

# Integração de serviços via Sistemas Multiagentes: Redes de Computadores e Sistemas Ubíquos

Alexandre Zamberlan

alexz@ufn.edu.br

Universidade Franciscana

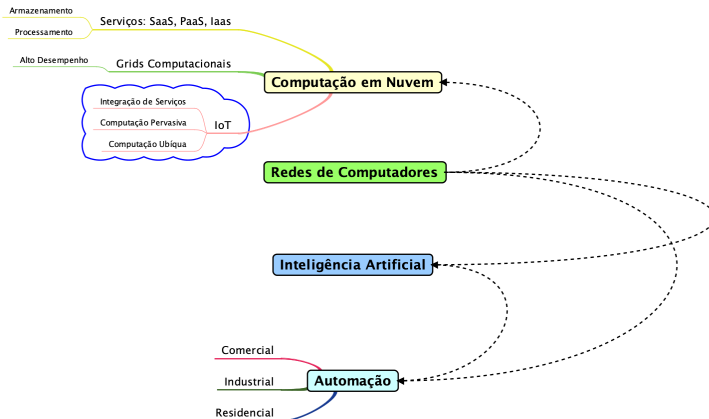
# Agenda

- 1 Contexto
- 2 Sistemas de Comportamento Inteligente
- 3 Automação
- 4 Considerações

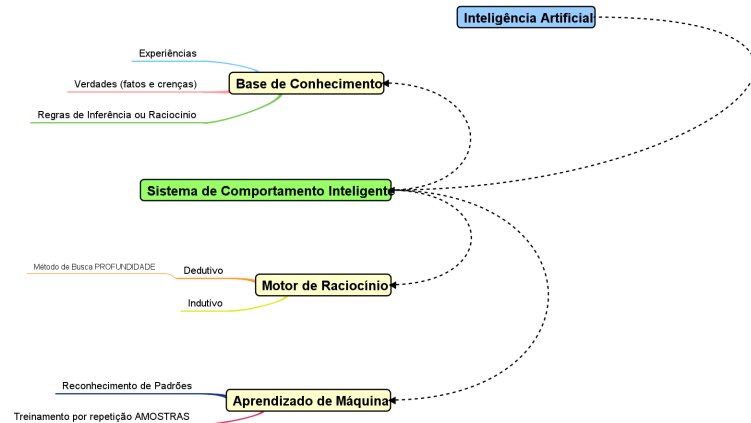
# Agenda

- 1 Contexto
- 2 Sistemas de Comportamento Inteligente
- 3 Automação
- 4 Considerações

# Redes de Computadores e seus impactos



# IA e Sistemas de Comportamento Inteligente



# IA e Sistemas de Comportamento Inteligente

Q\_\_M N\_\_ CH\_R\_ N\_O M\_M\_!!

# Agenda

- 1 Contexto
- 2 Sistemas de Comportamento Inteligente
- 3 Automação
- 4 Considerações

# Sistemas Multiagentes e a vida



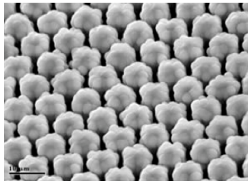
(A)



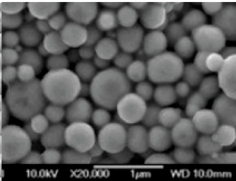
(B)



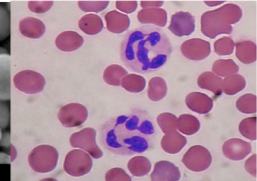
(C)



(D)



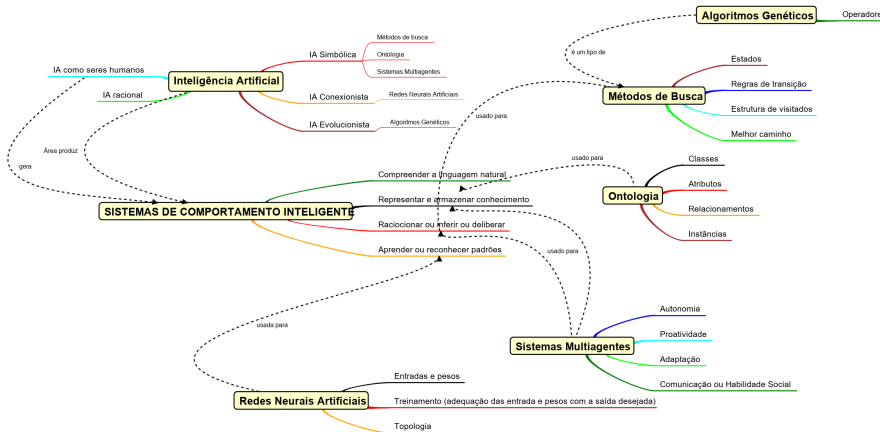
(E)



(F)



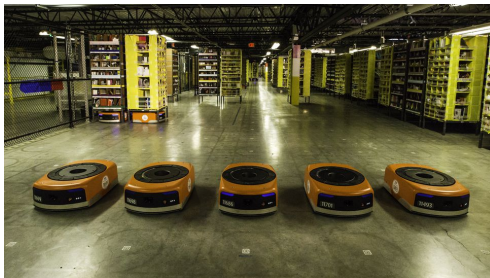
# IA e suas conexões



# Algumas aplicações de SCI



# Algumas aplicações de SCI



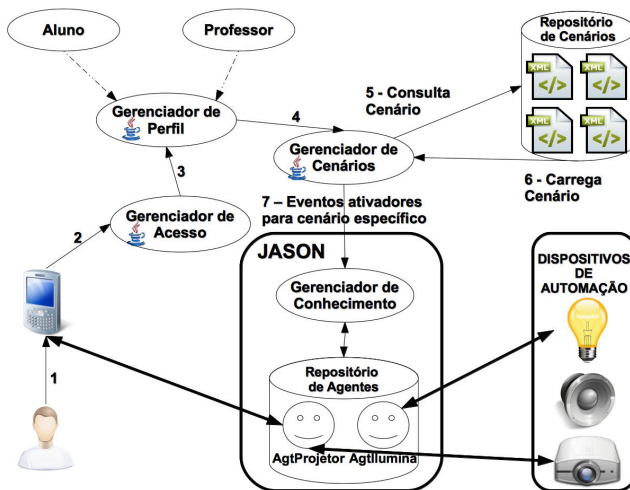
# Algumas aplicações de SCI



# Algumas aplicações de SCI



# Algumas aplicações de SCI



# Sistemas de Comportamento Inteligente

- Deve ser capaz de adaptar-se a novas situações
- Raciocinar relações entre fatos
- Descobrir significados
- Reconhecer e aprender com base em experiências

# Sistemas Multiagentes

- Método baseado em comportamento coletivo
- Agentes como elementos centrais
  - autônomos
  - flexíveis e adaptáveis
  - reativos: percebem e atuam (ambiente)
  - **habilidade social: interagem entre si (REDES)**
- Metodologias e ferramentas consolidadas
- Abordagem permite grau de abstração elevado
  - simulações  $\implies$  maior número de variáveis, restrições, operações, exceções
  - *Bottom-up* e *Top-down*  $\implies$  como sistemas particulados

[Bordini and Hübner 2009]



## Kits ou ferramentas de desenvolvimento

Possuem uma variedade de características e funcionalidades:

- ambiente de desenvolvimento integrado;
- linguagem de programação;
- sistema operacional;
- suporte ao usuário (manuais e exemplos);
- integração com outras bibliotecas;
- possibilidade de executar o sistema com visualização 2D e 3D;
- propriedade de visualização de cenários de simulação.

### Principais ferramentas

FLAME, JASON, MASON, NetLogo, Repast, SeSAm, SWARM

# JASON: Kit De desenvolvimento SW Livre e Aberto



The screenshot shows the official website for JASON, a Java-based interpreter for an extended version of AgentSpeak. The header features a classical painting of a winged figure and the text 'Jason by Gustave Moreau (1882)'. The main title 'Jason' is in a large, stylized font, followed by the subtitle 'a Java-based interpreter for an extended version of AgentSpeak'. A navigation bar includes links for Home, Description, Documents, Examples, Demos, Teaching, and Projects, along with a search bar. The main content area is divided into three columns. The left column contains the JASON logo and an 'About Jason' section. The middle column has a 'Download the latest version of Jason!' section with a 'DOWNLOAD' button, a 'Read the tutorial for installing Jason as an Eclipse plug-in!' section with an 'ECLIPSE PLUG-IN' button, and a 'News' section. The right column contains a 'Links' section with links to GitHub, SourceForge, and Screenshots, an 'Authors' section, and a 'Watch replays of the matches and see the results here.' link. The footer of the website shows the date '20/08/2017'.

**Jason**  
a Java-based interpreter for an  
extended version of AgentSpeak

Jason  
by Gustave Moreau (1882)  
Oil on canvas, 204 x 113 cm  
Musée d'Orsay, Paris  
© Photo: Getty Images by  
Hans Landerbach

Home Description Documents Examples Demos Teaching Projects Search

**Jason**  
Agent Programming

**About Jason**

Jason is an interpreter for an extended version of AgentSpeak. It implements the operational semantics of that language, and provides a platform for the development of multi-agent systems, with many user-customisable features. Jason is available Open Source, and is distributed under GNU LGPL. See more in the [Description](#) page.

**Links**

- Jason on [Github](#) (latest code);
- Jason on [Sourceforge](#);
- [Screenshots](#).

**Authors**

Jason is developed by [Jomi F. Hübner](#) and [Rafael H. Bordini](#), based on previous work done with many colleagues, in particular Michael Fisher, Joyce Martins, Álvaro Moreira, Renata Vieira, Willem Visser, Mike Wooldridge, but also many others, as acknowledged in

Download the latest  
version of Jason!

**DOWNLOAD**

Read the tutorial for installing  
Jason as an Eclipse plug-in!

**ECLIPSE PLUG-IN**

**News**

25/08/2017  
**The Multi-Agent Programming  
Contest 2017** had two teams using  
Jason:

- 2nd place: Jason-DTU
- 4th place: SMART-JaCaMo (using  
[JaCaMo](#))

Watch replays of the matches and  
see the results [here](#).

20/08/2017

# JASON: Modelagem

- evento ativador (*trigger*)
- contexto ou condição
- plano ou planos contingenciais
- atualização da base de conhecimento (fatos/crenças)
- Vídeo no Youtube sobre modelagem de Agentes
  - [https://youtu.be/IBNh7j\\_F4yc](https://youtu.be/IBNh7j_F4yc)

# JASON: Modelagem

- Unidade básica é o AGENTE
  - sensores
  - conjunto de planos
  - atuadores
- Características fundamentais:
  - Autonomia - *threads*
  - Proatividade - planos/métodos sobrecarregados
  - Adaptação ou flexibilidade - tratamento de exceções
  - Comunicação - *socket*
    - enviar um fato/crença
    - enviar um questionamento/pergunta
    - enviar uma ação ou plano

# JASON: arquivos

```
1 Jason - três tipos de arquivo
2
3 .mas2j
4     -> descrição do projeto do SWA
5
6
7 .java
8     -> código java que implementa o ambiente em que os agentes estão inseridos
9     -> principais métodos:
10         init() -> inicialização do SWA -> com percepções iniciais
11         executeAction() -> tratamento das ações solicitadas pelos agentes
12
13     -> métodos secundários:
14         addPercept() -> ambiente avisando todos os agentes de uma percepção
15         removePercept() -> ambiente retirando uma percepção gerada
16
17
18 .asl
19     -> código AgentSpeak(L), que é um Prolog melhorado, contendo:
20         - crenças ou fatos do ambiente (base de conhecimento de um agente)
21         - planos ou regras (conjunto de ações que são disparadas quando um evento
22           ocorre no ambiente - evento externo)
23         - subplano (conjunto de ações que são disparadas quando um evento interno
24           ocorre)
25     -> alguns comandos de apoio que são do Jason
26         - .print() -> exibe alguma coisa no terminal de verbose
27         - .send() -> diretiva de comunicação entre agentes
28             .send(agente,tell,crenca/fato)
29             .send(agente,achieve,plano)
30             .send(agente,untell,crenca/fato)
31
32         - .broadcast -> diretiva de comunicação de um agente para muitos agentes
33             .broadcast(tell,crenca/fato)
34             .broadcast(untell,crenca/fato)
35
```

# Agenda

- 1 Contexto
- 2 Sistemas de Comportamento Inteligente
- 3 Automação**
- 4 Considerações

# Automação

# Agenda

- 1 Contexto
- 2 Sistemas de Comportamento Inteligente
- 3 Automação
- 4 Considerações



# Referências

 Bordini, R. H. and Hübner, J. F. (2009).

Agent-based simulation using bdi programming in jason.

In *M. Uhrmacher ; Danny Weyns. (Org.). Multi-Agent Systems: Simulation and Applications, Modeling and Simulation*, pages 451–476. CRC Press.

 Perozzo, R. F. (2011).

*Framework para integração entre ambientes inteligentes e o sistema brasileiro de TV Digital.*

PhD thesis, Universidade Federal do Rio Grande do Sul - UFRGS, Porto Alegre.

 Zamberlan, A. (2018).

*Sistema Multiagente para avaliação do efeito de aglomeração em nanopartículas poliméricas.*

PhD thesis, Universidade Franciscana - UFN, Santa Maria.

# Integração de serviços via Sistemas Multiagentes: Redes de Computadores e Sistemas Ubíquos

Alexandre Zamberlan

alexz@ufn.edu.br

Universidade Franciscana