

# **AORML uma linguagem para modelagem de uma aplicação Multi-agentes: Uma Aplicação no Sistema Expertcop.**

Hebert de Aquino Nery, Daniel Gonçalves de Oliveira e Vasco Furtado.

Universidade de Fortaleza – UNIFOR

[vasco@unifor.br](mailto:vasco@unifor.br)

[hebartaquino@bol.com.br](mailto:hebartaquino@bol.com.br)

[goncalvesmail@yahoo.com.br](mailto:gonalvesmail@yahoo.com.br)T

## **1. Introdução**

A fase de modelagem conceitual é um passo cada vez mais importante no desenvolvimento de softwares. Um software bem modelado se torna fácil de implementar, de realizar manutenções e de ser compreendido.

A UML é uma linguagem de modelagem amplamente utilizada no mundo todo para sistemas orientados a objetos. No entanto ela não suporta o conceito de agentes criando assim uma certa ambigüidade na modelagem. A UML não consegue retratar de forma coerente a troca de mensagens entre os agentes e nem os estados de um agente, sendo uma linguagem apropriada para modelagem de sistemas orientados a objetos, e um objeto não pode ser tratado de forma igual a um agente. Por isso foi criada uma extensão para a UML chamada AORML.[WAGNER, 2002].

A AORML utiliza uma representação mais detalhada de agentes.Os agentes podem ser agentes humanos, agentes artificiais ou agentes institucional, onde cada agente possui uma representação diferenciada. Eles podem se comunicar, perceber, agir, fazer um compromisso e satisfazer uma reivindicação.Podendo ser representado de duas formas uma representação interna na qual essa representação refere-se aos próprios agentes e uma representação externa na qual ela enfoca o sistema como um todo.[WAGNER, 2003].

Neste trabalho descreveremos brevemente a AORML e mostrar sua aplicação em um sistema multi-agentes chamado ExpertCop.

## **2. Sistemas Multi-Agentes (*Multi-Agent System - MAS*)**

Um agente é “uma entidade que reside em um ambiente onde interpreta dados através de sensores que refletem eventos no ambiente e executam ações que produzem efeitos no ambiente. Um agente pode ser software ou hardware puro...”.[FIPA,2003]

Sistemas Multi-agentes são sistemas que possuem várias entidades denominadas de agentes. Estes sistemas possuem a características de que cada agentes tem o poder de realizar tarefas em conjuntos ou não dependendo da finalidade de cada aplicação que o sistema está inserido. Interagindo os agentes com um ambiente, tornando mais próximo de uma organização artificial.[SILVA,2003]

JADE (*Java Agent Development*) é um ambiente de desenvolvimento baseado na linguagem Java. Trata-se de um framework de desenvolvimento que facilita a criação

dos agentes inclusos. O Jade possui um ciclo de vida para cada agente, realiza o monitoramento de qual é o comportamento de cada agente e qual tipo de mensagens que os agentes estão realizando e qual é o destino de cada mensagem no ambiente. [JADE,2003]

### 3. AORML (Agent – Object Relationship Model Language)

AORML é uma extensão da linguagem de modelagem unificada (UML) que relaciona agentes e objetos. Ela representa os comportamentos dos agentes, as ações, troca de mensagens que cada agente realiza e determina diferenças entre os agentes do sistema e os objetos. As entidades estão divididas em agentes, eventos, ações, compromissos, reivindicações e objetos.

A AORML consiste de um modelo de análise e um modelo de design.

O modelo Externo, como é conhecido o modelo de análise, modela uma visão externa, de quem esta observando os agentes e suas interações considerando o domínio do problema de uma forma abrangente focando sempre no ambiente. No modelo externo as ações são também eventos, as reivindicações são também compromissos, ou seja, os modelos de ações e eventos são representados pelo mesmo diagrama de evento e a representação para reivindicação e compromisso são também representadas pelo mesmo diagrama de compromisso.

O modelo Interno, como é conhecido o modelo de design, modela o problema na visão de cada agente, como ele se comunica com os outros agentes, como ele é notificado dos eventos ao ambiente, como ele enxerga o “mundo”. [WAGNER, 2002]

No modelo externo temos os seguintes diagramas:

- Diagrama de agentes: destacando as classes agentes, os objetos de classes mais relevantes e as relações entre eles.
- Diagrama de interação de quadros: destacando as classe de eventos e as classes de *commitment/claim* que determinam possíveis interações entre dois tipos de agentes. Este diagrama possui quatro tipos de relacionamento:
  - Relacionamento entre agentes com ações e eventos comunicativos: enviar e receber mensagens são eventos ou ações do tipo que relata ao agente um tipo de relacionamento que gera comunicação.
  - Relacionamento entre agentes com ações e eventos não – comunicativos: fazer e perceber os agentes são eventos que não geram comunicação entre os agentes.
  - Relacionamento entre agentes e compromisso e reivindicação: Este relacionamento é visualizado com um conector particular que acompanha uma ação ou evento.

- Relacionamento entre evento sem ação: Este relacionamento é percebido pelo agente no qual ele não executa nenhuma ação, ou seja, apenas uma chamada a um método.
- Diagrama de seqüência: destacando a interação entre processos. Interação entre processos é uma seqüência de ações e eventos feitos e recebidos pelos agentes que seguem um conjunto de regras. Essa interação pode ocorrer entre o agente e o ambiente ou entre os próprios agentes.[WAGNER, 2002]

No modelo interno temos os seguintes diagramas:

- Diagrama de Quadros da Reação: destacando outros agentes e as classes de ações e eventos, e também as classes de *commitment/claim* que determinam uma possível interação com ele.
- Diagrama de Seqüência de Reação: destacando instâncias de interações de processos numa perspectiva interna ao agente.

## 4. UM ESTUDO DE CASO NO EXPERTCOP

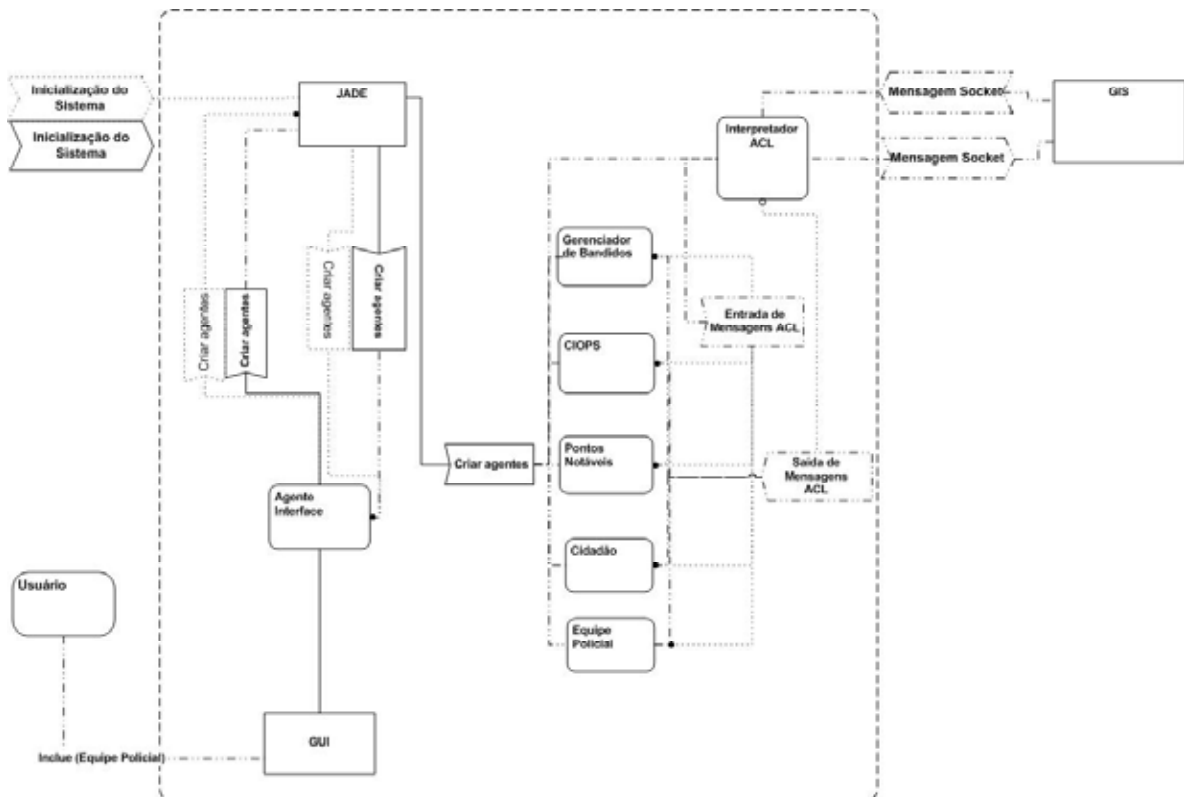
### 4.1. O Sistema ExpertCop

O sistema ExpertCop é um geosimulador que visa apoiar a educação através da simulação de uma região urbana, onde nesta região ocorrem crimes durante algum intervalo de tempo. O sistema permite que o usuário do sistema faça alocação dinâmica de recursos policiais que serão usados para evitar que os crimes ocorram. Estes exemplos são frutos da interação dos agentes que representam as entidades reais de uma região urbana em um ambiente de geoprocessamento. [FURTADO & FILHO, 2004].

### 4.2. MODELAGEM DO SISTEMA EXPERTCOP EM AORML

Na figura 1 apresentamos o sistema Expertcop numa visão de modelagem externa.

Os agentes são representados por retângulos com bordas arredondadas, os objetos são retângulos que são acionados via método. O sistema é iniciado através de um agente humano (usuário do sistema). Ele cria através de uma interface gráfica uma instância do agente de interface no qual é o agente artificial. O agente de interface dispara um evento para o JADE que é um container de agentes para a criação dos agentes no sistema. O JADE ao receber essa requisição de criação de novos agentes dispara um evento, representado por uma seta sem ponta no qual o sistema cria várias instancias de agentes. Esses agentes criados serão os principais agentes do sistema ExpertCop. Os agentes se comunicam via troca de mensagens ACL, essas mensagens são enviadas ao Interpretador ACL que são responsáveis por promover uma comunicação entres os agentes. O interpretador ACL se comunica com o geoprocessamento (GIS) enviando mensagens via socket, representadas através de seta tracejadas passando a localização do agente (Latitude e Longitude no mapa) e ao receber essas mensagens pelo GIS, essas mensagens são processadas e retorna ao agente Interpretador ACL. As mensagens processadas são enviadas para os agentes para realizar uma ação.



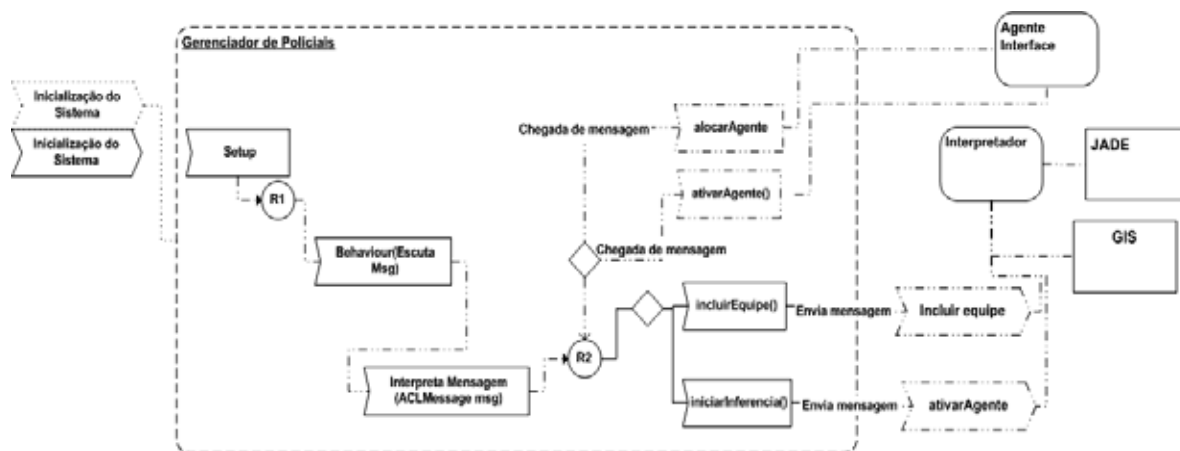
**Figura 1: Representação do Sistema ExpertCop utilizando AORML numa visão de modelo Externo**

Na figura 2 mostramos o agente gerenciador de Policiais do sistema Expertcop que é responsável em criar os policiais de acordo com a solicitação do agente de Interface.

Quando o sistema é inicializado é criado um evento *setup*, representado por uma seta sem ponta fechada, que representa um evento no qual chama uma regra de interação R1, representada por uma seta tracejada seguida de um círculo que cria outro evento, ou seja, um comportamento pelo agente responsável em escutar as mensagens que chegaram ao agente gerenciador de policiais.

O agente de interface será o responsável em enviar mensagens internas ao agente gerenciador de policiais que é representada através de uma seta sem ponta tracejada dentro do gerenciador de policiais para alocar um novo agente ou ativar um agente já criado. O losango representa a expressão *booleana* “OU”. O gerenciador de policiais chama a segunda regra de interação R2 e caso a mensagem seja incluir equipe é criado o evento incluir equipe que envia uma mensagem, representada por uma seta tracejada fora do gerenciador de policiais para o interpretador ACL requisitando ao JADE a criação de um novo agente ou caso a mensagem seja ativar agente é criado um outro evento iniciar inferência que envia uma mensagem, representada por uma seta tracejada fora do gerenciador de policiais para o JADE que ativa o agente já criado através do GIS.

A mensagem incluir equipe, representada pela seta tracejada, é um exemplo de entidade que não possui uma representação clara na UML. Para representar tal entidade na UML seria preciso usar uma classe estereotipada, o que deixaria o modelo ambíguo e confuso. Já na AORML podemos usar a simbologia e o seu significado correto para este tipo de entidade e assim criamos um modelo legível.



**Figura 2: Representação do agente Gerenciador de Policiais no Sistema ExpertCop baseado num modelo interno**

## 5. Resultados

No ExpertCop ganhamos mais na parte de documentação e manutenção, pois utilizando essa ferramenta houve uma maior compreensão e entendimento de quais são as funcionalidades de cada agente no sistema. Logo existe uma documentação gráfica da apresentação do sistema.

Na AORML há uma visualização mais detalhada na troca de mensagens entre os agentes envolvidos no sistema e como os comportamentos dos agentes são visualizados através das ações e eventos do sistema. Podendo diferenciar um modelo externo do modelo interno, onde o modelo externo engloba o sistema como um todo enquanto o modelo interno abrange apenas uma visão de como agente está se comportando diante das ações criadas pelo sistema.

Na modelagem externa o sistema possui 1 diagrama de interação com 8 agentes e 3 objetos. Na modelagem interna o sistema possui 9 diagramas com 6 agentes distintos e 4 objetos de classe diferentes.

## 6. Conclusão

Neste trabalho conclui-se que para um sistema multi-agentes a melhor solução para modelarmos seria a AORML, pois nele podemos ver as trocas de mensagens com os agentes, as ações que cada agente pode promover, os eventos possíveis de serem realizados pelos agentes, os compromissos que cada agente possui e as reivindicações que neles são necessárias para ser realizado um evento.

No Sistema ExpertCop, que é um sistema multi-agentes, podemos ver que utilizando uma linguagem de modelagem UML apresentavam-se falhas pois quando queríamos representa uma diagrama seqüência não haveria uma maneira concreta para representar as trocas de mensagens entre os agentes propostos e a única opção foi adotar a AORML para modelar o sistema, pois nos dois diagramas utilizados podemos observar a troca de mensagens entre eles.

## 7. Bibliografia

**Davidsson, P.** *Multi Agent Based Simulation: Beyond Social Simulation*. Department of Software Engineering and Computer Science, University of Karlskrona/Ronneby, 2000.

**WAGNER, Gerd.** The Agent-Object-Relationship Meta-Model: Towards a Unified View of State and Behavior. **Information Systems 28:5 (2003), pp. 475-504**

**WAGNER, Gerd.** A UML Profile for Agent-Oriented Modeling. Technical Report. Eindhoven Univ.of Technology, 2002.

**WEISS, Gerhard.** *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.

**JADE, Csel** <http://sharon.csel.it/projects/jade> **Parma - Itália**

**UML**, <http://www.uml.org/>

**FIPA**, <http://www.fipa.org/repository/index.html>

**FERBER, Jacques.** *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley Pub Co; 1st Edition, February 25 1999.

**SILVA, Leonardo Ayres Morais.** *Estudo e Desenvolvimento de Sistemas Multi-agentes usando JADE: Java Agent Development framework*. Fortaleza, UNIFOR, 2003. Monografia de conclusão de curso.

**FURTADO, Vasco & FILHO, José Eurico de Vasconcelos.** A Utilização de um Agente Pedagógico na Análise e Explicação de Dados Geosimulados: Uma Aplicação no Sistema ExpertCOP. Fortaleza, UNIFOR, 2004.