

Agent-oriented Programming

p.sernani@univpm.it

<http://airtlab.dii.univpm.it/>

Trend

- ubiquità
- interconnessione
- intelligenza
- delega
- “human-orientation”

Agente

Agente [Wooldridge]

An agent is a computer system that is capable of *independent* action on behalf of its user or owner.

Sistema Multi-Agente [Wooldridge]

A Multi-Agent System is one that consists of a number of agents, which *interact* with one another, typically by exchanging messages through some computer network infrastructure.

Sistemi Multi-Agente: influenze di altri campi

- Intelligenza Artificiale
- Ingegneria del Software
- Scienze Sociali
- Teoria dei giochi ed Economia
- Sistemi Distribuiti

Agent Oriented Programming (AOP)(1)

Modella un programma come un insieme di componenti (*gli agenti*) dotati di autonomia, proattività e abilità comunicative.

AOP e OOP:

- incapsulamento;
- ereditarietà;
- message passing.

Agent Oriented Programming (AOP)(2)

AOP vs OOP:

- oggetti invocati, agenti no;
- ogni agente ha il proprio thread di controllo;
- solo le informazioni passano i “confini” di un agente, non il controllo.

Perché AOP?(1)

Sistemi distribuiti:

- unità di elaborazione ovunque, in qualunque oggetto;
- sempre connesse e attive;
- comunicazione.

Autonomia:

- scenari non adatti a umani (es. task ripetitivi o ad alto rischio);
- scenari con ritardi di comunicazione.

Perché AOP?(2)



Distributed thinking

a complex problem can be divided into several small problems



Robustness

losing one decision node doesn't imply the system failure



Reusability

old components can be re-used to develop new components or new systems

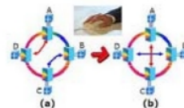
Modularity

building the system by pieces like using LEGO



Reconfigurability

changes can be performed on the fly



Smooth migration

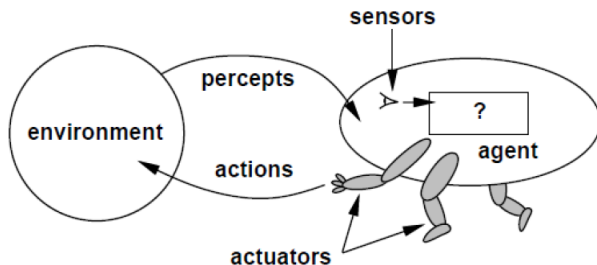
from old technologies to new ones



Agente(1)

[Russell & Norvig]

«An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.»



Agente(2)

[Wooldridge]

«We consider agents to be systems that are situated in some environment. By this, we mean that agents are capable of sensing their environment (via sensors), and have a repertoire of possible actions that they can perform (via effectors or actuators) in order to modify their environment.»

[Maes]

«Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed.»

Agente(3)

[Genesereth & Ketchpel]

«...software “components” that communicate with their peers by exchanging messages in an expressive *agent communication language*. Agents can be as simple as subroutines; but typically they are larger entities with some sort of persistent control.»

Agenti: proprietà (1)

Autonomia

Prendere decisioni senza l'intervento diretto di un essere umano.

Abilità Sociali

Comunicare con gli altri agenti per cercare informazioni.

Reattività

Reagire ad eventi o cambiamenti nell'ambiente circostante.

Proattività

Mostrare comportamenti ed eseguire azioni guidati da *obiettivi*.

Agenti: proprietà (2)

Mobilità

Migrare da un nodo all'altro della rete

Razionalità

Agire *sempre* per perseguire i suoi goal, *mai* per evitare il loro raggiungimento.

Capacità di apprendimento

Adattarsi all'ambiente e ai bisogni degli utenti.

Architettura reattiva

Mapping diretto tra i dati provenienti dai sensori e le azioni. Nessuna rappresentazione simbolica esplicita.

- semplicità di design e tempi di risposta minori
- incapacità di pianificare azioni future

Architettura deliberativa (logic-based)

L'ambiente è rappresentato simbolicamente e manipolato grazie a meccanismi di reasoning

- semplice capire la logica, spiegare le decisioni
- può essere difficile rappresentare accuratamente il mondo e i tempi di risposta potrebbero essere troppo elevati

es.: Architettura BDI (Rao & Georgeff)(1)

Basata su *practical reasoning* [Bratman]:

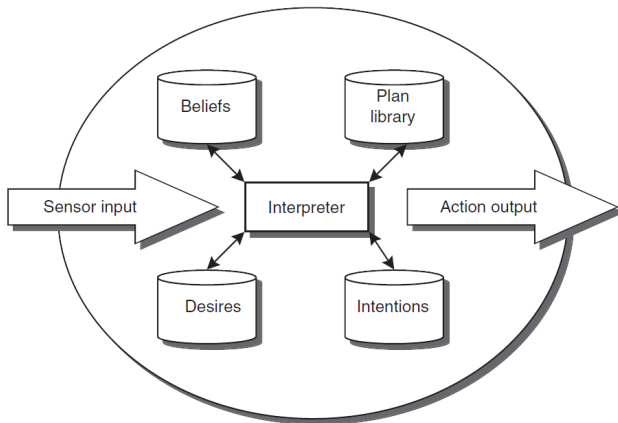
- deliberare l'obiettivo da raggiungere;
- stabilire la sequenza di azioni per raggiungerlo.

Definisce 3 attitudini mentali:

- **belief**, informazioni che l'agente ha sul proprio ambiente;
- **desire**, obiettivi che l'agente dovrebbe perseguire;
- **intention**, obiettivi che l'agente si è impegnato a perseguire.

es.: Architettura BDI (Rao & Georgeff)(2)

Procedural Reasoning System (PRS):



Agente Intelligente

Pu un “agente intelligente” esibire comportamenti che non siano preprogrammati, ma **appresi** dalle “Percezioni”?

Deve un “agente intelligente” esibire comportamenti che non siano preprogrammati, ma **appresi** dalle “Percezioni”?

Apprendimento e autonomia

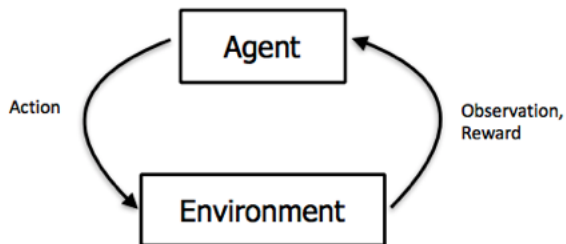
- Se non abbiamo a disposizione un dataset annotato del tipo (ingresso, uscita desiderata)?
- Se, anzich imitare la conoscenza di dominio di un esperto, volessimo trovare la strategia migliore?
- Se l'agente deve restare autonomo?

Apprendimento basato su rinforzo

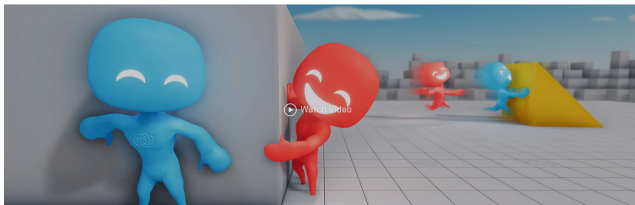
L'agente apprende attraverso un meccanismo basato su un “reward”

- l'agente calcola autonomamente l'output;
- l'obiettivo trovare una sequenza di azioni che massimizzi il reward dell'agente

Reinforcement Learning



Reinforcement Learning

[ABOUT](#) [PROGRESS](#) [RESOURCES](#) [BLOG](#)

SEPTEMBER 17, 2019 • 9 MINUTE READ

Emergent Tool Use from Multi-Agent Interaction

We've observed agents discovering progressively more complex tool use while playing a simple game of hide-and-seek. Through training in our new simulated hide-and-seek environment, agents build a series of six distinct strategies and counterstrategies, some of which we did not know our environment supported. The self-supervised emergent complexity in this simple environment further suggests that multi-agent co-adaptation may

<https://openai.com/blog/emergent-tool-use/>

MAS

Comunità di agenti: Sistema Multi-Agente (MAS)

- Risolvere problemi *complessi* mediante interazioni tra agenti;
- Cooperazione o competizione
- Ogni agente ha una visione parziale del *mondo* e integra la sua conoscenza comunicando con gli altri agenti



FIPA-IEEE

Foundation for Intelligent, Physical Agents (1996)

Definisce standard per:

- la gestione di piattaforme per l'esecuzione degli agenti;
- la comunicazione tra agenti.



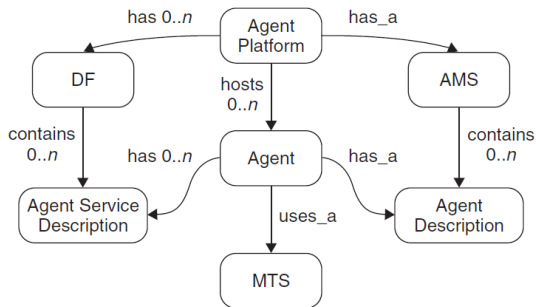
Agent Management System (AMS)

- Gestisce le operazioni all'interno dell'AP: creazione, cancellazione e migrazione di agenti;
- mantiene una directory con tutti gli AID degli agenti nell'Agent Platform (servizio di pagine bianche);
- può richiedere specifiche operazioni di gestione sugli agenti e forzarle (es. la loro terminazione);
- componente obbligatorio di un'Agent Platform.

Directory Facilitator (DF)

- Permette agli agenti di pubblicare i loro servizi (servizio di pagine gialle);
- mantiene una lista accurata, completa e aggiornata degli agenti che richiedono la registrazione dei propri servizi;
- componente opzionale di un'Agent Platform.

FIPA Agent Management



Modello logico di riferimento per la gestione di creazione, registrazione, posizione, comunicazione, migrazione di agenti.

Agent Communication

FIPA-Agent Communication Language (ACL)

Basato sulla *Speech Act Theory* [Searle]: i messaggi sono azioni - *atti comunicativi* - che permettono di stabilire una conversazione tra agenti.

Performative (1)

FIPA communicative act	Description
Accept Proposal	The action of accepting a previously submitted proposal to perform an action
Agree	The action of agreeing to perform some action, possibly in the future
Cancel	The action of one agent informing another agent that the first agent no longer has the intention that the second agent performs some action
Call for Proposal	The action of calling for proposals to perform a given action
Confirm	The sender informs the receiver that a given proposition is true, where the receiver is known to be uncertain about the proposition
Disconfirm	The sender informs the receiver that a given proposition is false, where the receiver is known to believe, or believe it likely that, the proposition is true
Failure	The action of telling another agent that an action was attempted but the attempt failed
Inform	The sender informs the receiver that a given proposition is true

Performative (2)

Inform If	A macro action for the agent of the action to inform the recipient whether or not a proposition is true
Inform Ref	A macro action allowing the sender to inform the receiver of some object believed by the sender to correspond to a specific descriptor, for example a name
Not Understood	The sender of the act (for example, <i>i</i>) informs the receiver (for example, <i>j</i>) that it perceived that <i>j</i> performed some action, but that <i>i</i> did not understand what <i>j</i> just did. A particular common case is that <i>i</i> tells <i>j</i> that <i>i</i> did not understand the message that <i>j</i> has just sent to <i>i</i>
Propagate	The sender intends that the receiver treat the embedded message as sent directly to the receiver, and wants the receiver to identify the agents denoted by the given descriptor and send the received <i>propagate</i> message to them
Propose	The action of submitting a proposal to perform a certain action, given certain preconditions
Proxy	The sender wants the receiver to select target agents denoted by a given description and to send an embedded message to them

Performative (3)

Query If	The action of asking another agent whether or not a given proposition is true
Query Ref	The action of asking another agent for the object referred to by a referential expression
Refuse	The action of refusing to perform a given action, and explaining the reason for the refusal
Reject Proposal	The action of rejecting a proposal to perform some action during a negotiation
Request	The sender requests the receiver to perform some action One important class of uses of the request act is to request the receiver to perform another communicative act
Request When	The sender wants the receiver to perform some action when some given proposition becomes true
Request Whenever	The sender wants the receiver to perform some action as soon as some proposition becomes true and thereafter each time the proposition becomes true again
Subscribe	The act of requesting a persistent intention to notify the sender of the value of a reference, and to notify again whenever the object identified by the reference changes

Parametri Messaggi ACL

Parameter	Description
Performative	Type of the communicative act of the message
sender	Identity of the sender of the message
receiver	Identity of the intended recipients of the message
reply-to	Which agent to direct subsequent messages to within a conversation thread
content	Content of the message
language	Language in which the content parameter is expressed
encoding	Specific encoding of the message content
ontology	Reference to an ontology to give meaning to symbols in the message content
protocol	Interaction protocol used to structure a conversation
conversation-id	Unique identity of a conversation thread
reply-with	An expression to be used by a responding agent to identify the message
in-reply-to	Reference to an earlier action to which the message is a reply
reply-by	A time/date indicating by when a reply should be received

L'unico obbligatorio è *performative*.

Messaggio FIPA-ACL - Esempio

```
(request
  :sender (agent-identifier :name alice@mydomain.com)
  :receiver (agent-identifier :name bob@yourdomain.com)
  :ontology travel-assistant
  :language FIPA-SL
  :protocol fipa-request
  :content
    ""( (action
      (agent-identifier :name bob@yourdomain.com)
      (book-hotel :arrival 15/10/2006
                  :departure 05/07/2002 ... )
    ) ) ""
)
```

Java Agent DEvelopment framework

Piattaforma software che fornisce un middleware per la realizzazione di applicazioni basate sul paradigma ad agenti.



- <http://jade.tilab.com/>

Autonomia e Proattività

Un agente incapsula il proprio flusso di controllo:

- non viene *invocato* (non fornisce il proprio riferimento);
- può decidere quali azioni compiere in quale momento;
- può dire **no**.

Comunicazione

Gli agenti sono entità lascamente accoppiate:

- comunicazione attraverso messaggi asincroni;
- nessuna dipendenza temporale tra mittente e destinatario;
- il ricevente può scegliere quali messaggi processare e quali scartare;
- il mittente non deve necessariamente rimanere bloccato in attesa della risposta.

Agenzia peer-to-peer.

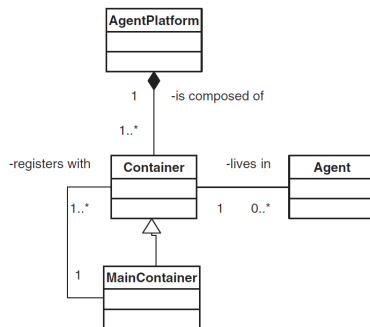
JADE Features

- completamente distribuito
- conforme agli standard FIPA
- comunicazione tramite messaggi asincroni
- implementazione di pagine bianche e gialle
- agent life-cycle management
- supporto alla *migrazione* di agenti
- tool grafici
- ...

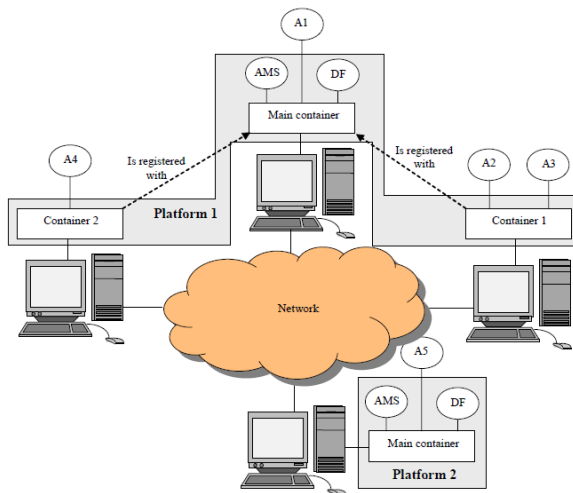
Jade: architettura(1)

Container

Processi Java che forniscono i servizi per ospitare ed eseguire gli agenti.



Jade: architettura(2)



Jade: packages

- jade.core
 - ▶ jade.core.Agent
 - ▶ jade.core.behaviours
- jade.lang.acl
- jade.proto
- jade.tools
- jade.wrappers

Riferimenti

- M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley, 2009.
- F. Belfrage, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE*, Wiley, 2007.
- S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009.
- A. S. Rao and M. P. Georgeff, "BDI Agents: From Theory to Practice," in Proceedings of the 1st Conference on Multi-Agent Systems, 1995, vol. 92, pp. 312219.
- M. R. Genesereth and S. P. Ketchpel, "Software agents," Commun. ACM, vol. 37, no. 7, pp. 4859, 1994
- P. Maes, "Artificial life meets entertainment: Lifelike autonomous agents," Communications of the ACM, vol. 38, no. 11, pp. 108114, 1995.

Risorse

- <http://jade.tilab.com/>
- Jade Programming for Beginners
- Jade Programmer's Guide
- API
- F. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE*, Wiley, 2007.

Repository Git

<https://gitlab.com/psernani/jadesotrdexamples.git>