

# Extensão da Sintaxe Abstrata da Linguagem MAS-ML para a Modelagem de Sistemas Multi-Agentes Normativos

Emmanuel S. S. Freire<sup>1</sup>, Enyo J. T. Gonçalves<sup>2</sup>, Mariela I. Cortés<sup>1</sup>, Yrleyjânder S. Lopes<sup>1</sup>

<sup>1</sup>Departamento de Computação – Universidade Estadual do Ceará (UECE)  
Avenida Paranjana, 1700 – 60740-000 – Fortaleza – CE – Brazil

<sup>2</sup>Departamento de Computação – Universidade Federal do Ceará (UFC)  
Estrada do Cedro, Km 5 – 63900-000 – Quixadá – CE – Brazil

{savio.essf, yrleyjander}@gmail.com, mariela@larces.uece.br, enyo@ufc.br

**Abstract.** *In Multi-Agent Systems (MAS), the agent's behavior is governed by norms, that their static elements need to be modeled at design time. In this context, the language MAS-ML (Multi-Agent System Language) provides several mechanisms for modeling of the properties of the entities that compose a MAS, but the support of the norm concepts is limited. This paper aims to extend the abstract syntax of the MAS-ML to support the modeling of MAS along with the static elements of the norms. Thus, the language will provide a more complete view of the system through a single modeling language.*

**Resumo.** *Nos Sistemas Multi-Agentes (SMAs), o comportamento de agentes é governado por normas, cujos elementos estáticos precisam ser modelados adequadamente em tempo de design. Neste contexto, a linguagem MAS-ML (Multi-Agent System Modeling Language) fornece diversos mecanismos para a modelagem das propriedades das entidades que compõem um SMA, porém o suporte aos conceitos das normas é limitado. Este artigo tem como objetivo estender a sintaxe abstrata de MAS-ML para dar suporte à modelagem de SMA considerando os elementos estáticos das normas. Com isso, a linguagem permitirá uma visão mais completa do sistema através de uma única linguagem de modelagem.*

## 1. Introdução

Os sistemas centrados em agentes [Padilha e Jacome, 2002] [Casillo, 2008] vêm sendo amplamente explorados pela comunidade científica como uma abordagem adequada para o desenvolvimento de sistemas computacionais complexos. No entanto, o desenvolvimento deste tipo de sistemas não é trivial, requerendo de um esforço de engenharia no intuito de dar suporte adequado às diversas atividades de desenvolvimento.

Segundo Russell e Norvig (2004), um agente de software é uma entidade capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores. Os agentes inteligentes podem ser classificados de acordo com a maneira que eles coletam informações e agem no ambiente. No caso de vários agentes cooperando ou disputando entre si, inseridos em um mesmo ambiente e trocando informações, chamamos esse sistema de multi-agente (SMAs).

De forma geral, agentes são inseridos em ambientes e, se relacionam com outros agentes para alcançar seus objetivos. Neste contexto, o comportamento dos agentes é governado por um conjunto de normas especificado para restringir suas ações. Com isso, a pesquisa em normas para SMAs tem como foco o comportamento de um agente ou grupo de agentes, definindo o que é obrigado, permitido ou proibido de fazer [Figueiredo e Silva, 2010]. As Normas podem ser definidas em tempo de *design* ou em tempo de execução. No entanto, a consideração de normas em tempo de modelagem propicia uma visão mais completa do sistema e pode influenciar na modelagem das entidades que integram o sistema. Adicionalmente, a verificação de conflitos pode ser realizada durante a fase de definição das normas e problemas detectados antes mesmo do sistema ser implementado podem ser solucionados [López Y López, 2003].

Dada a importância da modelagem de normas em forma concomitante com a modelagem das demais entidades que participam de um SMA, se torna indispensável à existência de linguagens de modelagem que possibilitem o projeto de SMAs com normas. Poucas são as linguagens de modelagem para SMAs que fornecem suporte à modelagem de normas em associação com as entidades previstas em um SMA. Dentre elas, a linguagem MAS-ML (*Multi-agent System Modeling Language*) [Silva, 2004] permite a modelagem de todas as entidades previstas em um SMA a partir dos conceitos definidos no *framework* conceitual TAO (*Taming Agents and Objects*) [Silva et. al., 2003], porém, o suporte à modelagem dos conceitos referentes a normas ainda é limitado. O objetivo desse artigo consiste na extensão da linguagem MAS-ML no intuito de dar suporte à modelagem de normas, com base no *framework* conceitual definido em [Figueiredo, 2011]. Com isso, além de possibilitar a modelagem de entidades de um SMA, também possibilita a modelagem das normas que serão utilizadas para guiar o comportamento destas entidades.

Este trabalho está organizado da seguinte maneira: a Seção 2 apresenta alguns trabalhos relacionados. Na Seção 3 é apresentado o referencial teórico, no qual é apresentada a linguagem de modelagem MAS-ML e os conceitos relacionados a normas para SMAs. Em seguida, na Seção 4, é descrita a extensão da linguagem MAS-ML proposta para modelar SMAs com normas. Um estudo de caso referente à extensão é mostrado na Seção 5 e, finalmente, as conclusões e trabalhos futuros são descritos na Seção 6.

## 2. Trabalhos Relacionados

Algumas linguagens de modelagem têm sido propostas para a modelagem das entidades que compõem um SMA, no qual o comportamento dessas entidades é regido por normas.

Wagner (2003) propõe a linguagem de modelagem AORML (*Agent-Object-Relationship Modeling Language*). As entidades definidas em AORML são: evento, ação, alegação, compromisso, agente ou objeto. Esta linguagem permite a modelagem dos três conceitos deônticos das normas, juntamente com ações não-comunicativas. Podemos destacar os seguintes pontos fracos: (i) não define ambientes, logo não é possível modelar agentes que estejam se movendo de um ambiente para outro; (ii) não permite a definição de normas para grupos de agentes; (iii) não oferece suporte para a definição de sanções.

A linguagem de modelagem para agentes AML (*Agent Modeling Language*) [Danc, 2008] é uma linguagem de modelagem visual semi-formal para especificação, modelagem e documentação de sistemas que incorpora os conceitos da teoria de SMAs.

As entidades definidas por AML são agentes, recursos e ambientes. Devido à necessidade de modelar aspectos sociais, esta linguagem utiliza conceitos de unidades organizacionais, relacionamento social, papéis e propriedades de papéis. AML possibilita descrever normas para um papel específico e para um grupo de indivíduos. Proporciona também, uma forma de restringir tanto ações comunicativas como não-comunicativas e a definição de um período no qual as normas ficarão ativas. Algumas desvantagens de AML são descritas a seguir: (i) não oferece suporte para a definição de sanções; (ii) não permite a definição de normas para um ambiente; (iii) não prevê a ativação de restrições pelo cumprimento ou violação de normas.

NormML (*Normative Modeling Language*) [Figueiredo, 2011] é uma linguagem de modelagem de normas para SMAs. Seu metamodelo é resultado da extensão do metamodelo da SecureUML [Basin, Doser e Lodderstedt, 2006]. Com a linguagem é possível modelar todos os elementos que compõem as normas (i) conceitos deonticos, (ii) entidades envolvidas, (iii) ações, (iv) restrições de ativação, (v) sanções e (vi) contexto. Assim, normas podem ser definidas para agentes, organizações, suborganizações, papéis de agente e ambiente, juntamente com seus períodos de ativação e sanções. NormML possui ferramenta de suporte para a modelagem de normas e para a checagem de conflitos entre as normas. Como a linguagem foca na modelagem das propriedades das normas, algumas deficiências podem ser levantadas: (i) não define as propriedades das entidades do sistema; (ii) não define a troca de papéis por um agente; (iii) não define elementos gráficos para as entidades definidas na linguagem; (iv) não permite a modelagem das entidades que fazem parte de um SMA, como por exemplo, do agente.

Silva (2003) propõe a linguagem MAS-ML cuja definição é baseada no framework conceitual TAO [Silva et al., 2003], possibilitando a modelagem de todas as entidades, suas propriedades e relacionamentos tipicamente encontrados em SMAs. A linguagem (i) possibilita a restrição das ações de um agente, de uma organização e/ou de uma suborganização por meio de um papel ativo ao qual o mesmo se encontra associado; (ii) permite a definição de normas de obrigação no contexto de organizações e suborganizações; (iii) possui ferramenta de suporte para a modelagem dos conceitos definidos na linguagem. Por outro lado, a linguagem apresenta algumas deficiências, tais como: (i) não permite a definição de normas aplicadas no contexto de um ambiente; (ii) não permite a definição de normas de proibição para papéis de agente; (iii) não permite a definição de normas de proibição e permissão para agentes, organizações e suborganizações; (iv) não prevê a modelagem de sanções nem de ativações das normas.

A partir da análise apresentada e considerando a necessidade de uma linguagem de modelagem que possibilite a modelagem de normas juntamente com as entidades previstas em um SMA, destacamos MAS-ML por (i) possuir uma ontologia adequada, (ii) modelar adequadamente ambientes e a interação entre agente e ambiente, (iii) apresentar suporte parcial à modelagem de alguns dos elementos que compõem normas para SMAs, e (iv) possuir ferramenta de suporte. Com isso, o metamodelo de MAS-ML se torna apto a receber os conceitos relacionados a normas previstos em Figueiredo (2010).

### 3. Referencial Teórico

Nesta seção, a linguagem MAS-ML, a definição e as características das normas são apresentados de maneira sucinta.

### 3.1. MAS-ML

MAS-ML [Silva, 2004] é uma linguagem de modelagem que estende a UML para permitir a modelagem de SMAs, com base no *framework* conceitual TAO [Silva et al., 2003]. A partir da ontologia definida no TAO a linguagem MAS-ML é capaz de oferecer suporte ao desenvolvimento de sistemas multi-agentes (SMAs) de larga escala.

Silva (2004) define os seguintes diagramas estáticos na linguagem MAS-ML: (i) diagrama de classes, estendido a partir da UML a fim de representar os relacionamentos entre as classes e outras entidades do SMA.; (ii) diagrama de organização, que tem como objetivo modelar todas as organizações de um sistema.; (iii) diagrama de papéis, que é responsável pela representação dos relacionamentos entre os papéis do agente e os papéis de objeto identificados nos diagramas de organização. Adicionalmente, em MAS-ML também foram definidos os diagramas de seqüência e de atividade, para representar os aspectos dinâmicos de SMAs, ou seja, representar as interações entre as instâncias do SMA e as ações de cada instância.

Para a modelagem dos diagramas definidos na linguagem, Farias et al. (2009) desenvolveram a ferramenta MAS-ML tool como um *plug-in* da plataforma Eclipse [ECLIPSE, 2011]. Isso implica que os usuários podem trabalhar com modelagem de SMAs ao mesmo tempo em que fazem uso dos recursos oferecidos pela plataforma Eclipse. Com esta ferramenta é possível a modelagem dos diagramas estáticos de classes e organização definidos em MAS-ML.

### 3.2. Normas para Sistemas Multi-agentes

As normas são utilizadas para restringir o comportamento de agentes, organizações e suborganizações durante um período de tempo, e definir as sanções aplicadas quando violadas ou cumpridas [Silva, Braga e Figueiredo, 2010]. A seguir são apresentados os principais elementos que compõem a norma com base no levantamento realizado por Figueiredo (2010) após a análise de especificações e linguagens de implementação utilizadas para descrever e implementar normas.

- **Conceitos deônticos:** a lógica deôntica refere-se à lógica de pedidos, ordens, regras, leis, princípios morais e julgamentos [Meyer e Wieringa, 1993]. Em SMA, tais conceitos têm sido utilizados para descrever as restrições de comportamento para os agentes na forma de obrigações (o que o agente deve executar), permissões (o que o agente pode executar) e proibições (o que o agente não pode executar).
- **Entidades Envolvidas:** desde que as normas são sempre definidas para restringir o comportamento das entidades, a identificação das entidades afetadas é fundamental. A norma pode regular o comportamento dos indivíduos (por exemplo, de um determinado agente, ou de um agente, enquanto está desempenhando um papel determinado), ou o comportamento de um grupo de indivíduos (por exemplo, todos os agentes desempenhando um papel determinado, grupos de agentes, grupos de agentes desempenhando papéis ou todos os agentes do sistema).
- **Ações:** uma vez que uma norma é definida para restringir a execução das entidades, é importante que a ação a ser regulamentada seja claramente especificada. Tais ações podem ser de comunicação, normalmente representado pelo envio e recebimento de uma

mensagem, ou ações não-comunicativas (como acessar e modificar um recurso, entrar em uma organização, se deslocar para outro ambiente, etc.).

- **Restrições de Ativação:** As normas têm um período em que suas restrições devem ser cumpridas, porém somente quando elas, as normas, estiverem ativas. As normas podem ser ativadas por uma restrição ou um conjunto de restrições que podem ser: a execução das ações, a especificação de intervalos de tempo (antes, depois, entre elas), a realização de estados do sistema ou aspectos temporais (como datas), e também a ativação / desativação de outra norma e cumprimento / violação de uma norma.
- **Sanções:** quando uma norma é violada a entidade responsável pode sofrer uma punição, e quando for cumprida uma norma, a entidade envolvida poderá receber uma recompensa. Recompensas e punições são chamadas de sanções e devem ser descritas juntamente com a especificação da norma.
- **Contexto:** as normas são geralmente definidas em um determinado contexto que determina a sua área de aplicação. A norma pode, por exemplo, ser descrita no contexto de um determinado ambiente e deve ser cumprida apenas pelos agentes em execução no ambiente, ou pode ser definida no contexto de uma organização e cumprida apenas pelos agentes que desempenham um papel na organização.

#### 4. Estendendo a Sintaxe Abstrata de MAS-ML para SMAs com Normas

Nesta seção é apresentada uma extensão à linguagem MAS-ML de modo que as entidades em SMA possam ser modeladas considerando os principais elementos que compõem as normas a elas associadas. A estratégia utilizada para a incorporação do conceito de normas em MAS-ML é baseada nos mecanismos de extensão da UML, isto é, por meio da definição de novas metaclasses e estereótipos.

##### 4.1. Novas Características em MAS-ML

As normas são baseadas nos conceitos deônticos que têm sido utilizados para descrever as restrições de comportamento para os agentes. Considerando a complexidade dos elementos envolvidos na definição de normas e dos relacionamentos requeridos entre normas e as outras entidades envolvidas na sua definição (metaclasses), foi necessária a criação da metaclasses *Norm* em MAS-ML, juntamente com os estereótipos `<<obligation>>`, `<<permission>>` e `<<prohibition>>`. Esta nova metaclasses estende a metaclasses *Element*, originária da UML, para representar as normas. Com a nova metaclasses podemos definir normas para: (i) agentes, (ii) organizações, (iii) suborganizações, (iv) ambiente e (v) papel de agente.

Na sua concepção original, MAS-ML contempla a definição de (i) obrigações e permissões do agente enquanto executa um papel por meio dos estereótipos `<<duty>>` e `<<right>>` associados à metaclasses *AgentAction* e (ii) restrições globais definidas em organizações, às quais agentes e suborganizações devem obedecer. Essas restrições são denotadas pelo estereótipo `<<axiom>>` associado à metaclasses *Property* que equivale conceitualmente aos estereótipos `<<obligation>>` e `<<prohibition>>`. Com a extensão, todos esses estereótipos foram eliminados.

Quando uma norma é violada ou cumprida, a entidade que violou/cumpriu essa norma pode receber uma sanção [Figueiredo, 2011]. A introdução destes conceitos em

MAS-ML levou à criação da metaclasses *Sanction* juntamente com dois estereótipos, <<*punishment*>> e <<*reward*>>, que representam, respectivamente, a punição e a recompensa relativas a uma determinada norma.

Segundo Figueiredo (2011), as normas têm um período em que suas restrições devem ser cumpridas, porém somente quando elas, as normas, estiverem ativas. De forma a modelar estes conceitos em MAS-ML, foi criada a metaclasses *NormConstraint*, responsável por definir o período de restrição de uma norma, e suas especializações (Figura 1): (i) *Before*, indica que uma norma está ativa antes da execução da ação e/ou da realização a partir de uma determinada data; (ii) *After*, indica que uma norma está ativa depois da execução da ação e/ou da realização a partir de uma determinada data; (iii) *Between*, indica que uma norma está ativa entre a execução de duas determinadas ações e/ou da realização entre duas datas definidas; (iv) *If*, ativa a norma de acordo com a comparação entre dois operandos ou por data. Um operando pode ser um objetivo, uma crença, um atributo ou um valor; e (v) *Date*, criada para armazenar as datas de ativação das normas.

O acesso aos recursos do sistema pode ser controlado através de um conjunto de ações, definidos no contexto de normas. De acordo com Figueiredo (2011) um recurso pode ser (i) um atributo, (ii) um método, (iii) uma associação, (iv) um agente, (v) um papel, (vi) uma organização, (vii) um ambiente, (viii) uma ação, (ix) uma mensagem, (x) um protocolo, (xi) uma crença, (xii) um objetivo e (xiii) um plano. Neste contexto, foi criada a metaclasses *Resource* como uma extensão da metaclasses *Element* da UML, para modelar recursos.

Um recurso pode estar vinculado a: (i) uma característica estrutural ou comportamental, por meio da associação entre as metaclasses *Resource* e *Feature*. Desta forma, tanto as características estruturais (objetivos, crenças, atributos), quanto comportamentais (métodos, ações, planos e protocolos) se tornam recursos; (ii) um classificador, por meio da associação entre as metaclasses *Resource* e *Classifier*. Com isso, as diversas entidades que compõem um SMA são definidas como recurso (um agente, uma organização, um papel de agente ou um ambiente); (iii) um relacionamento, por meio da associação entre as metaclasses *Resource* e *Association*, de forma a restringir o acesso a leitura e atualização; (iv) uma mensagem, por meio da associação entre as metaclasses *Resource* e *AgentMessage* estabelece que mensagens de agente também sejam recursos do sistema que podem ter seu acesso restringido.

Uma norma é definida para restringir a execução das entidades ou acesso aos recursos do sistema, através de um conjunto de ações. Portanto, as ações controladas sobre recursos são vinculadas a normas. Com isso, as ações a serem regulamentadas precisam ser claramente definidas. Para atender a essas especificações em MAS-ML, foi criada a metaclasses *NormAction*. A criação desta metaclasses tem como objetivo definir os recursos que serão restringidos pelas normas, diferentemente da metaclasses *AgentAction*, definida em MAS-ML, que tem como objetivo representar as ações executadas por agentes.

A metaclasses *NormAction* foi definida como extensão da metaclasses da UML *BehavioralFeature*, e é especializada através das metaclasses *AtomicAction* e *CompositeAction*. A metaclasses *AtomicAction* tem como objetivo mapear diretamente as operações do sistema modelado (excluir, atualizar, ler, criar e executar), enquanto a metaclasses *CompositeAction* mapeia as operações das entidades do sistema e mantém a hierarquia tanto de *AtomicActions* como de *CompositeActions* [Basin et al., 2006].

A Figura 1 mostra o metamodelo da MAS-ML estendido para que todas as características das normas descritas na Seção 3.2 possam ser modeladas.

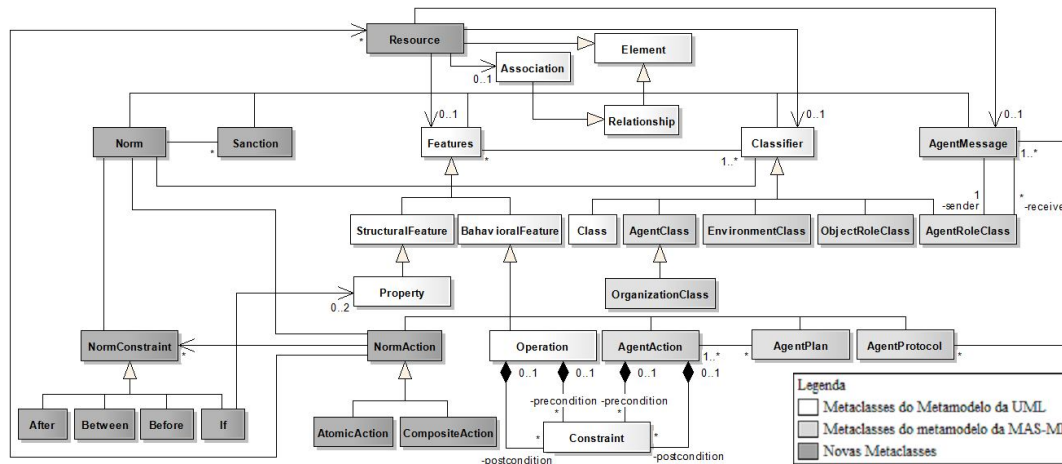


Figure 1. O metamodelo de MAS-ML estendido para incorporar as entidades definidas para normas e suas propriedades.

## 5. Estudo de Caso

Nesta seção é apresentada a modelagem do mercado virtual utilizando a extensão proposta da linguagem MAS-ML. Este estudo de caso, definido e modelado em Silva (2004) utilizando MAS-ML na sua versão original, foi adaptado para que as normas especificadas em Figueiredo (2011) possam ser definidas nas entidades que compõem o mercado virtual.

### 5.1. Mercado Virtual (*Virtual Marketplace*)

Mercados virtuais são mercados localizados na Web em que os usuários compram e vendem itens. Cada um é composto por um mercado principal em que os usuários podem negociar qualquer tipo de item. Além disso, o mercado principal define dois tipos: (i) mercados de produtos especiais que negociam itens caros e de alta qualidade, e (ii) mercados de produtos usados que negociam itens de baixa qualidade e preço baixo.

Os usuários podem: (i) comprar itens no mercado principal, em mercados de produtos especiais e em mercados de produtos usados, e (ii) vender seus itens nos mercados de produtos usados. No mercado principal e nos mercados de produtos especiais, os usuários compram os itens disponíveis no mercado. O mercado principal avalia os lucros, portanto, os mercados de produtos especiais e usados devem enviar as informações relativas às vendas para o mercado principal.

### 5.2. Identificação das Entidades do Mercado Virtual

No ambiente *Virtual Marketplace* é possível identificar a organização principal *General Store* e suas duas suborganizações, *Imported Bookstore* e *Second-hand Bookstore* que exercem os papéis *Market of Special Goods* e *Market of Used Goods*, respectivamente. Ademais, foram modelados nesse sistema dois tipos de agente: *user agent*, que pode exercer o papel *buyer*, e *store agent*, que pode exercer os papéis *manager* e *seller*. Estes papéis foram definidos pela organização principal, juntamente com os papéis de objeto *desire* e *offer*. As instâncias desses

papéis são exercidas pelas instâncias da classe *Book*, que herda da superclasse *Item* e possui duas subclasses *SecondHandBook* e *ImportedBook*.

### 5.3. Definição das Normas para o Mercado Virtual

Para o mercado virtual definido em Figueiredo (2011) foram estabelecidas as seguintes normas:

- N1: Os compradores da organização *General Store* são obrigados a pagar pelos itens que eles compraram.
- N2 (Punição para a violação da N1): Os compradores que violaram N1 são proibidos de comprar itens.

No estudo de caso apresentado em Silva (2004), o papel de agente *Buyer* determinava que todos os agentes associados a esse papel teriam (i) a obrigação de procurar por agentes vendedores e (ii) o direito a aceitar ou rejeitar uma proposta enviada pelo agente vendedor. Essas restrições eram especificadas por meios dos estereótipos `<<duty>>` e `<<right>>`. Entretanto, estes estereótipos foram removidos na extensão. Logo, essas restrições são modeladas por meio de normas, como explicado na Seção 4.1. As normas N3 e N4 representam a obrigação e a permissão associada ao papel de agente *Buyer*:

- N3: Os agentes compradores vinculados ao papel *Buyer* devem procurar por agentes vendedores.
- N4: Os agentes compradores vinculados ao papel *Buyer* podem enviar mensagem sobre a situação da proposta para um agente vendedor.

A Figura 2 mostra a modelagem das normas N1 e N2 e a Figura 3 mostra a modelagem da norma N3. Em ambas, utilizamos a extensão realizada na sintaxe abstrata de MAS-ML. A Norma N4 não foi modelada devido ao reduzido número de páginas. A modelagem de todas as normas deste estudo de caso se encontra em: <https://sites.google.com/site/uecegeessi/estudo-de-caso/estudodecaso-cotb2012>.

A possibilidade de modelar normas e a integração desta visão com os outros diagramas fornecidos por MAS-ML, tais como classe, organização e papéis, fornece uma visão ampla e abrangente do SMA.

## 6. Conclusão e Trabalhos Futuros

Considerando o limitado suporte fornecido por MAS-ML para a modelagem de normas para SMAs, este artigo apresenta a extensão da linguagem MAS-ML de forma a possibilitar a modelagem dos conceitos relacionados a normas, juntamente com conceitos deônticos, entidades envolvidas, ações, restrições de ativação, sanções e contexto.

A extensão realizada envolve a definição de 11 novas metaclasses: (i) *Norm*, (ii) *Sanction*, (iii) *Resource*, (iv) *NormConstraint*, (v) *After*, (vi) *Between*, (vii) *Before*, (viii) *If*, (ix) *NormAction*, (x) *AtomicAction* e (xi) *CompositeAction*. Também foram criados novos estereótipos para as metaclasses *Norm*, *Sanction*, *AtomicAction* e *CompositeAction*. Com eles, conseguimos diferenciar normas de permissão, obrigação e proibição, sanções de recompensa e punição e os tipos de ações atômicas e de composição. Adicionalmente, os estereótipos `<<duty>>` e `<<right>>` da metaclasses *AgentAction* e `<<axiom>>` da metaclasses



Property foram eliminados. Consequentemente, a estrutura estática da metaclassa *AgentRole* e *OrganizationClass* foi alterada para que os mesmos possam ter normas associadas.

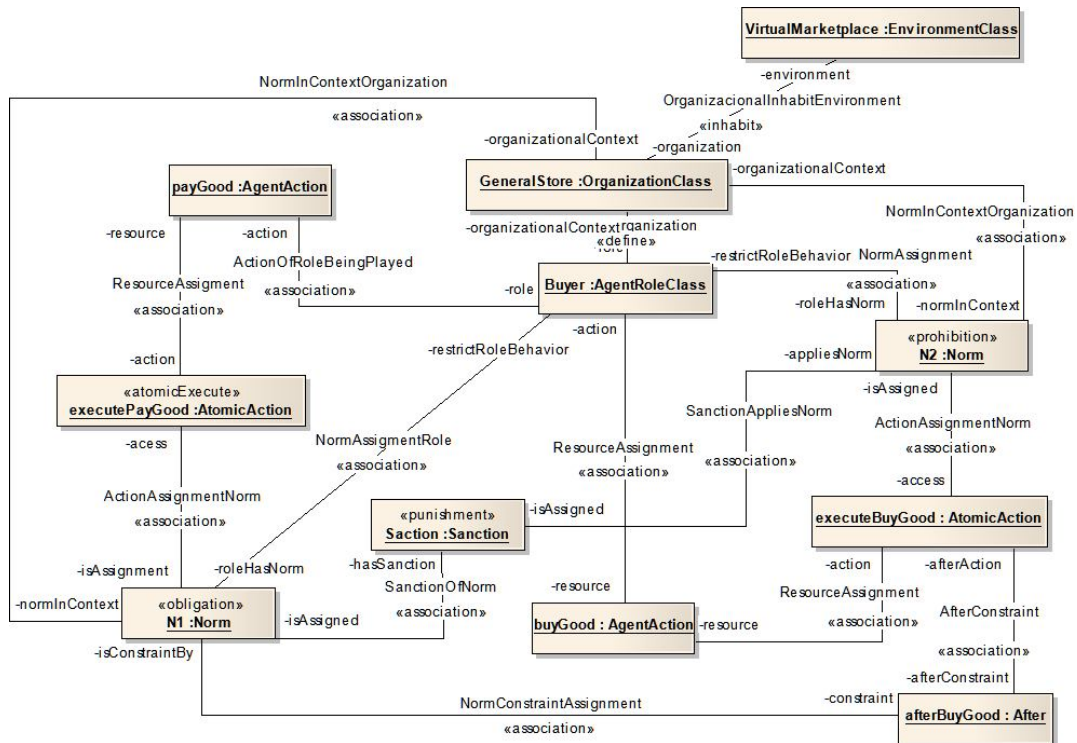


Figure 2. Modelagem das Normas N1 e N2 utilizando a sintaxe abstrata de MAS-ML estendida.

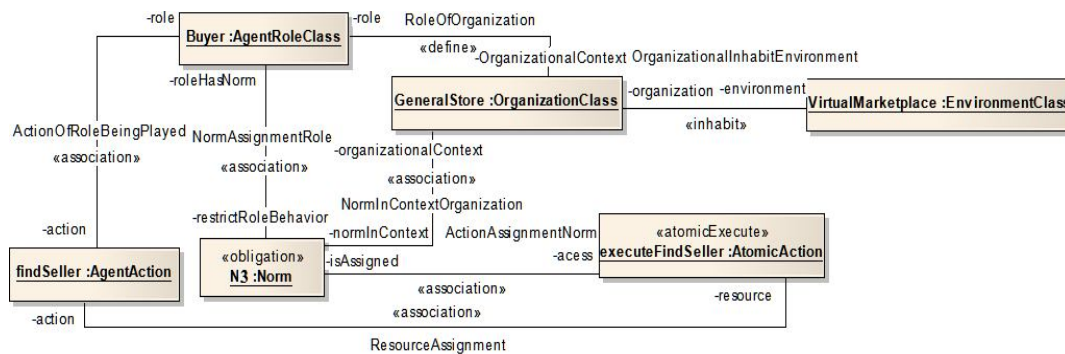


Figure 3. Modelagem da Norma N3 utilizando a sintaxe abstrata de MAS-ML estendida.

O estudo de caso ilustra, através da semântica abstrata, que a extensão realizada permite a modelagem de todas as entidades que a versão anterior conseguia modelar juntamente com os conceitos de normas que foram incluídos nessa nova versão.

Como trabalhos futuros, serão abordados os seguintes itens: (i) definir a sintaxe concreta das metaclasses inseridas na extensão para simplificar a modelagem das normas, (ii) adaptar e/ou definir diagramas para contemplar normas, (iii) estender a ferramenta MAS-ML tool para que a mesma seja capaz de dar suporte a modelagem dos novos elementos definidos na linguagem, verificar a checagem por conflitos entre normas e gerar código a partir dos diagramas definidos na linguagem MAS-ML com normas.

## Referências

- Basin, D., Doser, J. e Lodderstedt, T. (2006) “Model driven security: from uml models to access control infrastructures”, *ACM TSEM*, 15(1), pp.39-91.
- Casillo, B. H. (2008) “Agentes auxiliando ambientes de engenharia de software centrado em processos”, *Dissertação de Mestrado*. São José dos Campos: INPE.
- Danc, J. (2008) “Formal specification of AML”. Department of Computer Science Faculty of Mathematics, Physics and Informatics Comenius University Formal Specification of AML Master's Thesis Ján Danc Advisor: Mgr. Bratislava.
- ECLIPSE (2011), “Eclipse platform”, <http://www.eclipse.org/>, Acessado em 01 de Junho de 2011.
- Farias, K., Oliveira, K., Nunes, I., Silva, V. T. e Lucena, C.J.P. (2009) “MAS-ML tool: um ambiente de modelagem de sistemas multi-agentes”, *V Workshop on Software Engineering for Agent-oriented Systems (SEAS 2009)*, Fortaleza, Brasil, pp. 1-12.
- Figueiredo, K. e Silva, V. T. (2010) “NormML: a modeling language to model norms”. In: *1st Workshop on Autonomous Software Systems*. Salvador, Brazil.
- Figueiredo, K. (2011) “Modeling and validation norms in multi-agents systems”. *Dissertação de mestrado*. Niterói: UFF, Instituto de Computação.
- López y López, F. (2003) “Social Power and Norms: Impact on agent behavior”. PhD thesis, Univ. of Southampton, Faculty of Engineering and Applied Science, Department of Electronics and Computer Science.
- Meyer, J. J. e Wieringa, R. J. (1993) “Deontic logic in computer science: normative system specification”, *Deontic logic in computer science: normative system specification*, John Wiley and Sons Ltd. Chichester, UK.
- Padilha, T. P. P. e Jacome, T. F. (2002) “O uso de técnicas de modelagem de agentes em ambientes educacionais”, *VI Congresso Iberoamericano de Informática Educativa*.
- Russell, S. e Norvig, P. (2004) *Inteligência Artificial: uma Abordagem Moderna*, 2ª ed. Prentice-Hall: São Paulo.
- Silva, V. T., Braga, C. e Figueiredo, K. (2010) “A modeling language to model norms”. *The International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2010)*, 9th. Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems, Toronto, Canadá.
- Silva, V. T. (2004) “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.
- Silva, V. T., Garcia, A., Brandao, A., Chavez, C., Lucena, C. e Alencar, P. (2003) “Taming agents and objects in software engineering”, In: Garcia, A.; Lucena, C.; Zamboneli, F.; Omicini, A; Castro, J. (Eds.), *Software Engineering for Large-Scale Multi-Agent Systems*, Springer-Verlag, LNCS 2603, pp. 1-26, 2003, ISBN 978-3-540-08772-4.
- Wagner, G. (2003) “The Agent-Object-Relationship Meta-Model: Towards a Unified View of State and Behavior”, *Information Systems*. 28(5), pp. 475–504.