

Introduction to the Course

Agent-Oriented Software Engineering

A.A. 2019-2020

Prof. Paolo Giorgini



UNIVERSITY OF TRENTO - Italy

Department of Information
and Communication Technology

Where we are ...

- The success of the Internet has changed the way we think about computing (>'90)
- No longer is computing just about numerical calculation, or information processing, it is now about **interaction** and **coordination** between **distinct piece of software**
- More and more **independent software** able to coordinate with others (including humans)
 - The final objective is obtained as an emergent behaviour of multi-software behaviours
 - Open environment (new piece of software can be added or removed)
 - Multi-objective problem

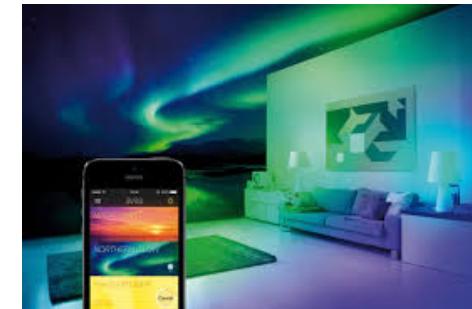
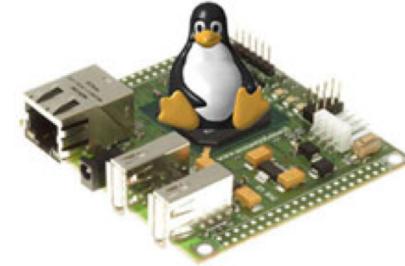


Amazon warehouse (Video)



Ubiquity

- The continual reduction in cost of computing capability has made it possible to introduce **processing power** into places and devices that would have once been uneconomic
- As processing capability spreads, sophistication (and intelligence of a sort) becomes **ubiquitous**
- What could benefit from having a processor embedded in it...?



Internet of Things

- The **Internet of Things** refers to uniquely identifiable objects (things) and their virtual representations in an Internet-like structure
 - If all objects of daily life were equipped with radio tags, they could be identified and inventoried by computers
- Ambient Intelligence (AI) and Autonomous Control (AC) are not part of the original concept of the Internet of Things
 - How to use the Internet of Things for AI and AC?

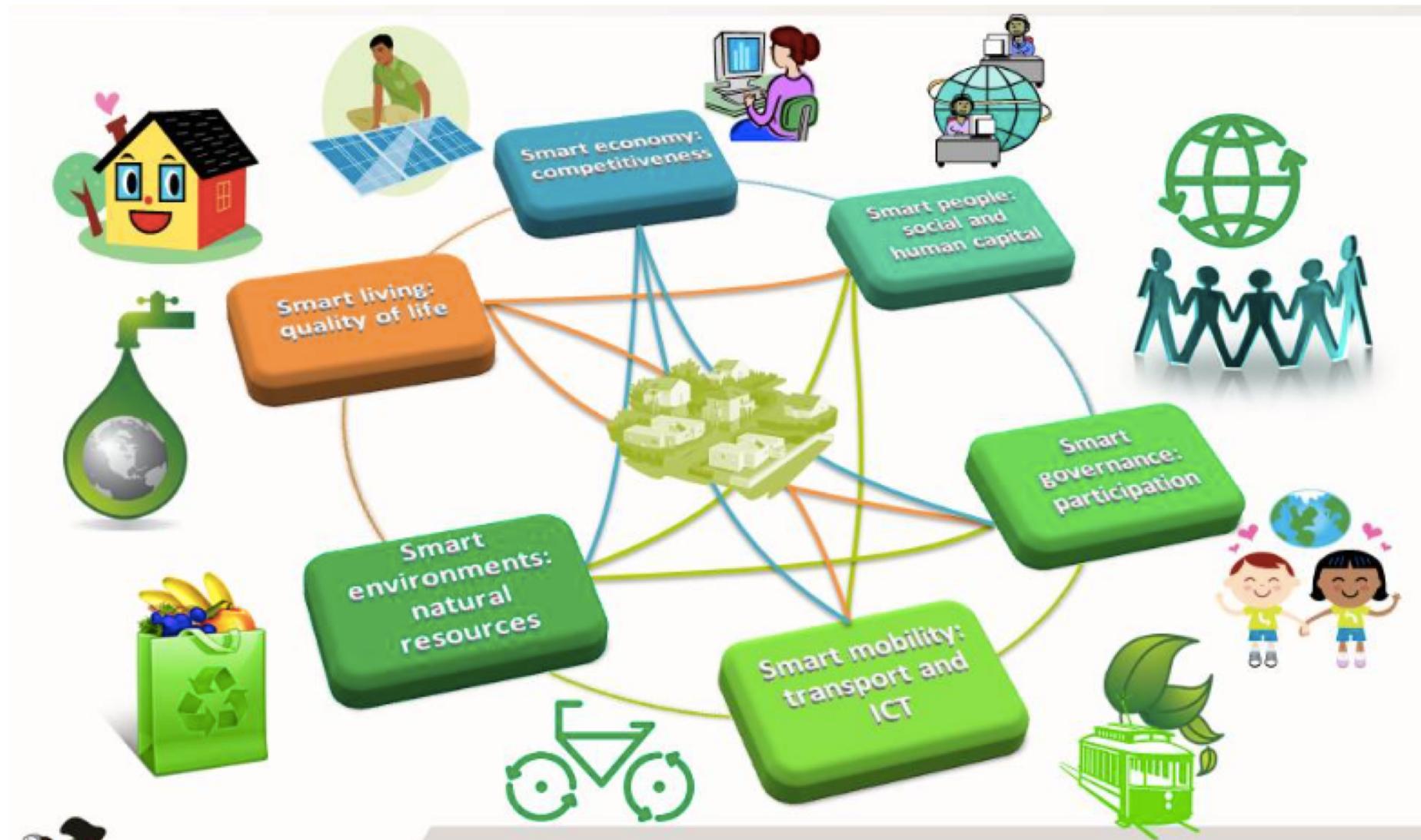


Interconnection

- Computer systems today no longer stand alone, but are networked into **large distributed systems**
 - The internet is an obvious example, but networking is spreading its ever-growing tentacles...
- Since distributed and concurrent systems have become the norm, some researchers are putting forward theoretical models that portray computing as primarily a **process of interaction**



Interconnection: smart city



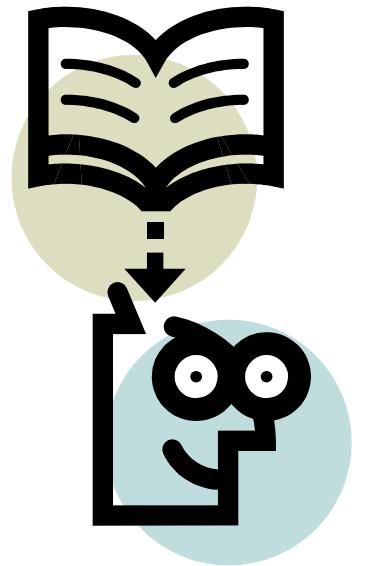
Services and Cloud

- Services are made available to be used and combined together
 - Data storage, software as service, platform as service, infrastructure as service
 - Interaction between different systems is the key issue
 - Quality and security control problems
 - Services availability
 - Run-time composition
 - Flexibility and scalability issues



Interconnection and distribution

- Interconnection and distribution have become core motifs in Computer Science
- But Interconnection and distribution, coupled with the need for systems to represent our best interests, implies systems that can **cooperate** and **reach agreements** (or even **compete**) with other systems that have different interests (much as we do with other people)



Human Orientation

- Movement away from machine-oriented views of programming toward concepts and metaphors that more closely reflect the way we ourselves understand the world
- Programmers (and users!) relate to the machine differently
- Programmers conceptualize and implement software in terms of **higher-level** – more human-oriented – **abstractions**
 - Abstractions to cope with complexity



Intelligence and delegation

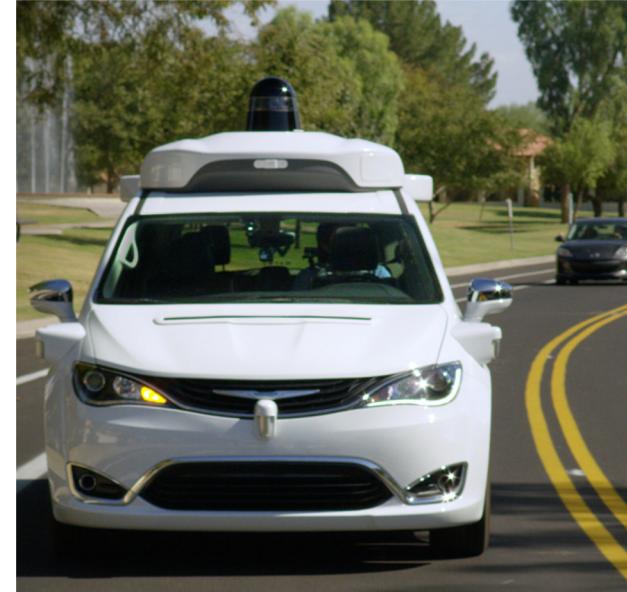
- The complexity of tasks that we are capable of **automating** and **delegating** to computers has grown steadily
- Computers are doing more for us – without our intervention
- We are **giving control** to computers, even in safety critical tasks

Examples:

fly-by-wire aircraft, where the machine's judgment may be trusted more than an experienced pilot

fly-by-wire cars, intelligent braking systems, cruise control that maintains distance from car in front...

Google's Driverless Car



How safe driver is your average robot? Safer than your average American, at least by one measure.

- Since 2009, more than 5 million miles, mostly on city streets. That would take the average American driver nearly 300 years to complete.

Where does it bring us?

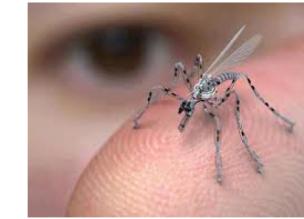
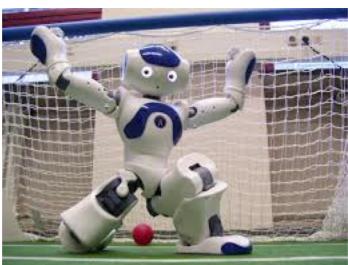


- Delegation and Intelligence imply the need to build computer systems that can act effectively on our behalf
- This implies:
 - The ability of computer systems to act **independently**
 - The ability of computer systems to act in a way that *represents our best interests* while **interacting** with other humans or systems

Examples



Robocup



Swarm intelligence



Distributed Sensing

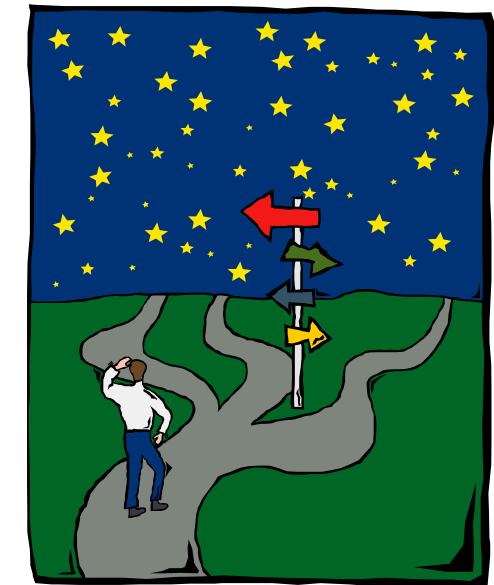


So Computer Science expands...

All of these trends have led to the emergence of a new field in Computer Science: **Multi-Agent Systems**

Agent systems provide us with the means to design and implement interactive computing, whether between machines, or people, or both

[Michael Luck]



Programming progression...



Programming has progressed through:

- machine code;
- assembly language;
- machine-independent programming languages;
- sub-routines;
- procedures & functions;
- abstract data types;
- objects;
- services;

to *agents*

Agent, a definition

- An agent is a computer system that is capable of **independent** actions on behalf of its user or owner (figuring out what needs to be done to satisfy design objectives, rather than constantly being told)
[Wooldridge&Jennigs]



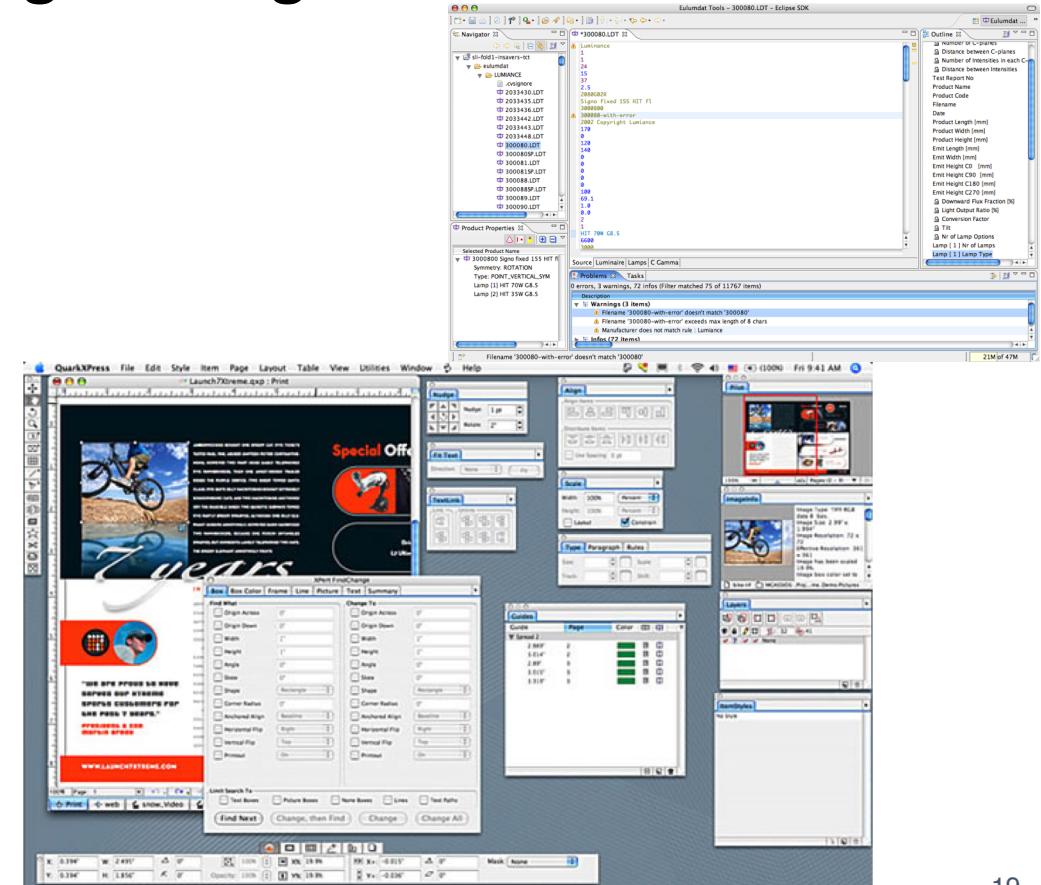
Multi-Agent System, a definition

- A multi-agent system is one that consists of a number of agents, which **interact** with one-another
- In the most general case, agents will be **acting on behalf of users** with different **goals** and motivations
- To successfully interact, they will require the ability to **cooperate**, **coordinate**, and **negotiate** with each other, much as people do



How can we develop such complex software systems?

- A good (Agent-Oriented) Software Engineering Methodology
 - Modeling languages
 - Analysis techniques
 - Design Techniques
 - Supporting Tools



Objective of the Course

- The objective of the course is to examine and explore the credentials of **agent-based approaches** as a **software engineering paradigm**, and to gain an insight into what agent-oriented software engineering will look like
 - Agents as building blocks of a software system
 - Agent as basic concept to develop a software system
 - Knowledge level Software Engineering
 - Agent concept and its mentalist notions (e.g., goal, plan, and belief) are used along all the software engineering process
 - Agent as main concept to
 - model and analyse and organizational setting
 - design a system
 - implement a system ...

Structure of the course

- Theory
- Laboratory
 - Class or laboratory?

Theory (preliminary)



Introduction

- Introduction to the agent paradigm and multi-agent systems
- Applications and problems

Interaction and cooperation

- Types of interaction
- Forms of Cooperation
- Methods of cooperation
- Organization and cooperation

Planning

Communication

- Aspects of communication
- Speech acts
- Conversation
- KQML/ACL

Collaboration and distributed tasks

- Models of tasks allocation
- Centralized allocation
- Decentralized allocation
- Emergent allocation

The BDI Architecture

Agent-Oriented Methodologies

- Gaia
- Tropos
- PASSI

Agent-oriented development tools

- JADE
- JACK

Serious Games and gamification

Lab (preliminary)

Hands-on with exercises

- Planning
- Jack (BDI) as core development framework



In collaboration with



Exam (two modalities)

- Written Exam
 - Questions about the whole program
- Project
 - Partially developed during the laboratory



Books and Material



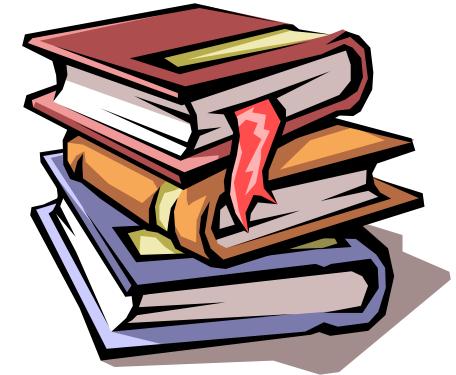
- Given the fact we will try and provide a broad coverage of various topics seems unrealistic an approach in which all the textbooks are adopted.

Therefore:

- We encourage the use of slides, papers, (on-line) material, etc.
- We provide a list of textbooks of which only parts are needed (no need to get all of them, though!)

Books...

- *Jacques Ferber*. [Multi-Agent System: An Introduction to Distributed Artificial Intelligence](#), Addison Wesley Longman, 1999
- *Michael Wooldridge*. [An Introduction to Multiagent Systems](#). John Wiley & Sons, 2002
- *Paolo Giorgini and Brian Henderson-Sellers*. [Agent-oriented Methodologies](#), Idea Group Inc, 2004.
- *Fabio Bellifemine, Giovanni Caire, Dominic Greenwood*. [Developing Multi-Agent Systems with JADE](#). John Wiley & Son Ltd, 2007
- *G. Weiss*. [Multiagent syestems](#), (second edition). MIT Press, 2013.



... and material

- JACK: <http://aosgrp.com/products/jack/>
- Tropos: <http://www.troposproject.org>
- FIPA: <http://www.fipa.org>
- detailed list provided at the end of lessons and...
- ... collected on esse3



...next

Monday (aula A223)

9.30 -11.30 → 9.30-11.00

Tuesday (aula A217)

14.30 -16.30 → 14.30-16.00



Overlaps with other courses?

(WE TRY TO RE-SCHEDULE THE TUESDAY CLASS ON THURSDAY (9.30) TO AVOID THE CONFLICT WITH THE COURSE ON SECURITY)

Contacts

Me

Email: pao.lo.giorgini@unitn.it

Web: <http://www.disi.unitn.it/~pgiorgio>

Skype: [pao.lo.giorgini](#)

Francesco Alzetta

Email: Francesco.alzetta@unitn.it

