

# From simulation to development in MAS: A JADE-based Approach

João P. C. Lopes

Dissertação desenvolvida sob orientação de *Henrique Lopes Cardoso*

23 de Junho 2014

---

## 1. Resumo

Os sistemas multi-agente (MAS, *Multi-Agent Systems*) exprimem uma abordagem interessante no desenvolvimento de sistemas modulares e eficientes, o compostos por elementos computacionais autónomos – chamados agentes. Existem ferramentas de software que facilitam o desenvolvimento desta classe de sistemas que podem variar entre ferramentas de âmbito geral até ferramentas focadas num domínio específico.

### 1.1. Motivation

Simulações baseadas em multi-agentes (MABS) são usadas no ciclo de desenvolvimento de MAS completos – por exemplo para realização de testes – pela performance superior das simulações. No entanto, as plataformas de desenvolvimento de MAS são geralmente pouco apropriadas para desenvolver MABS por serem pouco escaláveis [1]. Trabalhos estudados revelam a existência de interesse em integrar MABS no ciclo de desenvolvimento de MABS. Por fim, existe a possibilidade de automatizar parcialmente o desenvolvimento de MAS a partir de simulações previamente testadas.

O JADE [2] é uma framework para desenvolvimento de MAS distribuídos que implementa as especificações da FIPA. Quando o número de agentes é elevado, a sua arquitetura multi thread afeta negativamente a performance de simulações baseadas no JADE.

O Repast [3] é uma ferramenta de criação de MABS rica em GUIs e estatísticas em tempo real. Consegue facilmente gerir grandes números de agentes numa única simulação. Ao contrário do JADE, não dispõem de uma infraestrutura de interação e criação de agentes.

A motivação desta dissertação reside no potencial ganho de performance ao usar frameworks de MABS para simular MAS mais complexos que aqueles tipicamente criados usando essas frameworks. Alguns trabalhos [4, 5] propõem a integração de ferramentas de simulação no JADE, quer criando uma camada de simulação por cima deste, ou integrando-o com outras plataformas como o Repast.

### 1.2. Objetivos

Nesta dissertação é proposta uma solução integrada para integrar o desenvolvimento e simulação de MAS. Para tal, dois sub-objetivos foram identificados.

1. **Primeiro**, a criação de um adaptador ou API que permite que os programadores se abstraiam da ferramenta de simulação. Isto é possível reimplementando funcionalidades do JADE, incluindo especificações FIPA sobre gestão de e interação entre agentes. Uma arquitetura idêntica à do JADE permite uma conversão mais direta do código.
2. **Segundo**, o desenvolvimento de um mecanismo de conversão de código. Ao permitir a abstração da plataforma de simulação, é possível a criação de uma ferramenta que realize a conversão direta de um MABS num MAS equivalente.

### 1.3. SAJaS

O SAJaS (*Simple API for JADE-based Simulations*) implementa um conjunto de funcionalidades baseadas no JADE; foram reimplementadas de raiz de modo a simplificar algumas características internas e a aumentar a performance da simulação, preservando o comportamento do MAS. Entre as funcionalidades implementadas, destacam-se as especificações para gestão de agentes, serviço de mensagens e interação entre agentes da FIPA.

Um dos desafios ao desenvolver o SAJaS foi a adaptação da execução assíncrona do JADE ao ambiente síncrono baseado em “ticks” do Repast. Exceto quando ocorrem durante a iniciação e terminação dos agentes no JADE, as suas ações estão tipicamente encapsuladas em “Behaviours” que executam em concorrência ou em paralelo. No SAJaS, os behaviours são executados sequencialmente. Para emular a comunicação assíncrona do JADE no SAJaS, as mensagens são mantidas na fila de espera do destinatário até serem necessárias e processadas – ao contrário do que se verifica no JADE, em que as mensagens despoletam o “Behaviour” apropriado de imediato.

Os “behaviours” no SAJaS seguem o mesmo ciclo de vida que no JADE, incluindo os métodos **action**, **onStart**, **done** e **onEnd**. Também está presente a implementação de protocolos de interação, nomeadamente o “FIPA Request” e o “FIPA Contract Net”.

Foi feito um esforço consciente para manter o SAJaS genérico no que diz respeito a dependências ao Repast, abrindo as portas à integração futura em outras ferramentas de simulação sem a necessidade de modificar a API.

#### 1.4. MASSim2Dev

O MASSim2Dev (*MAS Simulation to Development code conversion tool*) é uma ferramenta de que, utilizando o SAJaS, estabelece a ponte entre o desenvolvimento e simulação de MAS.

O JDT ([www.eclipse.org/jdt/](http://www.eclipse.org/jdt/)) foi selecionado para desenvolver esta ferramenta. Entre as suas capacidades, destacam-se a capacidade de (dinamicamente) clonar projetos Java, gerir classes, “imports”, métodos e variáveis como objetos e de realizar manipulações de código complexas sem analisar todo o código (“parsing”). Também permite a utilização de uma árvore sintática (AST) de alto nível para manipulação direta do código fonte.

Após a conversão do código, não restam dependências à plataforma anterior no projeto gerado. Naturalmente, isto apenas se verifica para as funcionalidades comuns ao JADE e ao SAJaS presentemente.

#### 1.5. Validação

Foram criados três testes para demonstrar que o comportamento das funcionalidades baseadas no JADE que foram reimplementadas no SAJaS é equivalente ao das correspondentes no JADE. Além disso, foi possível obter ganhos na performance de simulações baseadas no SAJaS.

O primeiro exemplo consiste numa “Contract Net” entre um comprador e múltiplos vendedores. No segundo exemplo, múltiplas “Contract Nets” correm concorrentemente e alguns dos compradores utilizam valores de confiança para estabelecer contratos. No terceiro exemplo, um jogo de tabuleiro chamado Risk, desenvolvido em JADE antes da conceção desta dissertação foi sujeito à conversão do seu código. O objetivo deste último era testar o sistema num exemplo real que não tivesse sido desenvolvido de propósito para esta dissertação.

Estes cenários cobrem todas as funcionalidades do SAJaS. Foi possível demonstrar que é possível traçar a ponte entre o JADE e o Repast e que os ganhos de performance na simulação são significativos.

#### 1.6. Conclusions

O estudo da bibliografia demonstrou que existe um interesse em desenvolver ferramentas para simulação de MAS devido a limitações existentes em plataformas de desenvolvimento.

A solução proposta, composta pelo SAJaS e pelo MASSim2Dev, permite o desenvolvimento de simulações usando funcionalidades baseadas no JADE numa plataforma de simulação como o Reapast.

Neste momento, um subconjunto das funcionalidades do JADE estão presentes no SAJaS. No entanto, os testes de validação mostram que é já possível criar MAS com alguma complexidade e que a sua conversão com o MASSim2Dev preserva o comportamento original.

Algum trabalho futuro sugerido inclui expandir o conjunto de funcionalidades do JADE presentes no SAJaS, estender o suporte nativo da API a outras plataformas de simulação, complementar o plugin com configurações do utilizador e permitir a criação de gráficos e visualizações através do SAJaS.

#### References

- [1] D Mengistu, P Troger, L Lundberg, e P Davidsson. Scalability in distributed multi-agent based simulations: The jade case. Em *Future Generation Communication and Networking Symposia, 2008. FGCNS'08. Second International Conference on*, volume 5, páginas 93–99. IEEE, 2008.
- [2] Fabio Luigi Bellifemine, Giovanni Caire, e Dominic Greenwood. *Developing multi-agent systems with JADE*, volume 7. John Wiley & Sons, 2007.
- [3] N Collier. Repast: An extensible framework for agent simulation. *The University of Chicago's Social Science Research*, 36, 2003.
- [4] E García, S Rodríguez, B Martín, C Zato, e B Pérez. Misia: Middleware infrastructure to simulate intelligent agents. Em *International Symposium on Distributed Computing and Artificial Intelligence*, páginas 107–116. Springer Berlin Heidelberg, 2011.
- [5] J Gormer, G Homoceanu, C Mumme, M Huhn, e J Muller. Jrep: Extending repast symphony for jade agent behavior components. Em *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02*, páginas 149–154. IEEE Computer Society, 2011.