

Agentes e Sistemas Multiagentes

Daniela Maria Uez

dani.uez@gmail.com

Programa de Pós-Graduação em
Engenharia de Automação e Sistemas

16 de setembro 2014

1 Introdução

2 Agentes

3 Sistemas Multiagentes

4 Considerações Finais

Sumário

- 1 Introdução
- 2 Agentes
- 3 Sistemas Multiagentes
- 4 Considerações Finais

Mundo de hoje...

- Conhecimento e informações mudam rapidamente
- Ubiquidade: computação em diversos lugares
- Interconexão (ex: internet)
- Tarefas complexas sendo delegadas aos computadores
- Orientação a humanos (linguagem natural, interface gráfica, etc)

Necessidades de software

- Independência
- Agir de modo a representar os melhores interesses (de quem delega as funções)
- Interagir com outros seres humanos ou sistemas
- Distribuição
- Interconexão

Como resolver esses problemas?

- Paradigmas atuais tornam a resolução desses problemas muito complicada
- É necessário um novo paradigma que utilize metáforas mais próximas à nossa visão de mundo (humana)

Orientação a agentes

- Se utiliza de conhecimentos de diversas áreas para desenvolver software, como
 - Teoria dos jogos
 - Inteligência artificial distribuída
 - Filosofia
 - Lógica
 - Economia
 - Ciências sociais
 - ...

Paradigmas de programação

| Paradigma | Unidade Estrutura | Linguagem |
|-------------------------|-------------------|-------------------|
| Linguagem de Máquina | - | Assembly |
| Programação Estruturada | Subrotina | C, Pascal |
| Orientação a Objetos | Objeto | C++, Java |
| Orientação a Agentes | Agente | Jason, 2APL, GOAL |

Sumário

- 1 Introdução
- 2 Agentes**
- 3 Sistemas Multiagentes
- 4 Considerações Finais

O que é um agente?

Um agente é...

“...Um sistema computacional que está situado em um ambiente e pode agir autonomamente nesse ambiente para atingir seus objetivos”

Wooldridge [2009].

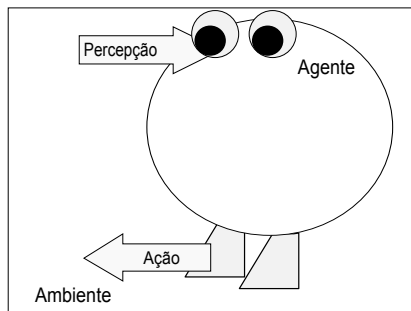
Agente - Características

- Situado em um ambiente
- Capacidade de agir sobre o ambiente
- Possui objetivos
- É autônomo
 - Autonomia é a principal característica de um agente
 - O agente decide qual ação vai executar sem intervenção (humana ou de outros sistemas)

Agentes - Outras Características

- **Reatividade:** age conforme as mudanças percebidas no ambiente
- **Proatividade:** toma a iniciativa de agir para atingir seu objetivo
- **Habilidade social:** interage com outros agentes ou com humanos para satisfazer seus objetivos

O agente e o ambiente



- O agente percebe o ambiente
- Decide o que vai fazer (delibera)
- Age sobre o ambiente, alterando-o
- O ambiente pode ser não-determinístico
- Nem sempre o agente conhece todo o seu ambiente

Ações do agente

- Cada agente pode realizar um conjunto definido de ações
- Ações representam alterações que o agente quer executar no ambiente
- As ações dependem de pré-condições para serem executadas - nem todas as ações podem ser executadas em todas as situações
- O ambiente é não-determinístico: a mesma ação pode gerar resultados diferentes cada vez que for executada.

Agentes X Objetos

Qual a diferença?

Objetos fazem de graça.
Agentes fazem porque querem.

Agentes X Objetos

Qual a diferença?

Objetos fazem de graça.
Agentes fazem porque querem.

- Objetos não têm autonomia. Quando um método é invocado, a ação é executada (independente da vontade do objeto)
- Agentes são autônomos. Executa as ações quando é do seu interesse executar (mesmo que outro agente tenha solicitado.)

Wooldridge [2009]

Tipos de agentes

- Agentes reativos: utiliza regras tipo percepção-ação
(Ex: Se ambiente sujo, então limpa)
- Agentes cognitivos (ou deliberativos): possuem uma capacidade de raciocínio mais complexa. Permitem a representação do conhecimento e planejam sua ação futura com base no conhecimento adquirido
Possuem **estado mental**
(Ex.: Arquitetura BDI)

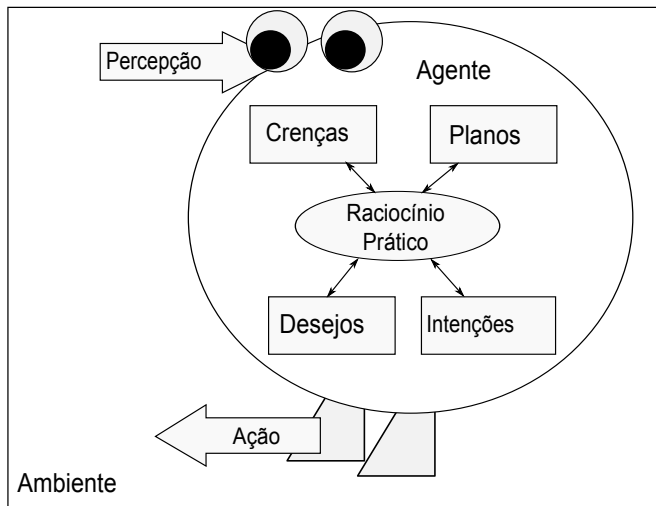
Arquitetura BDI (Belief - Desire - Intention)

- Baseada no modelo prático de raciocínio humano Bratman [1987]
- Define o estado mental do agente com base em suas crenças, desejos e intenções
 - Crenças (Beliefs): aquilo que o agente sabe sobre o mundo
 - Desejos (Desires): estados do mundo que o agente pode atingir
 - Intenções (Intentions): estados do mundo que o agente decidiu trabalhar para assistir

Raciocínio Prático

- É o processo de descobrir o que fazer
- Realizado em duas fases
 - Deliberação: com base nas crenças, nos desejos e nas intenções (já definidas), define o que o agente vai fazer (intenções)
 - Raciocínio meio-fim (means-ends reasoning): é o processo que decide como atingir um "fim" (intenção) usando os "meios" (ações) disponíveis
Define qual dos planos do agente deve ser executado para realizar a intenção

Raciocínio Prático (II)



Sumário

- 1 Introdução
- 2 Agentes
- 3 Sistemas Multiagentes**
- 4 Considerações Finais

Sistemas multiagentes (SMA)

- Maioria dos problemas não pode ser resolvida por um único agente
- Normalmente, diversos agentes dividem o mesmo ambiente
- Cada agente conhece uma parte do ambiente
- Agentes heterogêneos
- Os agentes precisam interagir e coordenar suas ações para atingir seus objetivos
- SMA pode ser definido como um “conjunto organizado de agentes” (organizado = regras de convivência)
Demazeau [2001]

Sistemas Multiagentes - Vogais

Dimensões do SMA

Um sistema multiagentes é formado por um **conjunto de agentes**, um **ambiente**, um **conjunto de possíveis interações** e uma **organização**

Demazeau [1995]

Sistemas Multiagentes - Vogais

Dimensões do SMA

Um sistema multiagentes é formado por um **conjunto de agentes**, um **ambiente**, um **conjunto de possíveis interações** e uma **organização**

Demazeau [1995]

As quatro dimensões:

- Agente (A)
- Ambiente (E)
- Interação (I)
- Organização (O)

Sistemas Multiagentes - Exemplo



Ambiente

- Visão clássica: externo ao SMA

Ambiente

- Visão clássica: externo ao SMA
- Entidade de primeira classe: faz parte do SMA, fornecendo funcionalidades e serviços que podem ser utilizados pelos agentes (modelo Agents & Artifacts [Ricci *et al.* [2007]])
Ex.: calculadora
- Framework CArtAgO (Ricci *et al.* [2011]) permite a implementação de ambientes baseados no modelo A&A.

Interações entre agentes

- Refere-se basicamente à estratégias de comunicação entre agentes
- Comunicação baseada na troca de mensagens
 - Linguagens padrão de comunicação (KQML e Fipa ACL)
 - Baseada na teoria dos atos de fala: “Linguagem é ação”
 - Força ilocutória - tipo de interação (inquire, inform...)
- Protocolos de comunicação
 - Sequencias de mensagens trocadas entre os agentes em forma de “conversa”
 - Auction, Contract Net, etc..

Organização

- Conjunto de regras comportamentais que o agente aceita quando entra no sistema
- Regula o comportamento e a interação entre os agentes para auxiliar o atingimento dos objetivos
- Define papéis que são exercidos pelo agente dentro da organização
- Os papéis encapsulam restrições (normas, requisitos, aptidões) que devem ser seguidas pelo agente

A organização pode ser analisada...

- Do ponto de vista do agente: agente conhece ou não a organização
- Do ponto de vista de um observador externo: implícita ou explícita
- Modelos organizacionais: permitem a representação explícita da organização
 - *Moise* (Hubner *et al.* [2007]; Hübner *et al.* [2010])
 - TOVE
 - TAEMS (Decker [1995])
 - OPERA (Argente *et al.* [2006])

Programação SMA

- Jason - <http://jason.sf.net/>
- GOAL - <http://mmi.tudelft.nl/trac/goal>
- 2APL - <http://apapl.sourceforge.net/>
- JIAC - <http://www.jiac.de/>
- Jadex - <http://www.activecomponents.org/>
- JADE - <http://jade.tilab.com/>
- JACK - <http://www.agent-software.com.au/products/jack/>

Exemplo Jason - Wumpus

```
// crenças iniciais  
pos(1,1).           // pos inicial  
orientation(east). // orientação  
borders(1, 1, 4, 4). // cenário
```

```
// objetivos iniciais  
!main.
```

```
//planos  
+!main  
  <- !find(gold).
```

```
+!find(gold)  
: has_gold.
```

```
+!find(gold)  
  <- !explore;  
    !find(gold).
```


Exemplo Jason - Wumpus (II)

+glitter

```
<- grab;  
+has_gold;  
.succeed_goal(explore).
```

+!explore

```
: borders(BottonX, BottonY, TopX, TopY) &  
.range(X,BottonX,TopX) & .range(Y,BottonY,TopY) &  
not visited(X,Y)  
<- .print("going to ",X,",",Y);  
!goto(X,Y).
```

JaCaMo

- O framework JaCaMo Boissier *et al.* [2013] permite o desenvolvimento integrado de três dimensões do SMA: agente, ambiente e organização (<http://jacamo.sourceforge.net/>)
- Cada dimensão é desenvolvida usando linguagem e conceitos próprios
- Os agentes são programados em Jason
- Os ambiente são programados em CArtaGO (<http://cartago.sourceforge.net/>)
- A organização é desenvolvida em Moise (<http://moise.sourceforge.net/>)
- Eclipse plug-in:
<http://jacamo.sourceforge.net/eclipseplugin/tutorial/>

Engenharia de software orientada a agentes

- Ainda sem a estrutura da engenharia de software OO
- Diversos métodos e linguagens de modelagens desenvolvidos
- Principais métodos
 - Prometheus (Padgham and Winikoff [2004]
<http://www.cs.rmit.edu.au/agents/prometheus/>)
 - O-MaSE (DeLoach and García-Ojeda [2010]
<http://agenttool.cis.ksu.edu/>)
 - Tropos (Cossentino and Seidita [2005]
<http://www.troposproject.org/node/93>)
 - Prometheus AEOLus (Uez *et al.* [2013])
<http://www.uez.com.br/aeolus/metodo.html>

Outras áreas de estudo

- Normas e instituições para SMA
- SMA abertos e agentes móveis
- Confiança e reputação
- Interação ambiente X organização - regras *count-as*
- Ontologias
- Raciocínio - Modelo Huginn
- ...

Sumário

- 1 Introdução
- 2 Agentes
- 3 Sistemas Multiagentes
- 4 Considerações Finais**

Aplicações SMA



- Robo de resgate (iRobot 510 - Usado em Fukushima)
- Exploração espacial (Opportunity - Marte)
- Veículos não tripulados (VANT)
- Robôs humanoides - (Robo NAO)

Aplicações SMA (II)

- Sistemas tutores
- Recuperação e gerenciamento de informações
- Agentes pessoais e para e-commerce
- Controle de tráfego aéreo
- Gerenciamento de produção
- Jogos
- ...

Quando usar?

- Tarefas complexas
- Ambiente não-determinístico
- Necessidade de comportamento “humano” (autonomia, aprendizagem, interação social, etc)

Competições

- Multi-agent Programming contest

<http://multiagentcontest.org/>

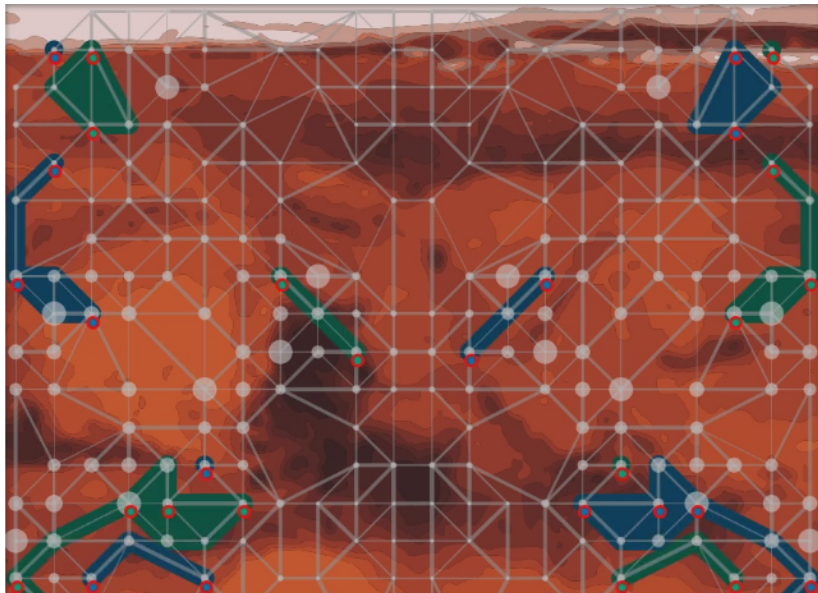
- Time SMADAS venceu em 2012 e 2013
- Competição 2014 - de 15 a 17 de setembro
- Rumo ao tricampeonato!

<https://www.facebook.com/video.php?v=565435513487352>

- RoboCup (Rescue and Soccer)

<http://www.robocup.org/>

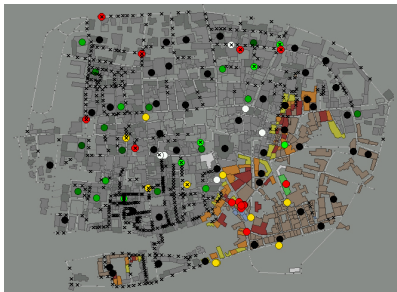
MAPC



RoboCup Soccer



RoboCup Rescue



SMADAS

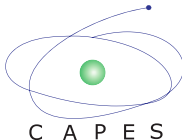
- Grupo de pesquisas em SMA do departamento de Automação e Sistemas
- Hoje conta com 10 membros:
 - 4 doutorandos
 - 3 mestrandos
 - 2 ICs
 - Prof. Jomi Fred Hübner,
- Algumas pesquisas:
 - Método AEOLus para AOSE
 - Modelo Huginn de raciocínio
 - Interação como entidade de primeira classe
 - Regras *Count-as*
 - Vant controlado por sistema autônomo
 - JaCaMo

That's all folks!

Obrigada!
dani.uez@gmail.com



DAS



Bibliografia I

- Estefania Argente, Vicente Julian, and Vicente Botti. Multi-agent system development based on organizations. *ENTCS.*, 150:55–71, 2006.
- Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley, 2007.
- Federico Bergenti and Michael N. Huhns. On the use of agents as components of software systems. In Federico Bergenti, Marie-Pierre Gleizes, and Franco Zambonelli, editors, *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, volume 11 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, chapter 2, pages 273–296. Kluwer Academic, 2004.

Bibliografia II

- Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*, 78(6):747 – 761, 2013.
- Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.
- M. Bratman. *Intention, plans, and practical reason*. Harvard University Press, 1987.
- Lars Braubach, Winfried Lamersdorf, and Alexander Pokahr. *Jadex: Implementing a bdi-infrastructure for jade agents*, 2003.
- Paolo Busetta, Ralph Ronnquist, Andrew Hodgson, and Andrew Lucas. *Jack intelligent agents - components for intelligent agents in java*, 1999.

Bibliografia III

- M. Cossentino and V. Seidita. Tropos: Processo e frammenti, 2005.
- Mehdi Dastani. 2apl: A practical agent programming language. *Autonomous Agents and Multi-Agent Systems*, 16(3):214–248, June 2008.
- K. Decker. Environment centered analysis and design of coordination mechanisms. *Ph.D. Thesis, Department of Computer Science, University of Massachusetts, Amherst*, 1995.
- Scott A. DeLoach and Juan Carlos García-Ojeda. O-mase: a customisable approach to designing and building complex, adaptive multi-agent systems. *Int. J. Agent-Oriented Softw. Eng.*, 4(3):244–280, 2010.
- Yves Demazeau. From interactions to collective behaviour in agent-based systems. In *1st. ECCS*, pages 117–132, 1995.

Bibliografia IV

- Y. Demazeau. *Principes et architecture des systèmes multi-agents*. IC2: Série Informatique et systèmes d'information. Hermes Science Publications, 2001.
- David Harel, Assaf Marron, Guy Wiener, and Gera Weiss. Behavioral programming, decentralized control, and multiple time scales. In *Proceedings of the Compilation of the Co-located Workshops on DSM'11, TMC'11, AGERE!'11, AOOPEs'11, NEAT'11, & VMIL'11, SPLASH '11 Workshops*, pages 171–182, New York, NY, USA, 2011.
- Brian Henderson-Sellers. Agent-oriented methodologies: an introduction. In Paolo Giorgini and Brian Henderson-Sellers, editors, *Agent-Oriented Methodologies*. Idea Group Publishing, 2005.

Bibliografia V

- Jomi F. Hubner, Jaime S. Sichman, and Olivier Boissier. Developing organised multiagent systems using the MOISE+ model: Programming issues at the system and agent levels. *Int. J. Agent-Oriented Softw. Eng.*, 1(3/4):370–395, 2007.
- Jomi F. Hübner, Olivier Boissier, and Rafael H. Bordini. From organisation specification to normative programming in multi-agent organisations. In *Proceedings of the 11th international conference on Computational logic in multi-agent systems*, CLIMA'10, pages 117–134. Springer, 2010.
- Lin Padgham and Michael Winikoff. *Developing Intelligent Agent Systems: A Practical Guide*. Halsted Press, New York, NY, USA, 2004.
- Alessandro Ricci, Mirko Viroli, and Andrea Omicini. The A&A programming model and technology for developing agent environments in MAS. In *PROMAS*, pages 89–106. 2007.

Bibliografia VI

Alessandro Ricci, Michele Piunti, and Mirko Viroli. Environment programming in multi-agent systems: an artifact-based perspective. *AAMAS*, 23:158–192, 2011.

Daniela Maria Uez, Jomi Fred Hübner, and Carine G. Webber. Método para modelagem de agentes, ambiente e organização de sistemas multiagentes. In *Anais IV Workshop Brasileiro sobre Sistemas de Software Autônomos - Autosoft 2013*, pages 41–50, 2013.

Michael J. Wooldridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2 edition, 2009.