Extending deontic interpreted systems with action logic

Raquel de Miranda Barbosa, Antônio Carlos da Rocha Costa
Programa de Pós-Graduação em Modelagem Computacional - PPGMC
Programa de Pós-Graduação em Computação - PPGCOMP
Universidade Federal do Rio Grande (FURG)
Rio Grande - RS, Brasil
{raq.mbarbosa,ac.rocha.costa}@gmail.com

***Abstract*—This paper presents an extension to deontic interpreted systems with the use of action logic for the specification of normative aspects in multiagent systems. The paper presents a preliminary formalization of operators required and describes a simple example of the application of this formalization, using the segregation model (a simulation model available in NetLogo platform) through the formalization and proof of some properties of this system.**

***Keywords*—Deontic Interpreted Systems; Action Logic; Multiagent Systems; Norms**

I. Introdução

Aspectos formais são frequentemente considerados importantes no processo de desenvolvimento de software, visto que permitem a descrição de especificações não ambíguas e verificações e correções em diferentes etapas do desenvolvimento, entre outros aspectos. Quando se trata do desenvolvimento de sistemas baseados em agentes, não é diferente. Há uma grande preocupação na formalização de diferentes aspectos do sistema. Existem diversas teorias formais na área, as mais comuns baseadas em lógica, porém nem sempre fica claro o que tais teorias devem representar, como são aplicadas ao problema em questão e como podem ser integradas em um processo de especificação e verificação de sistemas.

Quando pensamos em sistemas multiagentes, devemos considerar dois aspectos importantes: i) seus aspectos organizacionais, que refletem a estrutura e funcionamento dos sistemas através da descrição de papéis de agentes, grupos, instituições/organizações e seus relacionamentos; e ii) a questão de normas (características regulatórias do sistema), que ditam como os agentes devem se comportar no sistema, o que eles devem ou não fazer e até mesmo casos nos quais os padrões são violados para se atingir um objetivo mais importante (cf. [1]). Um sistema multiagente que usa normas (ou baseado em normas) é chamado sistema multiagente normativo, conforme apresentado em [2]. De acordo com Boella [2], sistemas normativos são um exemplo de uso de teorias sociológicas em sistemas multiagentes, i.e., o relacionamento entre teoria de agentes e ciências sociais como Sociologia, Economia e Ciência Jurídica. Estes conceitos são importantes em sistemas multiagentes sociais, visto que existe o interesse no comportamento dos agentes tanto no nível individual como no social.

A representação de normas em SMA normalmente é feita através de fórmulas da lógica deôntica, utilizando-se operadores de obrigação (O-*obligation*), permissão (P-*permission*) e proibição (F-*forbidden*) e conceitos relacionados [3].

Um dos problemas encontrados é que a maioria das linguagens de especificação não usa operadores deônticos, dificultando o processo de verificação de compatibilidade do sistema em relação às suas normas.

Em [4] é apresentada uma formalização para a noção de conhecimento em sistemas distribuídos, onde, embora o foco seja sistemas distribuídos de processadores, estes "processadores" podem ser pensados como pessoas ou agentes, por exemplo. A ideia está baseada no modelo clássico de mundos possíveis (cf. [5]), onde diz-se que um agente conhece um fato φ se φ é verdadeiro em todos os mundos que ele considera possíveis; porém o autor salienta que em sistemas distribuídos não se tem um conhecimento global do sistema. Em contrapartida, é apresentada uma interpretação concreta para a noção de mundos possíveis, identificando o sistema como um conjunto de *runs*, onde um *run* é uma descrição completa do que acontece no sistema ao longo do tempo. São definidos pontos como sendo um par (r, t), consistindo de um *run* e um tempo e estes pontos do sistema são vistos como mundos possíveis. Em qualquer ponto o sistema está em algum estado global o qual pode ser visto apenas como uma tupla contendo os estados locais de cada processador.

Esta formalização deu origem à ideia de sistemas interpretados, (cf. [6] e [7]), onde adiciona-se uma função de interpretação para proposições que descrevem fatos do sistema e, posteriormente, foi estendida por Lomuscio [8] com a introdução de conceitos deônticos, permitindo a utilização de fórmulas com um operador similar ao de obrigação, mas que representa a ideia de um comportamento ideal/correto de um agente.

Neste contexto, este artigo apresenta uma aplicação destes operadores deônticos, definidos por Lomuscio e Sergot para a descrição de aspectos normativos em sistemas multiagentes, estendendo-os através do uso de lógica de ações. O artigo apresenta uma formalização preliminar dos operadores necessários e descreve um simples exemplo de aplicação desta formalização em um modelo de Segregação (cf. e.g [9]) através da prova de algumas propriedades deste sistema.

O modelo de Segregação é um modelo de simulação disponível na ferramenta NetLogo [10], inspirado nos artigos de Thomas Schelling sobre sistemas sociais [9]. Este modelo apresenta dois tipos de agentes que se dão bem uns com os outros em um determinado ambiente. Cada um deles quer ter certeza de que vive próximo de algum outro do mesmo tipo. A simulação mostra o que acontece em uma população com estas características [11].

O artigo está estruturado conforme descrito a seguir. Na Seção 2 é apresentada a teoria de sistemas deônticos interpretados, na qual este trabalho baseia-se. A Seção 3 descreve a extensão proposta para tratar sistemas multiagentes normativos. Na Seção 4 é apresentado um exemplo de utilização desta proposta, seguida pelas conclusões e trabalhos futuros.

II. Sistemas Deônticos Interpretados

A noção básica de sistemas interpretados foi introduzida em [7], onde um sistema é visto como um conjunto de *runs*. Se observarmos o sistema em qualquer ponto no tempo, cada um dos agentes está em algum estado (chamado estado local do agente), que encapsula toda a informação a qual o agente tem acesso. O sistema é dividido conceitualmente em dois componentes: agentes e ambiente (que representa tudo o que é relevante). O estado global de um sistema descreve o sistema em um dado ponto no tempo. O estado global com n agentes é uma tupla (n+1) da forma $(s_e, s_1, ..., s_n)$ onde s_e é o estado do ambiente e $s_1, ..., s_n$ são os estados locais do agente i. Considerando-se que os sistemas não são estáticos, define-se *run* como a descrição completa de como o estado global do sistema evolui no tempo (por exemplo, *r(0)* representa o estado global do sistema em uma possível execução 0).

Em [8], [12] descreve-se a adaptação desta noção para fornecer um fundamento básico para questões deônticas.

Para a caracterização de sistemas interpretados, assume-se um conjunto P de átomos proposicionais e Ag = $\{1, .., n\}$ de agentes.

A linguagem $\mathcal{L}$ é definida como segue:

φ ::= **false** | qualquer elemento de P | $\neg\varphi$ | $\varphi \wedge \varphi$ | $\mathcal{O}_i\varphi(i \in Ag)$

O operador modal indexado $\mathcal{O}_i$ é usado para representar circunstâncias de **funcionamento correto** do agente i onde: $\mathcal{O}_i\varphi$ pode ser lido como "em todas as alternativas de funcionamento corretamente possíveis do agente i, φ é o caso", ou "quando o agente i está funcionando corretamente (em relação a algum protocolo ou especificação), φ é o caso".

A fórmula φ pode se referir tanto a propriedades locais quanto globais ou a ambas ao mesmo tempo.

O operador modal $\mathcal{P}_i$ é o dual de $\mathcal{O}_i$ de forma que $\mathcal{P}_i\varphi =_{def} \neg\mathcal{O}_i\neg\varphi$. $\mathcal{P}_i\varphi$ pode ser lido como "em alguns dos estados nos quais o agente i opera corretamente φ ocorre", ou "φ acontece em algumas das alternativas de funcionamento correto do agente i".

A escolha do operador $\mathcal{O}$ deve-se ao fato de que sua semântica é similar àquela do operador de obrigação da lógica deôntica padrão. No entanto, não é correto ler $\mathcal{O}_i\varphi$ como "é obrigatório para o agente i que φ".

No exemplo da Segregação, podemos pensar que o comportamento correto do agente é "ser feliz", ou seja, ele está vivendo perto de, no mínimo, alguns vizinhos do mesmo tipo (se o agente não está feliz, ele deve mover-se para uma nova posição). O conhecimento que um agente pode ter em um determinado estado, por exemplo, é sua posição corrente, se ele está sozinho e se está feliz. Utilizando o operador $\mathcal{O}_i$ podemos escrever $\mathcal{O}_1 happy$ para representar a proposição "em todas as alternativas possíveis de funcionamento correto do agente 1, ele está feliz".

Sistemas Deônticos Interpretados de Estados Globais (*IDS - Deontic Interpreted Systems*) são definidos em [8], [12], da seguinte forma:

Um IDS para n agentes é um par $IDS = (DS, \pi)$, onde DS é um *sistema deôntico de estados globais* e π é uma *interpretação* para os átomos. Os autores não tratam a noção de tempo, desconsiderando *runs* e trabalhando apenas com estados.

Sistemas deônticos de estados globais são definidos, assumindo-se que, para cada agente, seu conjunto de estados locais pode ser dividido em estados permitidos e proibidos (chamados de estados verdes e vermelhos, respectivamente). Desta forma, dados n agentes e $n + 1$ conjuntos mutuamente disjuntos e não vazios G_e, G_1, ..., G_n, um *sistema deôntico de estados globais* é qualquer sistema de estados globais definido em conjuntos quaisquer $L_e \supseteq G_e$, $L_1 \supseteq G_1$, ..., $L_n \supseteq G_n$.

Um sistema de estados globais para n agentes S é um subconjunto não vazio de um produto cartesiano $L_e \times L_1 \times ... \times Ln$, sendo $L_1, ..., L_n$ os conjuntos de estados locais para cada agente do sistema e L_e o conjunto de estados para o ambiente.

G_e é chamado o conjunto de *estados verdes* para o ambiente e, para cada agente i, G_i é chamado o conjunto de estados verdes para o agente i. O complemento de G_e em relação a L_e (respectivamente, G_i em relação à L_i) é chamado o conjunto de *estados vermelhos* para o ambiente (respectivamente para o agente i).

Uma coleção de estados verdes e vermelhos identifica uma classe de estados globais. A classe de sistemas deônticos de estados globais é denotada por $\mathcal{DS}$.

No caso da Segregação, os estados verdes seriam aqueles em que o agente está sozinho em uma determinada posição e está feliz, enquanto os demais seriam considerados vermelhos.

Esta ideia de sistemas deônticos interpretados também é utilizada em outros artigos que descrevem pesquisas sobre a representação e raciocínio sobre estados de funcionamento correto e incorreto dos agentes e o sistema como um todo [8], verificação automática de sistemas deônticos interpretados através de *model checking* ([13], [14]), um sistema de *tableaux* para sistemas deônticos interpretados [15], entre outros.

III. Estendendo Sistemas Deônticos Interpretados com Lógica de Ações

Em trabalhos anteriores [16], foi investigada a possibilidade de utilização de métodos formais tradicionais de engenharia de software para a especificação de organizações de sistemas multiagentes (mais especificamente o método RAISE e sua linguagem de especificação RSL) e observou-se que esta linguagem, bem como as demais linguagens de especificação tradicionais não oferecem suporte ao uso de operadores deônticos para descrever características de regulação dos sistemas. No caso particular de sistemas multiagentes, é necessário (ou desejável) formalizar aspectos, tais como: o que o agente sabe em um determinado estado, o que ele pode fazer, o que acontece quando ele executa certas ações, entre outras coisas.

Para tratar estas questões, utilizando-se a noção de sistemas deônticos interpretados, é necessária uma extensão das definições propostas por [17], de forma a tratar os aspectos relacionados à realização de ações pelos agentes, permitindo a representação de situações do tipo "Em todas as alternativas de funcionamento correto do agente i, se ele executar a ação α, ele vai para um estado verde".

O modelo M é definido como uma tupla

$M = (G, \pi, \mathcal{R}_i^K, \mathcal{R}_i^O, \mathcal{R}_i^\alpha, \Sigma_A, P)$, sendo A um conjunto de ações e i = 1..n (agentes), onde:

- G é um conjunto de estados globais (i.e. g = $\langle l_e, l_1, ..., l_n \rangle$)
- π é uma interpretação que associa a cada estado em G uma atribuição de verdade para as proposições primitivas de P (i.e., $\pi(g) : P \rightarrow \{true, false\}$ para cada estado $g \in G$).
- $\mathcal{R}_i^K$ é uma relação binária sobre G (i.e., $\mathcal{R}_i^K \subseteq G \times G$), tal que para todo $g \in G$ existe pelo menos um $g' \in G$ com $g \rightarrow g'$, ou seja, $(g, g') \in \mathcal{R}_i^K$.
- $\mathcal{R}_i^O$ é uma relação binária sobre G (i.e., $\mathcal{R}_i^O \subseteq G \times G$), tal que para $g \in G$, $g\mathcal{R}_i^O g'$, se $l_i' \in \mathbf{G}_i$ (estados verdes do agente i).
- $\mathcal{R}_i^\alpha$ (onde $\alpha \in A$) é uma relação binária (de transições rotuladas) sobre $G \times G$ (i.e., $\mathcal{R}_i^\alpha \subseteq G \times G$), tal que para todo $g \in G$, $g\mathcal{R}_i^\alpha g'$ (escrita como g $\stackrel{\alpha_i}{\rightarrow}$ g').
- Σ_A é um conjunto de ações ou rótulos.
- P é um conjunto de proposições primitivas.

Considerando-se P um conjunto de proposições atômicas, A um conjunto de ações e $i \in \{1..n\}$ (agentes), o conjunto $\mathcal{L}$ é definido por:

1) $\varphi \in P$ implica $\varphi \in \mathcal{L}$
2) $\alpha \in A$ implica $\alpha \in \mathcal{L}_{Act}$
3) φ e $\psi \in \mathcal{L}$ implica $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi, \varphi \leftrightarrow \psi$ $\in \mathcal{L}$.
4) $\varphi \in \mathcal{L}$ implica $\mathcal{O}_i\varphi$, $\mathcal{P}_i\varphi$, $K_i\varphi \in \mathcal{L}$
5) $\mathbf{g}_i \in \mathcal{L}$
6) $\varphi \in \mathcal{L}$, $\alpha \in \mathcal{L}_{Act}$ implica $[\alpha_i]\varphi$, $\langle \alpha_i \rangle \varphi \in \mathcal{L}$
7) $\varphi \in \mathcal{L}$ implica $\varphi? \in \mathcal{L}_{Act}$
8) $\alpha, \beta \in \mathcal{L}_{Act}$ implica $\alpha; \beta$, $\alpha + \beta$, $\alpha* \in \mathcal{L}_{Act}$

Note que o operador ; significa composição sequencial, o operador + representa uma escolha não-determinística e o operador * é usado para uma representação finita arbitrária. A construção desta linguagem está baseada em [18].

A interpretação para os operadores $\mathcal{O}_i, \mathcal{P}_i, \mathbf{g}_i, K_i, [\alpha_i]$ e $\langle \alpha_i \rangle$ é dada da seguinte maneira:

- $\mathcal{O}_i\varphi$: em todas as alternativas possíveis de funcionamento correto do agente i, φ é o caso.
- $\mathcal{P}_i\varphi$: em alguns dos estados nos quais o agente i opera corretamente φ ocorre.
- $\mathbf{g}_i$: o agente i está em um estado local de funcionamento correto de acordo com seu protocolo.
- $K_i\varphi$: o agente i conhece φ.
- $[\alpha_i]$ φ: após a ação α ser executada pelo agente i, necessariamente se obtém φ.
- $\langle \alpha_i \rangle \varphi$: após a ação α ser executada pelo agente i, possivelmente se obtenha φ.

Para verificar a satisfação deste sistema deôntico interpretado de estados globais deve-se provar que:

$$M \models_g \varphi,$$

para todo $\varphi \in \mathcal{L}$.

Desta forma, tem-se que:

1) $M \models_g$ **true**
2) $M \models_g p$ se $g \in \pi(p)$
3) $M \models_g \neg\varphi$ se $\neg$ $M \models_g \varphi$
4) $M \models_g \varphi \wedge \psi$ se $(M \models_g \varphi)$ e $(M \models_g \psi)$
5) $M \models_g \varphi \vee \psi$ se $(M \models_g \varphi)$ ou $(M \models_g \psi)$
6) $M \models_g \varphi \rightarrow \psi$ se (não $M \models_g \varphi$) ou $(M \models_g \psi)$
7) $M \models_g \varphi \leftrightarrow \psi$ se $(M \models_g \varphi$ implica $M \models_g \psi)$ e $(M \models_g \psi$ implica $M \models_g \varphi)$
8) $M \models_g \mathcal{O}_i\varphi$ se, $\forall g'$, temos que $gR_i^O g'$ implica $M \models_{g'} \varphi$
9) $M \models_g \mathcal{P}_i\varphi$ se, $\exists g'$, tal que $gR_i^O g'$ implica $M \models_{g'} \varphi$
10) $M \models_g \mathbf{g}_i$, se $g \in R_i^O(g)$ $(i \in A)$
11) $M \models_g K_i\varphi$, se para todo g' temos que $gR_i^K g'$ implica $M \models_{g'} \varphi$
12) $M \models_g [\alpha_i]\varphi$, se $\forall g'$ temos que $gR_i^\alpha g'$ implica $M \models_{g'} \varphi$
13) $M \models_g \langle \alpha_i \rangle \varphi$, se $\exists g'$ tal que $gR_i^\alpha g'$ implica $M \models_{g'} \varphi$
14) $M \models_g [\alpha_i; \beta_j]\varphi$, se $\forall g', g''$ temos que $gR_i^\alpha g'$ e $g'R_j^\beta g''$ implica $M \models_{g''} \varphi$
15) $M \models_g [\alpha_i + \beta_j]\varphi$, se $\forall g'$ temos que $gR_i^\alpha g'$ implica $M \models_{g'} \varphi$ ou $gR_j^\beta g'$ implica $M \models_{g'} \varphi$
16) $M \models_g [\alpha_i^*]\varphi$, se $\forall g'$ temos que $gR_i^\alpha g'$ implica $M \models_{g'} [\alpha_i^*]\varphi$
17) $M \models_g \langle \alpha_i; \beta_j \rangle \varphi$, se $\exists g', g''$ tal que $gR_i^\alpha g'$ e $g'R_j^\beta g''$ implica $M \models_{g''} \varphi$
18) $M \models_g \langle \alpha_i + \beta_j \rangle \varphi$, se $\exists g'$ tal que $gR_i^\alpha g'$ implica $M \models_{g'} \varphi$ ou $gR_j^\beta g'$ implica $M \models_{g'} \varphi$
19) $M \models_g \langle \alpha_i^* \rangle \varphi$, se $\exists g'$ tal que $gR_i^\alpha g'$ implica $M \models_{g'} [\alpha_i^*]\varphi$

A fim de ilustrar esta formalização em um sistema multiagente é utilizado o Modelo de Segregação. Na Seção IV, este modelo é especificado como um sistema deôntico interpretado e algumas propriedades são formalizadas e verificadas.

IV. Exemplo: o Modelo de Segregação

O modelo de Segregação, inspirado nos artigos de Thomas Schelling sobre sistemas sociais [9] e disponível na ferramenta de simulação NetLogo [10], apresenta dois tipos de agentes (representados na simulação por tartarugas verdes e vermelhas) que convivem em um determinado ambiente. Cada um deles quer ter certeza de que vive próximo de algum outro do mesmo tipo (mesma cor). A simulação mostra o que acontece em uma população com estas características [11].

Este modelo pode ser formalizado como um sistema deôntico interpretado. Para isto deve-se fazer escolhas a respeito de como modelar os estados locais de cada agente e do ambiente. Aqui optou-se pelas seguintes alternativas (observa-se que as cores das tartarugas, definidas no modelo como verdes e vermelhas, foram alteradas para evitar a confusão com os estados verdes (desejáveis) e vermelhos (indesejáveis)):

- O estado local de cada agente é representado por uma tupla `(i,p,c,sn,on,sw)`, onde `i` indica o índice do agente, `p` indica a posição representada pelo par (linha,coluna), `c` a cor da tartaruga: branca (W) ou laranja (O), `sn` (*similar nearby*) indica o número de vizinhos de mesma cor, `on` (*other nearby*), indica o número de vizinhos de cor diferente e `sw` (*similar wanted*) o percentual de similaridade desejado para o agente.
- O estado local do ambiente é representado pela tupla `(tp,ta,OP,FP)`, onde `tp` (*total positions*) indica o número de posições existentes no ambiente, `ta` (*total agents*), indica o número de agentes da simulação, `OP` o conjunto de posições ocupadas e `FP` o conjunto de posições livres.

O conjunto de proposições atômicas que podem ser observadas neste exemplo é representado por:

P = {position$_i$(x,y), happy$_i$, alone$_i$, occupied(x,y) }

O conjunto de ações que os agentes podem realizar em cada estado é representado por:

Act = {ϵ, move$_i$(x,y) },

onde ϵ representa a falta de movimentação do agente, ou seja, ele permanece na mesma posição, e move$_i$(x,y), representa a movimentação do agente i para a posição (x,y) no ambiente.

Um estado correto para o agente i (**g**$_i$) é aquele em que o agente está feliz (i.e. com o percentual de vizinhos da mesma cor de acordo com o desejado) e sozinho na posição que ocupa.

O comportamento correto de um agente nesta simulação segue o seguinte protocolo:

$\boxed{\text{enquanto } \neg(happy)_i : move_i(x,y)}$

Isto significa que os agentes que ainda não estão em uma posição com o percentual desejado de vizinhos da mesma cor, devem saltar para outra posição aleatória até que fiquem "felizes". Uma alternativa de funcionamento correto, neste caso, é aquela em que, partindo do estado inicial, após alguns saltos, o agente chega a um estado correto (verde).

Considere o exemplo mostrado na Tabela I, onde o ambiente possui 9 posições e 4 agentes. Cada agente está representado pela sua cor (W-*white* ou O-*orange*) e um índice que o identifica.

Neste contexto, o estado Global Inicial do Sistema ($L_e \times L_i \times ... \times L_n$) é descrito como:

TABLE I. Exemplo de Segregação

W_1	W_2	
	O_3	
O_4		

g$_0$ = ((9,4, {(1,1),(1,2),(2,2),(3,1)}, {(1,3),(2,1),(2,3),(3,2),(3,3)}), (1,(1,1),W,1,1,50), (2,(1,2),W,1,1,50), (3,(2,2),O,1,2,50), (4,(3,1),O,1,0,50)).

Os possíveis estados locais de cada agente i podem ser divididos em estados desejáveis - verdes (G_i) e não desejáveis - vermelhos (R_i) e são representados como:

$G_i = \{(i,(x,y),sn,on,50)$ onde x e $y \in \{1..3\}\}$, sn e $on \leq 3$, para $i = 1..4$

$R_i = \{(i,(x,y),sn,on,50)$ onde x e $y \in \{1..3\}\}$, sn e $on \leq 3$, para $i = 1..4$

Uma possível configuração final do ambiente pode ser vista na Tabela II, cujo estado global seria representado por:

g$_n$ = ((9,4, {(1,1),(1,2),(3,1),(3,2)}, {(1,3),(2,1),(2,2),(2,3),(3,3)}), (1,(1,1),W,1,1,50), (2,(1,2),W,1,1,50), (3,(3,2),O,0,1,50), (4,(3,1),O,1,0,50))

onde pode-se observar que nos estados locais de todos os agentes o número de vizinhos de mesma cor é maior ou igual ao número de vizinhos de cor diferente, ou seja, todos estão felizes e, assim, não precisam deslocar-se para outro ponto.

TABLE II. Exemplo de Segregação

W_1	W_2	
O_4	O_3	

Para exemplificar este sistema deôntico, observe a Figura 1.

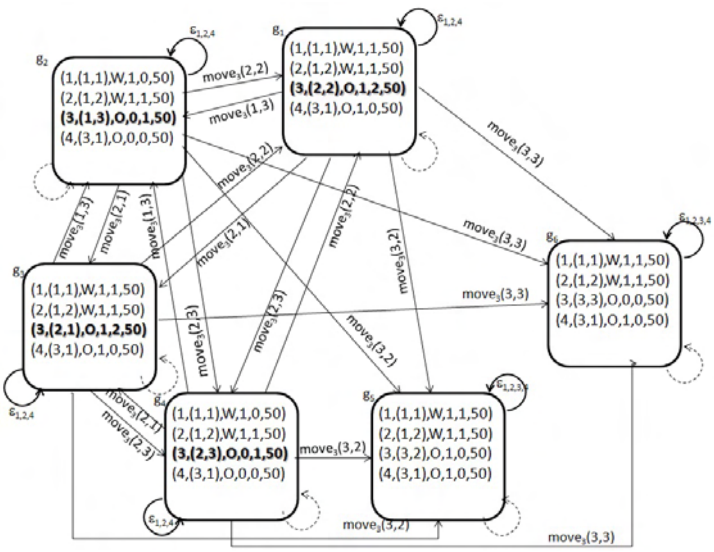

Fig. 1. Frame gerado a partir do sistema deôntico global.

Nesta figura o ambiente não é considerado e os estados locais para os agentes estão identificados dentro de cada estado global $g_1, ..., g_6$.

A Figura apresenta um subconjunto de SD que apresenta configurações para os agentes 1, 2, 3 e 4, respectivamente. Os links rotulados indicam as relações R_i^α (para i $\in \{1,2,3,4\}$), onde os rótulos indicam as ações executadas. Neste exemplo as ações de movimentação são apenas para o agente 3, visto que ele é o único que precisa mudar de posição para chegar a um estado final correto.

As setas tracejadas representam as relações de conhecimento R_i^K (para i $\in \{1,2,3,4\}$). Na figura, não estão representados os conhecimentos de cada agente, porém cabe salientar que em cada um dos estados globais, este refere-se a fatos como sua posição, cor, felicidade e se o agente está sozinho nesta posição. Por exemplo, considerando-se o agente 1 no estado global g_1 pode-se afirmar que ele conhece os fatos $position_1(1,1), color_1(W), happy_1$ e $alone_1$.

A relação $R_i^\mathcal{O}$ não está explícita na figura, porém pode ser explicada observando-se os estados locais dos agentes. Os agentes 1, 2 e 4 já estão em estados corretos em todos os estados globais analisados ($g_1,...,g_6$), portanto os estados g' alcançáveis para os agentes 1, 2 e 4, a partir de $g \in \{g_1,...,g_6\}$, são eles mesmos (e.g. $g_1 R_i^\mathcal{O} g_1$, para $i \in 1,2,4$). O agente 3 não está em estado correto nos estados g_1, g_2, g_3 e g_4, representado em negrito na figura. Por exemplo, no estado g_1 (configuração inicial do sistema apresentada na Tabela I, o agente 3 tem dois vizinhos W e apenas um vizinho O). Este agente somente vai obter um estado correto quando, após alguns movimentos, chegar a um dos estados g_5 ou g_6 (e.g $g_1 R_3^\mathcal{O} g_5$).

A. Propriedades

Com base no sistema deôntico especificado, é possível verificar algumas propriedades, tais como:

1) $M \models_{g_1} \mathcal{O}_1\ happy_1$: em todas as alternativas de funcionamento correto do agente 1, partindo do estado g_1, ele está feliz.

$$\begin{array}{c} M \models_{g_1} \mathcal{O}_1\ happy_1 \\ | \\ M \models_{g_1} happy_1 \\ | \\ g_1 \in \pi(happy_1) \\ g_1 \in \{g_1,g_2,g_3,g_4,g_5,g_6\} \checkmark \end{array}$$

2) $M \models_{g_1} \mathcal{O}_1(\mathbf{g}_1 \rightarrow happy_1 \wedge alone_1)$: em todas as alternativas de funcionamento correto do agente 1, partindo do estado g_1, se ele está em um estado verde, então ele está feliz e sozinho.

$$\begin{array}{c} M \models_{g_1} \mathcal{O}_1(\mathbf{g}_1 \rightarrow happy_1 \wedge alone_1) \\ | \\ M \models_{g_1} (\mathbf{g}_1 \rightarrow happy_1 \wedge alone_1) \\ | \\ \neg M \models_{g_1} \mathbf{g}_1 \ \vee \ M \models_{g_1} (happy_1 \wedge alone_1) \\ | \\ g_1 \in \pi(happy_1 \wedge alone_1) \\ g_1 \in \{g_1,g_2,g_3,g_4,g_5,g_6\} \checkmark \end{array}$$

3) $M \models_{g_1} K_1(position_1(1,1))$: o agente 1, no estado g_1, sabe que está na posição (1,1).

$$\begin{array}{c} M \models_{g_1} K_1(position_1(1,1)) \\ | \\ M \models_{g_1} (position_1(1,1)) \\ | \\ g_1 \in \pi(position(1,1)) \\ g_1 \in \{g_1,g_2,g_3,g_4,g_5,g_6\} \checkmark \end{array}$$

V. Conclusão

Este artigo apresentou uma extensão à abordagem de sistemas deônticos interpretados para a especificação de sistemas multiagentes, incluindo operadores relacionados às ações executadas pelos agentes. A ideia é que se possa utilizá-la para especificar organizações de sistemas multiagentes onde existem regras que regem o comportamento dos seus elementos.

Outros aspectos ainda necessitam ser desenvolvidos como, por exemplo, um método de cálculo para a validação de propriedades. Estão sendo desenvolvidas, pelos autores, regras para um sistema de tableaux que envolvam estes novos operadores de forma que se possa demonstrar a prova destas propriedades. Pretende-se aplicar estas definições em outros exemplos mais complexos e com aspectos organizacionais bem definidos e verificar a possibilidade de inclusão destes operadores em uma linguagem de especificação formal.

Agradecimentos

O trabalho conta com apoio financeiro do CNPq - Edital PDI, através do Projeto MSPP - Modelagem e Simulação de Políticas Públicas. Raquel de Miranda Barbosa é bolsista PosDoc/CAPES junto ao PPGMC/FURG.

Título em Português: Estendendo sistemas deônticos interpretados com lógica de ações.

References

[1] F. Dignum, "Autonomous agents with norms," *Artificial Intelligence and Law*, vol. 7, pp. 69–79, 1999.

[2] G. Boella and L. V. D. Torre, "Introduction to normative multiagent systems," *Computational and Mathematical Organization Theory*, vol. 12, pp. 71–79, 2006.

[3] G. H. von Wright, "Deontic logic," in *Mind*. Oxford University Press, 1951, vol. 60, no. 237, pp. 1–15.

[4] J. Y. Halpern, "Using reasoning about knowledge to analyze distributed systems," *Annual Review of Computer Science*, vol. 2, no. 1, pp. 37–68, 1987. [Online]. Available: http://www.annualreviews.org/doi/abs/10.1146/annurev.cs.02.060187.000345

[5] J. Hintikka, *Knowledge and Belief*. Ithaca: Cornell Univ., 1962.

[6] J. Y. Halpern, "Reasoning about knowledge: A survey," in *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford University Press, 1995, pp. 1–34.

[7] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning about Knowledge*. The MIT Press, Cambridge Massachusetts, 1995.

[8] A. Lomuscio and M. Sergot, "On multi-agent systems specification via deontic logic," in *Proceedings of ATAL 2001*. Springer Verlag, 2001.

[9] T. C. Schelling, *Micromotives and Macrobehavior*. New York: Norton, 1978, see also a recent Atlantic article: Rauch, J. (2002). Seeing Around Corners; *The Atlantic Monthly*; April 2002;Volume 289, No. 4; 35-48. http://www.theatlantic.com/issues/2002/04/rauch.htm.

[10] U. Wilensky, "Netlogo," http://ccl.northwestern.edu/netlogo/, Evanston,IL, 1999.

[11] ——, "Netlogo segregation model," http://ccl.northwestern.edu/netlogo/models/Segregation, Evanston,IL, 1997.

[12] A. Lomuscio and M. Sergot, “Extending interpreting systems with some deontic concepts,” in *Proceedings of TARK 2001*. Morgan Kauffman, 2001, pp. 207–218.

[13] F. Raimondi and A. Lomuscio, “Automatic verification of deontic interpreted systems by model checking via obdd's,” 2004.

[14] B. Wozna, A. Lomuscio, W. Penczek, and W. Penczek, “Bounded model checking for deontic interpreted systems,” in *Proc. of the 2nd Workshop on Logic and Communication in Multi-Agent Systems (LCMAS 04)*. Elsevier, 2004, pp. 93–114.

[15] G. Governatori, A. Lomuscio, and M. J. Sergot, “A tableux system for deontic interpreted systems,” in *AI 2003: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, T. Gedeon and L. Fung, Eds., vol. 2903. Berlin, Springer, 2003, pp. 339–350.

[16] R. M. Barbosa, “Especificação formal de organizações de sistemas multiagentes.” Ph.D. dissertation, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS, 2011.

[17] A. Lomuscio and M. Sergot, “Deontic interpreted systems,” *Studia Logica*, vol. 75, no. 1, pp. 63–92, 2003. [Online]. Available: http://dx.doi.org/10.1023/A%3A1026176900459

[18] J.-J. Meyer, “Dynamic logic reasoning about actions and agents,” in *LogicBased Artificial Intelligence*. Kluwer Academic Publishers, 2000, pp. 281–311.