Introduction to Multi-Agent Systems

Michal Pechoucek & Branislav Bošanský

AE4M36MAS Autumn 2016 - Lect. 1

General Information

Lecturers: Prof. Michal Pěchouček and Dr. Branislav Bošanský

Tutorials: Branislav Bošanský and Karel Horak

14 lectures and 14 tutorials

Course web page:
https://cw.fel.cvut.cz/wiki/courses/be4m36mas/start

Recommended reading:

- J. M. Vidal: Multiagent Systems: with NetLogo Examples (on-line)
- Y. Shoham and K. Leyton-Brown: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations (on-line)
- Russel and Norvig: Artificial Intelligence: Modern Approach

Outline of Lecture 1

1. Motivational Introduction
2. Defining Agency
3. Specifying Agents
4. Agent Architecturess

Introduction to Multiagent Systems

Motivational Introduction

Autonomous Agents and Multiagent Systems

Multiagent system is a collection of multiple autonomous agents, each acting towards its objectives while all interacting in a shared environment, being able to communicate and possibly coordinate their actions.

Autonomous agent ~ intelligent agent (see later).

Why Intelligent Agents?

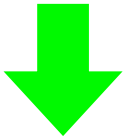
1992: **computers everywhere**

- lots of computerised data
- computer driven manufacturing, production planning, diagnostics

Why Intelligent Agents?

1992: **computers everywhere**

- lots of computerised data
- computer driven manufacturing, production planning, diagnostics



- AI: expert systems, automated planning, machine learning

Why Intelligent Agents?

1992: computers everywhere

Y2K: **internet everywhere**

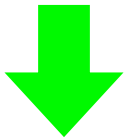
- data provisioning via internet, search (Google from 1998, in 2001 3B of documents)
- an explosion of internet shopping (Amazon from 1995, Ebay from 1996)

Why Intelligent Agents?

1992: computers everywhere

Y2K: **internet everywhere**

- data provisioning via internet, search (Google from 1998, in 2001 3B of documents)
- an explosion of internet shopping (Amazon from 1995, Ebay from 1996)



- parallel computing (map-reduce)
- statistical data analysis and machine learning
- networking, servers

Why Intelligent Agents?

1992: computers everywhere

Y2K: internet everywhere

NOW: **internet of everything**

- mobile computing
- cloud computing
- wireless enabled devices

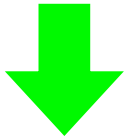
Why Intelligent Agents?

1992: computers everywhere

Y2K: internet everywhere

NOW: **internet of everything**

- mobile computing
- cloud computing
- wireless enabled devices



- Intelligent Agents and Multiagent systems

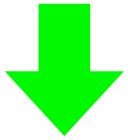
Why Intelligent Agents?

1992: computers everywhere

Y2K: internet everywhere

NOW: **internet of everything**

- mobile computing
- cloud computing
- wireless enabled devices



- Intelligent Agents and Multiagent

Latest trends in computing

Ubiquity: Cost of processing power decreases dramatically (e.g. Moore's Law), computers used everywhere

Interconnection: Formerly only user-computer interaction, nowadays distributed/networked machine-to-machine interactions (e.g. Web APIs)

Complexity: Elaboration of tasks carried out by computers has grown

Delegation: Giving control to computers even in safety-critical tasks (e.g. aircraft or nuclear plant control)

Human-orientation: Increasing use of metaphors that better reflect human intuition from everyday life (e.g. GUIs, speech recognition, object orientation)

Agents briefly

multi-agent system is a decentralized multi-actor (software) system, often geographically distributed whose behavior is defined and implemented by means of complex, peer-to-peer interaction among autonomous, rational and deliberative entities.

autonomous agent is a special kind of a intelligent software program that is capable of highly autonomous rational action, aimed at achieving the private objective of the agent – can exists on its own but often is a component of a multi-agent system – agent is autonomous, reactive, proactive and social

agent researchers study problems of integration, communication, reasoning and knowledge representation, competition (games) and cooperation (robotics), agent oriented software engineering, ...

agent technology is software technology supporting the development of the autonomous agents and multi-agent systems **agent-based computing** is a special research domain, subfield of computer science and artificial intelligence that studies the concepts of autonomous agents

Key properties of Intelligent Agent

Autonomy: Agent is fully accountable for its given state. Agent accepts requests from other agents or the environment but decides individually about its actions

Reactivity: Agent is capable of near-real-time decision with respect to changes in the environment or events in its social neighbourhood

Intentionality: Agent maintain long term intention. the agent meets the designer's objectives. It knows its purpose and executes even if not requested.

Rationality: Agent is capable of intelligent rational decision making. Agent can analyze future course of actions and choose an action which maximizes his utility

Social capability: Agent is aware of the either:

(i) existence,

Agents vs. Objects

agent's behaviour is unpredictable as observed from the outside, agent is *situated* in the environment, communication model is *asynchronous*, agent is autonomous, ...

Agents vs. Objects

agent's behaviour is unpredictable as observed from the outside, agent is *situated* in the environment, communication model is *asynchronous,* agent is autonomous, ...

agents are programs, they are build out of objects
$\rightarrow$ while objects often consist of objects, and object make together an object, agents never contain other agents, agents build together a **multiagent system**

Multiagent Systems Engineering &
Agent Oriented Software Engineering

Novel paradigm for building robust, scalable and extensible control, planning and decision-making systems

- *socially-inspired computing*
- *self-organized teamwork systems*
- *distributed (collective) artificial intelligence*

MAS become increasingly relevant as the connectivity, intelligence and autonomy of devices grows!

Software engineering methodology for designing MAS

Multiagent Systems Engineering &
Agent Oriented Software Engineering

Novel paradigm for building robust, scalable and extensible control, planning and decision-making systems

- *socially-inspired computing*
- *self-organized teamwork systems*
- *distributed (collective) artificial intelligence*

MAS become increasingly relevant as the connectivity, intelligence and autonomy of devices grows!

Software engineering methodology for designing MAS

Multiagent Design Problem

Traditional design problem: *How can I build a system that produces the correct output given some input?*

- Each system is more or less isolated, built from scratch

Multiagent Design Problem

Traditional design problem: *How can I build a system that produces the correct output given some input?*

- Each system is more or less isolated, built from scratch

Multiagent design problem: *How can I build a system that can operate independently on my behalf in a networked, distributed, large-scale environment in which it will need to interact with different other components pertaining to other users?*

- Each system is built into an existing, persistent but constantly evolving *computing ecosystem* – it should be robust with respect to changes
- No single owner and/or central authority

Types of Agent Systems

single-agent

multi-agent

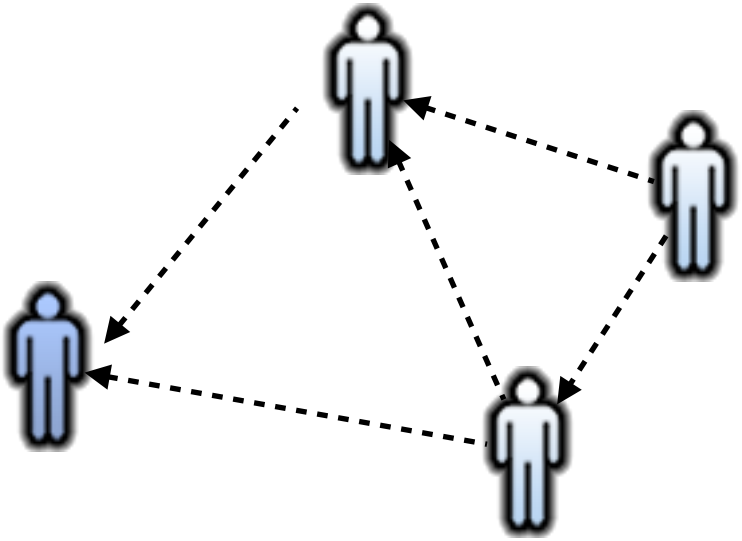
cooperative

single shared utility

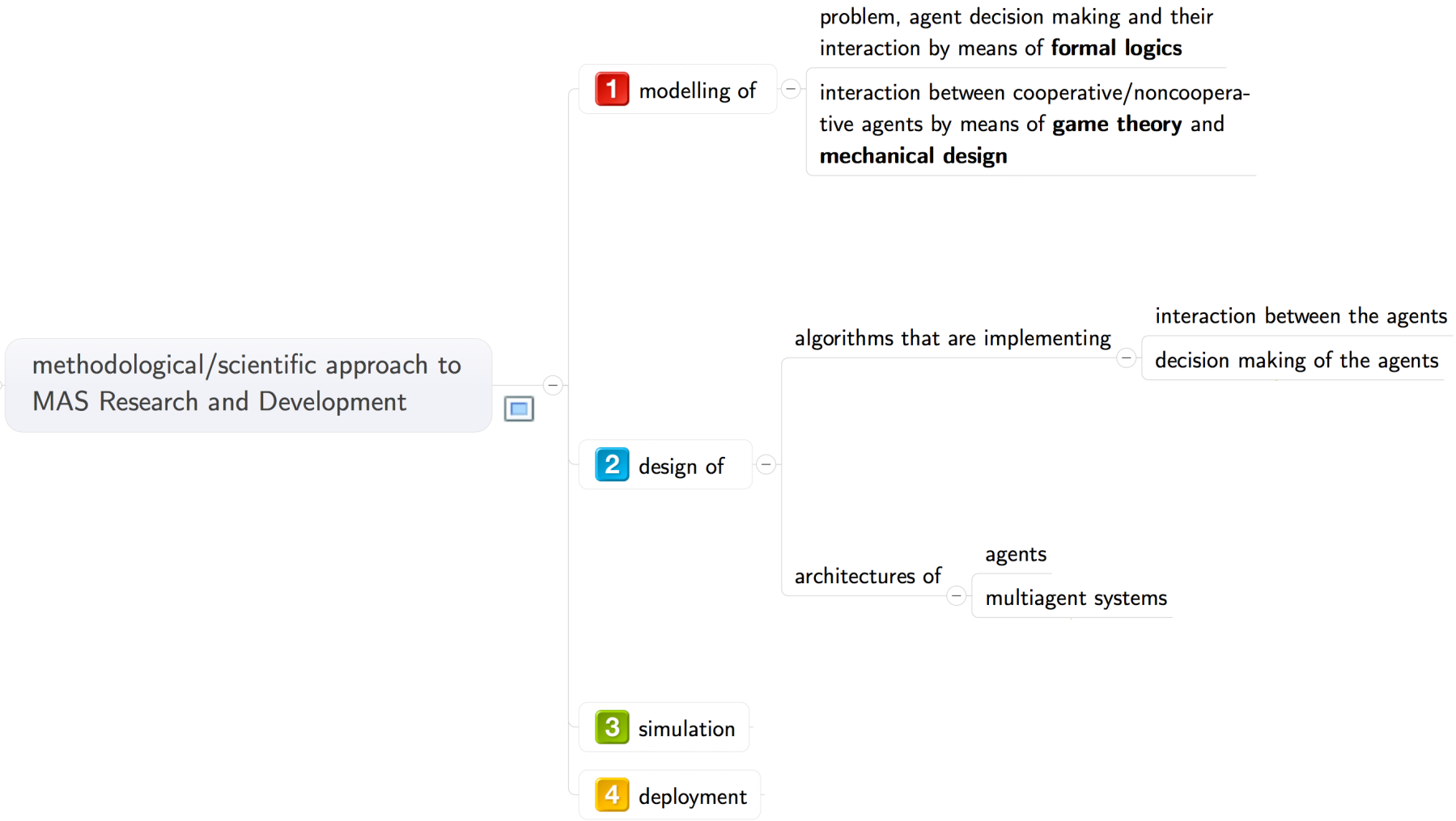
competitive

multiple different utilities

Micro vs. Macro MAS Engineering

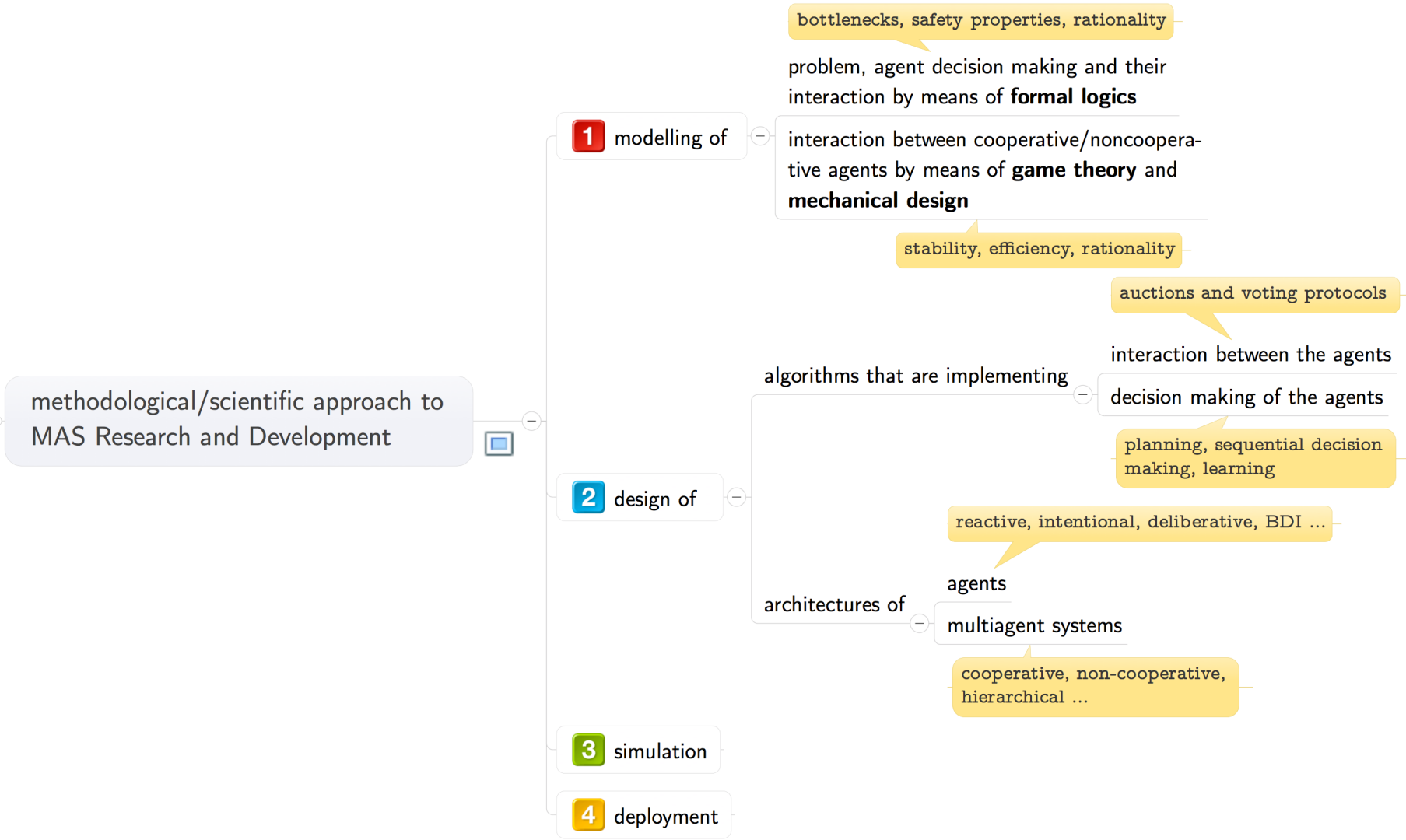


1. The agent design problem (micro perspective):
   How should agents act to carry out their tasks?
2. The society design problem (macro perspective):
   How should agents interact to carry out their tasks?


methodological/scientific approach to MAS Research and Development
1 modelling of
problem, agent decision making and their interaction by means of **formal logics**
interaction between cooperative/noncooperative agents by means of **game theory** and **mechanical design**
2 design of
algorithms that are implementing
interaction between the agents
decision making of the agents
architectures of
agents
multiagent systems
3 simulation
4 deployment


methodological/scientific approach to MAS Research and Development
1 modelling of
problem, agent decision making and their interaction by means of formal logics
bottlenecks, safety properties, rationality
interaction between cooperative/noncooperative agents by means of game theory and mechanical design
stability, efficiency, rationality
2 design of
algorithms that are implementing
interaction between the agents
decision making of the agents
architectures of
agents
multiagent systems
3 simulation
4 deployment


methodological/scientific approach to MAS Research and Development
1 modelling of
problem, agent decision making and their interaction by means of formal logics
bottlenecks, safety properties, rationality
interaction between cooperative/noncooperative agents by means of game theory and mechanical design
stability, efficiency, rationality
2 design of
algorithms that are implementing
interaction between the agents
decision making of the agents
planning, sequential decision making, learning
auctions and voting protocols
architectures of
agents
multiagent systems
3 simulation
4 deployment


methodological/scientific approach to MAS Research and Development
1 modelling of
problem, agent decision making and their interaction by means of formal logics
bottlenecks, safety properties, rationality
interaction between cooperative/noncooperative agents by means of game theory and mechanical design
stability, efficiency, rationality
auctions and voting protocols
2 design of
algorithms that are implementing
interaction between the agents
decision making of the agents
planning, sequential decision making, learning
architectures of
agents
reactive, intentional, deliberative, BDI ...
multiagent systems
cooperative, non-cooperative, hierarchical ...
3 simulation
4 deployment

Opportunities for MAS Deployment

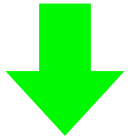
Agent-based computing have been used:

1. **Design paradigm** – the concept of decentralized, interacting, socially aware, autonomous entities as underlying software paradigm (often deployed only in parts, where it suits the application)
2. **Source of technologies** – algorithms, models, techniques architectures, protocols but also software packages that facilitate development of multi-agent systems
3. **Simulation concept** – a specialized software technology that allows simulation of natural multi-agent systems, based on (1) and (2).

Opportunities for MAS Deployment

Agent-based computing have been used:

1. **Design paradigm** – the concept of decentralized, interacting, socially aware, autonomous entities as underlying software paradigm (often deployed only in parts, where it suits the application)

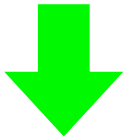


Agent Oriented Software Engineering – provide designers and developers with a way of structuring an application around autonomous, communicative elements, and lead to the construction of software tools and infrastructures to support this metaphor

Opportunities for MAS Deployment

Agent-based computing have been used:

1. **Design paradigm** – the concept of decentralized, interacting, socially aware, autonomous entities as underlying software paradigm (often deployed only in parts, where it suits the application)
2. **Source of technologies** – algorithms, models, techniques architectures, protocols but also software packages that facilitate development of multi-agent systems

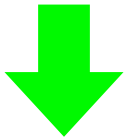


Multi-Agent Techniques – provide a selection of specific computational techniques and algorithms for dealing with collective of computational processes and complexity of interactions in dynamic and open environments.

Opportunities for MAS Deployment

Agent-based computing have been used:

1. **Design paradigm** – the concept of decentralized, interacting, socially aware, autonomous entities as underlying software paradigm (often deployed only in parts, where it suits the application)
2. **Source of technologies** – algorithms, models, techniques architectures, protocols but also software packages that facilitate development of multi-agent systems
3. **Simulation concept** – a specialized software technology that allows simulation of natural multi-agent systems, based on (1) and (2).



Multi-Agent Simulation – provide expressive models for representing complex and dynamic real-world environments, with the emphasis on capturing the interaction related properties of such systems

Intelligent Agents Applications

Manufacturing and production

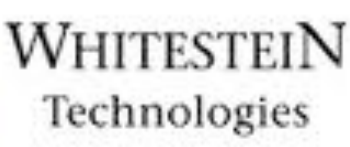
Traffic and logistics

Robotics, autonomous systems

Air traffic and space

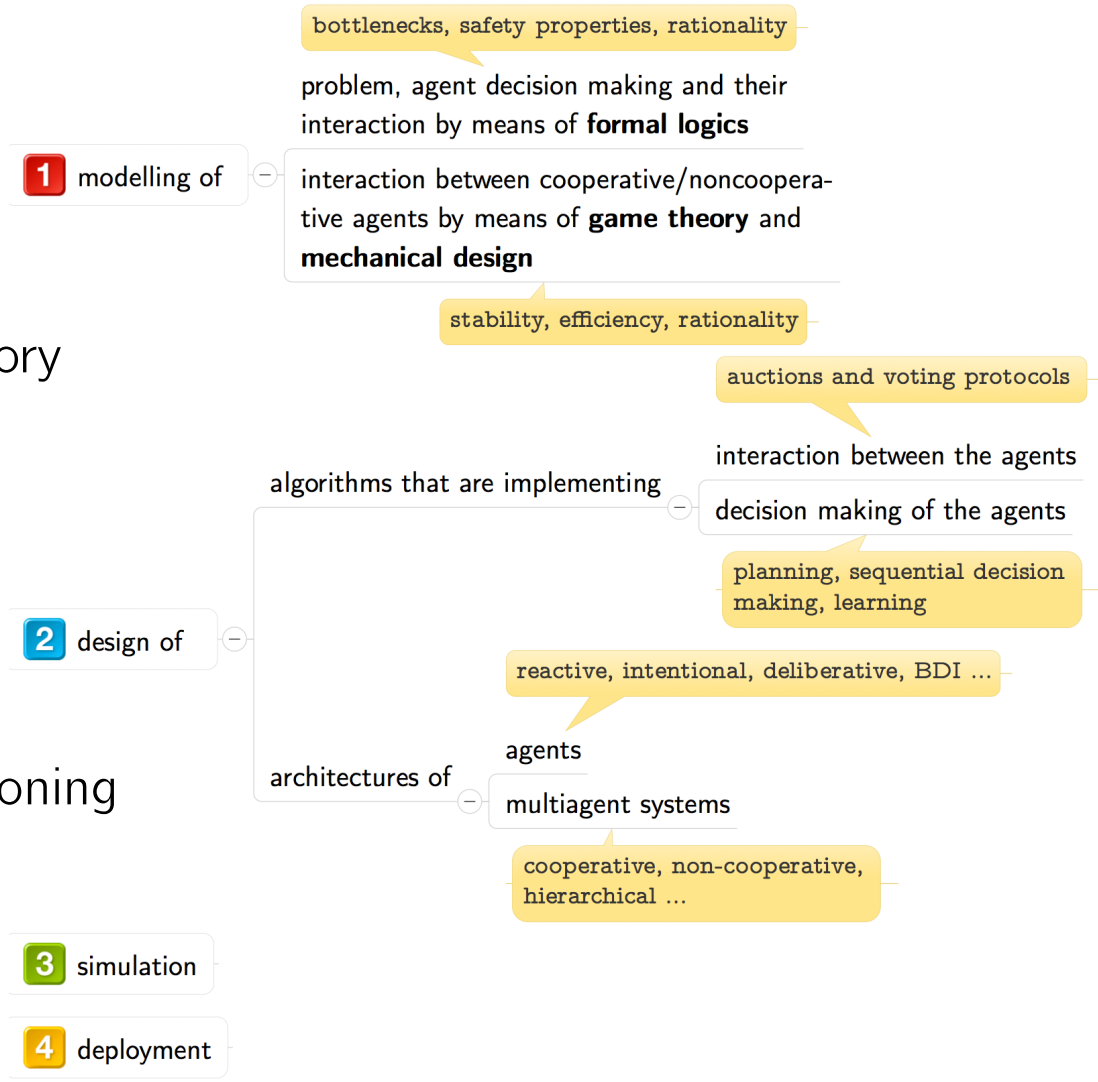
Security applications

Energy and smart grids



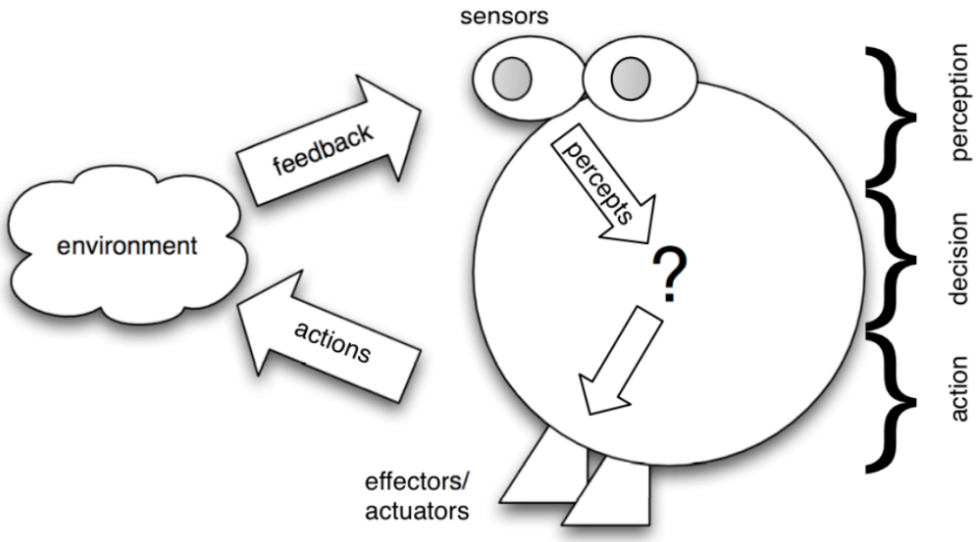
Course Content

- Agent architectures
- Non-cooperative game theory
- Coalition game theory
- Mechanism design
- Auctions
- Social choice
- Distributed constraint reasoning
- Agent based simulation



Defining Agency

What is Agent?



Definition (Russell & Norvig): An agent is anything that can perceive its environment (through its sensors) and act upon that environment (through its effectors)

Focus on situatedness in the environment (embodiment)

The agent can only influence the environment but not fully control it (sensor/effector failure, non-determinism)

What is Agent?

Definition (Wooldridge & Jennings): An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives/delegated goals.

Adds a second dimension to agent definition: the relationship between agent and designer/user

- agent is capable of independent action
- agent action is purposeful

Autonomy is a central, distinguishing property of agents

Introduction to Multiagent Systems

Specifying Agents

Agent Behaviour

$$f : \mathscr{P} \mapsto \mathscr{A}$$

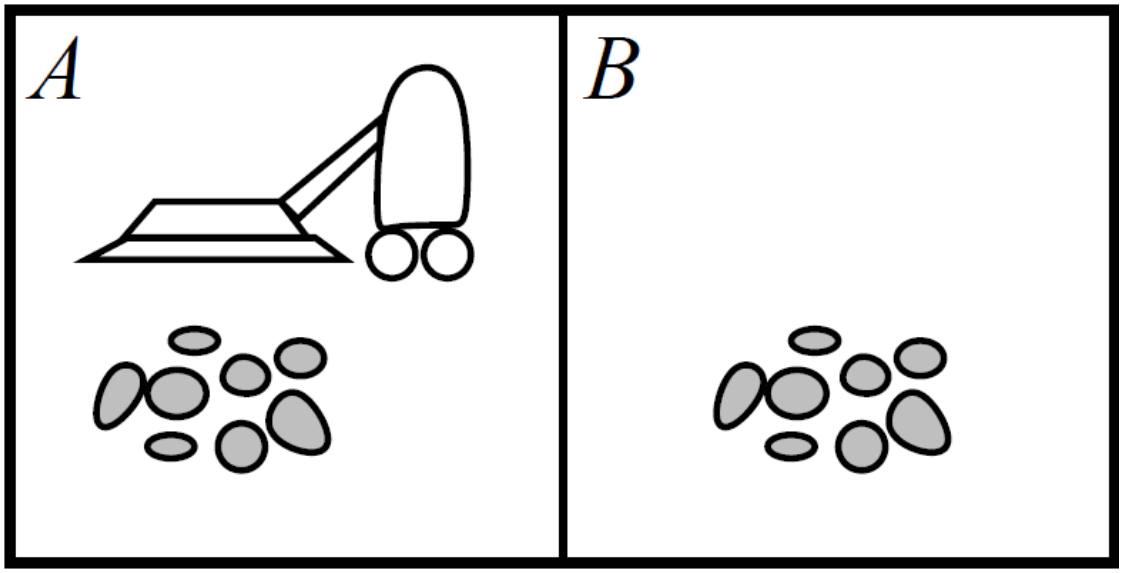
Agent's behaviour is described by the agent function that maps percept sequences to actions

The agent program runs on a physical architecture to produce f

Key questions:

What is the right function?
Can it be implemented in a small agent program?

Example: Vacuum Cleaner World



Percepts: location and contents, e.g. [A, Dirty]

Actions: Left, Right, Suck, NoOp

Vacuum Cleaner Agent

Percept sequence	Action
[A,Clean]	Right
[A, Dirty]	Suck
[B,Clean]	Left
[B, Dirty]	Suck
[A,Clean], [A,Clean]	Right
[A,Clean], [A, Dirty]	Suck
…	…
[A,Clean], [A,Clean], [A,Clean]	Right
[A,Clean], [A,Clean], [A, Dirty]	Suck
…	…

Is this a good agent function?

Rational Behaviour

Definition (Russell & Norvig): Rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date and whatever bulit-in knowledge the agent has.

Rationality is relative and depends on four aspects:

1. **performance measure** for the degree of success
2. **percept sequence** (complete perceptual history)
3. **agent's knowledge** about the environment
4. **actions** available to the agent

Specifying Task Environments (PEAS)

To design a rational agent, we must specify the task environment:

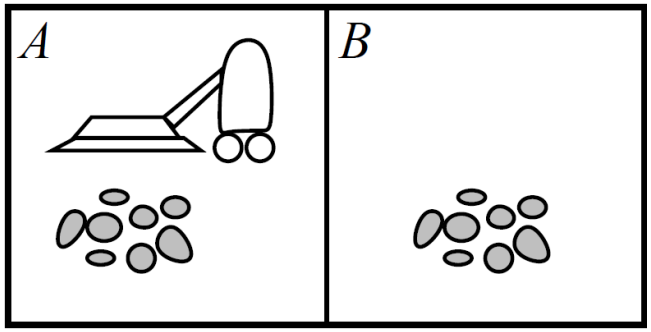
1. **Performance measure**
2. **Environment**
3. **Actuators**
4. **Sensors**

Task environments define problems to which rational agents are the solutions

Rationality of Vacuum Cleaner Agent

Agent programme:

Cleans a square if it is dirty and moves to the other square if not. Is it rational?



PEAS:

- The performance measure awards one point for each clean square at each time step, over a "lifetime" of 1000 time steps.
- The "geography" of the environment is known a priori but the dirt distribution and the initial location of the agent are not. Clean squares stay clean and sucking cleans the current square. The Left and Right actions move the agent left and right except when this would take the agent outside the environment, in which case the agent remains where it is.
- The only available actions are Left, Right, and Suck.
- The agent correctly perceives its location and whether that location contains dirt.

PEAS Examples

Agent	Performance measure	Environment	Actuators	Sensors

Properties of Environments

Fully observable vs. partially observable – can agents obtain complete and correct information about the state of the world?

Deterministic vs. stochastic – Do actions have guaranteed and uniquely defined effects?

Episodic vs. sequential – Can agents decisions be made for different, independent episodes?

Static vs. dynamic – Does the environment change by processes beyond agent control?

Discrete vs. continuous – Is the number of actions and percepts fixed and finite?

Single-agent vs. multi-agent – Does the behavior of one agent depends on the behavior of other agents?

Example Environments

	Solitaire	Backgammon	Shopping	Taxi
Observable				
Deterministic				
Episodic				
Static				
Discrete				
Single-agent	Yes	No	auctions)	No

Rational Behaviour

Definition (Russell & Norvig): Rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date and whatever bulit-in knowledge the agent has.

Rationality is relative and depends on four aspects:

1. **performance measure** for the degree of success
2. **percept sequence** (complete perceptual history)
3. **agent's knowledge** about the environment
4. **actions** available to the agent

Rationality

The agents rationality is given by the choice of actions based on expected utility of the outcome of the action. The rational agent selects an action a that provides the maximal expected outcome:

$$a = \arg\max_{l \in \mathcal{L}} \sum_{p_i : o_i \in l} p_i u(o_i)$$

Bounded Rationality: capability of the agent to perform rational decision (to choose the lottery providing maximal expected outcome) given bounds on computational resources:

- bounds on time complexity
- bounds on memory requirements

Calculative Rationality: capability to perform rational choice earlier than a fastest change in the environment can occur.

Rationality

Let us have a community of agents $A_j \in \mathcal{A}$ each choosing to play an action a_j, executing the lottery l_j. providing the agents with the utility $u(a_i)$.

- **Self-interested rational agent:** selects the action that optimizes its individual utility

$$a = \arg\max_{l \in \mathcal{L}} \sum_{p_i : o_i \in l} p_i U(o_i)$$

- **Cooperative rational agent:** selects the action that optimizes collective utility of the whole team:

$$a = \arg\max_{l \in \mathcal{L}} \sum_{\forall a_j \in \mathcal{A} - a} \sum_{p_{i,j} : o_{i,j} \in l_j} p_{i,j} u(o_{i,j}) + \sum_{p_i : o_i \in l} p_i u(o_i)$$

Summary

Multiagent systems approach ever more important in the increasingly interconnected world where systems are required to cooperate flexibly

→ "socially-inspired computing"

Intelligent agent is autonomous, proactive, reactive and sociable.

Agents can be cooperative or competitive (or combination thereof).

There are different agent architectures with different capabilities and complexity.

Related reading:

- Russel and Norvig: Artificial Intelligence: A Modern Approach – Chapter 2
- Wooldrige: An Introduction to Multiagent Systems – Chapters 1 and 2

→ Next: Belief-Desire-Intention Architecture