

Extensão do Metamodelo do Diagrama de Casos de Uso para a Modelagem de Requisitos em Projetos de Sistemas Multiagente Normativos

Emmanuel Sávio Silva Freire

Departamento de Ensino – Instituto Federal do Ceará (IFCE) /Campus Iguatu
Rua Deoclécio Lima Verde, s/n – Areias II – CEP 63.507-110 – Iguatu – CE – Brasil

savio.essf@gmail.com, savio.freire@ifce.edu.br

***Abstract.** Through requirements modeling is possible to understand the user's needs and map them in intended system. In this way, Unified Modeling Language (UML) has the use case diagram to support this phase. However, for multiagent systems (MAS) modeling, Guedes (2012) proposed some adjustments in this diagram's metamodel to allow the modeling of the typical entities found in these systems without regarding to the norms that regulate the behavior of these entities. This paper presents an extension of the metamodel of Guedes (2012) to allow the modeling of actors and use cases along with norms presented in the normative multiagent systems. In addition, a study case is used to represent this new version.*

***Resumo.** Por meio da modelagem de requisitos é possível entender as necessidades dos usuários e mapeá-las para o sistema que será desenvolvido. Neste sentido, a Unified Modeling Language (UML) possui o diagrama de casos de uso para dar suporte a essa fase. Entretanto, para a modelagem de sistemas multiagente, Guedes (2012) propôs algumas alterações para possibilitar a modelagem das entidades típicas encontradas nesses sistemas sem considerar as normas que regulam o comportamento dessas entidades. Este artigo tem como objetivo apresentar uma extensão do metamodelo de Guedes (2012) para possibilitar a modelagem de atores e casos de uso juntamente com as normas presentes nos sistemas multiagente normativos. Adicionalmente, um estudo de caso é utilizado para ilustrar a nova versão.*

1. Introdução

Os sistemas centrados em agente têm se tornado uma alternativa para o desenvolvimento de sistemas complexos (CASILLO, 2008). Esses sistemas, chamados de sistemas multiagente (SMAs), permitem que vários agentes de *software* interajam entre si para o compartilhamento ou disputa de um recurso do sistema (RUSSELL E NORVIG, 2004). Por isso, a Engenharia de Software tem definido técnicas, métodos e ferramentas para auxiliar o processo de desenvolvimento desses sistemas. Entretanto, vários autores (SILVA, 2004) (FIGUEIREDO e SILVA, 2010) (FREIRE et al., 2012) têm focado nas fases de modelagem e implementação sem considerar a fase de análise de requisitos.

Segundo Sommerville (2011), a fase de análise de requisitos possibilita que as necessidades do usuário sejam entendidas pelos analistas de requisitos e, posteriormente, sejam mapeadas para o sistema que será desenvolvido. Consequentemente, essa fase é

essencial para o entendimento das funcionalidades e deve anteceder as fases de modelagem e implementação de sistemas (BEZERRA, 2007). Portanto, os artefatos gerados durante essa fase devem servir como *input* para a fase de modelagem.

Neste contexto, a *Unified Modeling Language* (UML) (UML, 2016) possui o diagrama de casos de uso que dá suporte a fase de análise de requisitos. Por meio dele, é possível modelar os requisitos funcionais utilizando casos de uso e identificar quais os papéis que terão acesso a essas funcionalidades por meio de atores. Vale ressaltar que os atores devem ser externos ao sistema (UML, 2016). Por meio da análise desse diagrama e de seu respectivo metamodelo, Guedes (2012) verificou que o mesmo não dava suporte a modelagem de agentes. Conseqüentemente, essa autora estendeu o metamodelo da UML por meio da criação de novas metaclasses e relacionamentos. Assim, foi definido o conceito de ator interno e caso de uso interno que representam uma funcionalidade e o papel de agente que serão executados por um agente, respectivamente.

A nova versão do diagrama permite a modelagem de papéis de agente juntamente com suas características estruturais (objetivos e crenças) e comportamentais (planos e ações), considerados como casos de uso internos por Guedes (2012). Entretanto, o diagrama não representa as outras entidades típicas dos sistemas multiagente nem as normas que restringem o seu comportamento. Portanto, este diagrama não permite a modelagem de requisitos para sistemas multiagente normativos (SMANs). Vale ressaltar que esses sistemas são compostos pelas entidades típicas dos SMA cujo comportamento é regulado por meio de normas.

Segundo Freire e Cortés (2016), as normas podem ser vistas na fase de requisitos como regras de negócio. Alguns autores (SULL et al., 2011) (MORGAN, 2011) tentaram definir modelos para que a representação dessas regras não fosse feita de forma textual, mas sim por meio de elementos gráficos. Portanto, considerando que a versão de Guedes (2012) não dá suporte a modelagem de requisitos de sistemas multiagente normativos, o presente artigo tem como objetivo apresentar a extensão conservativa do metamodelo de Guedes (2012) para permitir a modelagem do diagrama de casos de uso utilizando papéis de agente, organizações e normas.

Com isso, será possível representar os requisitos funcionais que serão executados pelos atores externos e os requisitos associados aos atores internos (papel de agente e organização). Além disso, ao incluir as normas no diagrama de casos de uso, tanto o analista de requisitos quanto o usuário poderão verificar quais as funcionalidades que estarão sendo restringidas por cada uma das normas. Assim, espera-se que as atividades de elicitação e de validação de requisitos possam ser facilitadas por meio dessa nova versão do diagrama. Além da extensão do metamodelo, foi definida a sintaxe concreta (elementos gráficos) e modelado um estudo de caso para ilustrar a nova versão do diagrama. As contribuições desse artigo são as seguintes:

- Definição das metaclasses que representam as entidades Organização e Normas no metamodelo definido por Guedes (2012)
- Inclusão de elementos gráficos para representar as metaclasses definidas no item anterior,
- Modelagem de um estudo de caso para exemplificar a utilização do novo diagrama de casos de uso.

Este artigo está organizado como segue. A seção 2 apresenta o conceitual referente aos sistemas multiagente normativos. Em seguida, o metamodelo proposto por Guedes (2012) é abordado na seção 3. Na seção 4, o metamodelo proposto nessa pesquisa para a modelagem de requisitos para SMANs é detalhado. Um estudo de caso modelado utilizando o metamodelo definido na seção anterior é apresentado na seção 5. Na seção 6, são discutidos os trabalhos relacionados. Finalmente, as conclusões e trabalhos futuros são abordados na seção 7.

2. Sistemas Multiagente Normativos

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 meio de atuadores. Quando existem vários agentes inseridos em um mesmo ambiente, cooperando ou disputando entre si, e trocando informações, tem-se um sistema multiagente (SMA).

Entretanto, os agentes inseridos em um SMA precisam ter o seu comportamento regulado para que o sistema possa atingir os seus objetivos. Portanto, foram definidas normas nesses sistemas. Segundo Figueiredo e Silva (2010), uma norma é um elemento que restringe o comportamento de agentes, organizações e suborganizações durante um período de tempo, e aplica sanções quando violadas ou cumpridas. Vale ressaltar que, esse conceito de normas está relacionado com a abordagem *enforcement*, ou seja, os agentes podem violar as normas, maximizando a sua autonomia (HOLLANDER e WU, 2011). Neste sentido, a junção das entidades de um SMA com as normas formam um sistema multiagente normativo (SMAN).

A restrição de comportamento envolve a definição das ações que são obrigadas, permitidas e proibidas de serem executadas pelas entidades do sistema. Segundo Figueiredo e Silva (2010), uma norma é composta pelos seguintes elementos estáticos:

- Conceito Deontico: representa o conceito deontico (obrigação, permissão ou proibição) associado à norma.
- Contexto: uma norma deve ser definida em um determinado contexto (área de restrição), que pode ser um ambiente ou uma organização. Assim, os agentes que estejam no contexto da norma terão o seu comportamento restringido.
- Entidades Envolvidas: representam as entidades (indivíduos ou grupo de indivíduos) as quais possuem seu comportamento restringido pela norma.
- Ações: representam as ações comunicativas (envio e recebimento de mensagens) e não-comunicativas (acessar e modificar um recurso, entrar em uma organização, se deslocar para outro ambiente etc.) que serão restringidas pela norma.
- Restrição de Ativação: representa o período em que a norma deverá estar ativa, ou seja, o período em que a restrição de comportamento deve ser cumprida.
- Sanções: quando uma norma é violada/cumprida, a entidade que violou/seguiu essa norma pode receber uma punição/uma recompensa, respectivamente.

3. Modelagem de Requisitos de Sistemas Multiagente

Segundo Sommerville (2011), a fase de requisitos possibilita que as necessidades do usuário possam ser conhecidas e discutidas para, posteriormente, serem mapeadas para

um sistema computacional. Para Berenbach (2009), um bom requisito deve (i) ser implementável, (ii) ter uma única interpretação, (iii) ser testável, (iv) ser consistente, (v) apresentar a dependência com outros requisitos e (vi) ser rastreável.

Além disso, Sommerville (2011) indica que a fase de requisitos deve anteceder as fases de modelagem e projeto. Com isso, os resultados oriundos da fase de requisitos podem ser utilizados como *inputs* para essas fases. Neste sentido, a UML possui o diagrama de casos de uso para suporte a fase de requisitos. Por meio dele, é possível representar os requisitos funcionais como casos de uso e os usuários do sistema como atores (UML, 2016). Vários autores (RODRIGUEZ, INSFRAN e CERNUZZI, 2011) (BERENBACH, 2009) (SOMMERVILLE, 2011) (BEZERRA, 2007) indicam a utilização do diagrama de casos de uso para a modelagem de requisitos.

Entretanto, a modelagem de requisitos para sistemas multiagente tem sido pouco abordada pela literatura. Dentre os principais autores, Yu (2002) ressalta a importância na elicitação de requisitos para tais sistemas. Além disso, o comportamento dos agentes precisa ser especificado para dar suporte às fases de modelagem e de implementação (PAPASIMEON e HEINZE, 2003). Neste sentido, Guedes (2012) propôs uma extensão do metamodelo do diagrama de casos de uso para dar suporte à modelagem de requisitos funcionais oriundos de sistemas multiagente. Assim, foi definido um conjunto de metaclasses para representar os agentes e suas ações. Para os agentes, foi definida a metaclassa *AgentRole_Actor* e para as ações, foi criada a metaclassa *InternalUseCase*. A figura 1 apresenta as metaclasses (em verde) propostas por Guedes (2012).

Por consequência, a nova versão do diagrama permite a modelagem de papéis de agente juntamente com suas características estruturais (objetivos e crenças) e comportamentais (planos e ações), considerados como casos de uso internos. Vale ressaltar que essas alterações foram necessárias para representar os agentes como entidades internas do sistema, visto que, no diagrama original de casos de uso, apenas agentes externos podiam ser representados.

4. Extensão do Diagrama de Casos de Uso para a Modelagem de Requisitos para Sistemas Multiagente Normativos

Considerando o metamodelo estendido por Guedes (2012), foram identificadas as seguintes limitações:

- O diagrama de casos de uso estendido possibilita a modelagem de casos de uso internos para agentes. Entretanto, não suporta a modelagem de organizações e suas ações. Vale ressaltar que, segundo Silva (2004), uma organização possui comportamento semelhante a um agente. Logo, podem ser definidas ações (casos de uso interno) para a mesma;
- A versão não suporta a modelagem de normas e suas interações com as entidades que fazem parte do diagrama. Consequentemente, não é possível modelar um diagrama de casos de uso para sistemas multiagente normativos;
- Guedes (2012) não propôs uma ferramenta para dar suporte à modelagem do diagrama estendido. Foi utilizada uma ferramenta já existente para modelar as alterações propostas por ela.

Diante do exposto, foram definidas as metaclasses *Organization*, *Ownership*, *Norm*, *Restrict*, *Resource* e *Context* com o objetivo de possibilitar a modelagem de requisitos para SMAN. Adicionalmente, as metaclasses *AgentRole_Action* e *InternalUseCase* foram alteradas. Vale ressaltar que foi realizada uma extensão conservativa, ou seja, todos os conceitos presentes na versão anterior continuam válidos nessa nova versão. A figura 1 apresenta a nova versão do metamodelo do diagrama de casos de uso após essas alterações.

Inicialmente, foi alterada a metaclasses *AgentRole_Action*. Essa metaclasses foi definida por Guedes (2012) para representar papéis de agente. Entretanto, Silva (2004) afirma que um papel de agente pode ser exercido por um agente ou por uma organização. Portanto, foi alterada a semântica dessa metaclasses para possibilitar a representação tanto de papéis de agente exercidos por cada um deles quanto por organizações. Além disso, por conta dessa alteração, é possível definir casos de uso internos (*InternalUseCase*) para papéis de agente a serem exercidos por organizações.

Em seguida, a metaclasses *Organization* foi definida como uma especificação das metaclasses *Element* e *NameSpace* da UML. Por meio dessa metaclasses, é possível representar organizações e suborganizações de agentes. De acordo com Silva (2004), uma organização é composta por agentes e é responsável por definir os papéis de agente que serão exercidos por cada um deles. Logo após, foi incluída a metaclasses *Ownership*, que foi definida por Silva (2004) como uma especialização das metaclasses *DirectedRelationship* e *NamedElement*. A metaclasses *Ownership* representa os papéis de agente que estão subordinados a uma organização.

Do ponto de vista da engenharia de requisitos, a inclusão da metaclasses *Organization* possibilita que o analista de requisitos possa verificar o comportamento dessa entidade em conjunto com os agentes que a compõem. Com isso, a identificação e a definição das ações a serem executadas pelos agentes podem ser facilitadas por meio da presença desse novo elemento no diagrama. Além disso, a metaclasses *Ownership* permite que o diagrama apresente maiores detalhes acerca do comportamento entre os papéis de agente e a organização. Conseqüentemente, esses novos elementos podem auxiliar no entendimento do sistema por parte do usuário e na validação dos requisitos funcionais por parte do usuário e do analista de requisitos.

Após a definição dos elementos associados à organização, foram incluídos os elementos estáticos (seção 2) que compõem uma norma. Segundo Freire e Cortés (2016), uma norma pode ser vista como uma regra de negócio, pois representa as restrições inerentes ao processo de negócio que será mapeado para o sistema. Neste sentido, a representação de regras de negócio ocorre, comumente, por meio de representação textual associadas ao diagrama por meio de anotações. Entretanto, essa visualização dificulta o entendimento da regra por parte do usuário. Portanto, decidiu-se incluir os principais elementos estáticos da norma no diagrama de casos de uso para facilitar a representação e a validação das normas tanto pelo analista de requisitos quanto pelo usuário.

Dentre os elementos estáticos, foram escolhidos: (i) o conceito deontico, pois representa o tipo de restrição que será imposta pela norma, (ii) as entidades restringidas, pois identifica quais as entidades que terão seu comportamento regulado por uma norma, (iii) o recurso, pois indica a propriedade da entidade que está sendo regulado por uma norma, e (iv) o contexto, pois apresenta a área na qual uma norma é válida. Essa decisão

de projeto foi tomada, pois esses elementos possuem relação com as entidades presentes no diagrama de casos de uso (*InternalUseCase*, *AgentRole_Action* e *Organization*). Além disso, os outros elementos (restrição de ativação e sanções) representam o período de validade de uma norma e suas recompensas e punições, respectivamente. Provavelmente, esses elementos podem ser modelados em outro diagrama (Freire et al., 2012) mais específico para a representação total das normas.

A seguir, é apresentada cada uma das metaclasses incluídas no metamodelo:

- *Norm*: Foi definida como uma metaclassa que possui uma relação de generalização com as metaclasses *Element* e *Namespace*. Assim, é possível representar uma norma que pode regular o comportamento de agentes quando formam um grupo ou agem individualmente. Além disso, uma norma possui um conceito deôntico associado (permissão, proibição e obrigação). Tais conceitos são associados à norma por meio de uma classe *Enumeration*.
- *Restrict*: Representa as entidades restringidas por uma norma. Neste caso, apenas as metaclasses *Norm* e *AgentRole_Actor* podem fazer parte desse relacionamento.
- *Resource*: Representa as propriedades das entidades que podem ser restringidas por uma norma. Neste caso, apenas casos de uso internos (*InternalUseCase*) e uma norma podem participar desse relacionamento, pois representam as funcionalidades que um papel de agente (*AgentRole_Actor*) pode executar.
- *Context*: Representa o contexto de aplicação de uma norma. Neste caso, apenas uma organização (*Organization*) pode ser o contexto de uma norma.

Além disso, as metaclasses *Restrict*, *Resource* e *Context* são especializações das metaclasses *DirectedRelationship* e *Namespace* da UML.

5. Estudo de Caso

Nesta seção, um estudo de caso é apresentado com o objetivo de ilustrar a nova versão do diagrama de casos de uso. Para tanto, utilizou-se o contexto do ambiente AME-A (Ambiente Multiagente de Ensino-Aprendizagem) (D'AMICO et al., 1998). Este ambiente foi escolhido, pois foi utilizado por Guedes (2012) para apresentar as alterações propostas na sua extensão. Além disso, foram propostas alterações para esse ambiente para representar os aspectos normativos abordados por essa pesquisa.

Vale ressaltar que a modelagem do AME-A utilizando as metaclasses propostas implicaria em um diagrama contendo muitos relacionamentos, dificultando o seu entendimento e sua validação. Portanto, optou-se por utilizar os seguintes elementos gráficos para representar a semântica definida em cada metaclassa:

- *Organization*: foi definida como um retângulo de bordas arredondadas preenchido (Figura 2-a).
- *Norm*: foi definida por Freire et al. (2012) como um retângulo sólido com um ângulo no canto superior direito e outro no canto inferior esquerdo (Figura 2-b). No compartimento superior contém o nome da norma juntamente com o conceito deôntico associado representado por um estereótipo (*<<permission>>*, *<<obligation>>* ou *<<prohibition>>*).

- *Ownership*: é apresentado por Silva (2004) como uma linha dupla que liga o proprietário e o membro (Figura 2-c).
- *Restrict*: foi definido por Freire et al. (2012) como uma linha simples com um quadrado preenchido em uma extremidade (Figura 2-d). O quadrado preenchido indica a entidade que terá o comportamento restringido pela norma.
- *Context*: foi definido por Freire et al. (2012) como uma linha simples com um triângulo não preenchido e invertido em uma extremidade (Figura 2-e). O triângulo não preenchido indica o contexto no qual a norma será aplicada.
- *Resource*: foi definido como uma linha simples com um triângulo preenchido e invertido em uma extremidade (Figura 2-f). O triângulo preenchido indica o recurso que está sendo restringido pela norma.

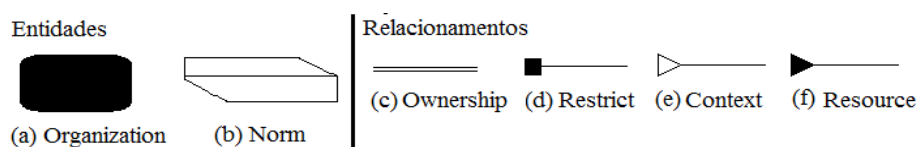


Figura 2. Elementos gráficos para o diagrama de casos de uso (Fonte: Próprio autor)

5.1. Ambiente Multiagente de Ensino-Aprendizagem (AME-A)

Segundo D’Amico et al. (1998), o ambiente AME-A possui um conjunto de agentes que cooperam para auxiliar no processo de aprendizagem dos alunos. Esses agentes podem interagir tanto com professores quanto com alunos. O objetivo principal do ambiente é possibilitar que os alunos consigam obter um melhor aprendizado por meio da ajuda de agentes que monitorem o processo de aprendizagem.

Neste sentido, o professor pode criar atividades de aprendizagem e/ou avaliar os seus alunos com o auxílio de um agente que estiver executando o papel *Teacher’s Tools*. Quanto aos alunos, os mesmos podem executar uma sessão de aprendizagem não monitorada ou monitorada. Por um lado, uma sessão de aprendizagem não monitorada é auxiliada pelo papel de agente *Unsupervised Learning*, apresentando o conteúdo a ser estudado.

Por outro lado, uma sessão de aprendizagem monitorada é auxiliada por meio dos seguintes papéis de agente: (i) *Student Modeling* (SM), que modela o perfil do aluno de forma dinâmica; (ii) *Methodology and Teaching Plan* (MTP), responsável por seleccionar e alterar a metodologia e o plano de ensino do aluno de acordo com o seu rendimento; (iii) *Learning Orientation* (LO), selecciona o material de aula considerando a metodologia escolhida; (iv) *Learning Analysis* (LA), verifica o rendimento do aluno e (v) *Knowledge Application Orienting* (KAO), aplica uma avaliação geral englobando todos os tópicos estudados durante a sessão de aprendizagem.

Adicionalmente, foi definida a organização *Learning Place* responsável por gerar relatórios contendo o rendimento dos alunos que realizaram sessões de aprendizagem monitorada. Todos os papéis descritos anteriormente pertencem a essa organização.

5.2. Identificação de Requisitos e Regras de Negócio para o AME-A

De acordo com a descrição do ambiente apresentada na seção anterior, foram definidos os requisitos funcionais (Quadro 1) e as regras de negócio (Quadro 2) para o AME-A.

Vale ressaltar que devido ao limite do número de páginas, foram listados apenas os requisitos funcionais referentes aos papéis de agente *Student Modeling* e *Methodology and Teaching Plan*.

Quadro 1. Requisitos funcionais para o ambiente AME-A (Fonte: Próprio autor)

Requisito	Descrição
RF001 – Execute a monitored learning session	O sistema deve permitir que o aluno possa executar uma sessão de aprendizagem monitorada com o auxílio dos agentes <i>Student Modeling</i> , <i>Methodology and Teaching Plan</i> , <i>Learning Orientation</i> , <i>Learning Analysis</i> e <i>Knowledge Application Orienting</i> .
RF002 – Verify learning beginning	O papel de agente <i>Student Modeling</i> deve perceber quando a sessão de aprendizagem é iniciada e disparar o plano de aplicação de questionário.
RF003 – Probe student behavior	O papel de agente <i>Student Modeling</i> deve perceber a mudança de comportamento do aluno e remodelar o perfil do aluno quando necessário.
RF004 – Student model or student performance change	O papel de agente <i>Methodology and Teaching Plan</i> deve perceber a mudança de modelo do aluno ou quando o desempenho for baixo.
RF005 – Create report about student performance	A organização <i>Learning Place</i> deve gerar relatórios contendo os resultados dos alunos.

Quadro 2. Regras de Negócio para o ambiente AME-A (Fonte: Próprio autor)

Regras de Negócio	Descrição
RN001 – Applying questionnaire	Os agentes vinculados ao papel <i>Student Modeling</i> da organização <i>Learning Place</i> devem aplicar o questionário assim que a sessão de aprendizagem for iniciada.
RN002 – Information chosen methodology and teaching plan	Os agentes vinculados ao papel <i>Methodology and Teaching Plan</i> da organização <i>Learning Place</i> devem informar sobre a mudança de metodologia e plano de ensino assim que estes forem alterados.
RN003 – Student Remodeling	Os agentes vinculados ao papel <i>Student Modeling</i> da organização <i>Learning Place</i> não devem remodelar o perfil de um estudante que está com notas acima de sete.

5.3. Diagrama de Casos de Uso para o AME-A

Considerando os requisitos funcionais e as regras de negócio definidas na seção anterior, a figura 3 apresenta o diagrama de casos de uso contendo os elementos apresentados por Guedes (2012) em conjunto com os elementos normativos. Vale ressaltar que as regras apresentadas no quadro 2 foram transformadas em normas conforme detalhamento a seguir:

(i) RN001: Os agentes vinculados ao papel *Student Modeling* (**entidade envolvida**) da organização *Learning Place* (**contexto**) podem (**conceito deôntico**)

aplicar (**ação**) o questionário (**recurso**) assim que a sessão de aprendizagem for iniciada (**restrição de ativação**).

(ii) RN002: Os agentes vinculados ao papel *Methodology and Teaching Plan* (**entidade envolvida**) da organização *Learning Place* (**contexto**) devem (**conceito deôntico**) informar (**ação**) sobre a mudança de metodologia e plano de ensino (**recurso**) assim que estes forem alterados (**restrição de ativação**).

(iii) RN003: Os agentes vinculados ao papel *Student Modeling* (**entidade envolvida**) da organização *Learning Place* (**contexto**) não devem (**conceito deôntico**) remodelar (**ação**) o perfil de um estudante (**recurso**) que está com média acima de sete (**restrição de ativação**).

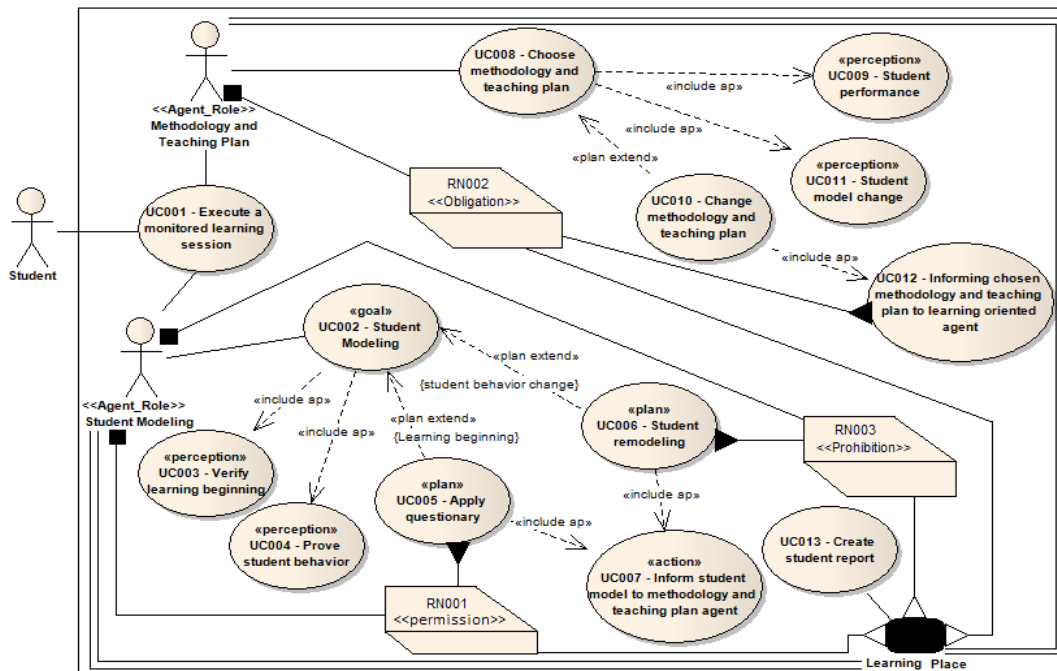


Figura 3. Diagrama de casos de uso para o ambiente AME-A (Fonte: Próprio autor)

6. Trabalhos Relacionados

Nesta seção, é apresentada a análise dos principais trabalhos que abordam a modelagem de requisitos para sistemas multiagente normativos utilizando como base o diagrama de casos de uso da UML.

Wagner (2003) definiu a linguagem de modelagem AORML (*Agent-Object-Relationship Modeling Language*), cujas entidades (evento, ação, alegação, compromisso, agente ou objeto) são definidas como estereótipos da metaclassa *Class* da UML. Mesmo possuindo um modelo externo, responsável por apresentar a visão mais externa e geral do sistema, não representa os requisitos por meio de diagrama de casos de uso, mas sim, utilizando diagrama de atividades. Por outro lado, a 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. Entretanto, mesmos apresentando bons mecanismos para representar papéis e agentes, essa linguagem não dá suporte a modelagem de requisitos por meio do diagrama de casos de uso.

A NormML (*Normative Modeling Language*) (FIGUEIREDO e SILVA, 2010) é uma linguagem de modelagem de normas para SMANs, definida a partir da extensão do metamodelo da *SecureUML*. Como essa linguagem foca apenas nos aspectos estáticos das normas, não considera as outras entidades que fazem parte de um SMAN. Logo, não possui suporte para a modelagem de normas como regras de negócio no diagrama de casos de uso. Entretanto, Silva (2004) propôs a linguagem MAS-ML, por meio de uma extensão conservativa da UML. Mesmo possuindo vários diagramas estruturais e dinâmicos, a linguagem não dá suporte à modelagem de requisitos.

Diante do exposto, apenas Guedes (2012) apresentou uma extensão do diagrama de casos de uso para a modelagem de requisitos para projetos de SMAs, porém não considerou as normas utilizadas para regular o comportamento dessas entidades. Portanto, a presente versão do diagrama de casos de uso consegue suprir essa limitação.

7. Conclusão e Trabalhos Futuros

Este artigo apresenta uma nova versão do diagrama de casos de uso para a modelagem de requisitos para projetos de sistemas multiagente normativos. Para tanto, o metamodelo proposto por Guedes (2012) foi estendido por meio da inclusão das metaclasses: (i) *Organization*, (ii) *Norm*, (iii) *Context*, (iv) *Restrict*, (v) *Resource*, e (vi) *Ownership*. Adicionalmente, o conceito de caso de uso interno foi alterado para possibilitar que um caso de uso possa ser relacionado com uma organização.

Em seguida, foram utilizados elementos gráficos para representar cada uma das novas metaclasses. Vale ressaltar que alguns dos elementos gráficos utilizados foram definidos por Silva (2004) e Freire et al. (2012). Finalmente, um estudo de caso, baseado no sistema AME-A, foi utilizado para representar as novas entidades em conjunto com as já existentes no diagrama. Por meio dessa extensão, é possível modelar os requisitos funcionais dos atores, dos agentes e das organizações juntamente com os principais elementos estáticos da norma. Com isso, o analista de requisitos pode incluir maiores detalhes acerca do sistema, facilitando o seu entendimento e a sua validação junto aos seus usuários.

Como trabalhos futuros, pode-se destacar: (i) a implementação de uma ferramenta para dar suporte à modelagem do novo diagrama de casos de uso utilizando a abordagem dirigida por modelos semelhante à proposta por Freire et al. (2012), e (ii) a integração entre o metamodelo apresentado nesse artigo com o de outra linguagem baseada em UML (por exemplo, MAS-ML (SILVA, 2004), NorMAS-ML (FREIRE et al., 2012)). Assim, pode-se combinar de forma automatizada as fases de requisitos e modelagem.

Referências

- Berenbach, B. (2009) *Software & Systems Requirements Engineering in Practice*, New York: McGraw-Hill.
- Bezerra, E. (2007) *Princípios de Análise e projeto de sistemas com UML*, 2ª Ed., Rio de Janeiro: Elsevier. p. 111.
- 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*. Master's Thesis: Department of Computer Science Faculty of Mathematics, Physics and Informatics Comenius University.

- D'Amico, C. B., Pereira, A. S., Geyer, C. F. R. e Vicari, R. (1998) Adpting Teaching Strategies in a Learning Environment on WWW. In: Proceedings of the WebNet Word Conference of the WWW, Internet & Intranet. Florida, USA.
- Figueiredo, K. e Silva, V. T. (2010) NormML: A Modeling Language to Model Norms. In: 1st Workshop on Autonomous Software Systems. Salvador, Brazil.
- Freire, E. S. S., Cortés, M. I., Gonçalves, E. J. T. e Lopes, Y. S. (2012) NorMAS-ML: A Modeling Language to Model Normative Multi-Agent Systems. In: 14th International Conference on Enterprise Information Systems (ICEIS), Wroclaw.
- Freire, E. S. S. e Cortés, M. I. (2016) Integrando Requisitos Organizacionais à Modelagem de Sistemas Multiagente Normativos. In: X Workshop-Escola de Sistemas de Agentes, seus Ambientes e apliCações. Maceió, Brasil.
- 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.
- Hollander, C. e Wu, A. (2011) The Current State of Normative Agent-Based Systems. *Journal of Artificial Societies and Social Simulation*, vol. 14, n. 2.
- Morgan, T. (2001) *Business Rules and Information Systems: Aligning IT with Business Goals*. Addison Wesley.
- Papasimeon, M. e Heinze, C. (2003) *Specifying Requirements in Multi-Agent System with Use Cases*. Technical Report February, Defence Science and Technology Organisation.
- Rodriguez, L., Insfran, E. e Cernuzzi, L. (2011) *Requirements Modeling for Multi-Agent Systems*. Multi-Agent Systems Modeling, Control, Programming, Simulations and Applications.
- Russell, S. e Norvig, P. (2003) *Artificial Intelligence: A Modern Approach*, 2nd Ed., Upper Saddle River, NJ: Prentice Hall, ISBN 0-13-790395-2.
- 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.
- Sommerville, I. (2011) *Engenharia de Software*. 9 ed. São Paulo: Pearson Addison-Wesley.
- Sull, R. D., Azevedo, L. G., Cappelli, C. e Baião, F. (2011) *Estudo de Linguagens para Representação de Regras de Negócio: Foco na SBVR*. Relatórios Técnicos do DIA/UNIRIO, N. 0014/2011.
- UML (2016) *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 2016.
- Wagner G. (2003) "The Agent-Object-Relationship Meta-Model: Towards a Unified View of State and Behavior", *Information Systems*. 28 (5), pp.475–504.
- Yu, E. (2002). *Agent-Oriented Modelling: Software Versus World*, In: *Proceedings of the Agent-Oriented Software Engineering (AOSE'01)*, Edited by Wooldridge, M., Weiss, G. and Ciancarini, P., LNAI, Vol. 2222, Springer-Verlag, p. 206 – 225.