

# From Simulation to Development in MAS: A JADE-based Approach

João Pedro Camacho Lopes  
Orientador: Henrique Lopes Cardoso

14 de Julho 2014

# AGENDA

Introdução

Estado da Arte

Proposta

SAJaS

MASSim2Dev

Validação

Conclusões

# CONTEXTO

- ▶ Sistemas multi-agente (MAS) focam-se na interação entre agentes.
  - ▶ Agentes são inteligentes e autónomos mas os MAS conseguem ter comportamentos complexos.
  - ▶ Jogos, negociação eletrónica, problemas não determinísticos, gestão de operações...
- 
- ▶ Simulações baseadas em multi-agentes (MABS) permitem testar sistemas complexos de agentes
  - ▶ Simulação de tráfego, contextos sociais, negociação..
  - ▶ Tipicamente mais simples, rápidas e sincronizadas
  - ▶ Ambiente controlado

# PROBLEMAS

- ▶ Múltiplos standards para desenvolvimento de MAS
- ▶ Uma única plataforma não cumpre a totalidade dos requisitos da maioria dos projetos
- ▶ Fraca performance na simulação de MAS usando plataformas de desenvolvimento.
- ▶ Plataformas para simulações baseadas em multi-agentes (MABS) têm infraestruturas mais simples.

# MOTIVAÇÃO

- ▶ Interesse em simulação durante o ciclo de desenvolvimento de MAS
- ▶ Ganhos de performance usando ferramentas de simulação
- ▶ Possibilidade de aproximar a simulação e o desenvolvimento

# OBJETIVOS

Aproximar simulação e o desenvolvimento de MAS:

1. Criar um adaptador para ferramentas de simulação
2. Desenvolver um mecanismo de conversão baseado em 1.

# OBJETIVOS

Plataformas selecionadas: JADE (MAS) e Repast (MABS)

- ▶ Amplamente utilizadas e populares
- ▶ Documentação abrangente
- ▶ Código livre e aberto



1. Replicar funcionalidades do JADE no Repast
2. Facilitar a simulação de modelos JADE
3. Permitir integração de plataformas além do Repast

# ESTADO DA ARTE

## Plataformas de desenvolvimento de MABS

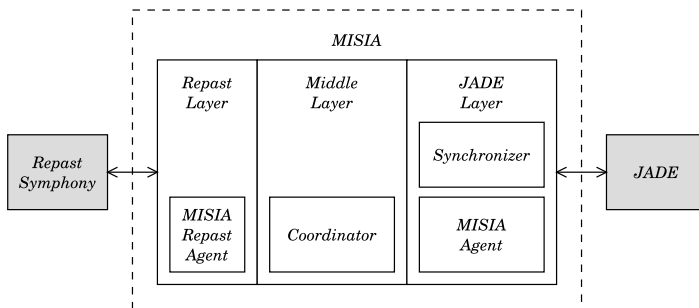
- ▶ Domínio específico (p.e. MASERaTi, MATSim, SUMO)
- ▶ Propósito geral (p.e. Repast, NetLogo, GALATEA)

Foco em integração de simulação no JADE (p.e. MISIA, JRep, PlaSMA)



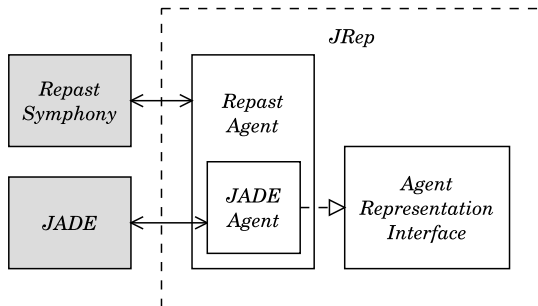
# MISIA

BISITE, Universidad de Salamanca ([bisite.usal.es](http://bisite.usal.es))



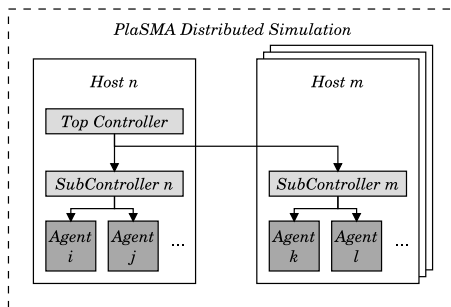
# JREP

Niedersächsische Technische Hochschule  
([www.nth-online.org](http://www.nth-online.org))



# PLASMA

TZI, Universität Bremen ([www.tzi.de](http://www.tzi.de))



# COMPARAÇÃO

- ▶ MISIA, JRep e PlaSMA dependem da plataforma JADE durante a simulação
- ▶ Agentes JADE correm em threads individuais - escalabilidade limitada
- ▶ JRep, MISIA não são projetos ativos

# JADE E REPAST

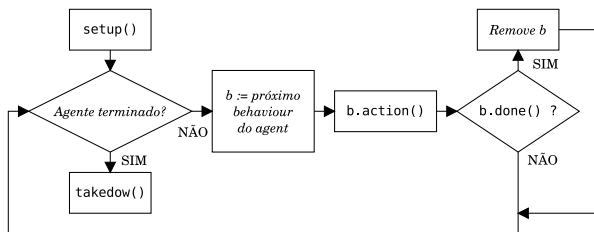
Comparação entre JADE e Repast.

	<b>JADE</b>	<b>Repast</b>
Distribuído	Sim	Não
Simulação	Não	Sim
Escalabilidade	Limitada	Elevada
Ontologias	Sim	Não
Execução	Concorrente Assíncrona	Consecutiva Síncrona
Comunicação	FIPA ACL <sup>1</sup>	Acesso direto Partilha de recursos
Código Livre	Sim	Sim
Linguagem	Java	Java

<sup>1</sup>Agent Communication Language

# FUNCIONALIDADES DO JADE

- ▶ Especificações FIPA:
  - ▶ Gestão de Agentes (DF, AMS, AID)
  - ▶ Serviço de Mensagens (MTS, ACL Message)
  - ▶ Protocolos de interação (FIPA Request, Contract Net...)
- ▶ Encapsulamento de ações em Behaviours
- ▶ Ontologias



# PROPOSTA

## Solução integrada

1. SAJaS (Simple API for JADE-based Simulations)
  - ▶ Reimplementa funcionalidades do JADE
  - ▶ *Standalone* ou em conjunto com MASSim2Dev
2. MASSim2Dev (MAS Simulation To Development conversion tool)
  - ▶ Plugin Eclipse
  - ▶ Utiliza o SAJaS

# CENÁRIOS DE UTILIZAÇÃO

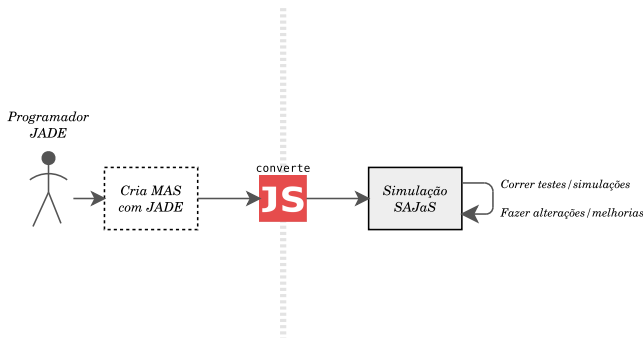
Principais fluxos de trabalho.





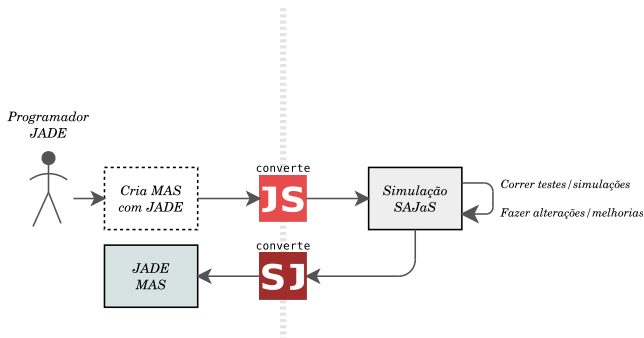
# CENÁRIOS DE UTILIZAÇÃO

Principais fluxos de trabalho.



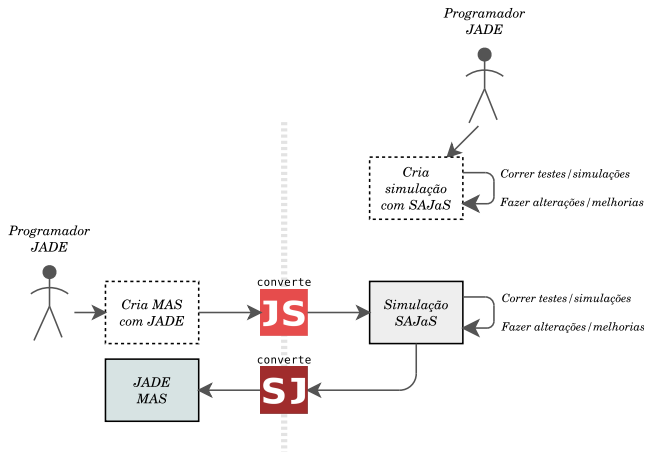
# CENÁRIOS DE UTILIZAÇÃO

Principais fluxos de trabalho.



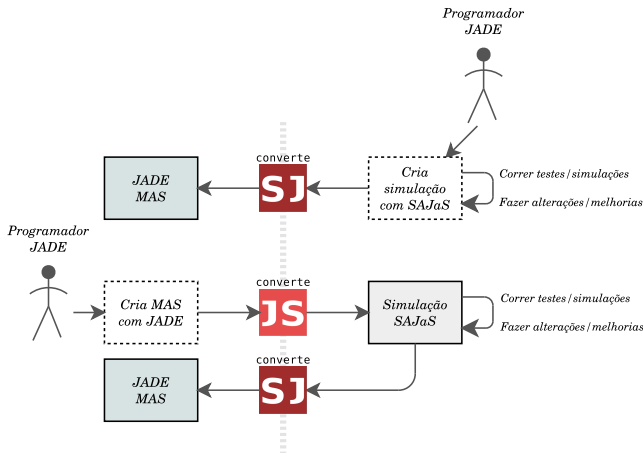
# CENÁRIOS DE UTILIZAÇÃO

## Principais fluxos de trabalho.



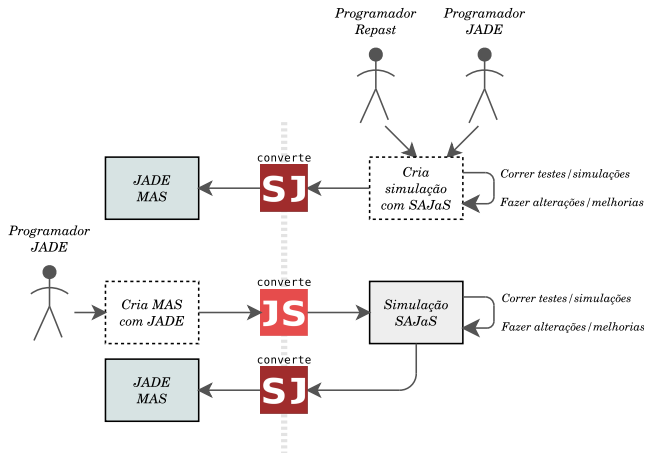
# CENÁRIOS DE UTILIZAÇÃO

## Principais fluxos de trabalho.



# CENÁRIOS DE UTILIZAÇÃO

## Principais fluxos de trabalho.

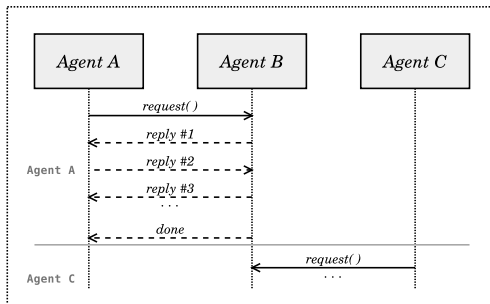


# SAJaS

Ações dos agentes são encapsuladas em “Behaviours”  
Duas possibilidades de execução de implementação da comunicação:

# SAJaS

Ações dos agentes são encapsuladas em “Behaviours”  
Duas possibilidades de execução de implementação da comunicação:

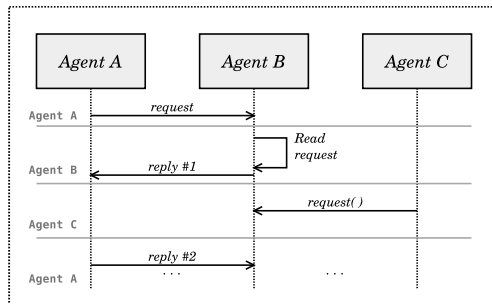


(a) Acesso direto



# SAJaS

Ações dos agentes são encapsuladas em “Behaviours”  
Duas possibilidades de execução de implementação da comunicação:



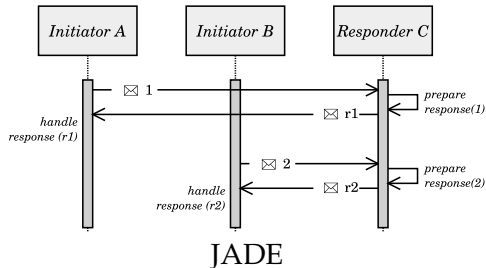
(b) Comunicação assíncrona

# SAJaS

## Solução para comunicação “assíncrona” no SAJaS

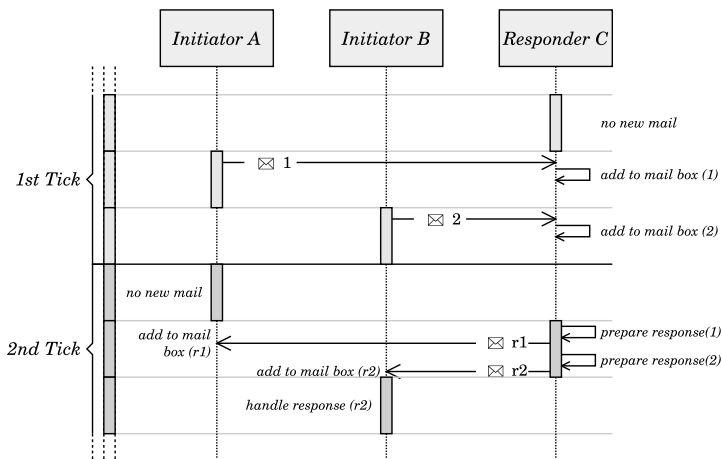
# SAJAS

## Solução para comunicação “assíncrona” no SAJaS



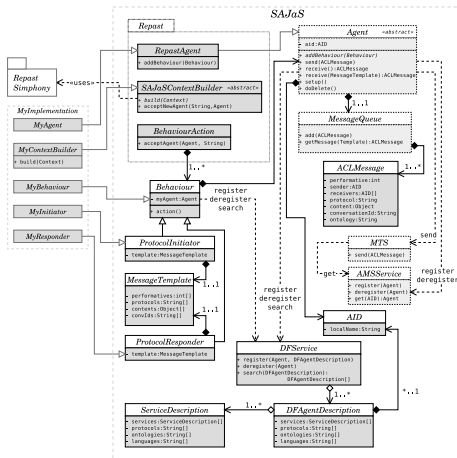
# SAJaS

## Solução para comunicação “assíncrona” no SAJaS

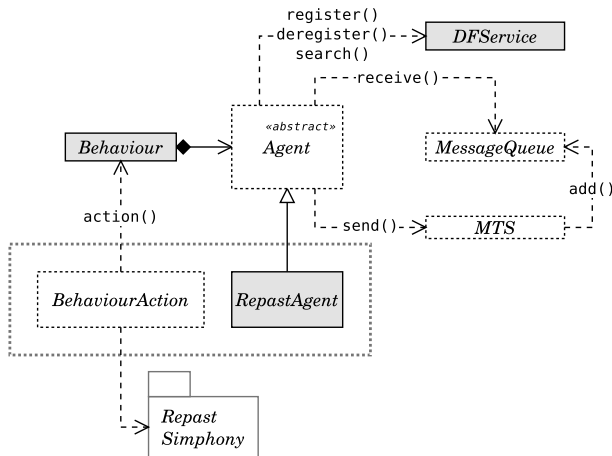


SAJaS

## Arquitetura do SAJaS

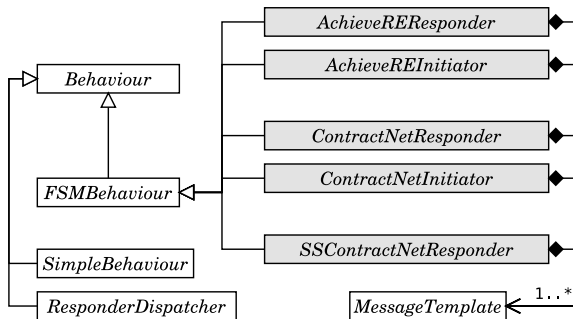


# ARQUITETURA



Arquitetura simplificada do SAJaS

# FIPA



## Behaviours e Protocolos no SAJaS

# MASSim2Dev - FERRAMENTA DE CONVERSÃO

- ▶ MASSim2Dev (*MAS Simulation to Development code conversion tool*)
- ▶ Estabelece a ponte entre o desenvolvimento e simulação de MAS utilizando o SAJaS



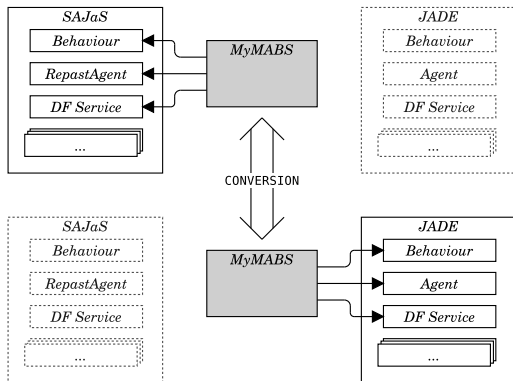
# MASSim2Dev - FERRAMENTA DE CONVERSÃO

Transformações de código Java: ferramentas estudadas

- ▶ ATL (“ATLAS Transformation Language”) - transformações de modelos através de AST usando linguagem específica;
- ▶ Spoon - transformações de código Java baseadas em anotações, usando Java
- ▶ JDT (“Eclipse Java Development Tools”) - criação de plugins para Eclipse, que permitem edições de alto nível, assim como via AST

# MASSim2Dev - FERRAMENTA DE CONVERSÃO

Após a conversão, não restam dependências da plataforma original.



# MASSim2Dev - FERRAMENTA DE CONVERSÃO

## Algoritmo

1. Clonar o projeto selecionado
2. Mudar as referências a *imports* em cada classe
3. Introduzir as bibliotecas necessárias (JADE, Repast...) e adiciona-las ao *build path* do projeto
4. Reparar hierarquia (p.e. classes que estendem RepastAgent passam a estender Agent)

# VALIDAÇÃO

Três testes de validação criados

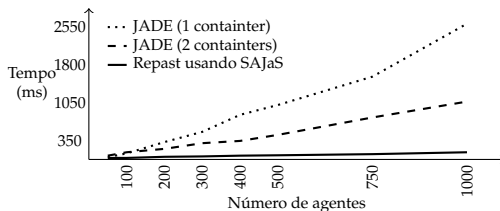
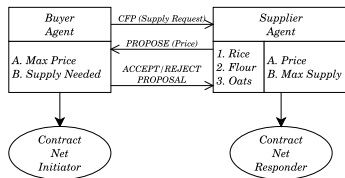
- ▶ Rede contratual simples
- ▶ Rede contratual com múltiplas sessões
- ▶ Jogo de tabuleiro RISK

Métricas avaliadas

- ▶ Resultado da execução do teste no SAJaS e no JADE
- ▶ Performance em ambos os casos

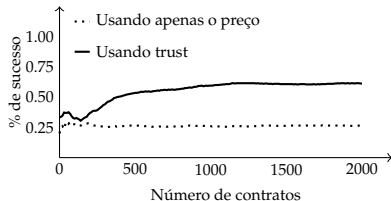
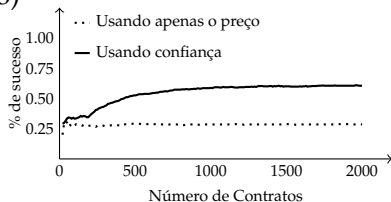
# VALIDAÇÃO

## Rede Contratual Simples



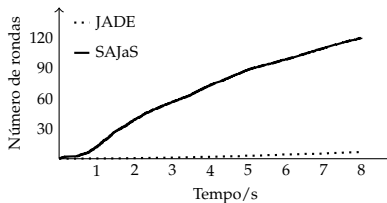
# VALIDAÇÃO

Rede contratual com múltiplas sessões. Gráficos: sucesso médio em 5 execuções ao longo 2000 contratos no SAJaS (cima) e no JADE (baixo)



# VALIDAÇÃO

Jogo de tabuleiro RISK. Gráfico: performance de um jogo de Risk com 5 agentes durante 8 segundos.



# CONCLUSÕES

- ▶ Existe interesse na aproximação do desenv. de MAS e MABS
- ▶ Solução integrada SAJaS + MASSim2Dev é possível integrando funcionalidades do JADE numa plataforma de simulação
- ▶ É possível automatizar parcialmente a geração de MAS a partir de simulações, ou vice versa.
- ▶ O ganho de performance ao utilizar plataformas de simulação é significativo



# TRABALHO FUTURO

- ▶ 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
- ▶ Permitir a criação de gráficos e visualizações através do SAJaS

# DOCUMENTAÇÃO

- ▶ Código/documentação - ([www.github.com/lopis](http://www.github.com/lopis))
- ▶ Artigo científico submetido para o MATES 2014 ([www.mates2014.de](http://www.mates2014.de)) descrevendo o desenvolvimento do SAJaS
- ▶ Artigo científico em produção, a submeter para o ESM'2014 ([www.eurosis.org](http://www.eurosis.org)), sobre o SAJaS e MASSim2Dev
- ▶ Relatórios, diapositivos e outras informações em [www.jlopes.pt/thesis](http://www.jlopes.pt/thesis)