

Dynamic Modeling of Multi-Agent Systems Using MAS-ML Tool

Francisco R. O. de Lima, Állan R. Feijó, Robert M. Rocha Jr, Igor B. Nogueira, Enyo J. T. Gonçalves, Emmanuel S. S. Freire, Mariela I. Cortés

Grupo de Engenharia e Sistemas Inteligentes (GESSI)
Departamento de Computação – Universidade Estadual do Ceará (UECE)
Fortaleza, Brasil

{us.robson7, allanfeijo1987, robstermarinho, igor.bnog, savio.essf}@gmail.com, enyo@ufc.br, mariela@larces.uece.br

Abstract — Given the diversity of entities comprising multi-agent systems (MAS), the modeling of the dynamic aspects is complex and error prone. Thus, the existence of a tool capable of modeling of these systems and validating them automatically, can be crucial in order to increase the productivity. The goal of this work is present the evolution of MAS-ML tool to provide the support to the dynamic diagrams of sequence and activities defined on the MAS-ML 2.0 language.

Keywords — multi-agent system; MAS-ML Tool; dynamic modeling;

I. INTRODUÇÃO

Em um cenário cada vez mais complexo, sistemas multi-agente (SMA) vem sendo cada vez mais utilizados para lidar com essa complexidade, tanto na indústria quanto na academia. O termo Sistema Multi-Agente (SMA) refere-se à subárea de Inteligência Artificial que investiga o comportamento de um conjunto de agentes autônomos, objetivando a solução de um problema que está além da capacidade de um único agente [1].

Neste contexto, várias linguagens de modelagem, frameworks de implementação e ferramentas foram criados para auxiliar o desenvolvimento de SMAs. Dentre eles, a linguagem MAS-ML (Multi-Agent System Modeling Language) [2] é capaz de modelar SMAs através dos diagramas estáticos de classe, papéis, organização, e dinâmicos de sequência e atividades [3]. A linguagem MAS-ML 2.0 [4] trata-se de uma extensão para permitir a modelagem de agentes com diversas arquiteturas internas [1], e seus papéis. Na sua versão atual, a ferramenta de suporte a modelagem MAS-ML tool [4] [5] não contempla nenhum diagrama dinâmico previsto na linguagem de modelagem MAS-ML 2.0, impossibilitando a modelagem do comportamento do sistema em tempo de execução. O presente artigo apresenta a evolução da ferramenta MAS-ML tool relacionada ao desenvolvimento dos diagramas dinâmicos de MAS-ML 2.0. O artigo é organizado como segue: Na Seção 2 é apresentado o referencial teórico. Na Seção 3, a evolução da ferramenta é apresentada. Na Seção 4, um estudo de caso é ilustrado. Na Seção 5, os trabalhos relacionados são comparados com as contribuições deste artigo. E por fim, na Seção 6, são apresentados as conclusões e os trabalhos futuros.

II. REFERENCIAL TEÓRICO

A. MAS-ML 2.0

MAS-ML [2] é uma linguagem de modelagem que estende a UML [6] e incorpora o conceito de agente definido no framework conceitual TAO (Taming Agents and Objects) [7] para a modelagem de SMAs. Originalmente, MAS-ML foi projetada para modelar apenas agentes pró-ativos orientados a objetivos e guiados por planos. Para possibilitar a modelagem das diversas arquiteturas de agente definidas por Russell e Norvig [1], Gonçalves [4] evoluiu a linguagem de maneira conservativa originando MAS-ML 2.0.

MAS-ML 2.0 contempla um conjunto de diagramas estáticos e dinâmicos. O diagrama de sequência de MAS-ML 2.0 consegue ilustrar diversas capacidades do SMA. Através de pathnames, podemos representar agentes mudando de um ambiente, organização, ou papel, e com uso dos estereótipos da linguagem podemos ilustrar a criação, a destruição e a interação entre as entidades. Por outro lado, o diagrama de atividades modela um fluxo de execução através de uma sequência de unidades subordinadas chamadas de ação. Na versão 2.0 de MAS-ML a representação de agentes reativos, baseados em objetivo com planejamento e baseados em utilidade foi proposta.

B. MAS-ML tool

A ferramenta MAS-ML tool [8] [5] é um ambiente de modelagem desenvolvido como um plug-in da plataforma Eclipse [9]. MAS-ML tool foi criada para dar suporte à modelagem dos diagramas contemplados na linguagem MAS-ML original, e na sua versão atual, a ferramenta fornece apoio para a construção dos diagramas de classe, organização, e papéis de acordo com MAS-ML 2.0. Visto que modelar SMAs sem o apoio de uma ferramenta torna o trabalho difícil de ser realizado e, podendo até certo ponto, ser considerado impraticável, é fundamental que ferramentas sejam propostas para o uso eficiente da linguagem.

MAS-ML tool foi gerada a partir dos plug-ins GMF (Graphical Modeling Framework) [10] e EuGENia [11]. O GMF é um framework para desenvolvimento de editores gráficos para modelos de domínio. Por outro lado, o EuGENia é capaz de automatizar os procedimentos necessários para o desenvolvimento de diagramas utilizando o GMF, cuja

abordagem é dirigida por modelos utilizando o próprio metamodelo da linguagem MAS-ML 2.0.

III. EVOLUÇÃO DA FERRAMENTA

A estratégia adotada para implementar as extensões propostas segue a abordagem dirigida por modelos, utilizada originalmente para desenvolver a própria ferramenta. Neste caso é utilizado como modelo central o metamodelo da linguagem MAS-ML 2.0.

A. Criação do Diagrama de Sequência

O processo de criação do diagrama de sequência para MAS-ML tool se deu primeiramente, transcrevendo todos os elementos presentes no metamodelo da linguagem MAS-ML 2.0 para a linguagem Emfatic. Com a transcrição desses elementos, podemos identificar os elementos presentes no diagrama de sequência, e associar a eles suas respectivas representações gráficas.

A modelagem do diagrama de sequência em MAS-ML tool prevê adicionalmente a checagem do modelo gerado de forma a verificar a sua correteude em relação à definição no metamodelo da linguagem. Desta forma, foi definido na ferramenta um conjunto de regras de validação implementadas na linguagem OCL (Object Constraint Language) [11], descritas na Tabela 1.

TABELA 1 REGRAS DE VALIDAÇÃO DOS MODELOS (DIAGRAMA DE SEQUÊNCIA).

Regra	Propósito e Definição em OCL	
Regra 1	Todos os elementos do modelo devem ter um nome.	<code>name.size() > 0</code>
Regra 2	Se o agente possui plano, então ele possui ação.	<code>self.ownedPlan->isEmpty() = false implies self.ownedAction->isEmpty() = false and self.ownedPlan->isEmpty() = false</code>
Regra 3	Se o agente possui plano, então não possui percepção.	<code>self.ownedPlan->isEmpty() = false implies self.ownedPerception->isEmpty() = true and self.ownedPlan->isEmpty() = false</code>
Regra 4	Se o agente possui planejamento, então ele possui percepção e ação.	<code>self.ownedPlanning->isEmpty() = false implies (self.ownedPerception->isEmpty() = false and self.ownedAction->isEmpty() = false) and self.ownedPlanning->isEmpty() = false</code>
Regra 5	Se o agente possui plano, então ele não possui planejamento.	<code>self.ownedPlan->isEmpty() = false implies self.ownedPlanning->isEmpty() = true and self.ownedPlan->isEmpty() = false</code>
Regra 6	Caso o agente possua planejamento, então ele não terá plano.	<code>self.ownedPlanning->isEmpty() = false implies self.ownedPlan->isEmpty() = true and self.ownedPlanning->isEmpty() = false</code>

B. Ferramenta Gerada

O ambiente implementado trata-se de um plug-in da plataforma Eclipse, como mencionado anteriormente. Isso permite utilizar os recursos oferecidos pela plataforma de forma concomitante com a modelagem de SMAs. Dado que muitas plataformas de agentes são implementadas em Java, tais

como JADE [13], Jadex [14], Jason [15]; o uso da plataforma Eclipse favorece uma possível geração de código dentro do mesmo ambiente de desenvolvimento. O projeto da ferramenta MAS-ML tool, juntamente com os plug-ins gerados encontram-se disponíveis em <https://sites.google.com/site/uecegessi/masmltool>.

IV. ESTUDO DE CASO

O ambiente de aprendizagem Moodle [16] é usado por instituições de ensino como um ambiente de aprendizagem colaborativa, facilitando a comunicação entre professor e aluno, estimulando a troca de informações e compartilhando recursos via internet. Os diagramas de sequência e atividades foram criados para cada um dos seis agentes descritos, porém, devido à questão de espaço, vamos ilustrar neste trabalho apenas os diagramas para o AgenteBuscadorDeInformacoes. A Figura 1 ilustra o diagrama de sequência para o AgenteBuscadorDeInformacoes, que foi modelado como um agente baseado em objetivo e guiado por plano.

Este agente conta com dois planos pré-definidos: i) `buscarInformacoesPessoas`, que tem as ações de `localizarPessoas`, `relacionarPessoas` e `exibirPessoasRelacionadas` que são executadas em sequência. Este plano visa relacionar pessoas no contexto do Moodle para que as mesmas possam interagir entre si; e ii) `buscarInformacoesDocumentos` com as ações `buscarDocumentos`, `relacionarDocumentos` e `exibirDocumentosRelacionados`. Este plano visa encontrar documentos relacionados às pessoas e exibi-los.



Fig. 1. Diagrama de sequência para o AgenteBuscadorDeInformacoes

A Figura 2 ilustra o diagrama de atividades para o plano `buscarInformacoesPessoas` do AgenteBuscadorDeInformacoes descrito no parágrafo anterior. Os demais agentes podem ser encontrados em <https://sites.google.com/site/uecegessi/masmltool/modelagemdiagramasdinamicosmoodle>.

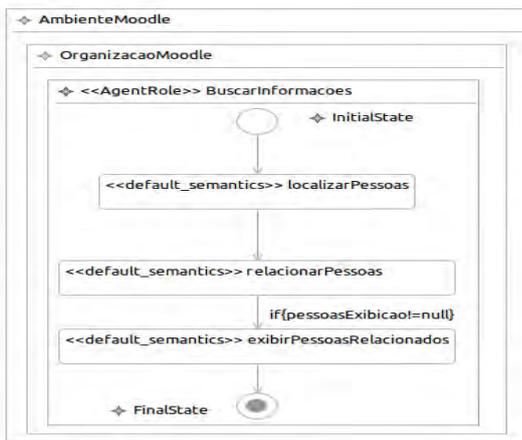


Fig.2. Diagrama de Atividade para o plano buscar informações do AgenteBuscadorDeInformacoes

V. TRABALHOS RELACIONADOS

Ferramentas de modelagem são normalmente projetadas com foco no suporte a uma linguagem de modelagem específica, propagando suas vantagens e desvantagens para as ferramentas que as implementam. Tanto a linguagem AUML [17] quanto Anote [18] descrevem adequadamente papéis e suas propriedades. Com isso, as respectivas ferramentas de suporte não são capazes de modelar tais entidades.

MAS-ML [2] possui duas ferramentas de modelagem para SMAs. O VisualAgent [19] é baseado no metamodelo original da MAS-ML, e consequentemente, o suporte à modelagem de agentes com diferentes arquiteturas internas é limitado. Adicionalmente, apenas os diagramas estáticos propostos na linguagem foram contemplados. Por outro lado, MAS-ML tool [4] [5] é um ambiente de modelagem específico de domínio que atende à modelagem de sistemas multi-agente por meio da linguagem de modelagem MAS-ML 2.0 [4] e contempla os diagramas de classes e organização e papel de acordo com a versão 2.0 de MAS-ML. Duas vantagens de MAS-ML tool em relação à VisualAgent podem ser notadas: o fato de ter sido desenvolvida como um plug-in da plataforma Eclipse [9], e a capacidade de realizar verificação do modelo em relação ao metamodelo da MAS-ML. Adicionalmente, a nova versão de MAS-ML tool possibilita a modelagem de todos os diagramas estáticos e dinâmicos em conformidade com MAS-ML 2.0. Entretanto, as demais ferramentas citadas nessa seção não são capazes de modelar todos os aspectos estáticos e dinâmicos dos SMAs.

VI. CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho foi apresentada a evolução da ferramenta MAS-ML tool para o suporte à modelagem dos diagramas de sequência e atividades definidos em MAS-ML 2.0. Com isso, tanto os diagramas estáticos como os dinâmicos previstos na linguagem podem ser gerados através da ferramenta. Adicionalmente, a extensão proposta prevê a validação da boa formação dos diagramas gerados reduzindo falhas e tornando a modelagem mais coerente com as definições feitas em MAS-ML 2.0. Como trabalhos futuros relacionados com a evolução

da ferramenta podem ser citados: (i) geração de código a partir dos diagramas construídos na ferramenta MAS-ML tool e (ii) realização de um estudo utilizando computação experimental com um público alvo formado por analistas/projetistas de sistemas multi-agente.

REFERÊNCIAS

- [1] RUSSELL, S.; NORVIG, P. Inteligência artificial: uma abordagem moderna, 2ª Ed. Prentice-Hall: São Paulo, 2004.
- [2] SILVA, V. T. Uma linguagem de modelagem para sistemas multi-agente baseada em um framework conceitual para agentes e objetos, Tese de doutorado. Rio de Janeiro: PUC, Departamento de Informática, 2004.
- [3] SILVA, V. T.; CHOREN, R.; LUCENA, C. J. P. MAS-ML: A Multi-Agent System Modeling Language, In: Conference on Object-oriented programming, systems, languages, and applications, 18th annual ACM SIGPLAN; USA, ACM Press, 2007.
- [4] GONÇALVES, E. J. T. (2009). Modelagem de Arquiteturas Internas de Agentes de Software Utilizando a Linguagem MAS-ML 2.0. Dissertação de Mestrado. Fortaleza: UECE, Centro de Ciência e Tecnologia.
- [5] GONÇALVES, E. J. T.; OLIVEIRA, K. S. F.; CORTÉS, M. I.; FEIJÓ, A. R.; OLIVEIRA, F. R.; SILVA, V. T. MAS-ML TOOL: A Modeling Environment for Multi-Agent Systems. In: Proceedings of 13th International Conference on Enterprise Information Systems, Beijing 2011.
- [6] UML, Unified Modeling Language Specification, versão 2.2, disponível em: <http://www.uml.org>, acessado em 2 de Junho de 2011.
- [7] SILVA, V.; Garcia, A.; Brandao, A.; Chavez, C.; Lucena, C.; 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.
- [8] FARIAS, K.; NUNES, I.; SILVA, V. T.; LUCENA, C. J. P. MAS-ML Tool: Um ambiente de modelagem de sistemas multi-agente, Fifth Workshop on Software Engineering for Agent-oriented Systems (SEAS@SBES 09), Brazil, 2009.
- [9] ECLIPSE, Eclipse Platform, disponível em: <http://www.eclipse.org>, acessado em 2 de Junho de 2011.
- [10] GMF, disponível em: <http://www.eclipse.org/modeling/gmf/>, acessado em Junho de 2011.
- [11] EuGENia disponível em: <http://www.eclipse.org/epsilon/>, acessado em Junho de 2011.
- [12] OCL, disponível em: <http://www.eclipse.org/modeling/mdt/?project=ocl>, acessado em 20 de Junho de 2011.
- [13] BELLIFEMINE, F. L.; CAIRE, G.; GREENWOOD, D. (2007). Developing Multi-Agent Systems with JADE. [S.l.]: Wiley (Wiley Series in Agent Technology).
- [14] POKAHR, A.; BRAUBACH, L.; LAMERSDORF, W. (2003). Jadex: Implementing a BDI-Infrastructure for JADE Agents. EXP - In Search of Innovation (Special Issue on JADE), vol. 3, no. 3, Telecom Italia Lab, Turin, Italy, S. 76-85.
- [15] BORDINI, R. H.; WOOLDRIDGE, M.; HÜBNER, J. F. (2007). Programming Multi-Agent Systems in AgentSpeak using Jason, John Wiley & Sons.
- [16] MOODLE, disponível em: <http://www.moodle.org.br>, acessado em Junho de 2011.
- [17] ODELL, J.; PARUNAK, H. V. D.; BAUER, B. (2000). Extending UML for Agents. Proc. Of the Agent-Oriented. Information Systems Workshop (AOIS'00) at the 17th National Conference on Artificial Intelligence (AIJ'00) (3-17).
- [18] CHOREN, R., LUCENA, C. Agent-Oriented Modeling Using ANote, 3rd International Workshop on Software Engineering for Large-Scale Multi-Agent Systems, 3rd; The Institution of Electrical Engineers, IEE, Stevenage, UK, 2004, pp. 74-80, 2004
- [19] DE MARIA, B. A.; SILVA, V. T.; LUCENA, C. J. P.; CHOREN, R. VisualAgent: A software development environment for multi-agent systems, Proceedings of the 19º Simpósio Brasileiro de Engenharia de Software, Tool Track, Brazil, 2005.