

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

Contexto

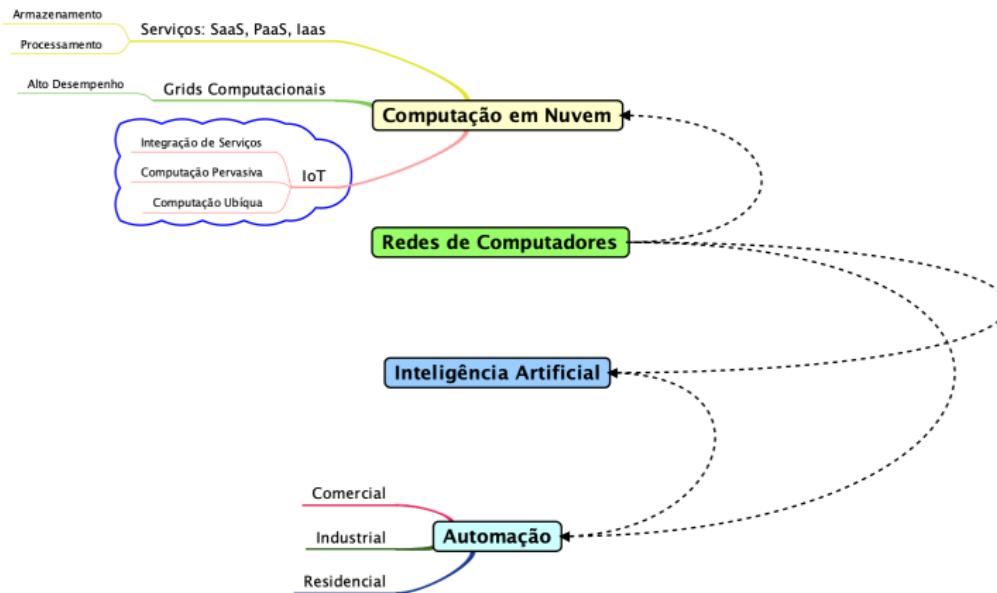
**5^a Semana Acadêmica
Redes de Computadores**

08, 09 e 10 de Novembro
Das 9:00 até as 17:30

**SARC
2023**

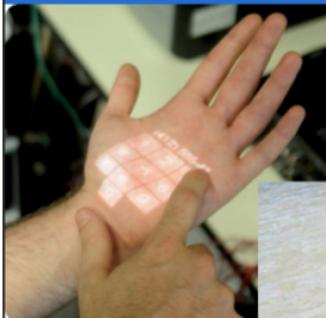
Universidade Federal de Santa Maria
UFSC
Colégio Técnico Industrial
UFSM
TECNOLOGIA EM REDES DE COMPUTADORES
UFSM

Redes de Computadores e seus impactos



Redes de Computadores e seus impactos

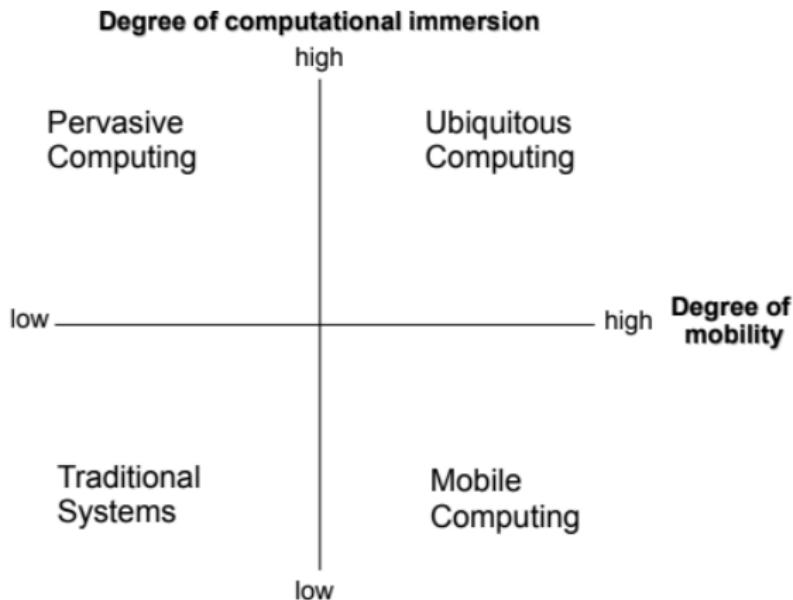
Background – pervasive computing



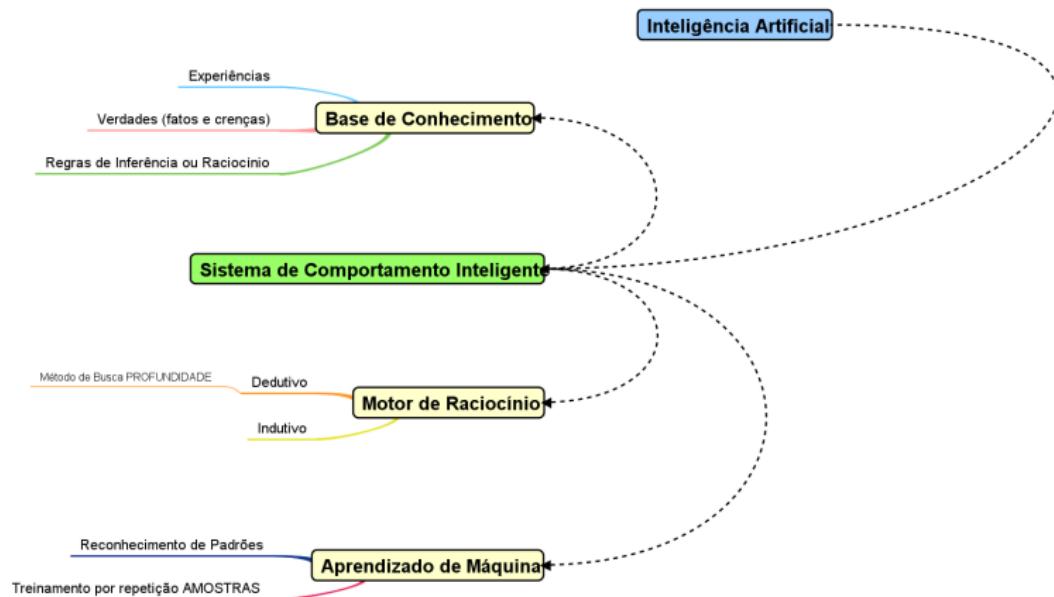
7

[Zamberlan et al. 2014]

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



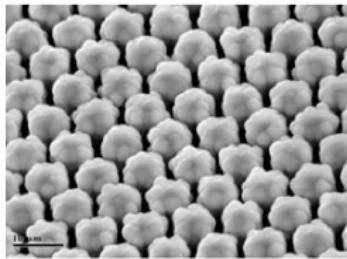
(A)



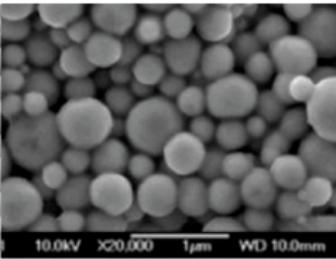
(B)



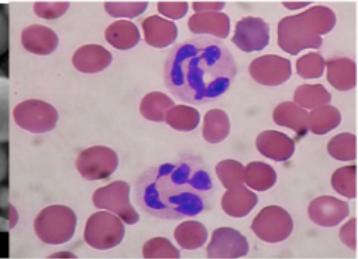
(C)



(D)

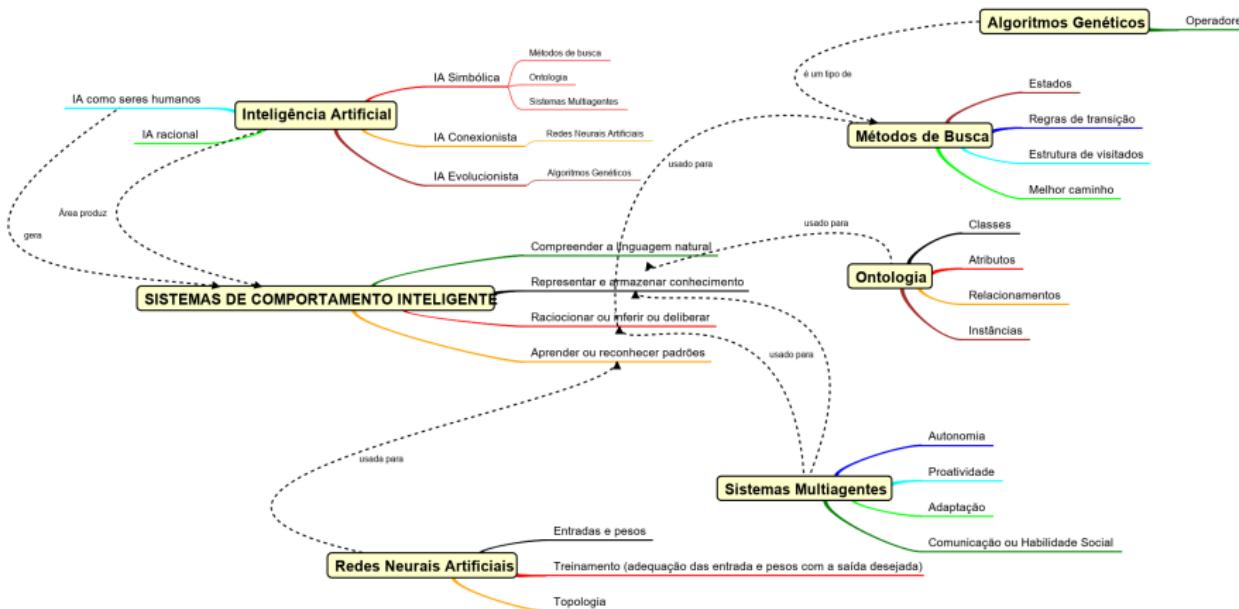


(E)



(F)

IA e suas conexões



Algumas aplicações de SCI



Algumas aplicações de SCI



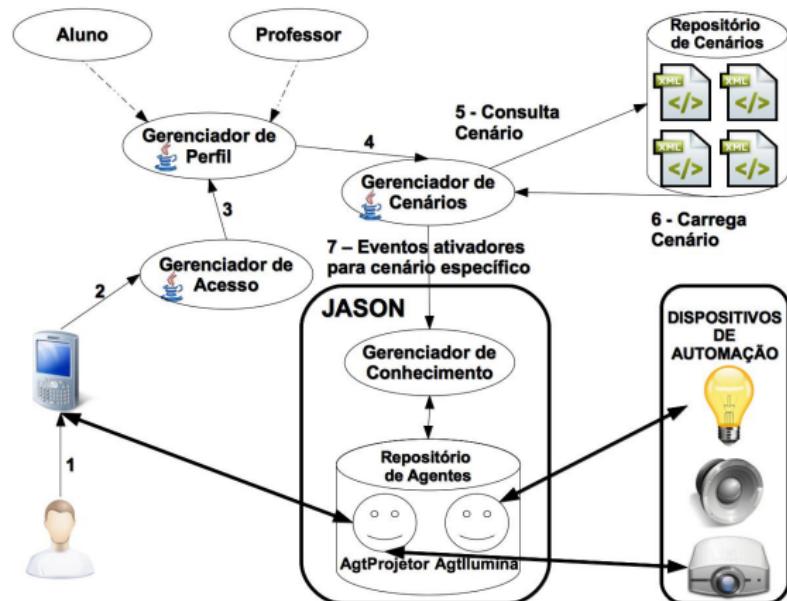
Algumas aplicações de SCI



Algumas aplicações de SCI



Algumas aplicações de SCI



Sistemas de Comportamento Inteligente

- 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 \Rightarrow possibilidade de testar antes
 - *Bottom-up* e *Top-down*

[Bordini and Hübner 2009]

Sistemas Multiagentes e a vida



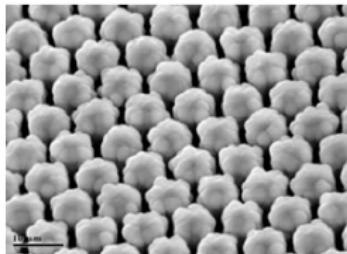
(A)



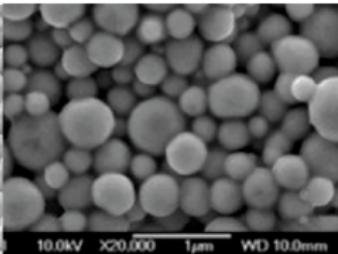
(B)



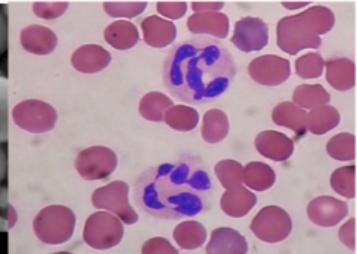
(C)



(D)



(E)



(F)

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 Jason project's website. At the top left is a painting by Gustave Moreau titled "Jason". To its right is the title "Jason" in a large serif font, followed by the subtitle "a Java-based interpreter for an extended version of AgentSpeak". Below this is a navigation bar with links for Home, Description, Documents, Examples, Demos, Teaching, Projects, and a search bar. The main content area features a logo for "Jason Agent Programming" with a portrait of a man. To the right of the logo are two download buttons: one for the "latest version of Jason!" and another for the "ECLIPSE PLUG-IN". Below these are sections for "About Jason", "Links", "Authors", and "News". The "About Jason" section describes Jason as an interpreter for an extended version of AgentSpeak, implemented with operational semantics, and available as Open Source under GNU LGPL. The "Links" section lists Jason on GitHub, Sourceforge, and Screenshots. The "Authors" section credits Jomi F. Hübner and Rafael H. Bordini. The "News" section contains two entries: one from 25/09/2017 about the Multi-Agent Programming Contest 2017, and another from 20/09/2017 announcing the results of the contest.

Jason

a Java-based interpreter for an extended version of AgentSpeak

Home Description Documents Examples Demos Teaching Projects Search

Jason Agent Programming

Download the latest version of Jason!

DOWNLOAD

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

ECLIPSE PLUG-IN

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

25/09/2017

The [Multi-Agent Programming Contest](#) 2017 had two teams using Jason:

- 2nd place: Jason-DTU
- 4th place: SMART-JaCoMo (using JaCoMo)

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

20/09/2017

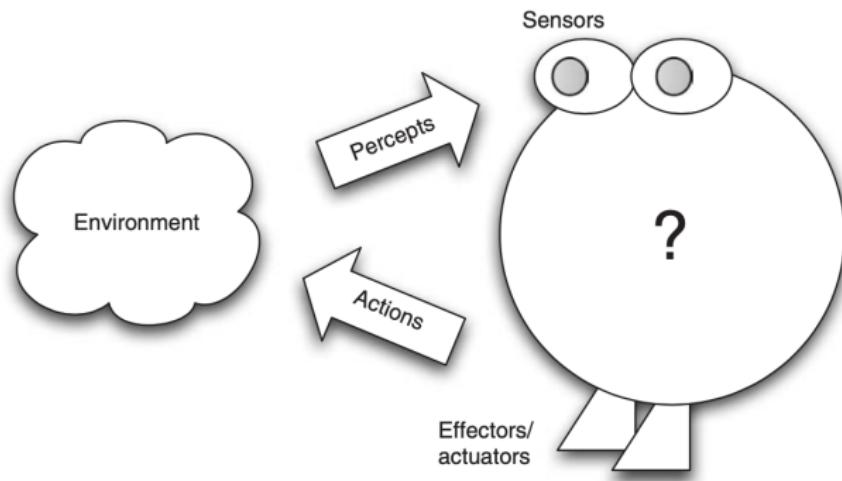
JASON: Modelagem

- evento ativador (*trigger*)
- contexto ou condições
- plano ou planos contigenciais
- atualização da base de conhecimento (fatos/crenças)
- Vídeo no Youtube sobre modelagem de Agentes
 - https://youtu.be/IBNh7j_F4yc

JASON: Modelagem

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

JASON: Modelagem



[Bordini and Hübler 2009]

JASON: arquivos

```
1 Jason - três tipos de arquivo
2
3 .mas2j
4 -> descrição do projeto do SNA
5
6
7 .java
8 -> código java que implementa o ambiente em que os agentes estão inseridos
9 -> principais métodos:
10     init() -> inicialização do SNA -> 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     removePercep() -> 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,crença/fato)
29         .send(agente,achieve,plano)
30         .send(agente,untell,crença/fato)
31
32     - .broadcast -> diretiva de comunicação de um agente para muitos agentes
33         .broadcast(tell,crença/fato)
34         .broadcast(untell,crença/fato)
35
```

Agenda

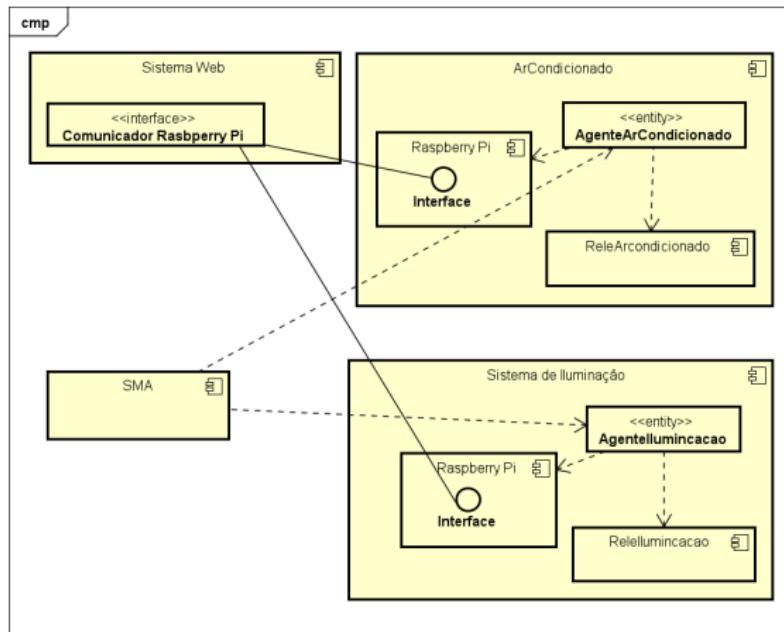
1 Contexto

2 Sistemas de Comportamento Inteligente

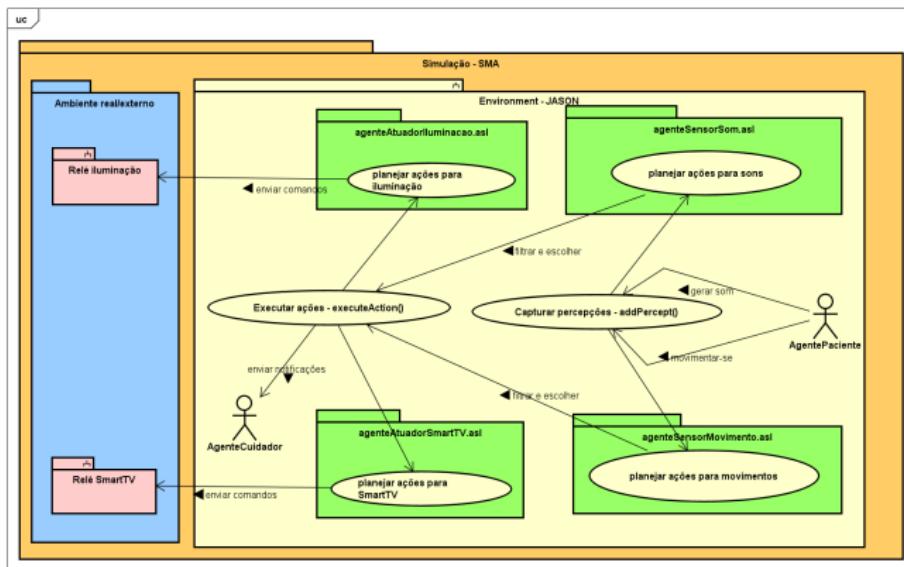
3 Automação

4 Considerações

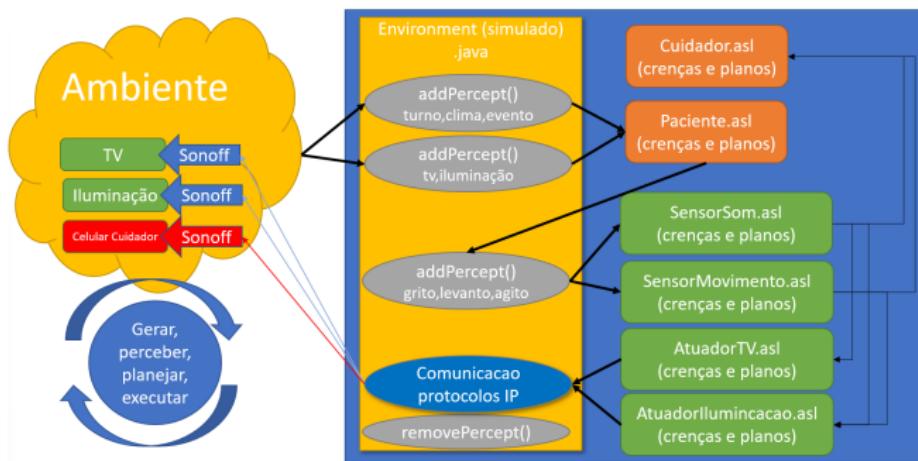
Automação



Automação



Automação



[Inácio and Zamberlan 2023]

Agenda

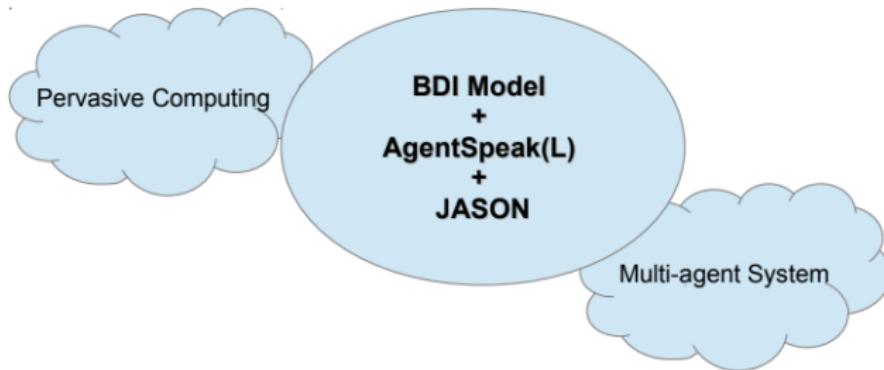
1 Contexto

2 Sistemas de Comportamento Inteligente

3 Automação

4 Considerações

Considerações



[Zamberlan et al. 2014]

Considerações

- Sistemas Pervasivos/Ubíquos são proativos
 - identificar o que o usuário deseja e fornecer ações apropriadas no tempo certo
- Sensibilidade ao contexto
 - sensores
 - atuadores
- Automação + Sistemas de Comportamento Inteligente
 - infraestrutura de comunicação de dados para troca de informações
 - entre dispositivos e entre usuários do ambiente
 - sensibilidade ao contexto - adaptação inteligente

Oportunidades e desafios

- Gerar as conexões e integrar serviços (Protocolos Internet ou Específicos)
- Garantir integridade e segurança

Onde encontrar a apresentação

- <https://github.com/alexandrezamberlan/apresentacoes>

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 Simulaton*, 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.



Santos, A. K. D. and Zamberlan, A. (2021).

Sistemas Pervasivos Integrados Por Agentes Inteligentes Em Jason E Raspberry Pi.

Trabalho de Conclusão de Curso Ciência Da Computação - Universidade Franciscana (UFN), Santa Maria RS, Brasil. Disponível em <https://tfgonline.lapinf.ufn.edu.br>.



Zamberlan, A. (2018).

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

PhD thesis, Universidade Franciscana - UFN, Santa Maria.



Zamberlan, A., Perozzo, R., Kurtz, G., Librelotto, G., and Fagan, S. (2014).

Integrando agentes AgentSpeak(L) em ambientes pervasivos educacionais.

In *WESAAC - Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações*, pages 1–13, Porto Alegre. SBC.

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

Alexandre Zamberlan

alexz@ufn.edu.br

Universidade Franciscana