

# Especificação de Casos de Uso para a Modelagem de Requisitos de Sistemas Multiagente Normativos<sup>†</sup>

Emmanuel Sávio Silva Freire<sup>1</sup>, Patrícia Maria Barbosa Ferreira<sup>2</sup>

<sup>1</sup>Departamento de Ensino – Instituto Federal do Ceará (IFCE/*Campus* Morada Nova)  
Av. Prefeito Raimundo José Rabelo, 2717 – 62940-000 – Morada Nova – CE – Brasil

<sup>2</sup>Departamento de Ensino – Instituto Federal do Ceará (IFCE/*Campus* Aracati)  
Rodovia CE-040, Km 137, s/n – Aeroporto – 62800-000 - Aracati – CE – Brasil

savio.freire@ifce.edu.br, {savio.essf, pati270297}@gmail.com

**Abstract.** *Normative multiagent systems (NMASs) has stood out in complex system development. Considering that gathering and analysis phase is essential in development process of these systems, an extension of use case UML diagram was proposed to allow the modelling of functionalities provided by agents. Nevertheless, it was not proposed a template to specify use cases. Therefore, this paper discusses the necessity to define a template of use case capable to specify the functionalities for NMASs along with the formalization of this template using VDM++ method. Because it is a preliminary work, it does not have results to compare with related work.*

**Resumo.** *Os sistemas multiagente normativos (SMANs) tem se destacado no desenvolvimento de sistemas complexos. Considerando que a fase de levantamento e análise de requisitos é essencial no processo de desenvolvimento desses sistemas, uma extensão do diagrama UML de casos de uso foi proposta para permitir a modelagem das funcionalidades providas por agentes. Entretanto, não foi proposto um template para a especificação dos casos de uso. Com isso, este artigo discute a necessidade da definição de um template de casos de uso capaz de especificar as funcionalidades para os SMANs juntamente com a formalização desse template utilizando o método VDM++. Pelo fato de ser um trabalho preliminar, não se dispõem de resultados para comparar com trabalhos relacionados.*

## 1. Introdução

Os sistemas multiagente normativos (SMANs) [Boella, van der Torre e Verhagen 2006] têm sido utilizados para o desenvolvimento de sistemas complexos. Esses sistemas são formados por um conjunto de agentes inteligentes que interagem entre si para alcançar seus objetivos, considerando as normas que regulam o comportamento desses agentes. Logo, uma norma define as ações que podem, que devem e que não devem ser executadas [Silva, Braga e Figueiredo 2010]. Neste contexto, a Engenharia de Software tem definido técnicas, métodos e ferramentas para auxiliar o processo de desenvolvimento desses

---

<sup>†</sup>Este trabalho faz parte do Projeto número 5924 intitulado “Suporte para a Modelagem de Requisitos para a Modelagem de Casos de Uso para Projetos de Sistemas Multiagente Normativos” do Edital IFCE/PIBIC/2017. A autora Patrícia Ferreira agradece pelo suporte financeiro recebido pelo IFCE para a realização deste trabalho.

sistemas. Entretanto, o maior foco é em linguagens de modelagem e de implementação [Freire 2017]. Assim, a fase de levantamento e análise de requisitos tem sido pouco abordada. Segundo Sommerville (2011), a fase de levantamento e análise de requisitos é essencial para que os usuários do sistema a ser desenvolvido possam identificar e fornecer as suas necessidades para os analistas de requisitos.

Considerando a importância dos requisitos, a Linguagem de Modelagem Unificada (do inglês, *Unified Modeling Language* - UML) [UML 2017] possui o diagrama de casos de uso que dá suporte a fase de análise de requisitos, porém não considera os agentes nem as normas presentes nos SMANs. Assim, Freire (2017) propôs a extensão do diagrama UML de casos de uso, baseado na primeira extensão realizada por Guedes (2012), possibilitando a representação visual das normas e da interação delas com as outras entidades previstas no diagrama.

No entanto, não foi proposto um *template* para casos de uso para permitir a especificação das funcionalidades providas pelos agentes nem das normas que regulam o comportamento dessas entidades. Vale ressaltar que a especificação de requisitos engloba a descrição dos requisitos de usuário e de sistema em um documento utilizado durante a fase de implementação do sistema. Além disso, esse documento pode ser especificado por meio de linguagem natural, notações gráficas ou especificações matemáticas [Sommerville 2011].

Assim, esse artigo teve como objetivo discutir a necessidade da definição de um *template* para a especificação de casos de uso no contexto de desenvolvimento de SMANs. A definição desse *template* estaria vinculada a versão do diagrama de casos de uso proposto por Freire [2017] para auxiliar a especificação e a validação dos requisitos junto aos *stakeholders*. Além disso, pretende-se formalizar o *template* definido por meio do *Vienna Development Method ++* (VDM++) [Larsen et al. 2018], possibilitando a redução da ambiguidade inerente à especificação de requisitos. Vale ressaltar que esse método foi escolhido pois (i) apresenta consolidação em relação a sua semântica e sintaxe, (ii) apresenta conceitos relacionados à orientação a objetos, podendo ser extensível para o paradigma orientado a agentes, e (iii) tem sido amplamente utilizado na indústria [Overture 2017].

O presente artigo está estruturado como segue: a Seção 2 apresenta o referencial teórico acerca do diagrama de casos de uso para SMANs proposto por Freire (2017), dos *templates* utilizados para a especificação de casos de uso e do método VDM++. A Seção 3 discute a necessidade da definição de um *template* e o seu mapeamento para VDM++ considerando o diagrama de casos de uso proposto por Freire (2017). Finalmente, a Seção 4 apresenta as conclusões e os trabalhos futuros.

## 2. Referencial Teórico

### 2.1. Diagrama de Casos de Uso para Sistemas Multiagente Normativos

O diagrama de casos de uso proposto por Freire (2017) foi gerado por meio da extensão do metamodelo de Guedes (2012) por meio da inclusão dos elementos normativos definidos por Figueiredo e Silva (2010). Assim, a nova versão do diagrama permite a modelagem de requisitos para SMANs, considerando seus atores internos, suas normas e suas organizações. Mais especificamente, a modelagem de normas permite a identificação do conceito deontológico (o que é permitido, obrigado ou proibido), do contexto (organização) no

qual a norma está relacionada, do recurso (caso de uso) que está sendo regulamentado pela norma e da entidade (ator) que está sendo restringida pela norma.

A extensão considerou alterações nas sintaxes abstrata e concreta. Na sintaxe abstrata, foram incluídas novas metaclasses para representar os elementos normativos com as entidades (ator e caso de uso) previamente existentes no diagrama de casos de uso proposto por Guedes (2012). Para a sintaxe concreta, foram utilizados elementos gráficos para representar os novos elementos (*Organization, Norm, Context, Restrict, Resource e Ownership*). Um exemplo (veja a Figura 1) baseado no sistema “Ambiente Multiagente de Ensino-Aprendizagem” (AME-A) foi modelado por Freire (2017) utilizando a nova versão do diagrama. Vale ressaltar que não foi desenvolvida uma ferramenta de suporte à nova versão nem foi proposto um *template* para a especificação de casos de uso. Diversos autores, como Sommerville (2011), sugerem a utilização de documentação textual para essa especificação.

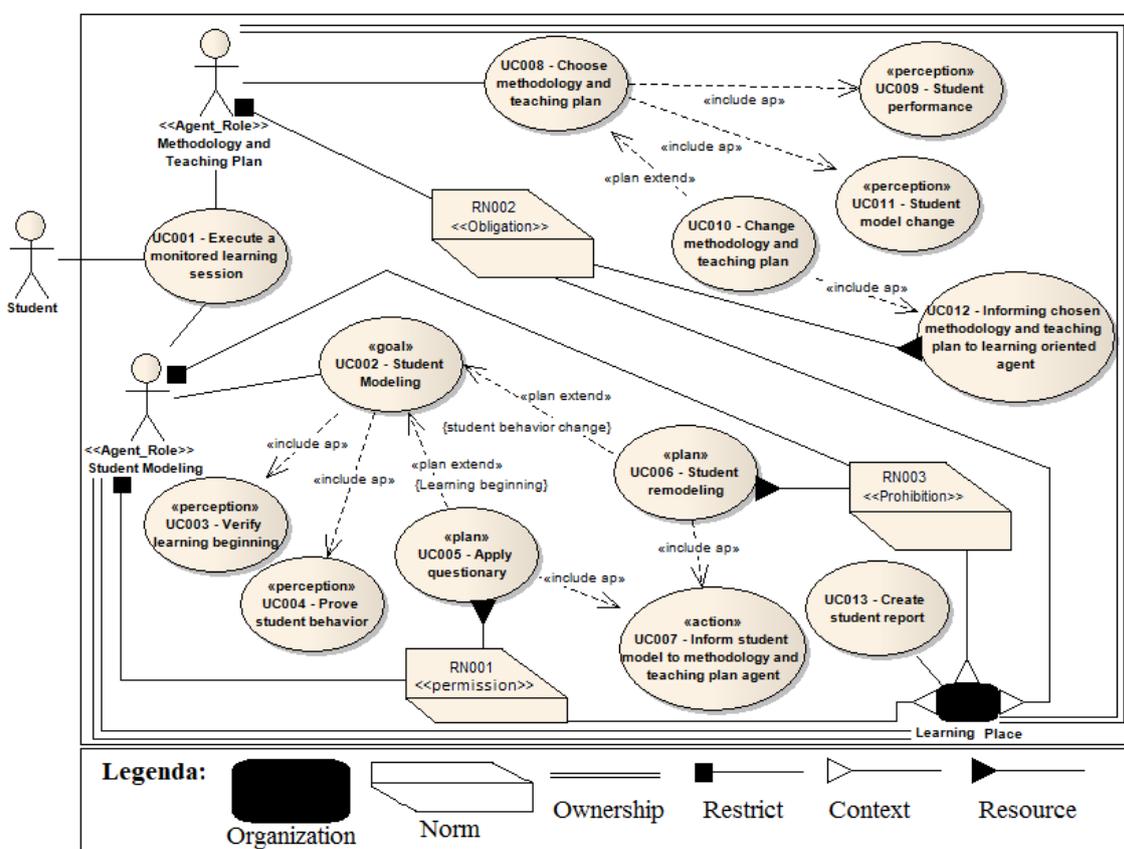


Figura 1. Um exemplo do diagrama de casos de uso proposto por Freire (2017)

## 2.2. Templates para Especificação de Casos de Uso

A especificação de requisitos engloba a descrição dos requisitos de usuário e de sistemas em um documento textual [Sommerville 2011]. Por um lado, os requisitos de usuário correspondem aos requisitos funcionais e não funcionais do sistema, descrevendo apenas o comportamento externo do sistema. Por outro lado, os requisitos de sistema são baseados nos requisitos de usuário e têm o intuito de serem utilizados para auxiliar no projeto do sistema, pois possuem uma especificação completa e detalhada do sistema. Vale ressaltar que um requisito funcional está relacionado com a funcionalidade que o sistema deve prover, enquanto o requisito não funcional representa as restrições aos serviços ou funções oferecidos do sistema.

Normalmente, a especificação de requisitos é realizada por meio de linguagem natural, notações gráficas (diagramas UML de caso de uso ou de sequência) em conjunto com anotações de texto e especificações matemáticas. Além disso, a especificação de um *template* auxilia na escrita dos requisitos. Mais especificamente, a linguagem UML [UML 2018] é utilizada para a notação gráfica de requisitos [Sommerville 2011], porém não possui uma estruturação específica para a documentação de um caso de uso. Por conta disso, existem diversas propostas de documentação por diferentes autores [Cockburn 2001] [Larman 2004] [Some 2010] [Sommerville 2011]. Entretanto, nenhuma delas dá suporte à especificação de casos de uso para sistemas multiagente (SMAs).

Consequentemente, Guedes (2012), considerando a sua versão do diagrama de casos de uso para SMAs, definiu um *template* para documentar os casos de uso internos, contendo os seguintes campos: (i) nome do caso de uso interno, (ii) seu estereótipo, (iii) o principal *AgentRole\_Actor* que o utiliza, (iv) os prováveis *AgentRole\_Actors* secundários, (v) outros atores que possam estar envolvidos no processo, (vi) as possíveis percepções associadas ao caso de uso interno, (vii) as possíveis ações associadas, (viii) os possíveis planos associados, (ix) pontos de extensão de plano (se existirem), (x) um resumo da função do caso de uso interno, (xi) as possíveis pré e pós-condições que devem ser satisfeitas antes e depois da execução do caso de uso interno, (xii) o fluxo principal, e (xiii) os possíveis fluxos alternativos de cada caso de uso interno. Vale ressaltar que esse *template* dá suporte parcial ao diagrama de casos de uso definido por Freire (2017), pois não contempla os elementos normativos incluídos na extensão desse autor.

### 2.3. Vienna Development Method++ (VDM++)

O *Vienna Development Method ++* (VDM++) é uma linguagem de especificação formal que permite a especificação de sistemas orientados a objetos com comportamento paralelo e *real-time* [Larsen et al. 2018]. Essa linguagem foi originada por meio da extensão de VDM-SL [Larsen et al. 1996]. A sintaxe concreta é descrita no Formalismo de Backus-Naur (BNF, do inglês *Backus-Naur Form*) e possui tipos de dados definidos. Além disso, é possível definir funções estáticas, ou seja, não é necessário a instância de objetos para a execução das funções. Os algoritmos definidos nesse método podem conter operações e funções. A diferença entre elas é em relação à utilização de variáveis locais e globais.

O trabalho de Wong, Mit e Sidi (2016) utilizou o método VDM++ para formalizar um *template* de caso de uso. Esse *template* era baseado nos elementos contidos no diagrama UML de casos de uso. As pré-condições e pós-condições presentes no *template* foram definidas para pré-condições e pós-condições contidas em VDM++. Com o objetivo de demonstrar a sintaxe do método, é apresentado o mapeamento de casos de uso para VDM++ proposto por Wong, Mit e Sidi (2016):

```
UcNameUML = {verb, noun}
UcNameUML. noun |→ classNameVDM++
if (genUcNameUML <> null) then
  genUcNameUML |→ subClsOfVDM++. className2
```

## 3. Discussão

Por meio dos *templates* de casos de uso, o analista de requisitos pode especificar detalhadamente as funcionalidades do sistema. Assim, é possível descrever as requisições dos usuários e as respostas correspondentes do sistema. Entretanto, essa especificação não

é retratada no diagrama de casos de uso, pois é possível modelar apenas as funcionalidades (casos de uso) em conjunto com os atores que irão interagir com o sistema. Mais especificamente, a versão do diagrama de casos de uso proposta por Freire (2017) permite a modelagem de normas, organizações, papéis de agente e atores. Os atores são externos ao sistema. Esse conceito já existia no diagrama UML de casos de uso. As entidades normas, organizações e papéis de agente foram incluídas no diagrama permitindo uma melhor visão do sistema e a relação entre as normas e as entidades do sistema. No entanto, não é possível detalhar as funcionalidades de cada caso de uso.

Neste sentido, a necessidade da elaboração de um *template* é requerida, visto que os *templates* existentes na literatura não são suficientes para o detalhamento das entidades contidas no diagrama de casos de uso proposto por Freire (2017). Mesmo o *template* definido por Guedes (2012) para SMAs, não possui campos específicos para o detalhamento dos elementos normativos, focando apenas nas percepções, nas ações e nos planos de agentes (representados por meio de papéis de agente) contidos em tais sistemas.

Portanto, a evolução do *template* de Guedes (2012) para contemplar os elementos normativos seria uma boa alternativa para a especificação de casos de uso e das normas que formam um SMAN. Logo, esse *template* passaria a ter um ou mais campos específicos para as normas e as relações entre elas e as outras entidades do diagrama de casos de caso. Neste caso, o *template* colaboraria para a especificação em linguagem natural das funcionalidades e das normas que as regulam. Os benefícios associados a esse novo *template* seriam: (i) um melhor detalhamento das normas e da organização, (ii) auxílio na especificação e na validação dos requisitos junto aos *stakeholders*, e (iii) uma melhor compreensão das funcionalidades que serão executadas pelos agentes e a relação delas com as normas.

Por se tratar de um *template* em linguagem natural, podem ocorrer erros oriundos da ambiguidade em relação à escrita e aos termos utilizados na especificação. Esses erros podem ser reduzidos pela formalização do *template* utilizando o método VDM++. Esse método foi utilizado por Wong, Mit e Sidi (2016) para formalizar o *template* utilizado para a especificação de casos de uso oriundos do diagrama UML de casos de uso. O resultado encontrado por esses autores foram as regras de mapeamento entre o *template* utilizado e o método VDM++, seguindo a sua sintaxe e a sua semântica. Esse mapeamento foi possível pois o método VDM++ possibilita a especificação de sistemas orientados a objetos.

Considerando que o paradigma orientado a agentes estende linguagens de modelagem e implementação, e *frameworks* criados especificamente para o paradigma orientado a objetos, a análise da semântica e da sintaxe de VDM++ é requerida com o intuito de identificar os pontos convergentes entre esse método e as entidades previstas no diagrama de casos de uso proposto por Freire (2017). Assim, o *template* definido para essa versão do diagrama seria transformado para o método formal VDM++. Finalmente, após a definição do *template* e do mapeamento para VDM++, será possível modelar um exemplo para validar o *template* e as regras de mapeamento.

#### 4. Conclusão e Trabalhos Futuros

Este artigo apresentou uma discussão acerca da definição de um *template* para a especificação de casos de uso para SMANs. Para tanto, é proposta a reutilização do *template* definido por Guedes (2012) com o intuito de incluir campos para as entidades previstas no diagrama de casos de uso proposto por Freire (2017). Além disso, após a definição do *template*, é proposta a

formalização do *template* utilizando o método VDM++. Esse método foi sugerido, pois já foi utilizado na formalização de *templates* de casos de uso para o diagrama UML de casos de uso. Espera-se como trabalhos futuros: (i) a definição do *template* de casos de uso, (ii) a análise do método VDM++ em relação às entidades dos SMANs, (iii) a formalização do *template* definido no item (i), e (iv) a verificação da consistência entre o *template* e as regras oriundas da formalização por meio de um exemplo de modelagem.

## Referências

- Boella, G., van der Torre, L. e Verhagen, H. (2006) “Introduction to normative multiagent systems”, Computational & Mathematical Organization Theory, vol. 12, no. 2, pp. 71–79.
- Cockburn, A. (2001) Writing Effective Use Cases. CA: Addison-Wesley, 2001.
- Freire, E. S. S. (2017) Extensão do Metamodelo do Diagrama de Casos de Uso para a Modelagem de Requisitos em Projetos de Sistemas Multiagente Normativos. In: Diana F. Adamatti; Mariela I. Cortés; Anarosa A. Brandão (Org.). E-book do WESAAC: 11th workshop-escola de sistemas de agentes, seus ambientes e aplicações. 1 ed. Rio Grande: Editora da FURG, 2017, v.1, p. 137-148.
- Guedes, G. T. A. (2012) Um Metamodelo UML para a Modelagem de Requisitos em Projetos de Sistemas Multiagentes, Tese de doutorado. Porto Alegre: UFRGS, Instituto de Informática.
- Larman, C. (2004) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd ed. USA: Prentice Hall, 2004.
- Larsen, P. G., Hansen B. S., Brunn, H., Plat, N., Toetenel, H., Andrews, D. J., Dawes, J., Parkin G. e others (1996) Information technology – Programming languages, their environments and system software interfaces – Vienna Development Method – Specification Language – Part 1: Base language. December 1996.
- Larsen, P. G., Lausdahl, K., Battle, N., Fitzgerald, J., Wolff, S., Sahara, S., Verhoef, M., Tran-Jørgensen, P.W. V., Oda, T. and Chisholm, P. (2018) Overture Technical Report Series – VDM-10 Language Manual. Disponível em: [https://raw.githubusercontent.com/overturetool/documentation/master/documentation/VDM10LangMan/VDM10\\_lang\\_man.pdf](https://raw.githubusercontent.com/overturetool/documentation/master/documentation/VDM10LangMan/VDM10_lang_man.pdf). Acessado em 20 de dezembro de 2017.
- Overture (2017) The Vienna Development Method. Disponível em: <http://overturetool.org/>. Acessado em 20 de dezembro de 2017.
- Silva, V., Braga, C. e 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, Toronto.
- Some, S. S. (2010) “Formalization of Textual Use Case Based on Petri Nets”, International Journal of Software Engineering and Knowledge Engineering, vol. 20, no. 5, pp. 695-737.
- Sommerville, I. (2011) Engenharia de Software. 9 ed. São Paulo: Pearson Addison-Wesley.
- UML (2017) Unified Modeling Language Specification, versão 2.5, OMG. Disponível em: <http://www.omg.org/spec/UML/2.5/PDF/>. Acessado em: 20 de dezembro de 2017.
- Wong, S. Y., Mit, E. and Sidi, J. (2016) Integration of Use Case Formal Template using Mapping Rules. In: Third International Conference on Information Retrieval and Knowledge Management, Malasya.