

# MAS-ML - Uma Linguagem para Modelagem de Sistemas Multi-Agentes: Uma Análise do Estado da Arte por Meio de uma Revisão Sistemática

Lukas Felipe Gaedicke, Gilleanes Thorwald Araujo Guedes, João Pablo Silva da Silva

<sup>1</sup>Universidade Federal do Pampa - Campus Alegrete (UNIPAMPA)  
Av. Tiarajú, 810 - Ibirapuitã, Alegrete – RS – Brazil

{lukasfgaedicke, gilleanesguedes, joaosilva}@unipampa.edu.br

**Abstract.** *In this paper we present and analyze MAS-ML, a language for modeling multiagent systems. We show the language evolution from its initial proposal to the present day and determine its strong and weak aspects. This study has as its goal to establish the state-of-art of this language and determine how it can be improved. To achieve this goal we conducted a systematic review of this language, retrieving all the papers considered relevant about MAS-ML, describing the language evolution and analyzing its features.*

**Resumo.** *Neste artigo apresentamos e analisamos a MAS-ML, uma linguagem para modelagem de sistemas multiagentes. Nós apresentamos a evolução da linguagem desde sua proposta inicial até os dias atuais e determinamos seus aspectos fortes e fracos. Este estudo tem como objetivo estabelecer o estado-da-arte dessa linguagem e determinar como ela pode ser melhorada. Para atingir este objetivo, conduzimos uma revisão sistemática da literatura sobre esta linguagem, recuperando todos os artigos considerados relevantes sobre a MAS-ML, descrevendo as evoluções da linguagem e analisando suas características.*

## 1. Introdução

O interesse em utilizar agentes como auxiliares nas mais diversas aplicações vem crescendo ao longo dos anos. Todavia, o desenvolvimento de sistemas que suportem agentes apresenta novos desafios para a engenharia de software, o que demonstrou a necessidade da criação de novas metodologias e processos de desenvolvimento para este tipo de sistema, bem como da criação de linguagens de modelagem que os suportem. Levando isto em consideração, várias linguagens foram derivadas a partir da UML (Unified Modeling Language), uma linguagem amplamente utilizada na área de engenharia de software, para o projeto de sistemas multiagentes (SMAs).

Dentre as linguagens que estendem a UML, a MAS-ML (MultiAgent Systems Modeling Language) em seu estado atual, está entre as que contemplam a maior quantidade de recursos para a modelagem das características estruturais e comportamentais dos agentes. Isto é corroborado por [Gonçalves et al. 2015], que a compara com as linguagens AUML de [Odell et al. 2000] e AML de [Cervenka and Trencansky 2007]. Isso posto, consideramos a MAS-ML como possuidora de um interesse particular de estudo.

Segundo [Silva et al. 2008b], a MAS-ML é uma linguagem de modelagem que tem como principal objetivo facilitar a abstração no desenvolvimento de SMAs. Esta linguagem foi estendida a partir da superestrutura da UML, no qual foram inseridos novos

conceitos de modelagem, representados por novas metaclasses e estereótipos, consideradas apropriadas para capturar as características típicas de SMAs.

Com o objetivo de estabelecer o estado-da-arte dessa linguagem e determinar como ela pode ser melhorada, realizamos uma revisão sistemática sobre MAS-ML, recuperando os artigos considerados relevantes. A partir disso, nós analisamos as características acrescidas à linguagem a cada evolução, procurando identificar as vantagens e limitações da MAS-ML. Esse trabalho se justifica pelo fato de não termos encontrado outra revisão sistemática sobre o tema.

A estrutura deste artigo está organizada da seguinte forma: na Seção 2, é apresentado uma breve fundamentação teórica, na Seção 3 apresentamos a metodologia e o protocolo utilizado para a revisão sistemática. Na Seção 4 apresentamos os resultados obtidos através da revisão sistemática e, após isso, as considerações finais desse estudo.

## **2. Fundamentação Teórica**

### **2.1. UML (Unified Modeling Language ou Linguagem de Modelagem Unificada)**

Segundo [Guedes 2018] a UML é uma linguagem padrão de modelagem adotada internacionalmente pela indústria de engenharia de software. Esta linguagem é utilizada principalmente para modelar software durante a sua especificação de requisitos e seu projeto.

A UML oferece diversos diagramas com objetivos e visões distintas, divididos em estruturais e comportamentais. Dentre estes podemos citar o diagrama de casos de uso, utilizado para identificar os requisitos do software; o diagrama de classes que permite representar as classes que compõem o sistema e como elas se relacionam; o diagrama de sequência que demonstra a ordem em que as mensagens são disparados entre os objetos envolvidos em um processo; e o diagrama de atividade utilizado para representar o fluxo de controle e de objetos de uma atividade, podendo esta ser um método, um algoritmo, ou mesmo um processo completo.

### **2.2. Noções Básicas de Agência**

De acordo com [Vicari and Gluz 2007], um agente é um processo computacional, situado em um ambiente, projetado para atingir um propósito através de um comportamento autônomo e flexível. O propósito de um agente pode ser especificado pela definição de suas crenças e desejos e o comportamento deste agente é fortemente influenciado por suas intenções. Agentes podem perceber eventos que ocorrem sobre o ambiente e realizar ações sobre o mesmo. Podem ser classificados em reativos ou cognitivos, os reativos apenas reagem a eventos, enquanto os cognitivos possuem crenças, desejos e intenções a respeito do ambiente em que se encontram. Um sistema composto por um conjunto de agentes é chamado sistema multiagente.

## **3. Protocolo da Revisão Sistemática**

Para realização deste estudo, realizamos uma revisão sistemática da literatura. De acordo com [Kitchenham 2004], uma Revisão Sistemática (RS) tem como objetivo identificar, avaliar e interpretar os resultados do estudo que estão relacionados com as questões, área temática ou fenômeno que se deseja coletar evidências para servir como fundamentação para conclusões.

De acordo com [Biolchini et al. 2005], uma RS deve basear-se em um protocolo previamente definido, que deve formalizar a execução da RS desde a estipulação das questões de pesquisa, passando pelo estabelecimento dos critérios de seleção (inclusão e exclusão de trabalhos), das bases digitais utilizadas para a extração dos estudos relacionados e das palavras chaves aplicadas durante a busca nestas bases, até a definição de como será o relatório final.

Dessa forma, elaboramos um protocolo<sup>1</sup>, no qual definimos duas questões de pesquisa (“QP1. *Quão abrangente é a MAS-ML para a modelagem de sistemas multiagentes?*”) e “QP2. *Quais são as lacunas da MAS-ML para a modelagem de sistemas multiagentes?*”). Neste protocolo também definiu-se a estratégia de seleção dos trabalhos e a forma de extração das informações nos estudos encontrados. A tabela 1 apresenta os critérios de inclusão e exclusão utilizados durante a revisão sistemática.

**Tabela 1. Critérios de inclusão e exclusão**

Tipo	Critério
Inclusão	O trabalho deve reportar explicitamente estudos sobre a linguagem MAS-ML.
Exclusão	Estudo não escrito em inglês.
Exclusão	Estudo não disponíveis para download.
Exclusão	Estudo com menos de 6 páginas.
Exclusão	Estudo não relacionado à modelagem de sistemas multiagentes.

Para encontrar os trabalhos, realizamos um processo de busca em seis bases de dados digitais (ACM Digital Library, IEEE Xplore, Engineering Village, ScienceDirect, SpringerLink e Scopus), todas elas amplamente utilizadas e conhecidas no meio acadêmico. Também definiram-se 3 palavras chave para a busca de estudos relacionados: MAS-ML, Modeling Language e Multi-Agent Systems. Com o intuito de aumentar a abrangência desse estudo, buscando identificar a maior quantidade de artigos sobre a MAS-ML, foram definidos mais 4 sinônimos derivados das palavras chave, sendo elas Multi-Agent System Modeling Language, Modeling Agents e MAS.

Após o levantamento de termos centrais e seus sinônimos foi construída uma *string* de busca genérica, que pode ser observada na figura 1. Contudo é importante ressaltar que para cada uma das bases de dados pesquisadas a string padrão sofreu modificações com o objetivo de adequá-la às particularidades de cada um dos mecanismos de busca adotados pelas bases online.

Para selecionar os estudos, definimos 4 etapas: (1) executar as strings de busca nas bases digitais; (2) remover os estudos duplicados; (3) aplicar os critérios de exclusão nos estudos restantes da etapa 2; (4) aplicar os critérios de inclusão nos estudos remanescentes da etapa 3; (5) ler e extrair as informações dos estudos remanescentes da etapa 4.

(MAS-ML OR Multi-Agent System Modeling Language) AND (Modeling Language OR Modeling Agents AND Multi-Agent Systems OR MAS)

**Figura 1. Ilustração da String genérica utilizada.**

<sup>1</sup>Artefatos gerados durante a execução do protocolo <https://github.com/LukasGaedicke/MAS-ML-SR-Artifacts/>

## 4. Resultados

A figura 2 ilustra as etapas do processo de pesquisa durante o qual houve uma redução na quantidade dos estudos relacionados com nosso objetivo a cada etapa do processo. Assim, na Etapa 1, a busca retornou 186 resultados e após isso, na execução do passo 2 restaram 151 estudos, remanescentes da eliminação de trabalhos duplicados. Já na aplicação dos critérios de exclusão, na Etapa 3, restaram 53 estudos. Na etapa 4 do processo, em que foram aplicados os critérios de inclusão, restaram apenas 8 estudos, os quais foram lidos e analisados. As informações extraídas destes trabalhos são apresentadas e discutidas na seção 4.1.

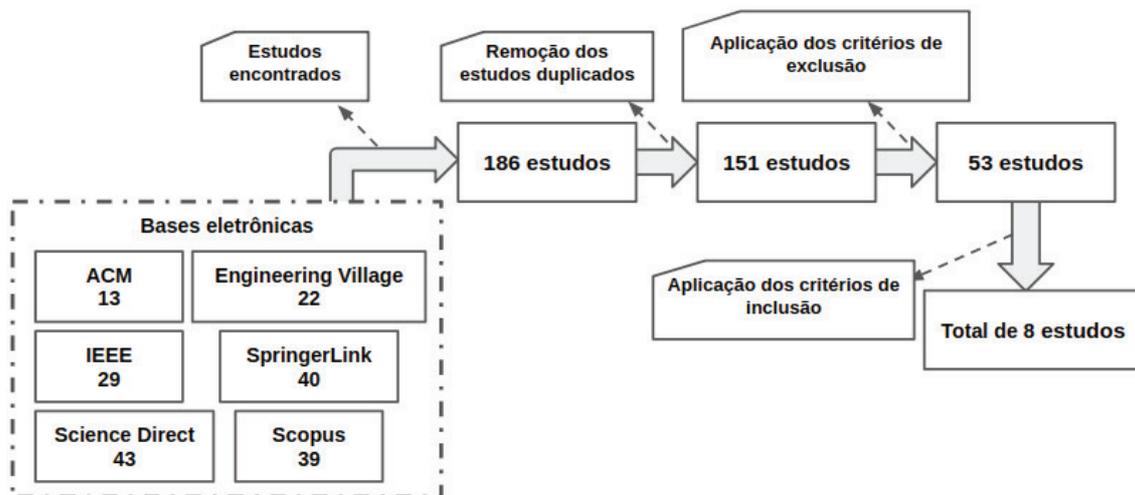


Figura 2. Ilustração do processo de seleção dos estudos.

### 4.1. Visão Geral dos Estudos

O estudo de [Silva and Lucena 2004], contém a primeira proposta da linguagem MAS-ML propriamente dita. Esta linguagem estendeu a UML baseada na descrição de propriedades estruturais e dinâmicas apresentadas em um *framework* conceitual chamado Taming Agents and Objects (TAO). Esta proposta inicial procurou identificar as características particulares de sistemas multi-agentes (SMA) que não eram cobertas pela UML. Assim, para sanar essa limitação, o estudo apresenta como solução um metamodelo UML estendido, descrevendo as novas metaclasses e estereótipos que podem ser usados para modelar as características dos agentes em um SMA.

A modelagem na MAS-ML foi dividida em dois aspectos, estruturais e dinâmicos. Os aspectos estruturais, propostos em [Silva and Lucena 2004], englobam os diagramas estáticos (diagramas de classes, organização e papéis). Enquanto os aspectos dinâmicos, propostos originalmente em [Silva and Lucena 2003] - um trabalho precursor da MAS-ML - envolvem os diagramas comportamentais (diagrama de sequência) e foram incorporados à MAS-ML para representar as interações entre os agentes.

A partir desses recursos, segundo [Silva and Lucena 2004], a MAS-ML seria capaz de abordar as características particulares de um SMA que não são suportadas pela UML, tornando-se possível representar abstrações associadas a um SMA e descrever as relações estáticas e dinâmicas entre essas abstrações. Mas apesar dos benefícios da

utilização da MAS-ML para modelagem de agentes, o estudo de [Silva and Lucena 2004], apenas se preocupava com os aspectos estruturais e parcialmente com os aspectos dinâmicos do sistema (apenas diagramas de sequência). Dessa forma, essa versão da linguagem não suportava a modelagem de planos associados aos objetivos dos agentes.

Em seu trabalho seguinte [Silva et al. 2004], estendeu o diagrama de sequência da MAS-ML, de maneira que a linguagem passasse a suportar a representação de envio e recebimento de mensagens FIPA [FIPA 2018] entre agentes. Posteriormente, no estudo de [Silva et al. 2005], o diagrama de atividade da UML foi estendido e incorporando aos diagramas dinâmicos da MAS-ML, para permitir a representação dos planos e ações dos agentes, além das funções desempenhadas pelos agentes durante a execução de um plano.

Após isso, no trabalho de [Silva et al. 2008a], foram definidas uma série de diretrizes para ajudar os projetistas a modelar orientação, interação, adaptação, distribuição, mobilidade e execução simultânea de objetivos relativos aos agentes. Em seguida [Silva et al. 2008b] apresenta com um maior grau de detalhamento, os elementos e mecanismos da MAS-ML descritos anteriormente nos estudos citados. Esse trabalho passou a ser considerado o modelo de referência padrão da linguagem, sendo utilizado por todas as extensões posteriores.

Em outro trabalho, [Freire et al. 2012], estendeu a MAS-ML de [Silva et al. 2008b] de forma a viabilizar sua integração com a NormML [Figueiredo and da Silva 2010], uma linguagem de modelagem para a especificação de normas. Segundo [Figueiredo and da Silva 2010], a grande maioria dos agentes possuem total autonomia para realizar a execução de tarefas que lhe são determinadas, porém em alguns cenários é necessário restringir o comportamento dos agentes por meio de normas, definindo sanções e aplicando-as quando normas são violadas ou cumpridas. Dessa forma, visando possibilitar a modelagem de elementos estáticos de normas de agentes ainda em fase de projeto, [Freire et al. 2012] produziu uma nova versão da MAS-ML com suporte a modelagem de normas de agentes, chamada de NorMAS-ML. O estudo apresenta como principal limitação a realização apenas da extensão dos aspectos estruturais da MAS-ML, deixando em aberto a extensão dos aspectos dinâmicos de normas na MAS-ML.

Seguindo uma outra perspectiva, o estudo de [Adamzadeh et al. 2014], estendeu a MAS-ML a partir da versão de [Silva et al. 2008b], definindo conceitos relacionados à mitigação de danos em ambientes de resposta a emergências (EREs). Todavia, a extensão ainda necessita de melhorias porque apenas alguns dos conceitos do metamodelo para gerenciamento de desastres de [Othman et al. 2014], foram considerados na extensão.

Finalmente, em [Gonçalves et al. 2015] a linguagem MAS-ML foi evoluída para a MAS-ML 2.0, o que envolveu a extensão dos aspectos estáticos e dinâmicos propostos por [Silva et al. 2008b], de forma que, além das características particulares de SMAs, a linguagem também contemple a modelagem das arquiteturas internas dos agentes propostas por [Russell and Norvig 2016]. Considerando que um agente faz uso de suas percepções para obter informações sobre meio ambiente e sobre outros agentes, além de planejar uma sequência de ações para atingir um objetivo. Assim, tomando como base que a MAS-ML não contemplava essas características dos agentes, foi realizada a definição de duas novas metaclasses, sendo elas: *AgentPlanningStrategy*, para representar a funcionalidade

de planejamento dos agentes, e *AgentPerceptionFunction*, responsável por representar a percepção dos agentes. Também foram definidos novos estereótipos para a representação do comportamento dos agentes nos diagramas da MAS-ML.

Após a extensão das metaclasses, os diagramas estáticos das MAS-ML sofreram modificações, sendo possível agora a representação de cinco estruturas de agentes baseados em arquiteturas, sendo eles: Agentes de Reflexo Simples, Agentes Baseados em Modelo, Agentes baseados em metas com estrutura de planejamento, Agentes baseados em metas com estrutura de plano e Agentes Baseados em Utilidade.

## 4.2. Respondendo às Questões

No que concerne à primeira pergunta “Quão abrangente é a MAS-ML para a modelagem de sistemas multiagentes?”), tomando como base os estudos encontrados, observamos que a MAS-ML original de [Silva et al. 2008b] suporta a modelagem e representação (I) das principais entidades de um SMA, apresentadas como propriedades estáticas como agentes, papéis de agente, organizações, sub-organizações e ambientes; (II) das propriedades dinâmicas, como a interação dos agentes por meio de troca de mensagens, o fluxo de execução dos planos e ações dos agentes. A MAS-ML 2.0 de [Gonçalves et al. 2015] estende a MAS-ML original com suporte a (III) modelagem das arquiteturas internas de agentes propostas tanto nos diagramas estáticos como dinâmicos. A NorMAS-ML de [Freire et al. 2012] estende a MAS-ML original com suporte a (IV) modelagem dos elementos estáticos das normas para agentes. A extensão de [Adamzadeh et al. 2014] estende a MAS-ML original com suporte a (V) modelagem dos aspectos estáticos dos conceitos relacionados à mitigação de danos em EREs.

Considerando que a modelagem de SMAs requer todas as características citadas e existem três extensões distintas, não existe hoje uma versão que suporte todas as propriedades necessárias para a modelagem de um SMA. Percebe-se portanto a necessidade de unir todas as versões em uma linguagem única.

Com relação à segunda pergunta, “Quais são as lacunas da MAS-ML para a modelagem de sistemas multiagentes?”), observamos que (I) a MAS-ML 2.0 ainda precisa sofrer outras extensões para fornecer suporte à modelagem de outras arquiteturas internas propostas pela literatura; (II) a MAS-ML ainda não possui recursos para a representação do aprendizado e a autonomia dos agentes em seus recursos dinâmicos; (III) as extensões de [Adamzadeh et al. 2014] e [Freire et al. 2012] apenas estenderam os aspectos estruturais da linguagem, sem incluir os aspectos dinâmicos.

Além disso, a MAS-ML e suas extensões não contém nenhum recurso específico para a modelagem de requisitos particulares para SMAs, tais como percepções, ações, objetivos e planos. Acreditamos que, como alternativa, se utilizaria para isto o diagrama de casos de uso da UML padrão sem qualquer extensão, mas não encontramos nenhum exemplo dessa aplicação. Finalmente, não encontramos nenhuma proposta de um processo de desenvolvimento para sistemas multiagentes que utilize a MAS-ML como notação.

## 5. Considerações Finais

Considerando o objetivo principal desse estudo, de identificar a abrangência e lacunas da MAS-ML, destacamos que a MAS-ML atualmente suporta parcialmente a modelagem

das principais entidades e arquiteturas internas de agentes de um SMA por meio de diagramas estáticos e dinâmicos, além de possibilitar a modelagem dos aspectos estáticos de normas e EREs.

Mas apesar dos benefícios, a linguagem ainda possui oportunidades de melhoria. Com base nos estudos dessa revisão, identificamos que atualmente a MAS-ML possui 3 versões disjuntas. De um ponto de vista de engenharia de software, isso é um grande problema, porque se torna difícil realizar a identificação de quais adaptações cada uma das linguagens implementou em sua versão. E mesmo que [Adamzadeh et al. 2014], em seu estudo utilize duas versões diferentes da linguagem MAS-ML em um mesmo exemplo de uso, ainda não é possível afirmar qual é a compatibilidade das linguagens entre si.

A partir destas conclusões, como trabalhos futuros, passamos à segunda etapa de nossa pesquisa (que já se encontra em andamento) durante a qual estamos aplicando a linguagem MAS-ML na modelagem de um sistema multiagente real com o objetivo de determinar se realmente a linguagem é adequada para a modelagem deste tipo de sistema e se ela possui mais limitações. Com isso almejamos determinar o que precisaria ser estendido nesta linguagem. Podemos destacar neste momento a falta de mecanismos para a modelagem de requisitos que esta linguagem apresenta, além da falta de um processo de desenvolvimento que suporte sua utilização.

Com relação à deficiência em modelar requisitos específicos para este tipo de sistema, poderíamos aplicar o metamodelo de Guedes [Guedes and Vicari 2012] com este objetivo, que foi projetado especificamente para a modelagem de requisitos de sistemas multiagentes e verificar sua compatibilidade com a MAS-ML. No que tange à falta de um processo, consideramos a sua elaboração muito importante, posto que é necessário estabelecer regras de como utilizar a notação da linguagem. Esta elaboração deverá também constituir um trabalho futuro.

## Referências

- Adamzadeh, T., Zamani, B., and Fatemi, A. (2014). A modeling language to model mitigation in emergency response environments. In *Computer e Knowledge Engineering (ICCKE), 2014 4th International eConference on*, pages 302–307. IEEE.
- Biolchini, J., Mian, P. G., Natali, A. C. C., and Travassos, G. H. (2005). Systematic review in software engineering. *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES*, 679(05):45.
- Cervenka, R. and Trencansky, I. (2007). *The Agent Modeling Language-AML: A Comprehensive Approach to Modeling Multi-Agent Systems*. Springer Science & Business Media.
- Figueiredo, K. and da Silva, V. T. (2010). Normml: a modeling language to model norms. In *1st Workshop on Autonomous Software Systems. Salvador, Brazil*.
- FIPA (2018). Fipa agent management specification. <http://www.fipa.org/specs/fipa00023/SC00023K.html>, Acessado em 2018-09-26.
- Freire, E. S. S., Cortés, Mariela Inés e Gonçalves, E. J. T., and Lopes, Y. S. (2012). Normas-ml-a modeling language to model normative multi-agent systems. In *ICEIS* (2), pages 113–119.

- Gonçalves, E. J. T., Cortés, M. I., Campos, G. A. L., Lopes, Y. S., Freire, E. S., da Silva, V. T., de Oliveira, K. S. F., and de Oliveira, M. A. (2015). Mas-ml 2.0: Supporting the modelling of multi-agent systems with different agent architectures. *Journal of Systems and Software*, 108:77–109.
- Guedes, G. T. A. (2018). *UML 2 - Uma Abordagem Prática - 3a. Edição*. Novatec Editora.
- Guedes, G. T. A. and Vicari, R. (2012). *Um metamodelo UML para a modelagem de requisitos em projetos de sistemas multiagentes*. PhD thesis, Universidade Federal do Rio Grande do Sul - UFRGS.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26.
- Odell, J., Parunak, H. V. D., and Bauer, B. (2000). Extending uml for agents. *Ann Arbor*, 1001:48103.
- Othman, S. H., Beydoun, G., and Sugumaran, V. (2014). Development e validation of a disaster management metamodel (dmm). *Information Processing & Management*, 50(2):235–271.
- Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.
- Silva, V. T., Choren, R., and De Lucena, C. J. (2004). A uml based approach for modeling and implementing multi-agent systems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 914–921. IEEE Computer Society.
- Silva, V. T., Choren, R., and De Lucena, C. J. (2008a). Modeling mas properties with mas-ml dynamic diagrams. In *Agent-Oriented Information Systems IV*, pages 1–18. Springer.
- Silva, V. T. and Lucena, C. J. (2004). From a conceptual framework for agents and objects to a multi-agent system modeling language. *Autonomous Agents and Multi-Agent Systems*, 9(1-2):145–189.
- Silva, V. T. and Lucena, C. J. P. (2003). *Extending the UML Sequence Diagram to model the dynamic aspects of Multi-Agent Systems*. PUC.
- Silva, V. T., Noya, R. C., and de Lucena, C. J. (2005). Using the uml 2.0 activity diagram to model agent plans and actions. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 594–600. ACM.
- Silva, V. T. D., Choren, R., and De Lucena, C. J. (2008b). Mas-ml: a multiagent system modelling language. *International Journal of Agent-Oriented Software Engineering*, 2(4):382–421.
- Vicari, R. and Gluz, J. C. (2007). An intelligent tutoring system (its) view on aose. *International Journal of Agent-Oriented Software Engineering*, 1(3-4).