Construção de Planos BDI a partir de Políticas Ótimas de POMDPs, com Aplicação na Programação em AgentSpeak usando o Jason

Diego R. Pereira, Luciano V. Gonçalves, Graçaliz P. Dimuro

[1]Programa de Pós-Graduação em Informática – Universidade Católica de Pelotas
Rua Félix da Cunha, 412 – 96.010-000 – Pelotas – RS – Brasil

{dpereira,llvarga,liz}@ucpel.tche.br

Abstract. *Based on the analisys of hybrid approach BDI-MDP found in the literature, this paper introduces the algorithm* ***policyToBDIplan*** *that builds AgentSpeak plans for BDI agents that obey optimal POMDP policies, presenting an application example using Jason. POMDP policies are mapped into BDI agents plans, following an intention, considering that plans extracted from a POMDP optimal policy are the ones adopted by the BDI agent that selects the plan with the greatest utility, and an optimal strategy reconsideration.*

Resumo. *Com base na análise da abordagem híbrida BDI-POMDP encontrada na literatura, este artigo introduz o algoritmo* ***policyToBDIplan*** *que constrói planos em AgentSpeak para agentes BDI que obedecem uma política ótima POMDP, apresentando um exemplo ilustrativo de sua aplicação utilizando o Jason. Políticas de POMDPs são mapeadas para planos de agentes BDI, de acordo com uma dada intenção, considerando que os planos derivados de uma política ótima de um POMDP são aqueles adotados pelo agente BDI que seleciona o plano com a maior utilidade, e uma reconsideração de estratégia ótima.*

1. Introdução

A arquitetura de agentes BDI (*Beliefs*, *Desires*, *Intentions*) [Wooldridge 2000, Rao and Georgeff 1992] considera agentes com conjuntos de crenças sobre o ambiente onde está inserido, e um conjunto de desejos que levam o agente aos seus objetivos. Através de um processo de deliberação, um agente BDI verifica quais destes desejos podem ser alcançados, selecionando-os como intenções, que são estados que o agente está comprometido a alcançar. Então, o agente constrói um plano para alcançar tal intenção, através de um processo de raciocínio meio-fim, e executa tal plano para alcançá-la.

Modelos de agentes BDI têm sido utilizados em aplicações diversas e complexas. Entretanto, como a abordagem BDI é heurística, o plano pode falhar, o mundo pode mudar ou a intenção pode não ser mais alcançável. Quando isto acontece, o agente normalmente passa por um processo de reconsideração de intenções para verificá-las se ainda são viáveis. Este processo geralmente diminui a sua performance, impossibilitando um comportamento ótimo, se comparado aos modelos de teoria da decisão. O balanço entre o tempo que o agente passa deliberando e o tempo que passa executando planos é crucial para um bom agente BDI.

A abordagem BDI não possui ferramentas para análise quantitativa sistemática (quer dizer, baseada em teoria matemática bem estabelecida) do desempenho dos agentes, principalmente quando se trata de ambientes onde ocorre incerteza sobre os resultados das ações e das percepções dos agentes.

Por outro lado, os modelos para a construção de agentes baseados em Processos de Decisão de Markov (MDP - *Markov Decision Processes*) [Puterman 1994] sempre oferecem uma solução ótima para os problemas colocados aos agentes. Salienta-se, entretanto, que a realização dos cálculos implicados pelos seus algoritmos só apresentam desempenho adequado quando o espaço de estados não é muito grande. Quando consideram-se modelos que representam ambientes reais, que produzem usualmente um espaço de estados muito grande, algumas vezes até parcialmente observáveis, o problema torna-se intratável, devido à própria natureza dos algoritmos usados para resolver o MDP.

Observa-se, entretanto, que agentes BDI podem resolver problemas que normalmente seriam intratáveis por modelos MDP, e podem ter um desempenho melhor do que Processos de Decisão Parcialmente Observáveis (POMDP - *Partially Observable Markov Decision Processes*) [Kaelbling et al. 1998, Lovejoy 1991], para problemas relativamente pequenos [Simari and Parsons 2006].

Modelos híbridos têm sido propostos para combinar as vantagens e superar as desvantagens dos dois modelos, existindo diversas abordagens diferentes que mostram a relação entre os dois modelos e como eles podem combinados para tratar problemas de naturezas diversas [Simari and Parsons 2006, Paruchuri et al. 2006, Nair and Tambe 2005]. Por exemplo, um agente BDI com planos baseados em POMDPs, ou um POMDP cuja política é construída a partir de planos BDI, têm sido apontados como soluções para melhorar o desempenho de agentes BDIs ou a tratabilidade de modelos POMDPs, respectivamente [Simari and Parsons 2006].

Neste artigo, avançando na proposta apresentada em [Pereira and Dimuro 2007], direcionada a modelos totalmente observáveis, discute-se a relação entre MDPs (e POMDPs) e arquiteturas BDI e apresenta-se um algoritmo para construção de planos em AgentSpeak para agentes BDI, extraídos de políticas de POMDPs, de tal forma que tais planos "obedeçam" essas políticas.

Este artigo está organizado da seguinte forma: na Seção 2, apresenta-se uma relação entre as descrições BDI e MDP; na Seção 3 discute-se a relação entre planos BDI e políticas ótimas de MDPs e POMDPs; na Seção 4 introduz-se o algoritmo para construção de planos BDI a partir de grafos de políticas de POMDPs; na Seção 5 apresenta-se um exemplo para estudo de caso; a Seção 6 é a Conclusão.

2. Relação entre as Descrições dos Modelos BDI e POMDP

Nesta seção, utiliza-se a notação adotada em [Simari and Parsons 2006], que, embora não seja padrão, simplifica as descrições BDI e MDP, de modo a estabelecer correlações entre seus elementos. Ambas as descrições consistem de um espaço de estados S, um conjunto de ações A, e uma função de transição T que depende do estado corrente e da ação a ser realizada. Assim, tem-se que:

Definição 2.1 *Um Processo de Decisão de Markov (MDP) é definido como uma tupla* (S, A, T, R, P, π)*, onde:*

- S *é o conjunto finito de estados;*
- A *é o conjunto finito de ações;*
- $T : S \times A \rightarrow \Pi(S)$ *é a função de transição de estados, que dada uma ação* $a \in A$ *sobre um estado* $s \in S$*, retorna uma distribuição de probabilidade sobre o conjunto de estados* S*;*
- $R : S \times A \rightarrow \mathbb{R}$ *é a função de recompensa, ou seja a recompensa por realizar uma ação* $a \in A$ *em um estado* $s \in S$*;*
- P *é uma distribuição inicial de probabilidades sobre o conjunto de estados;*
- $\pi : S \times A$ *é uma política, ou seja um mapeamento estado/ação.*

Um agente que usa a descrição MDP será chamado de agente MDP. Como o espaço de estados S é totalmente observável, P indica apenas o estado atual do agente com uma probabilidade de 100%, pois o agente sempre sabe o seu estado atual.

Definição 2.2 *Um Processo de Decisão de Markov Parcialmente Observável (POMDP) é definido como uma tupla* $< S, A, T, R, \Omega, O, P >$*, onde:*

- S, A, T, eR *são definidos como no MDP (Def. 2.1);*
- Ω *é um conjunto finito de observações que o agente tem do mundo;*
- $O : S \times A \rightarrow \Pi(\Omega)$ *é a função de observação que dá, para cada ação e estado resultante, uma distribuição de probabilidade sobre possíveis observações;*
- P *é uma distribuição inicial de probabilidades sobre o conjunto de estados.*

Na descrição parcialmente observável, considera-se que o agente possui um estado de crença, que depende da ação e da observação realizadas, bem como de seu estado de crença anterior, que atualizarão P.

Por outro lado, uma descrição BDI pode ser dada como:

Definição 2.3 *Uma descrição BDI é definida como uma tupla* $(S, A, T, B, D, I, Del, M)$*, onde* S*,* A *e* T *são, respectivamente o espaço de estados, o conjunto de ações e a função de transição (como na Def. 2.1), e*

- B *é o conjunto de crenças,* D *o de desejos, e* I *o de intenções;*
- Del *é o componente de deliberação;*
- M *é o componente de raciocínio meio-fim.*

Um agente que usa a descrição BDI será chamado de agente BDI.

Para um agente em um dado ambiente, considera-se que S, A e T são os mesmos para ambas as descrições (BDI e POMDP). Além disso, considera-se que B e P representam a mesma idéia, isto é, identificam em que estado o agente se encontra. Com estas equivalências definidas, tem-se de um lado recompensas e políticas, e de outro tem-se desejos, deliberação, raciocínio meio-fim, e intenções. Na verdade, como recompensas são meios de se determinar políticas, e desejos são um passo para se determinar intenções, estes componentes podem ser ignorados. Finalmente a relação que se considera com detalhes é a entre políticas e intenções.

A semântica da intenção de um agente varia de acordo com a literatura, os significados mais comuns são: "o foco atual do agente", um plano que o agente toma para alcançar um determinado estado. Neste artigo, intenção é o estado que o agente se comprometeu a alcançar, e portanto um plano BDI é uma seqüência de ações construídas para alcançar um determinado estado, ou seja para alcançar uma determinada intenção.

Em [Simari and Parsons 2006] foram estabelecidas diversas equivalências entre as descrições BDI e MDP, considerando que os agentes trabalham no mesmo espaço de estados. O resultado que interessa a este trabalho em particular é que, dada uma política, que tem uma ação para cada estado, é possível derivar um ou mais planos BDI, determinando uma trajetória através do espaço de estados. Portanto, pode-se dizer que uma política MDP incorpora um conjunto de planos BDI.

A utilidade esperada de um plano BDI pode ser obtida da mesma forma que uma política em um MDP, estabelecendo-se um valor para cada ação em cada estado em que é executada. A diferença é que em um plano BDI considera-se apenas uma seqüência no espaço de estados, e na avaliação de uma política consideram-se as ações em todos os estados.

A princípio, o agente BDI terá a mesma abordagem para atravessar o espaço de estados. Selecionará uma intenção, identificará um plano BDI para alcançar sua intenção e executará seu plano até perceber que seu plano não irá alcançar sua intenção ou que sua intenção não pode ser alcançada (ou não é a melhor intenção possível).

Compara-se este processo com um agente MDP (POMDP) no mesmo espaço de estados, no qual o agente sempre sabe, através de sua política, qual a melhor ação a tomar. A desvantagem é que calcular uma política ótima é mais custoso do que criar um plano simples. O preço pago pelo agente BDI é que este tem que computar o que fazer online enquanto o calculo de política do agente MDP (POMDP) pode ser feito offline e ainda seu plano pode ser sub-ótimo pois pode haver desvio em suas ações.

3. De Políticas para Planos BDI

Seja π uma política que é a solução para um MDP completamente especificado, e considera-se que π é ótima. Observa-se, entretanto, que de qualquer política π é possível extrair valores de utilidade para os estados que irão induzir π, e estes poderão ser usados para estabelecer planos BDI.

Uma seqüencia de ações é chamada de plano BDI se suas ações forem selecionadas com o objetivo de executá-las uma por vez em ordem para alcançar um dado objetivo. Um plano BDI *obedece* uma política π se, e somente se, as ações prescritas pelo plano BDI são as mesmas prescritas pela política através dos estados intermediários do plano BDI. Observa-se que assume-se que os planos BDI são lineares, que nenhuma consideração é feita devido a resultados inesperados de suas ações.

Assim, dado um agente BDI e um agente MDP com uma política ótima π, se o agente BDI está no estado inicial s, então um plano BDI com o maior valor de utilidade será aquele que obedece a π, começando em s. Em geral a afirmação somente se sustentará se estados com a mesma recompensa são considerados os mesmos nas duas abordagens, BDI e MDP. De outra forma, mesmo que as utilidades sejam equivalentes, as ações podem não ser exatamente as mesmas, pois a ordem em que os estados com a mesma recompensa são considerados pode afetar a seleção de ações [Simari and Parsons 2006].

Os resultados iniciais encontrados em [Simari and Parsons 2006] para relacionar políticas com intenções e planos BDI são estabelecidos para casos determinísticos e totalmente observáveis. Considera-se assim que: (i) o componente de deliberação Del é ótimo e sempre seleciona a intenção com a maior utilidade e que as ações são determinísticas

no ambiente; (ii) a política calculada é ótima e foi obtida através de algum algoritmo de resolução do MDP. Se o componente M do modelo BDI sempre selecionar o plano com o maior valor de utilidade e o ambiente for totalmente observável, então Del selecionará o estado com a maior utilidade, o mesmo estado selecionado pela política, e M irá escolher o plano BDI que percorre a mesma trajetória através do espaço de estados que a política recomenda. Portanto, assumindo-se que estados com recompensas iguais são considerados na mesma ordem pela política ótima π e pelo componente de raciocínio meio-fim M do modelo BDI, está claro que planos BDI gerados por M irão obedecer π [Simari and Parsons 2006].

Se as ações não forem determinísticas, a utilidade dos planos BDI não estará claramente definida. Ao invés de uma simples soma das recompensas através do caminho do plano, a falha nas ações deve ser considerada. Portanto, deve-se assumir que os componentes de deliberação Del e o raciocínio meio-fim M são ótimos sob um critério de máxima utilidade esperada, ao contrário de serem capazes de escolher a intenção e o plano BDI (respectivamente) com as maiores recompensas.

Pode-se, então, utilizar o mesmo argumento para ambientes determínisticos para afirmar que os planos BDI obedecem a uma política ótima, como pode ser visto em [Simari and Parsons 2006].

Estes resultados [Simari and Parsons 2006] mostram porque a abordagem BDI tem dificuldades para gerar um comportamento ótimo. Nos modelos BDI clássicos, a deliberação seleciona uma intenção e então a análise meio-fim constrói um plano para alcançar tal intenção. Para poder escolher um conjunto de ações ótimo, o componente de deliberação deve poder escolher a intenção que é ótima sob um critério de utilidade máxima esperada antes da análise de meio-fim escolha um plano BDI.

Analisa-se agora o caso em que o ambiente não é totalmente observável, em outras palavras, o agente não sabe em qual estado $s \in S$ ele está, e deve confiar em suas estimativas do estado atual do ambiente [Lovejoy 1991]. Modelos POMDP tratam este tipo de situação estendendo o conceito de estado. Ao invés de lidar com um estado que é $s \in S$, onde espaço de estados S descreve todos os estados no ambiente, um estado se torna uma probabilidade de distribuição sobre S, denotado por s'_i. Para enumerar todas as possíveis distribuições s'_i, então considera-se políticas e planos BDI que se preocupam com um novo espaço de estados $S' = \bigcup_i s'_i$.

Se S' é a contraparte parcialmente observável de S, e considera-se que S' é o novo espaço de estados onde os agentes BDI e MDP irão operar, então ambos B e P irão identificar algum $s' \in S'$ como estado atual do agente. Logo, pode-se então estender a afirmação de que planos BDI obedecem uma política [Simari and Parsons 2006].

4. Algoritmo para a obtenção de planos BDI a partir de uma política ótima

Neste trabalho, as políticas ótimas de POMDPs são obtidas através da aplicação do algoritmo *Witness* [Kaelbling et al. 1998].

Observa-se que para determinar uma política ótima seria necessário enumerar todos os estados de crenças e todas as distribuições de probabilidades de observações em cada estado de crença. Diferentemente de outras técnicas, o algoritmo *Witness* define regiões para o vetor de valores associados aos estados, e procura por um ponto onde este

vetor não é dominante. O algoritmo seleciona apenas uma ação de cada vez e tenta encontrar a melhor função de valor (utilidade) para cada uma das ações separadamente. De posse destes valores, o algoritmo combina os resultados e constrói a função de valor V'. Além de considerar uma ação de cada vez, também considera uma observação de cada vez. Um vetor em um ponto do estado de crença é construído a partir da seleção de um vetor transformado do vetor V para cada observação. O algoritmo começa com um ponto aleatório do estado de crença, e gera o seu vetor (de valor). Após o adiciona ao conjunto e considera que este vetor é V', e então tenta provar que este vetor é realmente o V'.

Na construção de um vetor, faz-se uma escolha para cada observação; esta escolha é a seleção de um dos vetores V representando uma estratégia futura. O algoritmo então procura pelas escolhas individuais feitas, observação por observação, para encontrar uma escolha diferente que possua um valor maior. Após, define uma região em que a sua escolha com certeza é a melhor. Se conseguir achar um ponto no estado de crença onde uma estratégia diferente seria melhor, então este estado de crença serve de testemunha (ou *"witness"*) para o fato de que a escolha de vetores não é o V' procurado.

A saída do algoritmo *Witness* é uma tabela, que é usada para construir o grafo de política. Como este grafo possui nodos que nunca serão visitados, é necessário ainda realizar a simplificação do grafo.

O algoritmo **policyToBDIplan** (apresentado em Alg. 1) recebe o grafo de política simplificado, e constrói regras para a plataforma *Jason* [Bordini et al. 2007] que constituirão planos para agentes BDI. O algoritmo se comporta da seguinte maneira:

- A entrada do algoritmo é o grafo de política (pg) gerado pelo algoritmo *Witness*;
- A saída é um conjunto de regras no formato *AgentSpeak* [Rao 1996];
- Para cada linha do arquivo que contém o grafo de política é extraída uma regra $(rule)$ e uma ação $(action)$, e para cada observação é extraída uma próxima regra. Nesta descrição tem-se um conjunto de n observações $R1, R2, ..., Rn$.

Algorithm 1 Algoritmo para extrair planos BDI de grafos de políticas POMDP

```
policyToBDIplan(Policygraph pg){
Policygraphline pl
BDIplan plan
BDIrule ruleR1, ruleR2, ... ,ruleRn
set pl to firstLine(pg)
repeat
    set ruleR1 to BDIrule(head={Node(pl)}, context={True}, body={act(Action(pl)), aGoal(Node(pl'))})
    set ruleR1 to BDIrule(head={Node(pl')}, context={ObsR1}, body={aGoal(R1(pl))})
    set ruleR2 to BDIrule(head={Node(pl')}, context={ObsR2}, body={aGoal(R2(pl))})
    ...
    set ruleRn to BDIrule(head={Node(pl')}, context={ObsRn}, body={aGoal(Rn(pl))})
    addBDIrules {ruleR1, ruleR2, ... , ruleRn} to plan
    advance ln
until ln beyond lastLine(pg)
return  Plan
}
```

5. Exemplo: O Problema do Tigre

Nesta seção, apresenta-se um exemplo da aplicação do algoritmo proposto a um exemplo clássico da literatura de modelos POMDPs, que é o *Problema do Tigre* [Kaelbling et al. 1998]. Neste exemplo, o agente é puramente reativo e o espaço

de estado é muito pequeno, não havendo vantagens em se utilizar uma descrição BDI para implementar o agente. Entretanto, a simplicidade do problema serve ao propósito de ilustrar o funcionamento do algoritmo.

O Problema do Tigre consiste de um agente em frente a duas portas. Atrás de uma existe um tigre, que irá atacá-lo e atrás da outra existe um baú com uma grande recompensa. Em vez de abrir uma porta, o agente pode apenas escutar para obter alguma informação sobre a localização do tigre. Mas escutar tem um preço e não é muito confiável. A descrição do POMDP para o Problema do Tigre é dada pela tupla $< S, A, T, R, \Omega, O >$, onde:

- $S = \{sl, sr\}$ é o conjunto de estados, onde sl representa o tigre à esquerda, e sr à direita;
- $A = \{listen, open - left, open - right\}$ é o conjunto das ações escutar, abrir a porta esquerda e abrir a porta direita, respectivamente;
- $T : S \times A \rightarrow \Pi(S)$ é a função de transição de estados, onde a ação $listen$ não causa transição no sistema e $open - left$ e $open - right$ reiniciam o problema;
- $R : S \times A \rightarrow \{-1, +10, -100\}$ é a função de recompensa, onde -1 é o custo de $listen$, $+10$ é a recompensa por abrir a porta sem tigre, e -100 é o custo de descobrir o tigre está atrás da porta que foi aberta;
- $\Omega = \{tl, tr\}$ é o conjunto de observações, onde tl indica que o agente escutou o tigre na esquerda, e tr que o escutou na direita;
- $O : S \times A \rightarrow \Pi(\Omega)$ é a função de observação que dá a probabilidade de se fazer uma observação o dado que o agente executou a ação a e chegou no estado s.

O estado de crença do agente, no início de cada rodada, é uma distribuição de probabilidades uniforme entre os estados, isto é, probabilidade de 0.5 para sl (tigre à esquerda) e 0.5 para sr (tigre à direita). Se o estado é sl, existe uma probabilidade de 0.85 de ocorrer a observação tl (escutar o tigre à esquerda), e de 0.15 de ocorrer a observação tr (escutar o tigre à direita). Se o estado é sr, existe uma probabilidade de 0.85 de ocorrer a observação tr, e de 0.15 de ocorrer a observação tl.

A política ótima para este problema, ou seja, o mapeamento observação/ação, pode ser descrita pelo grafo de política simplificado apresentado na Fig. 1 e pela Tab. 1, obtido com a utilização do algoritmo *Witness*. Neste caso a política diz que o agente deve escutar duas vezes o tigre do mesmo lado antes de abrir a porta, para aumentar a certeza de sua localização.

Tabela 1. Tabela do grafo de políticas para o problema do tigre.

regra	ação	Obs tl	Obs tr
0	0	3	1
1	0	0	2
2	1	0	0
3	0	4	0
4	2	0	0

O algoritmo `policyToBDIplans` obtém planos BDI descritos como as regras exemplificadas na Fig. 2. Por exemplo, na regra `0`, tem-se que, realizada a ação de escutar (`listen`), se a observação for `TR` então a próxima regra a ser considerada será

a regra `1` (`!State(1)`), caso contrário, se a observação for `TL` então a próxima regra a ser considerada será a `3` (`!State(3)`). Da mesma forma para a regra `1`, que se escutar `TR` mais uma vez levará o agente a regra `2`, em que ele deve abrir a porta (ação `act(open-left)`).

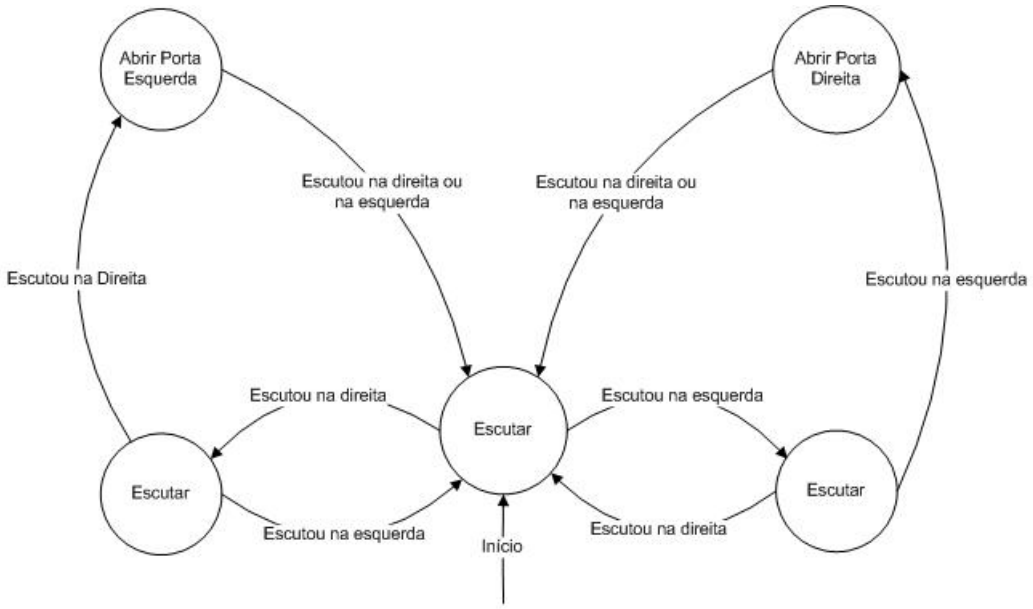

Figura 1. Grafo de Políticas

Para os testes foram realizadas 100 iterações do problema e calculada a média. Nos testes realizados, o agente acertou a porta correta em $79,5\%$ das vezes e errou $20,5\%$, dentro do esperado devido a probabilidade da observação estar nesta faixa (85% e 15%).

6. Conclusão e Considerações Finais

Este artigo analisou a relação entre as descrições BDI e MDP apresentada em [Simari and Parsons 2006], aprofundando a proposta de uma abordagem híbrida BDI-POMDP, onde políticas de POMDPs são mapeadas para planos de agentes BDI.

Para tanto, foi introduzido o algoritmo **policyToBDIplan** que constrói planos na linguagem *AgentSpeak* para agentes BDI, a partir de um grafo de política de um POMDP, obtido pela aplicação do algoritmo *Witness*. Para ilustrar o funcionamento do algoritmo, foi apresentado um exemplo de sua aplicação para o Problema do Tigre.

Uma aplicação imediata desta abordagem híbrida é na auto-regulação de trocas sociais em sistemas multiagentes baseados em personalidades [Dimuro et al. 2007], conforme proposto em [Pereira 2008]. O mecanismo de regulação de trocas busca manter o balanço das trocas equilibrados sem, entretanto, interromper a continuidade das interações. Quando este mecanismo é internalizado nos agentes, o processo de decisão sobre as trocas que um agente deve propor a seu parceiro na interação é parcialmente observável, pois um agente não tem acesso ao estado interno do outro agente. Uma abordagem híbrida BDI-POMDP foi adotada no simulador de trocas sociais implementado no Jason.

```
+!State(00): True ->
                act(listen),
                !State(02').
+!State(00'): obs==TL ->
                !State(3).
+!State(00'): obs==TR ->
                !State(1).

+!State(01): True ->
                act(listen),
                !State(01').
+!State(01'): obs==TL ->
                !State(0).
+!State(01'): obs==TR ->
                !State(2).

+!State(02): True ->
                act(open-left),
                !State(02').
+!State(02'): obs==TL ->
                !State(0).
+!State(02'): obs==TR ->
                !State(0).
```

Figura 2. Exemplo de Regras

Como trabalhos futuros, pretende-se avançar no estudo destes modelos híbridos, com o objetivo de desenvolver um algoritmo para obter planos BDI a partir de políticas ótimas para modelos de Processos de Decisão Fracamente Acoplados [Parr 1998, Meuleau et al. 1998], para a aplicação no desenvolvimento de um sistema multiagente de um "chão de fábrica", com o problema de alocação de recursos, processos e produtos.

Agradecimentos. Este trabalho é financiado pela Petrobrás (Projeto COPPETEC) e pelo CNPq (Proc. 473201/2007-0).

Referências

Bordini, R. H., Hübner, J. F., and Wooldrige, M. (2007). *Programming Multi-agent Systems in AgentSpeak Using Jason*. Wiley Series in Agent Technology. John Wiley & Sons, Chichester.

Dimuro, G. P., Costa, A. C. R., Gonçalves, L. V., and Hübner, A. (2007). Centralized regulation of social exchanges between personality-based agents. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, volume 4386 of *LNCS*, pages 338–355. Springer, Berlin.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134.

Lovejoy, W. S. (1991). A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28(1–4):47–66.

Meuleau, N., Hauskrecht, M., Kim, K.-E., Peshkin, L., Kaelbling, L. P., Dean, T., and Boutilier, C. (1998). Solving very large Weakly Coupled Markov Decision Processes. In *Proc. of the 15th Nat. Conf. on Artificial Intelligence, 10th Conf. on Innovative Applications of Artificial Intelligence*, pages 165–172, Menlo Park. AAAI Press.

Nair, R. and Tambe, M. (2005). Hybrid BDI-POMDP framework for multiagent teaming. *Journal of Artificial Intelligence Research*, 23:367–420.

Parr, R. (1998). Flexible decomposition algorithms for weakly coupled markov decision problems. In Cooper, G. F. and Moral, S., editors, *Proc. 14th Conf. Uncertainty in Artificial Intelligence, Madison*, pages 422–430. Morgan Kaufmann.

Paruchuri, P., Bowring, E., Nair, R., Pearce, J., Schurr, N., Tambe, M., and Varakantham, P. (2006). Multiagent teamwork: Hybrid approaches. In *Computer Society of India Communications*. CSI. (Invited Talk, avaliable at http://teamcore.usc.edu/publications.htm).

Pereira, D. R. (2008). Construção de planos BDI a partir de políticas ótimas de POMDPs, com aplicação na auto-regulação de trocas sociais em sistemas multiagentes. Dissertação de mestrado, PPGINF/UCPel, Pelotas, RS.

Pereira, D. R. and Dimuro, G. P. (2007). Um algoritmo para extração de um plano BDI que obedece uma política MDP Ótima. In *Anais do Workshop-Escola de Sistemas de Agentes para Ambientes Colaborativos*, Pelotas. PPGINF/UCPel.

Puterman, M. L. (1994). *Markov Decision Processes–Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York.

Rao, A. S. (1996). AgentSpeak(L): BDI agents speak out in a logical computable language. In van Hoe, R., editor, *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, volume 1038 of *LNCS*, pages 42–55. Springer, Berlin.

Rao, A. S. and Georgeff, M. P. (1992). An abstract architecture for rational agents. In Nebel, B., Rich, C., and Swartout, W. R., editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92), Cambridge, MA, October 25–29, 1992*, pages 439–449. Morgan Kaufmann.

Simari, G. I. and Parsons, S. (2006). On the relationship between MDPs and the BDI architecture. In Nakashima, H., Wellman, M. P., Weiss, G., and Stone, P., editors, *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*, pages 1041–1048. ACM.

Wooldridge, M. (2000). *Reasoning about Rational Agents*. Intelligent Robots and Autonomous Agents. The MIT Press, Cambridge, Massachusetts.