendo sujeira, ele limpa o local e se movimenta. Se o local estiver limpo, o agente se movimenta para outro local.

O programa de agentes reativo simples seleciona de maneira aleatória as ações que causam os movimentos para cima,

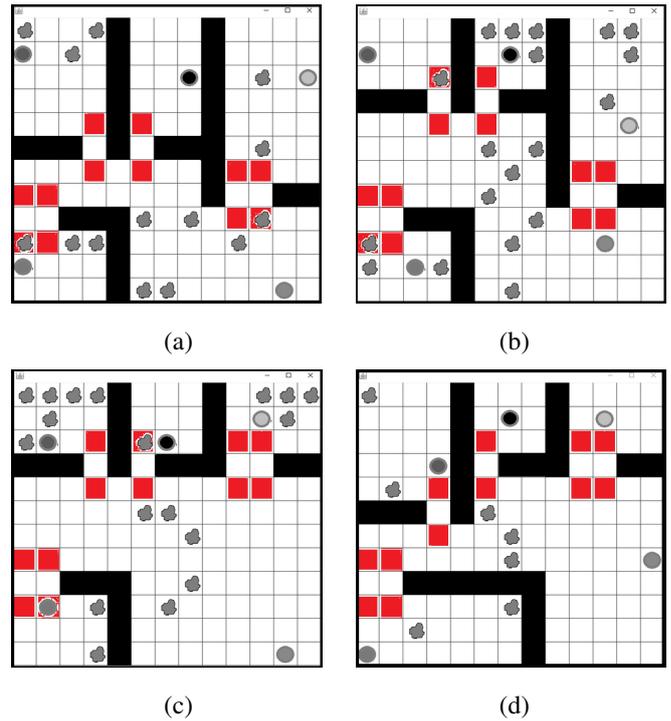


Figura 3. Ambientes de tarefa com a localização em vermelho dos pontos de ativação da norma de proibição.

baixo, direita ou esquerda. O programa de agentes reativos com estado interno armazena as posições por onde passa empregando uma fila encadeada limitada. Este programa seleciona suas ações dando prioridade aquelas ações que levam os agentes a locais que ainda não foram visitados ainda, ou seja, que não estão na sua memória interna. Inicialmente, foram realizados experimentos visando comparar os desempenhos dos dois tipos de programas de agentes citados, nas situações em que os sensores disponibilizam informações apenas sobre um e dois locais adiantes. Posteriormente, os experimentos foram realizados visando perceber o efeito de um tipo específico de norma de proibição no desempenho dos programas. Mais especificamente, estas normas são ativadas no ambiente sempre que algum agente estiver diante de uma porta, proibindo-os de sair dos compartimentos em que estão.

A condição de ativação será quando o agente se encontra na coordenada em vermelho na Figura 3. Em relação a condição de término, será no logo após o instante do movimento. Mais precisamente, esta norma irá inibir as ações: movimentar para cima e movimentar para baixo, fazendo com que a movimentação somente seja para esquerda ou para direita caso a norma esteja ativa, dessa maneira, a permanência no compartimento é alcançada. Tanto para agentes reativos como baseados em modelo, funcionará da mesma maneira, já que consideramos o tipo de norma sendo proibição de ação, descartando normas relacionadas ao alcance de estados e, conseqüentemente, criação de normas temporárias. A formalização é dada da seguinte forma:

$$\text{Norma: (Coordenadas}(X_v, Y_v), \text{Instante Atual, Proibido}(\text{MovimentoDireita}, \text{MovimentoEsquerda}))$$

Foram selecionados resultados referentes a quatro situações: agentes reativos simples com raio de visão de até um quadrado ao seu redor (visão 1), agentes reativos com estado interno e visão 1, agentes reativos simples com visão 2 e agentes reativos com estado interno e visão 2 para cada um dos 4 ambientes. A avaliação de desempenho levou em consideração o número de passos realizados por cada um dos agentes, significando que quanto menor o número, mais rápido foi a limpeza do ambiente. Cada programa foi executado vinte vezes e os valores médios foram obtidos.

IV. RESULTADOS INICIAIS

A Figura 4 apresenta a avaliação de desempenho dos agentes em cada um dos ambientes. As duas primeiras barras apresentam a média de passos dos **agentes reativos com o raio de visão 1** (A.R com Visão 1). As duas barras posteriores apresentam a média dos **agentes reativos com raio de visão 2** (A.R com Visão 2). A quinta e a sexta, apresentam a média de passos dos **agentes com estado interno e raio de visão 1** (A.B.M. com Visão 1). As últimas duas barras apresentam a média dos **agentes com estado interno e raio de visão 2** (A.B.M. com Visão 2). Para cada um dos quatro tipos de agentes, mediu-se a média de passos no ambiente com e sem normas (S/N e C/N).

Na Figura 4(a), a média de passos executados pelos agentes reativos simples com visão de raio 1 foi de 231.4, sem a norma, e de 143.35 com a norma, aumentando o desempenho do SMA em 0.61 vezes. Estendendo o raio de visão destes agentes para 2, a média obtida foi de 121 sem a norma e de 67.85 com a norma, tendo um aumento no desempenho de 0.78 vezes. Nos agentes baseados em modelo, com o raio de visão 1, a média de passos obtida, sem a norma, foi de 104.45 e, com a norma, de 42.9, alcançando um aumento de desempenho de 1.43 vezes, já com o raio de visão 2, a média de passos antes da norma foi de 29.45 e depois de 36.75, tendo uma diminuição no desempenho de 0.2 vezes.

A Figura 4(b) apresenta resultados obtidos dos quatro tipos de agentes no ambiente 2. Nos agentes reativos simples, com visão 1, sem a norma, a média de passos foi de 190.8, já introduzindo a norma, é obtida a média de 126.9 passos, alcançando um aumento de desempenho de 0.5 vezes. Aumentando o raio de visão para 2, a média de passos sem a norma fica de 147.4 e, com a norma, de 93.55, tendo um aumento no desempenho do SMA de 0.57 vezes. Já nos agentes com estado interno, com visão de raio 1, a média de passos, antes da introdução da norma, foi de 116.45 e, depois, de 64.75, aumentando o desempenho em 0.79 vezes. Estendendo o raio de visão para 2, destes agentes, a média de passos foi de 75.6 sem a norma e de 82.75 com a norma, diminuindo o desempenho em 0.09 vezes.

A Figura 4(c) vem mostrar os resultados obtidos no ambiente 3. Com os agentes reativos simples com visão de raio 1, sem a norma, a média ficou de 158.5 e, com a norma, de 151.65, aparecendo o aumento no desempenho de 0.04 vezes, já com o raio de visão 2, sem a norma, foi de 79.1 e, com a norma, de 44.25, tendo o aumento de desempenho de 0.78 vezes. Nos agentes com estado interno, os resultados das

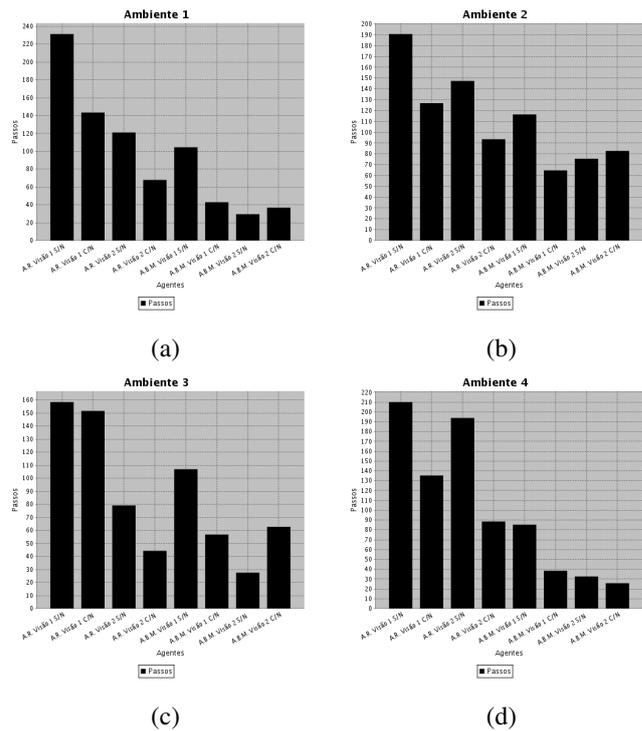


Figura 4. Gráficos com os resultados das médias de cada ambiente em cada caso.

médias para o raio de visão 1 foi de 106.95 e 56.8 sem e com a norma, respectivamente, obtendo o aumento de desempenho de 0.88 vezes. Estendendo o raio de visão destes agentes para 2, a média de passos sem a norma ficou de 27.5 e, com a norma, de 62.7, diminuindo o desempenho em 0.57 vezes.

Na Figura 4(d), o ambiente testado é o ambiente 4. Em agentes reativos, com o raio de visão 1, sem a norma, a média de passos ficou de 210.05, com a norma, de 135.35, tendo o aumento de desempenho de 0.55 vezes. Estendendo a visão de raio para 2, sem a norma a média foi de 193.85 e, com a norma, foi de 88.5, obtendo o aumento de desempenho de 1.19 vezes. Nos agentes dotados de estado interno, com o raio de visão 1, sem a norma, a média foi de 85.25 e, com a norma, de 38.4, conseguindo o aumento de desempenho de 1.22 vezes. Para o raio de visão 2 destes agentes, a média foi de 32.65 sem a norma e de 25.75 com a norma, tendo o aumento no desempenho de 0.26.

V. CONCLUSÃO

Ao testar os diversos tipos de agentes nestes quatro ambientes distintos, com e sem normas, é possível retirar conclusões importantes. Os agentes dotados de mecanismos para raciocinar com as normas, em sua grande maioria, tiveram um ganho de desempenho em ambientes parcialmente observáveis comparando aos agentes que não possuem a arquitetura refinada para reconhecimento e processamento de normas. Foi percebido que esse ganho de desempenho foi mais alto nos agentes menos robustos (reativos simples com uma pequena visão extra) em todos os ambientes de tarefa testados. Caso o ambiente seja mais heterogêneo, ou seja, os

compartimentos tendem a ter tamanhos diferentes, o ganho é pouco ou até tenha perdas de desempenho.

Visto isso, como trabalho futuro, mecanismos de comunicação entre os agentes irão ser implementados a fim de que um agente tenha influência no outro nas tomadas de decisão. Os agentes tomariam decisões de obediência ou não às normas baseados em suas atividades realizadas, assim como influências exteriores vindo de mensagens. Desta forma, quando um agente terminasse de limpar seu compartimento, poderia ajudar os outros compartimentos a serem limpos, evitando assim, desperdício de movimentos em compartimentos já limpos. Além disso, seriam definidas novas normas, como obrigação e permissão que não estão presentes neste trabalho.

AGRADECIMENTOS

O presente trabalho foi parcialmente suportado pelo Conselho Nacional Científico e Tecnológico, Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico e pela Universidade Estadual do Ceará.

REFERÊNCIAS

- [1] A. Ahmad, “An agent-based framework incorporating rules, norms and emotions”(OP-RND-E) [Ph.D. thesis], Universiti Tenaga Nasional, Selangor, Malaysia, 2012.
- [2] M. Alberti, A. S. Gomes, R. Gonçalves, J. Leite and M. Slota, “Normative systems represented as hybrid knowledge bases,” in *Computational Logic in MultiAgent Systems: Proceedings of the 12th International Conference on Computational Logic in Multi-Agent Systems (CLIMA '11)*, vol. 6814 of Lecture Notes in Computer Science, pp.330–346, Springer, Berlin, Germany, 2011.
- [3] P. Aragão, G. Campos, M. Cortés, F. Cruz, “Uma abordagem para o tratamento racional de normas de obrigação em ambientes de tarefas normativas” [*Autosoft Review*], Universidade Estadual do Ceará, Ceará, 2014.
- [4] G. Campos, E. Freire, M. Cortés, “Norm-based behavior modification in reflex agents”. In: *14th International Conference on Artificial Intelligence (ICAI)*, Las Vegas, Nevada, USA, Proceedings of the 14th International Conference on Artificial Intelligence, 2012.
- [5] K. Figueiredo, “Modeling and Validation Norms in Multi-Agents Systems”. Dissertação (Mestrado em Computação). Universidade Federal Fluminense, Rio de Janeiro, Niterói, 2011.
- [6] E. Freire, G. Campos, M. Cortés, W. Vasconcelos, “An approach for norm-based behavior modification in model-based reflex agents” .In: *15th International Conference on Artificial Intelligence (ICAI)*, Las Vegas, Nevada, USA, Proceedings of the 14th International Conference on Artificial Intelligence, 2013.
- [7] F. Meneguzzi and M. Luck, “Norm-based behaviour modification in BDI agents. In: *8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, 2009.
- [8] G. Boella and L. van der Torre, “Substantive and procedural norms in normative multiagent systems,” *Journal of Applied Logic*, vol. 6, no. 2, pp. 152–171, 2008. View at Publisher · View at Google Scholar · View at Scopus
- [9] S. Russel , P. Norvig, “Artificial Intelligence: a modern approach.” Prentice-Hall, New Jersey, 2004.
- [10] V. T. Silva, “Uma linguagem de modelagem para sistemas multi-agentes baseada em um framework conceitual para agentes e objetos”. Tese de doutorado. Rio de Janeiro: PUC, Departamento de Informática, 2004.
- [11] N. Oren, S. Panagiotidi, J. Vazquez-Salceda, S. Modgil, M. Luck, and S. Miles, “Towards a formalisation of electronic contracting environments”. In *Proc. 12th COIN Workshop*, pages 61–68, 2008.