

Engenharia Orientada a Agentes Open AEOLus

Daniela Maria Uez
dani.uez@gmail.com

06 de novembro de 2018

- 1 AOSE
- 2 Sistemas Multiagentes Abertos
- 3 Open AEOLus
 - Visão Geral
 - Análise
 - Projeto das dimensões
 - Projeto de abertura
 - Implementação
- 4 Considerações Finais

Sumário

- 1 AOSE
- 2 Sistemas Multiagentes Abertos
- 3 Open AEOLus
 - Visão Geral
 - Análise
 - Projeto das dimensões
 - Projeto de abertura
 - Implementação
- 4 Considerações Finais

Engenharia de software orientada a agentes - AOSE (I)

- Visa auxiliar no desenvolvimento de sistemas baseados em agentes e/ou de sistemas complexos que utilizam a noção de agentes
- Inclui
 - Métodos
 - Técnicas de modelagem
 - Ferramentas
 - Frameworks e linguagens de implementações
 - Linguagens de comunicação

AOSE II

- Ainda sem a estrutura da engenharia de software OO
- Diversos métodos e linguagens de modelagens desenvolvidos
- Não existem padrões de linguagens de modelagem, métodos ou ferramentas (como a UML e o UP na OO)

Métodos

- Os primeiros só levavam em conta a especificação dos agentes
- Atualmente existem métodos que permitem especificar a dimensão da organização e alguns que levam em conta o ambiente

Alguns Métodos AOSE

- Adelfe (Bernon *et al.* [2002])
<http://www.irit.fr/ADELFE/>
- ASPECS (Cossentino *et al.* [2010]) - (Janus)
<http://www.aspecs.org/Home>
- Ingenias (Pavón and Gómez-Sanz [2003]) - (JADE)
<http://ingenias.sourceforge.net/>
- O-MaSE (DeLoach and García-Ojeda [2010]) - (JADE)
<http://agenttool.cis.ksu.edu/>
- OperA+Environment (Dastani *et al.* [2004])
- Prometheus (Padgham and Winikoff [2004]) - (JACK)
<http://www.cs.rmit.edu.au/agents/prometheus/>
- Tropos (Cossentino and Seidita [2005]) - (JADEX / JACK)
<http://www.troposproject.org/node/93>
- Prometheus AEOLus (Uez *et al.* [2013]) - (JaCaMo)

Sumário

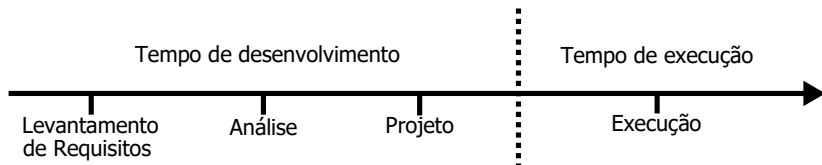
- 1 AOSE
- 2 Sistemas Multiagentes Abertos
- 3 Open AEOLus
 - Visão Geral
 - Análise
 - Projeto das dimensões
 - Projeto de abertura
 - Implementação
- 4 Considerações Finais

Sistemas Multiagentes

O SMA pode ser composto por:

- Agente (A) - entidades autônomas que agem em prol dos seus objetivos
- Ambiente (E) - disponibiliza recursos e ferramentas para os agentes
- Organização (O) - define normas e protocolos que permitem a cooperação entre os agentes

Conceitos Iniciais



Tempo de desenvolvimento

Todas as atividades desenvolvidas antes da execução do sistema: requisitos, análise, projeto ...

Tempo de execução

Momento em que o sistema está sendo executado

Definição clássica de abertura

- Um SMA aberto é aquele no qual os agentes podem entrar e sair livremente em tempo de execução
- Os agentes que farão parte do sistema em tempo de execução não são conhecidos durante o desenvolvimento do sistema
- Em outras palavras:
 - Os elementos não são desenvolvidos juntamente com o sistema
 - Não se tem conhecimento sobre como foram desenvolvidos, linguagem usada, etc
 - Normalmente, são desenvolvidos por outro desenvolvedor e integrados ao sistema quando este está em execução

A nova definição de abertura

SMA aberto

- Um SMA é aberto quando qualquer uma de suas dimensões puder ser alterada em tempo de execução
- Ou seja, elementos de qualquer dimensão (agentes, artefatos, protocolos, normas...) que são "desconhecidos" durante o desenvolvimento do sistema são integrados ao sistema durante a execução

Tipos de abertura

- Cada dimensão pode apresentar um diferente tipo de abertura:
 - Dimensão Aberta
Os elementos dessa dimensão não são conhecidos em tempo de desenvolvimento - só farão parte do sistema em tempo de execução
 - Dimensão Fechada
Todos os elementos são conhecidos durante o desenvolvimento do sistema
 - Dimensão Parcialmente aberta
Alguns elementos dessa dimensão não são conhecidos em tempo de desenvolvimento, outros são

Exemplo - Multiagent Programming Context

- Competição de agentes que acontece todos os anos (<https://multiagentcontest.org/>)
- O cenário e os objetivos dos times mudam de acordo com o ano
- Cada competidor desenvolve um time de agentes para tomar parte na competição
- No dia da competição, os times de agentes são integrados ao cenário do jogo para competir uns contra os outros

Exemplo - MAPC (II)

- O cenário é desenvolvido pelos organizadores da competição na Alemanha
- Os organizadores definem como será o ambiente do jogo, algumas normas e papéis
- Mas os organizadores não desenvolvem os agentes que farão parte do sistema: estes são desenvolvidos por competidores ao redor do mundo
- Organizadores não sabem como esses agentes serão desenvolvidos - podem inclusive ser desenvolvidos para atrapalhar a competição!
- Os agentes só farão parte do sistema no dia da competição, quando o sistema estiver sendo executado

Sumário

- 1 AOSE
- 2 Sistemas Multiagentes Abertos
- 3 Open AEOLus
 - Visão Geral
 - Análise
 - Projeto das dimensões
 - Projeto de abertura
 - Implementação
- 4 Considerações Finais

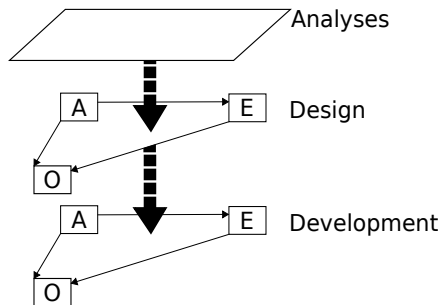
Open AEOLus



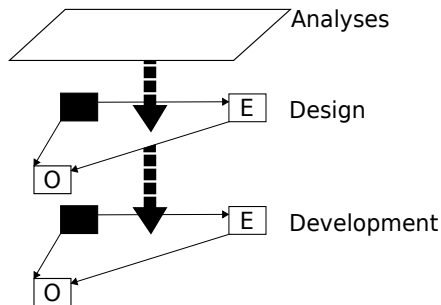
Open AEOLus

- Método para modelagem de SMA abertos
- Dimensões independentes - cada uma definida com abstrações e conceitos próprios
- Conceitos de borda - conceitos especiais que visam auxiliar a inclusão dos novos elementos em tempo de execução

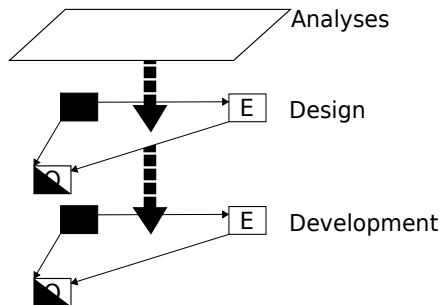
Visão Geral



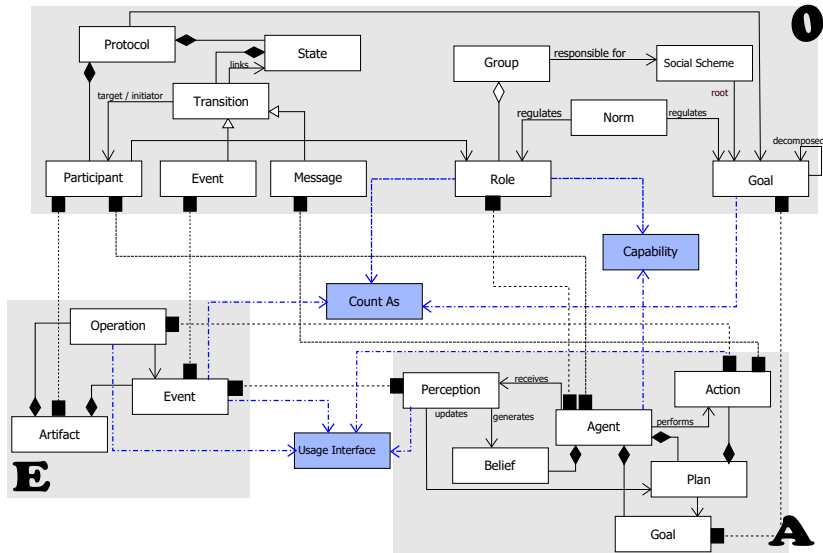
Visão Geral (II)



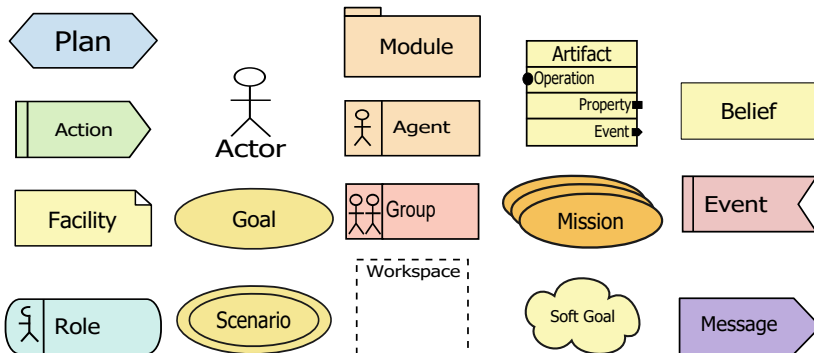
Visão Geral (III)



Metamodelo



Notação



Fases

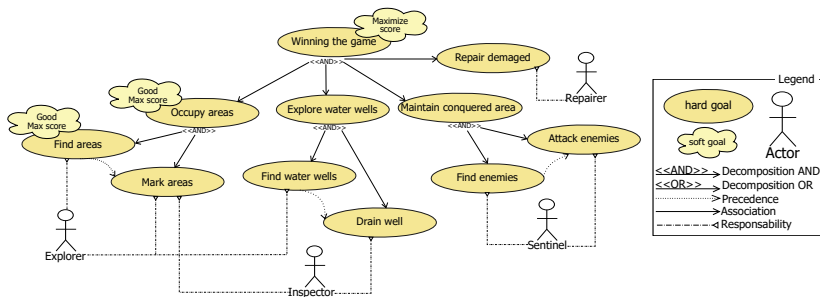
- Análise: o que o sistema deve fazer?
- Projeto: define cada dimensão independentemente e os conceitos de borda que permitem a integração das dimensões em tempo de execução
- Implementação: refina os conceitos da fase anterior levando em conta uma plataforma de implementação específica (JaCaMo) - geração de código

Análise

Parte de uma definição inicial do sistema

- 1 Define *use cases*
- 2 Define objetivos do sistema
- 3 Define atores envolvidos
- 4 Liga atores e objetivos
- 5 Caracterização das dimensões

Atores - Objetivos



Caracterização das dimensões

MAS Dimensinal Review

Agents

- ☐ Open ☐ Closed
☐ Half-Open

Organization

- ☐ Open ☐ Closed
☐ Half-Open ☐ Inexistent

Environment

- ☐ Open ☐ Closed
☐ Half-Open ☐ Inexistent

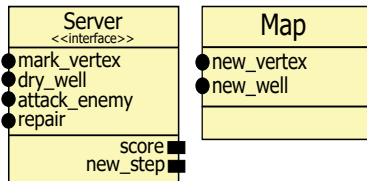
Projeto

Com base nos artefatos criados na fase anterior

- Projeto das dimensões: define cada dimensão (fechada ou parcialmente aberta) independentemente, usando conceitos e abstrações próprias
- Projeto de abertura: define os conceitos de borda para auxiliar na integração dos elementos das dimensões abertas

Projeto das dimensões - Ambiente

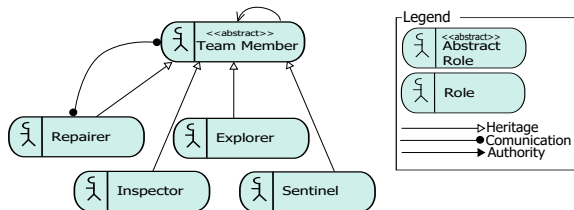
- Com base nos *use cases* e no diagrama de atores e objetivos, define os artefatos do ambiente (atores externos ou recursos)
- Refina os artefatos para inclusão de eventos e operações



Projeto das dimensões - Organização

- Identifica os papéis que farão parte da organização
- Refina os papéis definindo relações entre eles
- Define as responsabilidades de cada papel, ligando-o ao seu objetivo

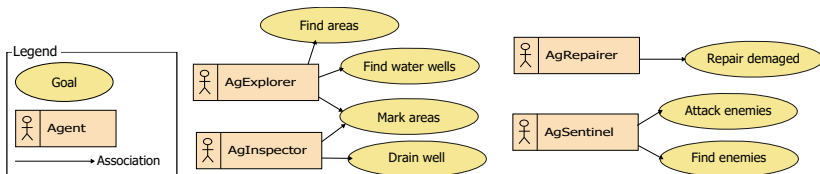
Refinamento de papéis



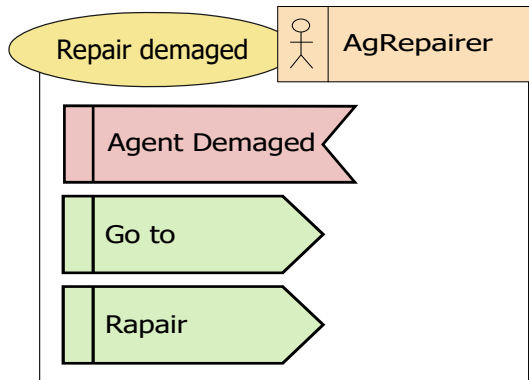
Projeto das dimensões - Agentes

- Identifica agentes
- Define responsabilidades dos agente ligando-os com os objetivos
- Define planos do agente para atingir os objetivos

Responsabilidade dos agentes



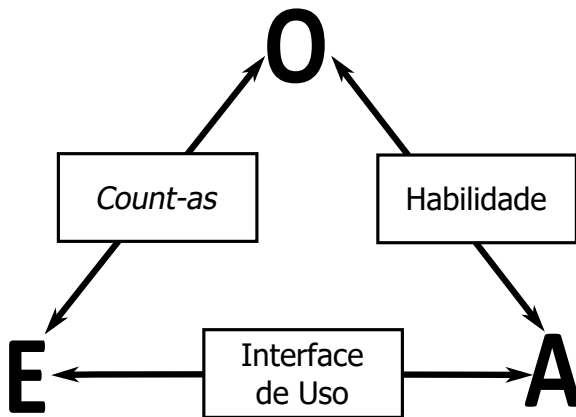
Planos dos agentes



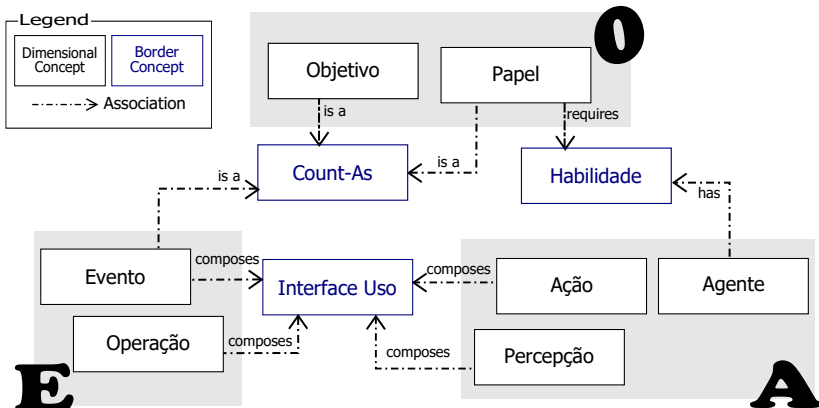
Conceitos de borda

- Usados para definir, em tempo de desenvolvimento, informações específicas sobre como as dimensões são ligadas em tempo de execução

Conceitos de Borda (II)



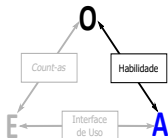
Conceitos de Borda - Metamodelo



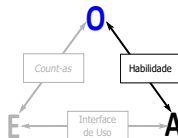
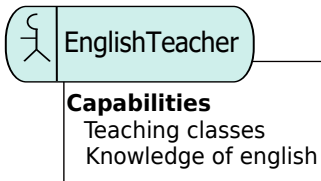
Conceitos X tipos de abertura

	Conceitos Dimensão	Conceitos Borda
Aberta	Não	Sim
Fechada	Sim	Não
Parcialmente Aberta	Sim	Sim
Inexistente	Não	Não

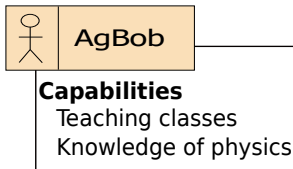
Conceito de borda - Habilidade



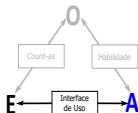
Aberto Dimensão Agentes



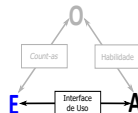
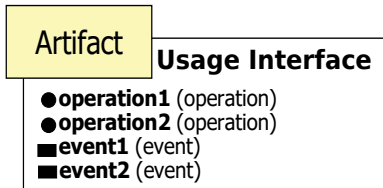
Aberto Dimensão Organização



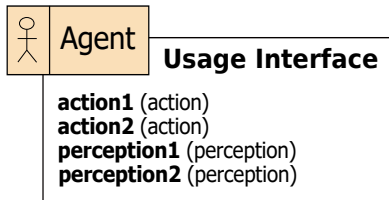
Conceito de borda - Interface de Uso



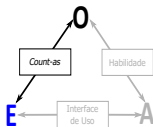
Abertura Dimensão Agente



Abertura Dimensão Ambiente



Conceito de borda - Count-as



Abertura Dimensão Ambiente

Take a test

Count-as

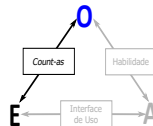
- **answerQuestions** (operation)
- **testScore** (event)



Student

Count-as

- **enterRoom** (event)



Abertura Dimensão Organização

Enrollment

studentEnrolled ■

Count-as

Enroll Student (goal)

Enrollment

studentUnenrolled ■

Count-as

Unenroll Student (goal)

Implementação

Refina os elementos definidos no projeto para geração de código do JaCaMo

Sumário

- 1 AOSE
- 2 Sistemas Multiagentes Abertos
- 3 Open AEOLus
 - Visão Geral
 - Análise
 - Projeto das dimensões
 - Projeto de abertura
 - Implementação
- 4 Considerações Finais

Para saber mais...

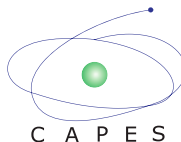
- Site (em construção constante):
<http://www.uez.com.br/aeolus>
- Ferramenta AEOLia
 - Ferramenta para uso do método AEOLus
 - Em desenvolvimento (plug-in pra o Eclipse)

That's all folks!

Obrigada!
dani.uez@gmail.com



DAS



Bibliografia I

- Federico Bergenti and Michael N. Huhns. On the use of agents as components of software systems. In Federico Bergenti, Marie-Pierre Gleizes, and Franco Zambonelli, editors, *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, volume 11 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, chapter 2, pages 273–296. Kluwer Academic, 2004.
- Carole Bernon, Pierre Glize, Gauthier Picard, and Pierre Glize. The ADELFE methodology for an intranet system design. In *AOIS 2002*, pages 27–28. 2002.
- M. Cossentino and V. Seidita. Tropos: Processo e frammenti, 2005.

Bibliografia II

- Massimo Cossentino, Nicolas Gaud, Vincent Hilaire, Stéphane Galland, and Abderrafiâa Koukam. ASPECS: an agent-oriented software process for engineering complex systems. *JAAMAS*, 20(2):260–304, 2010.
- Mehdi Dastani, Joris Hulstijn, Frank Dignum, and John-Jules Ch. Meyer. Issues in multiagent system development. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 922–929, Washington, DC, USA, 2004. IEEE Computer Society.
- Scott A. DeLoach and Juan Carlos García-Ojeda. O-mase: a customisable approach to designing and building complex, adaptive multi-agent systems. *Int. J. Agent-Oriented Softw. Eng.*, 4(3):244–280, 2010.

Bibliografia III

- Brian Henderson-Sellers. Agent-oriented methodologies: an introduction. In Paolo Giorgini and Brian Henderson-Sellers, editors, *Agent-Oriented Methodologies*. Idea Group Publishing, 2005.
- Lin Padgham and Michael Winikoff. *Developing Intelligent Agent Systems: A Practical Guide*. Halsted Press, New York, NY, USA, 2004.
- Juan Pavón and Jorge J. Gómez-Sanz. Agent oriented software engineering with INGENIAS. In *CEEMAS 2003*, volume 2691 of *LNAI*, pages 394–403. Springer, 2003.
- Daniela Maria Uez, Jomi Fred Hübner, and Carine G. Webber. Método para modelagem de agentes, ambiente e organização de sistemas multiagentes. In *Anais IV Workshop Brasileiro sobre Sistemas de Software Autônomos - Autosoft 2013*, pages 41–50, 2013.