

# Uma Proposta de Mapeamento entre i\* Normativo e o Diagrama NorMAS-ML de Normas\*

Emmanuel Sávio Silva Freire<sup>1</sup>, Thiago Gomes de Souza<sup>2</sup>

<sup>1</sup>Departamento de Ensino – Instituto Federal do Ceará (IFCE/Campus Morada Nova)  
Av. Prefeito Raimundo José Rabelo, 2717 – 62.940-000 – Morada Nova – CE – Brasil

<sup>2</sup>Departamento de Ensino – Instituto Federal do Ceará (IFCE/Campus Iguatu)  
Rua Deoclécio Lima Verde, s/n – Areias II – 63.500-000 – Iguatu – CE – Brasil  
savio.freire@ifce.edu.br, {savio.essf, thiagogsouza91}@gmail.com

**Abstract.** *Normative multi-agent systems have been used to develop complex systems. Thus, several modeling languages, such as NorMAS-ML, were defined in order to support the modeling phase, but a few number of works have been addressed the gathering and analysis of requirements phase. In this sense, normative i\* framework was defined in order to make possible that requirement analysts can model stakeholders as actors and theirs intentions as goals. Therefore, this paper has the objective the integration between gathering and analysis of requirements and modeling phases, proposing a mapping between normative i\* and NorMAS-ML. In addition, a modeling example was done in order to illustrate this mapping.*

**Resumo.** *Os sistemas multiagente normativos são utilizados para o desenvolvimento de sistemas complexos. Logo, várias linguagens de modelagem, dentre elas NorMAS-ML, foram definidas para dar suporte à fase de modelagem, porém poucos trabalhos têm abordado a fase de levantamento e análise de requisitos. Neste sentido, o framework i\* normativo foi definido para possibilitar que analistas de requisitos possam modelar os stakeholders como atores e suas intenções como objetivos. Assim, esse artigo tem como objetivo a integração entre as fases de levantamento e análise de requisitos e a fase de modelagem, propondo um mapeamento entre i\* normativo e NorMAS-ML. Além disso, um exemplo de modelagem foi realizado para ilustrar este mapeamento.*

## 1. Introdução

Os sistemas multiagente normativos (SMANs) são formados por um conjunto de agentes inteligentes que interagem entre si para alcançar os seus objetivos, verificando as normas definidas nesses sistemas para identificar quais as ações que são permitidas, obrigadas ou proibidas de serem executadas [Boella, van der Torre e Verhagen 2006] [Silva, Braga e Figueiredo 2010]. Assim, linguagens de modelagem e de implementação foram propostas para auxiliar o desenvolvimento desses sistemas. Dentre elas, destaca-se a linguagem *Normative Multiagent System Modeling Language* (NorMAS-ML) [Freire et al. 2012] que possibilita a modelagem das entidades encontradas nesses sistemas junto com as normas. Contudo, essa linguagem não dá suporte a fase de levantamento e análise de requisitos.

No processo de desenvolvimento de *software*, o levantamento de requisitos compreensíveis por todas as partes envolvidas no desenvolvimento do sistema (clientes,

\* Este trabalho faz parte do Projeto número 5925 intitulado “Integração de Requisitos Organizacionais à Modelagem de Sistemas Multiagente Normativos” do Edital IFCE/PIBIC Jr/2017. O autor Thiago Souza agradece pelo suporte financeiro recebido pelo IFCE para a realização deste trabalho.

analistas, desenvolvedores etc.) é um fator básico e extremamente importante, evitando falhas no entendimento do problema a ser solucionado [Franceto 2005]. Assim, a fase de levantamento e análise de requisitos propicia um melhor entendimento das reais necessidades dos usuários do sistema [Sommerville 2011]. Logo, é possível detalhar e priorizar os requisitos permitindo que aqueles essenciais ao sistema possam ser implementados inicialmente. Conseqüentemente, o usuário tem uma maior probabilidade de receber um sistema que supra as suas necessidades.

Neste sentido, o *framework i\** (iStar) [Yu 2002] foi proposto para auxiliar a fase de requisitos por meio de um modelo conceitual, permitindo que os analistas de requisitos possam modelar os *stakeholders* como atores e suas intenções como objetivos. Assim, consegue-se obter uma compreensão sobre os relacionamentos da organização (que define os requisitos, ou seja, uma empresa) e sobre as razões envolvidas nos processos de decisão [Yu 2002]. Entretanto, *i\** não permitia a modelagem de requisitos organizacionais para os SMAN, pois não abordava os elementos normativos nem a interação deles com as entidades de um SMAN. Por isso, Siena et al. (2008) propuseram a extensão de *i\**, chamada de *i\** normativo, para permitir a modelagem de aspectos normativos, indicando as entidades que são reguladas pelas normas.

Considerando que *i\** normativo permite a modelagem de requisitos organizacionais para SMAN e NorMAS-ML permite a modelagem das entidades típicas encontradas nesses sistemas, este artigo propõe um mapeamento entre *i\** normativo e NorMAS-ML com o intuito de identificar quais as semelhanças e as diferenças conceituais existentes entre eles. Assim, espera-se possibilitar a integração entre a fase de levantamento e análise de requisitos e a fase de modelagem para o desenvolvimento de SMANs, proporcionando uma visão mais completa do sistema. Vale ressaltar que a relevância dessa integração foi discutida em um trabalho inicial [Freire e Cortés, 2016].

Este artigo está organizado como segue. A seção 2 apresenta a linguagem NorMAS-ML, enquanto a seção 3 discorre sobre *i\** normativo. O mapeamento entre *i\** normativo e NorMAS-ML é detalhado na seção 4. Na seção 5, um exemplo de modelagem é apresentado. As discussões acerca do mapeamento e a sua comparação com os trabalhos relacionados são descritos na seção 6. Finalmente, as conclusões e os trabalhos futuros são abordados na seção 7.

## 2. NorMAS-ML

A *Normative Multiagent System Modeling Language* (NorMAS-ML) [Freire et al. 2012] é uma linguagem de modelagem que permite o *design* de SMANs. Ela surgiu por meio da extensão da MAS-ML para possibilitar a modelagem das entidades típicas encontradas nos SMANs (ambiente, organização, agente, papel de agente, objeto e papel de objeto) em conjunto com os seguintes elementos que compõem uma norma [Silva, Braga e Figueiredo 2010]: (i) **Conceitos deônticos**, descrevem as restrições de comportamento para os agentes na forma de obrigações, permissões e proibições; (ii) **Entidades Envolvidas**, que são as entidades que possuem o seu comportamento restringido por uma norma; (iii) **Ações**, que são as ações que estão sendo reguladas pela norma; (iv) **Restrições de Ativação**, que indicam a restrição ou um conjunto de restrições que ativam uma norma; (v) **Sanções**, que são punições/recompensas que uma entidade recebe quando uma norma é violada/seguida, respectivamente; e (vi) **Contexto**, indica a área de atuação (um ambiente ou uma organização) de uma norma.

Adicionalmente, Freire et al. (2012) definiram um novo diagrama estático que permite a modelagem de normas. Um exemplo do diagrama de normas pode ser encontrado

na seção 5. As entidades que podem participar do diagrama são: (i) classe de norma, (ii) classe, (iii) classe de papel de objeto, (iv) classe de agente, (v) classe de papel de agente, (vi) classe de organização e (vii) classe de ambiente. Além disso, dos relacionamentos definidos na UML, também podem ser utilizados os seguintes relacionamentos no diagrama: (i) *ownership*, (ii) *play*, (iii) *inhabit*, (iv) *context*, (v) *restrict*, e (vi) *sanction*.

### 3. Framework i\* Normativo

O *framework iStar* normativo ou *i\** normativo [Siena et al. 2008] é uma extensão do *framework i\** [Yu 2002] que possibilita a representação de leis e normas utilizando o modelo de ator, objetivo e dependências existente em *i\**. Para tanto, os autores consideram as seguintes propriedades das normas relevantes para a aquisição de requisitos: (i) **a relação de compromisso normativo** (*the normative commitment relation*), essa relação é formada por quem criou a norma (*source*), por aquele que é endereçado pela norma (*addressee*) e pelo artefato que será restringido pela norma (*legal artefact*); (ii) **o esquema de uma norma** (*the schema of the norm*), que está relacionado com o padrão de comportamento que a norma impõe para o *addressee*, como por exemplo, formas de atuação, objetivos e princípios a serem adotados; e (iii) **as intenções de conformidade** (*the compliance intentions*), que estão associadas ao impacto real da norma sobre os *stakeholders*, identificando o que eles colocaram em ação para cumprir a norma.

Por meio da análise das três propriedades incluídas na extensão, a Figura 1 apresenta o metamodelo<sup>1</sup> de *i\** normativo. Adicionalmente, os modelos que permitem que os engenheiros de requisitos possam modelar os *stakeholders* como atores e suas intenções como objetivos também foram alterados por conta da extensão. Esses modelos são: (i) o modelo de dependência estratégica (*Strategic Dependency – SD*), que foca na relação entre os agentes organizacionais, e (ii) o modelo estratégico de razão (*Strategic Rational – SR*), que possibilita a modelagem detalhada das relações intencionais internas de cada agente organizacional. Um exemplo de modelagem utilizando o *framework i\** normativo pode ser encontrado na seção 5.

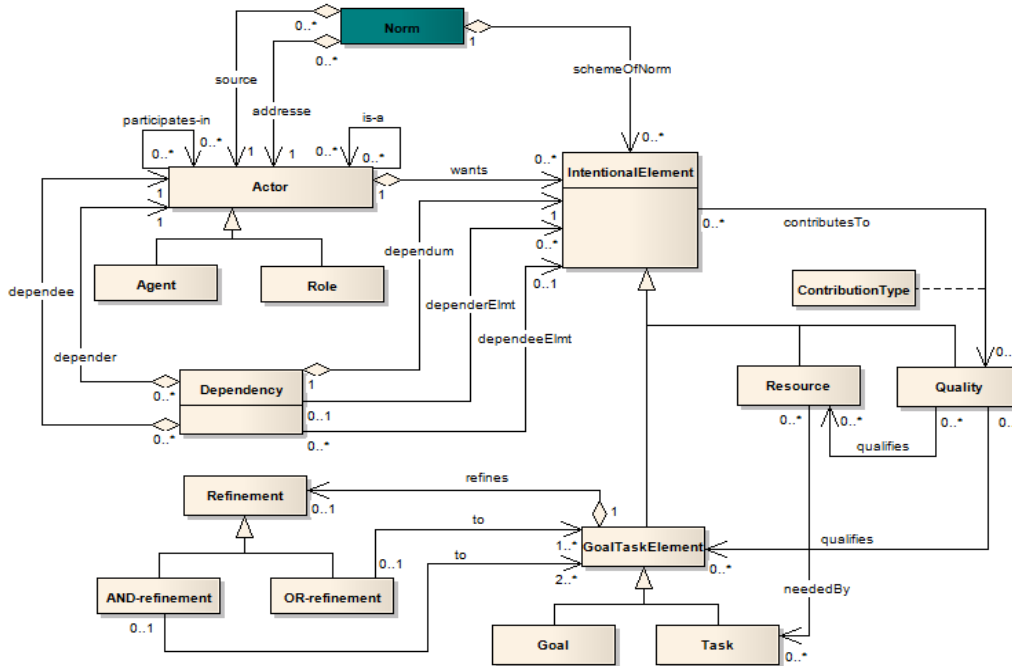


Figura 1. Metamodelo do *framework i\** normativo (adaptado pelos autores)

1 Vale ressaltar que esse metamodelo foi proposto pelos autores do presente artigo por meio da comparação entre o metamodelo original de *i\** e as propriedades das normas apresentadas na seção 3.

#### 4. Mapeamento entre i\* Normativo e o Diagrama NorMAS-ML de Normas

Nesta seção, são apresentadas as diretrizes de mapeamento entre i\* normativo e o diagrama NorMAS-ML de normas associado a exemplos para cada diretriz.

##### 4.1. Diretrizes de Mapeamento

As diretrizes de mapeamento determinam as regras pelas quais a associação entre o modelo i\* Normativo e o diagrama NorMAS-ML de normas é realizada corretamente. Para tanto, foram modificadas as diretrizes propostas por Alencar et al. (2003) e Melo et al. (2015) que possibilitavam o mapeamento entre i\* e o diagrama UML de classes.

O objetivo das diretrizes é manter a consistência e o rastreamento entre o sistema a ser desenvolvido e os objetivos da organização, considerando as normas que regulam o comportamento das entidades que compõem os SMANs. Desta forma, aumenta-se as chances do sistema que está sendo desenvolvido atender os objetivos da organização de uma forma mais precisa. Além disso, pode-se relacionar os artefatos gerados na fase de análise (modelo i\* normativo) com os gerados na fase de projeto (diagrama NorMAS-ML de normas).

As diretrizes foram subdivididas nos seguintes sete grupos: (i) D1, mapeamento de atores; (ii) D2, mapeamento de tarefas; (iii) D3, mapeamento dos recursos; (iv) D4, mapeamento de objetivos e objetivos *soft*; (v) D5, mapeamento de relação entre meios-fim (*means-end*); (vi) D6, mapeamento de decomposição de tarefas; e (vii) D7, mapeamento de normas. O Quadro 1 apresenta as diretrizes de mapeamento entre i\* Normativo e o diagrama NorMAS-ML de normas.

**Quadro 1. Diretrizes de mapeamento entre i\* normativo e o diagrama NorMAS-ML de normas**

Tipo	Número	i* Normativo	Diagrama NorMAS-ML de Normas
D1	D1.1	Papel	Classe de Papel de Agente
	D1.2	Agente	Classe de Agente
	D1.3	Posição	Classes de Organização
	D1.4	Relacionamento IS-PART-OF entre papel e posição	Relacionamento <i>ownership</i> entre Classes de Papel de Agente e Organização
	D1.5	Relacionamento IS-PART-OF entre posições	Relacionamento <i>aggregation</i> entre Classes de Posição
	D1.6	Relacionamento IS-PART-OF entre agente e posição	Não apresenta correspondente
	D1.7	Relacionamento IS-A	Generalização/Especialização de classes de Papel de Agente ou entre Classes de Agentes ou de Organizações
	D1.8	Relacionamento i* OCCUPIES entre um agente e uma posição	Não apresenta correspondente
	D1.9	Relacionamento i* COVERS entre uma posição e um papel	Relacionamento <i>ownership</i> entre Classes de Papel de Agente e Organização
	D1.10	Relacionamento i* PLAYS entre um agente e um papel	Relacionamento <i>play</i> entre Classes de Agente e de Papel de Agente
D2	D2.1	Tarefa no modelo SD	Atributo <i>agent action</i> com visibilidade pública ou <i>resource</i> , caso seja uma norma.
	D2.2	Tarefas no modelo SR	Atributo <i>agent action</i> com visibilidade privada ou <i>resource</i> , caso seja uma norma.
D3	D3.1	Recursos definidos no modelo SD	Classe de Objeto, se essa dependência tem a característica de um objeto

	D3.1	Recursos definidos no modelo SD	Atributo agent action com visibilidade privada
	D3.2	Recursos (sub-recursos) definidos no modelo SR	Atributo agent action com visibilidade privada na classe que representa o ator em que o sub-recurso pertence (se esse sub-recurso não pode ser entendido como um objeto).
	D3.2	Recursos (sub-recursos) definidos no modelo SR	Classe de Objeto, se esse sub-recurso tem a característica de um objeto
D4	D4.1	Objetivo (ou Objetivo soft) definido no modelo SD	Atributo goal com visibilidade pública na classe da entidade correspondente.
	D4.2	Objetivo (ou Objetivo soft) definido no modelo SR	Atributo goal com visibilidade pública na classe da entidade correspondente
D5	D5.1	Objetivos (ou Objetivos soft) - Objetivos (ou Objetivos soft)	Representada por regras (podem ser descritas em OCL) da entidade correspondente
	D5.2	Objetivos (ou Objetivos soft)-Tarefa, Recurso-Tarefa	Representada por regras (podem ser descritas em OCL) da entidade correspondente
	D5.3	Tarefa-Tarefa	Representada por regras (podem ser descritas em OCL) da entidade correspondente
D6	D6.1	Decomposição de tarefa no modelo SR	Representada pelas pré e pós-condições (podem ser expressas em OCL) da agent action correspondente
D7	D7.1	Fonte da Norma	Relacionamento context entre Classes de Norma e de Papel de Agente
	D7.2	Norma	Classe de Norma de Obrigação
	D7.3	Endereçado pela Norma	Relacionamento restrict entre Classes de Norma e de Agente ou entre Classes de Norma e de Papel de Agente ou entre Classes de Norma e de Organização.
	D7.4	Prescrições da Norma	Segue as diretrizes D2, D3, D4, D5 e D6.

#### 4.2. Detalhamento das Diretrizes

Nesta seção, as diretrizes de mapeamento serão detalhadas por meio da discussão conceitual dos elementos mapeados e da apresentação da transformação das entidades de i\* normativo para seu correspondente no diagrama NorMAS-ML de normas. De acordo com o Quadro 1, a **diretriz D1.1** menciona que um papel deve ser transformado em uma classe de Papel de Agente. Em i\* normativo, um papel é uma caracterização abstrata do comportamento de um ator dentro de algum contexto. Esse conceito é equivalente em NorMAS-ML, pois um papel de agente é responsável por restringir e orientar o comportamento de agentes quando executam o papel. A Figura 2 (a) apresenta essa transformação.

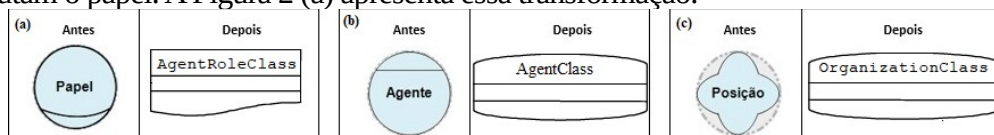


Figura 2. Transformação de atores de i\* normativo para entidades de NorMAS-ML

Considerando o conceito de agente em i\* normativo é equivalente em NorMAS-ML, a **diretriz D1.2** indica que um agente em i\* normativo deve ser transformado em uma classe de agente no diagrama de normas (ver Figura 2 (b)). Em relação à entidade posição, no *framework* i\* normativo, representa um conjunto de papéis normalmente executado por agentes. Esse conceito é encontrado na entidade Organização de NorMAS-ML. Uma organização é capaz de definir papéis que devem ser exercidos pelos agentes.

Logo, possui um conjunto de papéis. Consequentemente, a **diretriz D1.3** determina que uma posição em i\* normativo seja transformado em uma classe de organização no diagrama de normas. A Figura 2 (c) ilustra essa transformação.

A **diretriz D1.4** determina que o relacionamento IS-PART-OF entre uma posição e um papel seja mapeado para o relacionamento *ownership* de NorMAS-ML. Ele indica que um proprietário (organização) conheça todos os seus membros (papéis de agente) (Ver Figura 3). Além disso, a **diretriz D1.5** indica que o relacionamento IS-PART-OF entre posições seja transformado no relacionamento *aggregation* de NorMAS-ML. Vale ressaltar que, em NorMAS-ML, não há correspondente para o mapeamento do relacionamento IS-PART-OF entre uma posição e um agente (**diretriz D1.6**), pois esse relacionamento é feito de forma implícita quando um agente executa um papel de agente pertencente a uma organização.

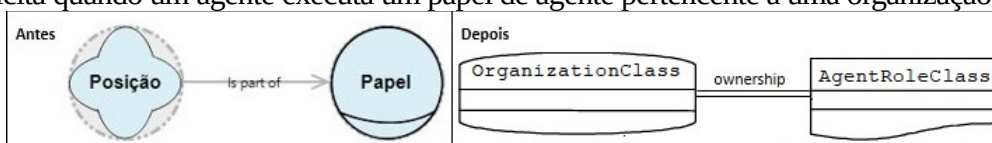


Figura 3. Transformação do relacionamento IS-PART-OF em ownership de NorMAS-ML

A **diretriz D1.7** indica que o relacionamento IS-A deve ser transformado em um relacionamento *specialization* em NorMAS-ML, pois indica relação de herança entre as entidades participantes. Esses relacionamentos possuem a mesma semântica. A Figura 4 apresenta essa transformação. Em relação à **diretriz D1.8**, não é possível representar o relacionamento OCCUPIES em NorMAS-ML pelo mesmo motivo apresentado para a diretriz D1.5.

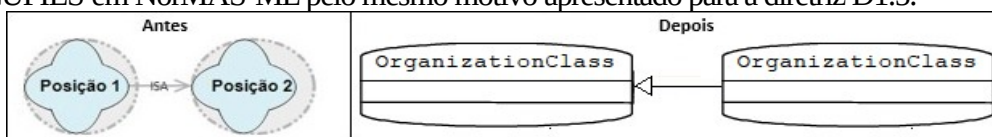


Figura 4. Transformação do relacionamento IS-A em specialization de NorMAS-ML

Considerando que o relacionamento COVERS de i\* normativo indica a relação entre uma posição e seus papéis, a **diretriz D1.9** estabelece que esse relacionamento seja mapeado para o relacionamento *ownership* entre classes de papel de agente e de organização. A figura 5 apresenta esse mapeamento. Em relação ao relacionamento PLAYS de i\* normativo, a **diretriz D1.10** determina que ele seja transformado no relacionamento *play* de NorMAS-ML, pois apresentam a mesma semântica. A figura 6 ilustra essa transformação.

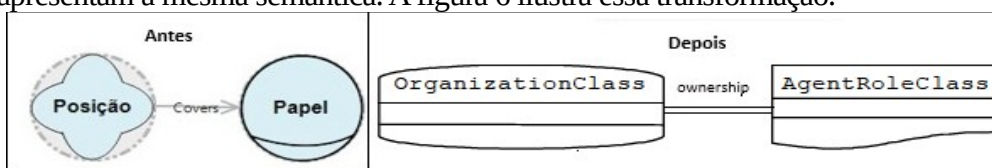


Figura 5. Transformação do relacionamento COVERS em ownership de NorMAS-ML

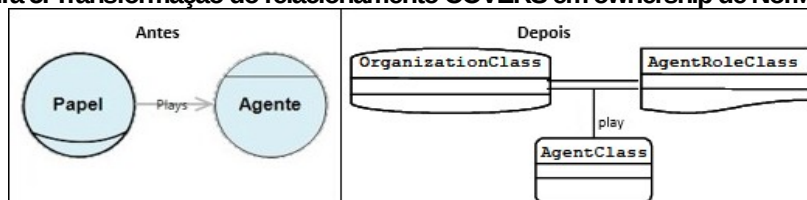


Figura 6. Transformação do relacionamento PLAYS em play de NorMAS-ML

É importante ressaltar que os relacionamentos IS-PART-OF e COVERS de i\* normativo foram mapeados para o relacionamento *ownership* de NorMAS-ML, conforme as diretrizes D1.4 e D1.9, respectivamente. Assim, por mais que IS-PART-OF e COVERS tenham semânticas diferentes no modelo i\*, o relacionamento *ownership* garante tanto a

associação entre atores (organização e papéis de agente) demonstrando que um ator faz parte de outro ator e a associação entre uma posição (organização) e seus papéis de agente.

A **diretriz D2.1** indica que as tarefas definidas no modelo SD devem ser convertidas em agent actions com visibilidade pública na classe do ator que relaciona com essas tarefas. Além disso, a **diretriz D2.2** estabelece que as tarefas definidas no modelo SR também devem se tornar agent actions, porém com visibilidade privada (Ver Figura 7). É importante ressaltar que um agent action de NorMAS-ML corresponde a uma ação que pode ser executada por um agente, por uma organização ou por um papel de agente.

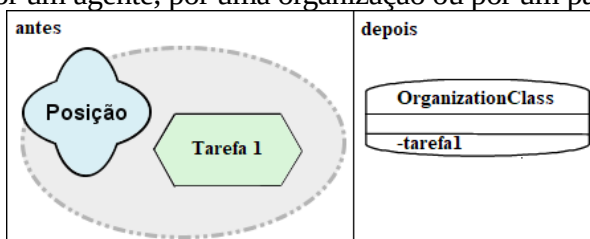


Figura 7. Transformação de tarefas em agent actions de NorMAS-ML

A **diretriz D3.1** afirma que os recursos feitos no modelo SD devem ser transformados em um objeto em NorMAS-ML, caso essa dependência tenha característica de objeto. Entretanto, se a dependência não tiver essa característica, o recurso deve ser transformado em um agent action com visibilidade privada na classe do ator. O mesmo ocorre para os recursos do modelo SR conforme a **diretriz D3.2**. A Figura 8 apresenta esses mapeamentos.

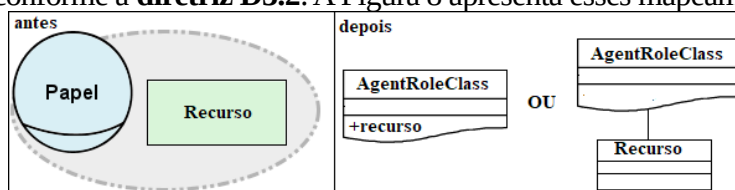


Figura 8. Transformação de recursos em objetos ou agent actions de NorMAS-ML

Em relação aos objetivos no modelo SD e SR, as **diretrizes D4.1** e **D4.2** indicam que devem ser transformados em *goals* em NorMAS-ML com visibilidade pública na classe do ator correspondente. Um atributo *goal* representa o objetivo de um agente, de uma organização ou de um papel de agente em NorMAS-ML. A Figura 9 ilustra essa transformação.

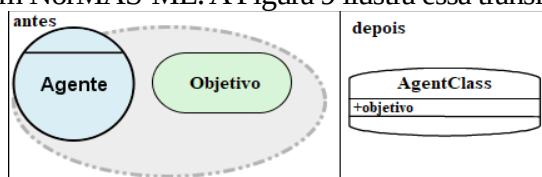


Figura 9. Transformação de recursos em objetos ou agent actions de NorMAS-ML

A **diretriz D5** refere-se aos meios-fim (*means-end*), que sugere que pode-se ter formas alternativas para alcançar um objetivo. Assim, as **diretrizes D5.1, D5.2 e D5.3** indicam que deve ser criada uma nota contendo a instrução correspondente, vinculada à classe do ator. A **diretriz D6** está relacionada com a decomposição de tarefas. Essa decomposição indica o que deve ser feito para a realização de uma determinada tarefa. Assim, por meio da **diretriz D6.1**, a decomposição deve também ter uma nota vinculada à classe do ator. A Figura 10 apresenta essas transformações de *means-end* e de decomposição de tarefas.

A **diretriz D7.1** indica que a fonte da norma deve ser transformada para o relacionamento *context* de NorMAS-ML entre classes de norma e de papel de agente. A Figura 11 apresenta esse mapeamento. Além disso, a **diretriz D7.2** estabelece que a norma no *framework* i\* normativo deve ser transformada na entidade norma de NorMAS-ML com conceito deontico de obrigação



(Ver Figura 11). Foi utilizado esse conceito deôntico pois em i\* normativo não é possível a diferenciação entre normas de obrigação, permissão e proibição no seu metamodelo.

A **diretriz D7.3** afirma que a entidade endereça pela norma deve ser representada pelo relacionamento *restrict* de NorMAS-ML (Ver Figura 11). Assim, pode-se vincular classes de agente, de papel de agente ou de organização a uma classe de norma. Já a **diretriz D7.4** estabelece que as prescrições da norma devem seguir as diretrizes D2, D3, D4, D5 e D6 (Ver Figura 10).

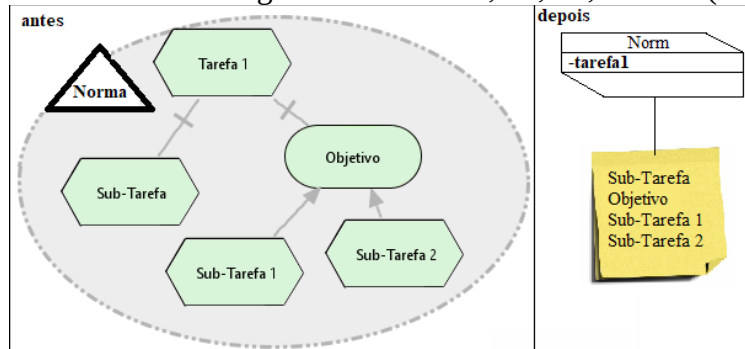


Figura 10. Transformação de means-end e de decomposição de tarefas em NorMAS-ML

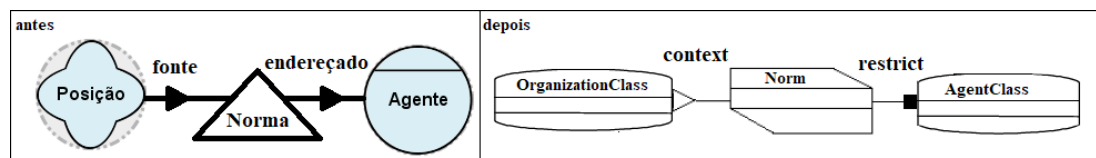


Figura 11. Transformação da norma e seus elementos em NorMAS-ML

## 5. Exemplo de Modelagem

Para ilustrar o uso das diretrizes propostas na seção anterior, o exemplo de modelagem escolhido foi um sistema *SmartCD* definido por Alencar et al. (2003). Vale ressaltar que esse sistema foi previamente modelado por Melo et al. (2015) para ilustrar as diretrizes de mapeamento entre i\* e o diagrama UML de classes. Consequentemente, esse sistema não possui normas associadas às suas entidades. Logo, foram definidas e incluídas normas no sistema *SmartCD* para possibilitar a utilização das diretrizes de mapeamento entre i\* normativo e o diagrama NorMAS-ML de normas.

O sistema *SmartCD* permite a realização de pedidos por meio de um catálogo online. Inicialmente, Melo et al. (2015) definiram três posições: *Client*, *Store Management* e *SmartCD*. A última posição corresponde ao sistema a ser desenvolvido que deve tratar as encomendas, notificar as chegadas de CD e fornecer o catálogo. A Figura 12 apresenta o modelo *Strategic Dependence* (SD) definido por esses autores. Adicionalmente, para a posição *SmartCD*, foram definidas as seguintes normas: (i) N1: A posição *SmartCD* definiu que a posição *Internet Sales* deve registrar o cliente antes de efetuar a venda de CDs; (ii) N2: A posição *SmartCD* definiu que a posição *Inventory* deve atualizar o estoque sempre que houver uma nova venda; e (iii) N3: A posição *Internet Sales* definiu que o ator *Office Boy* possui um prazo de duas horas, obrigatoriamente, para a entrega de CDs na cidade que este ator se encontra.

A Figura 13 apresenta o modelo *Strategic Rational* (SR) considerando a posição *SmartCD* em conjunto com as normas apresentadas acima. Esse diagrama foi baseado no exemplo de modelagem apresentado em Melo et al. (2015). Após a transformação de modelos, foi gerado o diagrama NorMAS-ML de normas apresentado na Figura 14, considerado as diretrizes de mapeamento definidas no Quadro 1. No diagrama, os papéis *CDReservation* e *CDDelivery*, o agente *OfficeBoy*, e as posições *Inventory*, *SmartCD*, *Financial*, *InternetSales* e *Client* foram transformados em classes de papel de agente, de agente e de organização, respectivamente,



segundo as **diretrizes D1.1, D1.2 e D1.3**. Além disso, o relacionamento IS-PART-OF foram transformados em agregações entre a organização *SmartCD* e as organizações Inventory e Financial, conforme a **diretriz D1.5**. Para o relacionamento COVERS foi transformado no relacionamento ownership (**diretriz D1.9**) entre a organização *InternetSales* e os papéis de agente *CDReservation* e *CDDelivery*. O relacionamento PLAYS foi transformado no relacionamento *play* de NorMAS-ML (**diretriz D1.10**) entre o papel de agente *CDDelivery* e o agente *OfficeBoy*. Vale ressaltar que o relacionamento OCCUPIES não foi mapeado para NorMAS-ML, conforme a **diretriz D1.8**. Entretanto, as entidades ligadas por esse relacionamento foram mapeadas para o diagrama de normas.

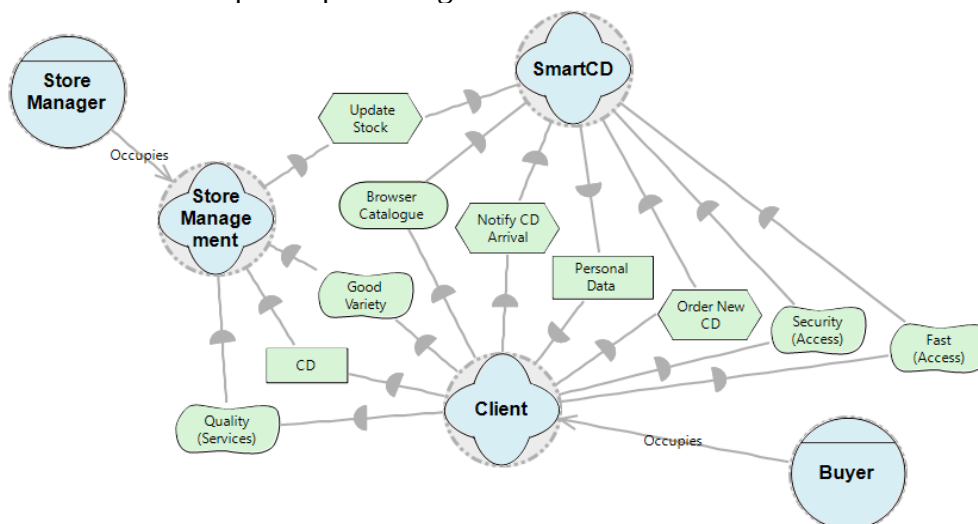


Figura 12. Modelo Strategic Dependence (SD) para o SmartCD [Melo et al., 2015]

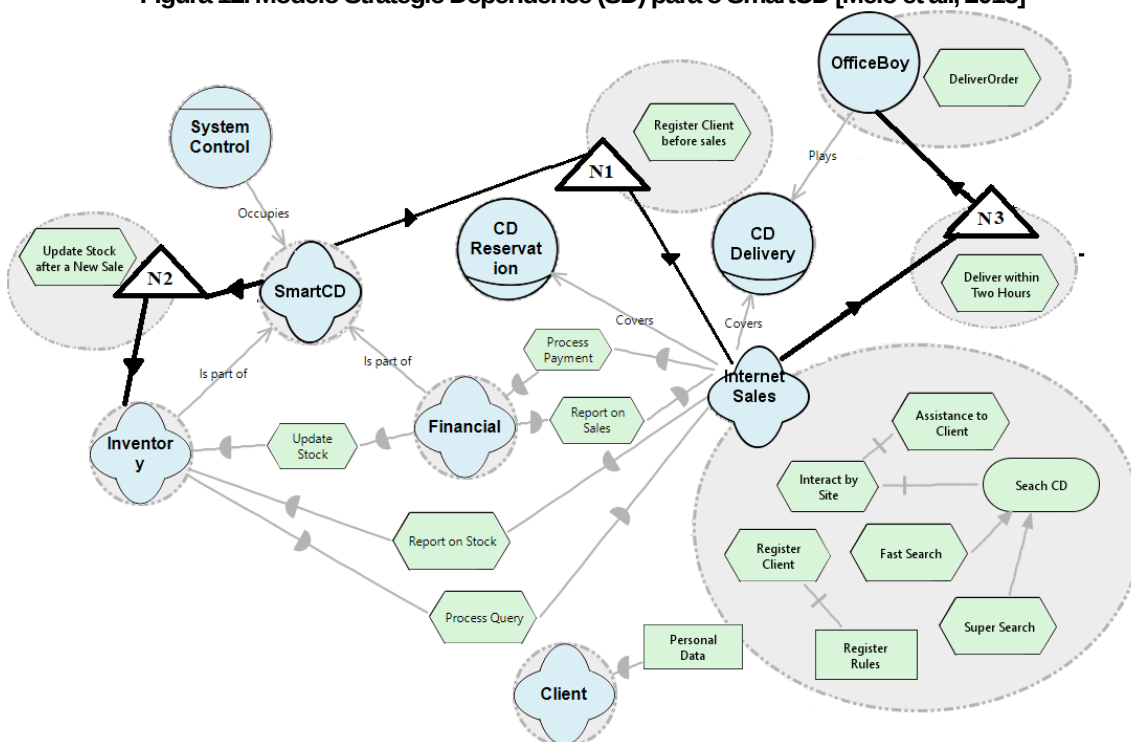


Figura 13. Modelo Strategic Rational (SR) para o SmartCD (Fonte: Próprios Autores)

As tarefas foram mapeadas para *agentActions* nas classes das entidades responsáveis por cada tarefa (**diretriz D2**). Além disso, o recurso *PersonalData*, foi mapeado como uma classe no

diagrama, pois apresentava característica de objeto (**diretriz D3.2**). Conforme as **diretrizes D5 e D6**, os links de decomposição de tarefas e de *means-end*, foram mapeados para um comentário associado à organização *InternetSales*. Em relação as normas, N1, N2 e N3 foram mapeadas para classes de norma (**diretriz D7.2**), suas fontes foram representadas pelo relacionamento *context* (**diretriz D7.1**) e seus endereçamentos pelo relacionamento *restrict* (**diretriz D7.3**). Finalmente, as prescrições de cada norma foram mapeados para *resource* seguindo a **diretriz D7.4**.

## 6. Discussão

Por meio das diretrizes de mapeamento apresentado na seção 4, é possível transformar modelos *i\** normativo no diagrama NorMAS-ML de normas. Essa transformação de modelos foi exemplificada na seção 5 através da modelagem do sistema *SmartCD*. Como vantagens, pode-se citar: (i) *O analista de requisitos pode concentra-se no entendimento das necessidades dos usuários do sistema a ser desenvolvido e mapeá-las para os modelos SD e SR de i\* normativo. Além disso, as normas que regulam o comportamento das entidades desse sistema também podem ser modelados;* (ii) *O projetista pode utilizar as diretrizes de mapeamento propostas no presente trabalho para auxiliar na transformação dos modelos i\* normativo no diagrama NorMAS-ML de normas. Com isso, o projetista se concentrará na identificação de possíveis melhorias para o diagrama em conjunto com os analistas de requisitos;* e (iii) *A integração entre as fases de requisitos e de modelagem permite que os erros associados à transformação de modelos entre essas fases seja diminuído, possibilitando uma maior consistência entre os modelos gerados.*

Entretanto, essa pesquisa apresenta as seguintes limitações: (i) *O diagrama i\* normativo apresentado na seção 5 apresenta uma modelagem parcial do sistema SmartCD. Visto que, a modelagem poderia ter abordados os demais atores e novas normas poderiam ter sido propostas;* (ii) *Não foi proposta uma ferramenta para automatizar a transformação de modelos i\* normativo para o diagrama de normas. Por consequência, deve ser feita de forma manual;* e (iii) *Ao transformar os modelos i\* normativo para o diagrama NorMAS-ML de normas, o projetista precisaria seguir as diretrizes e realizar a transformação de forma manual. Logo, possíveis inconsistências poderiam ser encontradas nesse processo.*

Em relação aos trabalhos relacionados, vários autores [Richard 2010] [Sun et al. 2010] propuseram métodos que permitem a transformação entre modelos. Mais especificamente, Lucena et al. (2011) e Aguilar et al. (2010) propuseram a transformação de modelos *i\** para modelos arquitetônicos Acme e para a linguagem QVT, respectivamente. Em relação ao escopo do presente trabalho, as abordagens propostas por Melo et al. (2015), Alencar, Giachetti e Pastor (2009), Filho, Zisman e Spanoudakis (2003) e Alencar et al. (2003) se aproximam ao objetivo deste trabalho pelo fato de proporem o mapeamento de *i\** para diagramas da UML. No entanto, nenhum dos trabalhos descritos acima consideram o contexto dos SMANs, pois não abordam as normas que regulam as entidades desses sistemas. Com isso, o presente trabalho se diferencia dos demais por utilizar o *framework i\** normativo em conjunto com o diagrama NorMAS-ML de normas, possibilitando a integração das fases de requisitos e modelagem via modelo.

## 7. Conclusão e Trabalhos Futuros

Este artigo teve como objetivo propor um mapeamento entre *i\** normativo e NorMAS-ML com o intuito de identificar quais as semelhanças e as diferenças conceituais existentes entre eles. Para tanto, foram analisadas conceitualmente as entidades previstas em *i\** normativo e do diagrama NorMAS-ML de normas. Assim, foi proposto um conjunto de diretrizes para a transformação de modelos entre esse *framework* e essa linguagem. Adicionalmente, um exemplo de modelagem, baseado no sistema *SmartCD*, foi realizado com o intuito de ilustrar as diretrizes de transformação propostas. Além das diretrizes de transformação, a contribuição

desse artigo foi a integração entre as fases de requisitos e de modelagem previstas no processo de desenvolvimento de *software*, considerando o paradigma orientado a agentes. Mais especificamente, essa integração colabora para que o entendimento de sistemas complexos (SMANs) seja facilitada por meio da utilização de linguagem visual (diagramas) e para a diminuição de inconsistências que normalmente ocorrem na transição de fases do processo de desenvolvimento de *software*. Como trabalhos futuros, utilizando as diretrizes propostas neste trabalho, pode-se destacar: (i) a formalização das diretrizes baseando na abordagem de Melo et al. (2015) e (ii) a implementação de uma ferramenta capaz de ler os modelos em i\* normativo e gerar o diagrama NorMAS-ML de normas correspondente.

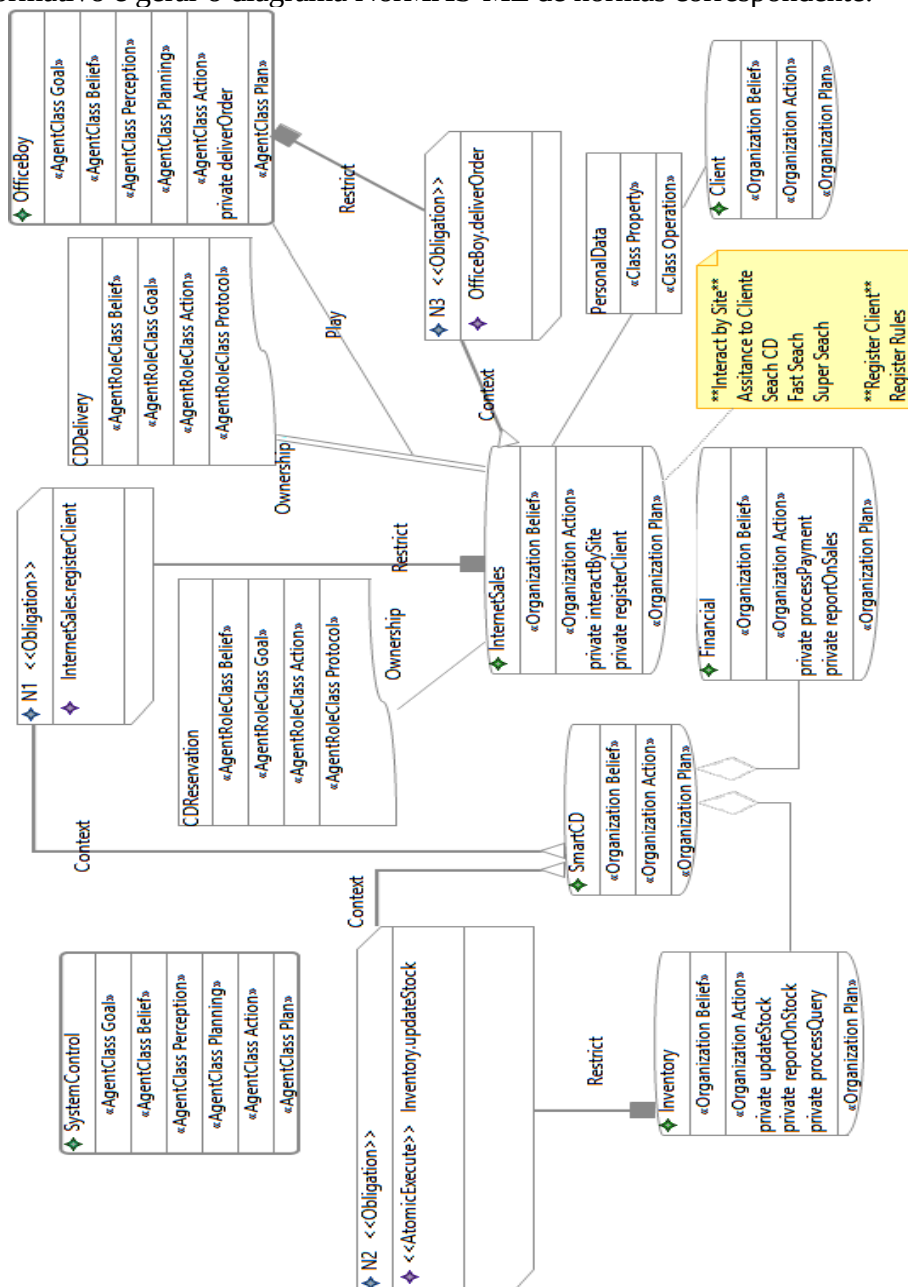


Figura 14. Diagrama NorMAS-ML de Normas para o SmartCD (Fonte: Próprios Autores)

## Referências

Aguilar, J., Garrigós, I., Mazón, J. and Trujillo, J. (2010) An MDA approach for goal-

- oriented requirement analysis in Web engineering, *J. Univers. Comput. Sci.*, vol. 16, no. 17, pp. 2475–2494, 2010.
- Alencar, F., Gianchetti, G. and Pastor, O. (2009). From i\* requirements models to conceptual models of a Model Driven Development process. In: *The Practice of Enterprise Modeling - Second IFIP WG 8.1 Working Conference, PoEM 2009, Proceedings*, pp. 99-114.
- Alencar, F.M.R., Pedroza, F., Castro, J.F.B. and Amorim, R.C.O. (2003) New Mechanisms for the Integration of Organizational Requirements and Object Oriented Modeling, VI WORKSHOP DE ENGENHARIA DE REQUISITOS, Piracicaba-SP, 2003.
- Boella, G., van der Torre, L. and H. Verhagen (2006) Introduction to normative multiagent systems, *Computational & Mathematical Organization Theory*, vol. 12, no. 2, pp. 71–79.
- Filho, C. G, Zisman, A. and Spanoudakis, G. (2003) Traceability Approach for i\* and UML Models. In: *Proceedings of 2nd International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'03)*, Portland.
- Franceto, S. (2005) Especificação e Implementação de uma Ferramenta para Elicitação de Requisitos de Software baseada na Teoria da Atividade. 2005. Disponível em: <<http://goo.gl/4wj9EZ>>. Acessado em 10 de outubro de 2017.
- Freire, E. S. S. and Cortés, M. I. (2016) Integrando Requisitos Organizacionais à Modelagem de Sistemas Multiagente Normativos. In: *10th Workshop-School on Agents, Environments, and Applications (WESAAC)*, 2016, Alagoas.
- Freire, E. S. S., Cortés, M. I., Gonçalves, E. J. T. and Lopes, Y. S. (2012) NorMAS-ML: A Modeling Language to Model Normative Multi-Agent Systems. In: *14th International Conference on Enterprise Information Systems (ICEIS)*, 2012, Wroclaw (Poland).
- Lucena, M., Castro, J., Silva, C., Alencar, F. and Santos, E. (2011) Stream: a strategy for transition between requirements models and architectural models. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 699-704. ACM, 2011.
- Melo, J., Sousa, A., Agra, C., Júnior, J., Castro, J. and Alencar, F. (2015) Formalization of Mapping Rules from iStar to Class Diagram in UML. In: *29th Brazilian Symposium on Software Engineering (SBES)*, 2015, Belo Horizonte.
- Richard, B. (2010) A deidealisation semantics for KAOS. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 267-274. ACM, March 2010.
- Siena, A., Maiden, N., Lockerbie, J., Karlsen, K., Perini, A. and Susi, A. (2008) Exploring the Effectiveness of Normative i\* Modelling: Results from a Case Study on Food Chain Traceability. In: *CAiSE '08 Proceedings of the 20th international conference on Advanced Information Systems Engineering* Pages 182 – 196.
- Silva, V., Braga, C. and Figueiredo, K. (2010) A Modeling Language to Model Norms. In: *Workshop on Coordination, Organization, Institutions and Norms in agent systems at International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS10)*, Toronto, p. 25-32.
- Sommerville, I. (2011) *Engenharia de Software*. 9 ed. São Paulo: Pearson Addison- Wesley, 2011.
- Sun, Z., Wang, J., He, K., Xiang, S. and Yu, D. (2010) A Model Transformation Method in Service-Oriented Domain Modeling. In *Proceedings of the 2010 21st Australian Software Engineering Conference*, pages 107-116. IEEE Computer Society, 2010.
- Yu. E. (2002) *Social Modeling and i\**. In: Faculty of Information, University of Toronto, Toronto, Canada M5S 3G6, 2002.