Zurich University
of Applied Sciences

Artificial Intelligence
V02: Intelligent Agents

Agents, environments and rationality
Environments types and properties
Agent types

Based on material by Stuart Russell, UC Berkeley

Zurich University of Applied Sciences and Arts
InIT Institute of Applied Information Technology (stdm)

Educational objectives

- **Remember** the **definition of** a **rational agent** and **PEAS**
- **Explain** **why** a **rational agent** might **neither** be **omniscient**, **prophetic** or **successful** and still be called «rational»
- **Argue** **how expressiveness** of an agent is **a mixed blessing**

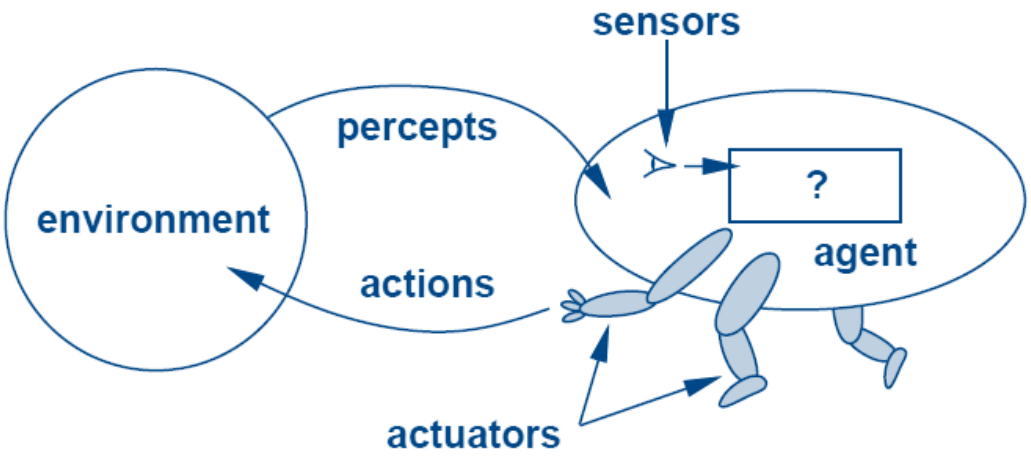
“In which we discuss the nature of agents, perfect or otherwise, the diversity of environments, and the resulting menagerie of agent types.”

➔ Reading: AIMA, ch. 2

Zurich University
of Applied Sciences

1. AGENTS, ENVIRONMENTS AND RATIONALITY

Agents and environments



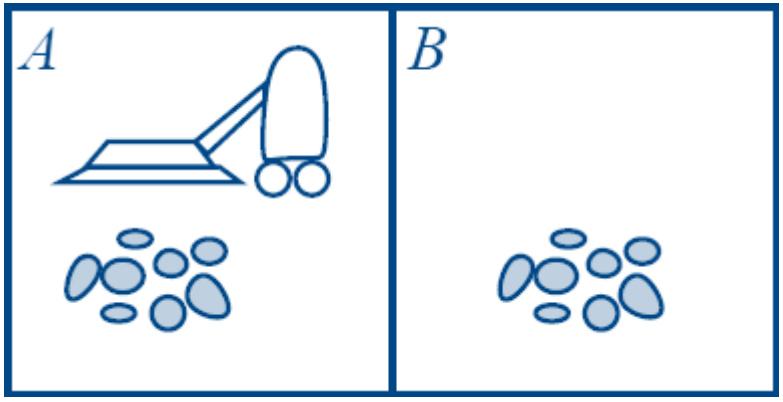
Vocabulary

- Agents include humans, robots, softbots, thermostats, etc.
- The agent function maps from percept sequence (complete history) to actions:

$$f: P^* \rightarrow A$$

- The agent program runs on the physical **architecture** to produce f

Example: Vacuum-cleaner world



Environment

- Percepts: location and content, e.g., $[A, Dirty]$
- Actions: $Left$, $Right$, $Suck$, $NoOp$

A vacuum-cleaner agent

Agent function f: tabulated

Percept sequences	Corresponding actions
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
…	…

Agent program: implementation of abstract functional description

```
function Reflex-Vacuum-Agent([location, status]) returns an action
  if status = Dirty then return Suck
  else if location = A then return Right
  else if location = B then return Left
```

Questions

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

Rationality

Need of a fixed performance measure to evaluate environment sequence

- One point per square cleaned up in time T?
- One point per clean square per time step, minus one per move?
- Penalize for $> k$ dirty squares?

➔ Is the sequence of results **desirable**?

A **rational agent** chooses whichever action maximizes the **expected** value of the performance measure **given the percept sequence to date**

- Rational ≠ omniscient
  Percept may not supply all relevant information
- Rational ≠ prophetic
  Action outcome may not be as expected

➔ Rational ≠ successful
  Rational = exploration, learning, autonomy

Zurich University
of Applied Sciences

2. ENVIRONMENTS TYPES AND PROPERTIES

Environment specifications: PEAS

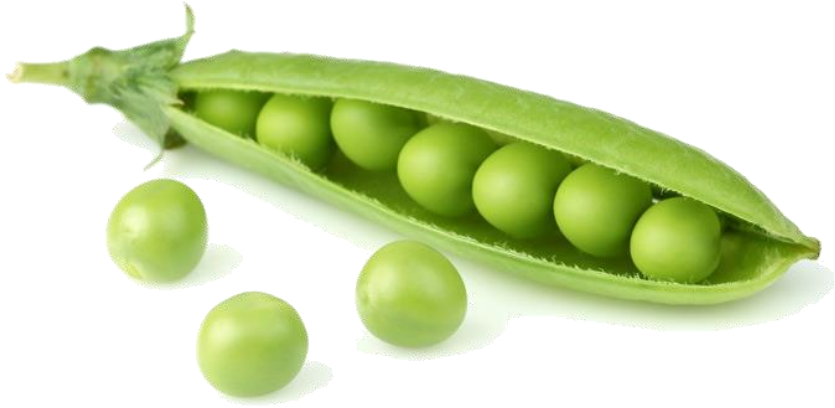
Performance measure, Environment, Actuators, Sensors

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

- PEAS specifies environment **types**: how we as humans would perceive ***external*** **features**

Example: the task of designing an automated taxi

- **Performance measure?**
- **Environment?**
- **Actuators?**
- **Sensors?**



Environment specifications: PEAS

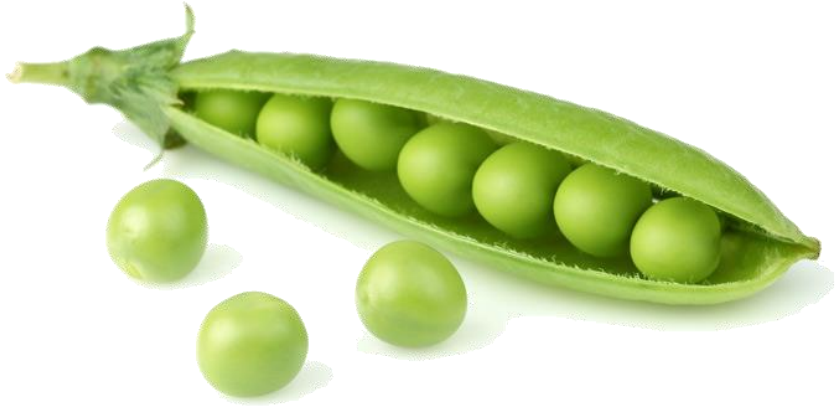
Performance measure, Environment, Actuators, Sensors

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

- PEAS specifies environment **types**: how we as humans would perceive ***external*** **features**

Example: the task of designing an automated taxi

- **Performance measure?** □ safety, destination, profits, legality, comfort
- **Environment?**
- **Actuators?**
- **Sensors?**



Environment specifications: PEAS

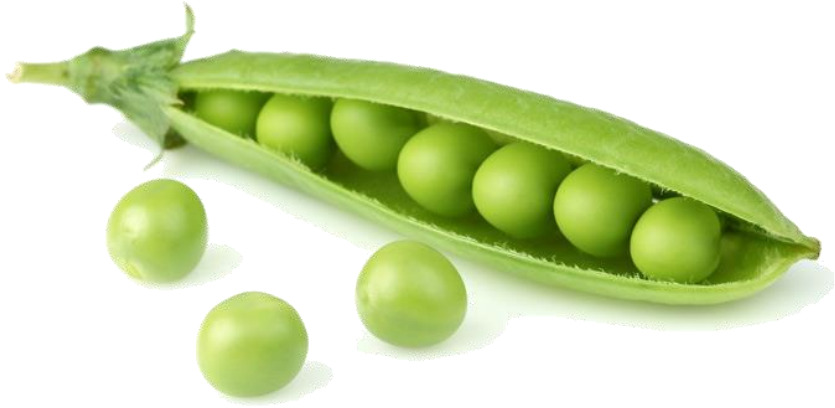
Performance measure, Environment, Actuators, Sensors

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

- PEAS specifies environment **types**: how we as humans would perceive ***external*** **features**

Example: the task of designing an automated taxi

- **Performance measure?** □ safety, destination, profits, legality, comfort
- **Environment?** □ streets/freeways, traffic, pedestrians, weather
- **Actuators?**
- **Sensors?**



Environment specifications: PEAS

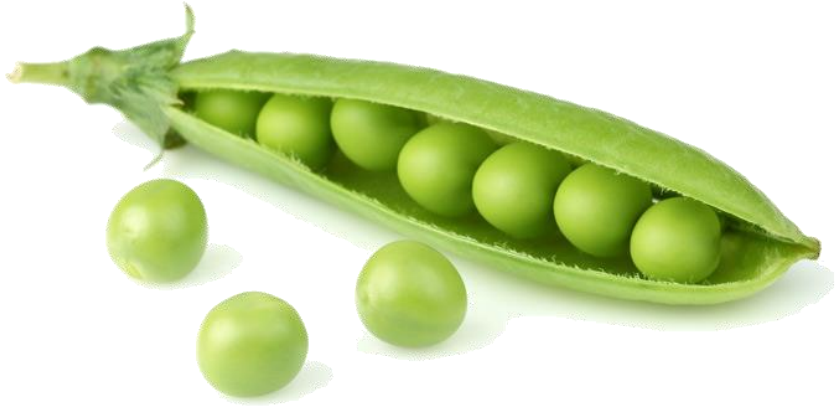
Performance measure, Environment, Actuators, Sensors

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

- PEAS specifies environment **types**: how we as humans would perceive ***external*** **features**

Example: the task of designing an automated taxi

- **Performance measure?** □ safety, destination, profits, legality, comfort
- **Environment?** □ streets/freeways, traffic, pedestrians, weather
- **Actuators?** → steering, accelerator, brake, horn, speaker/display
- **Sensors?**



Environment specifications: PEAS

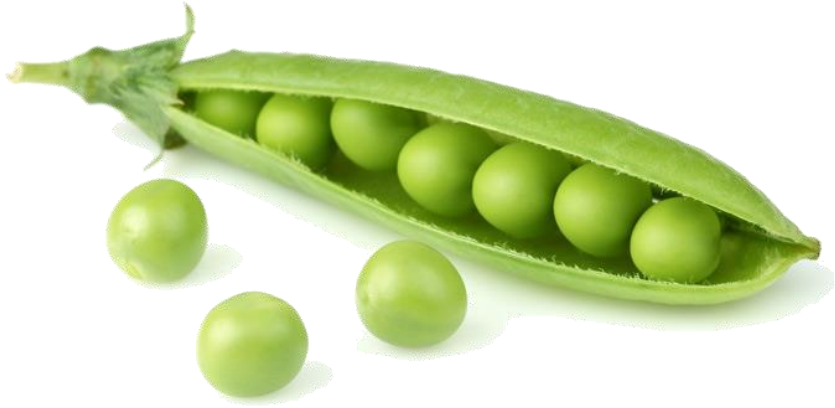
Performance measure, Environment, Actuators, Sensors

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

- PEAS specifies environment **types**: how we as humans would perceive ***external*** **features**

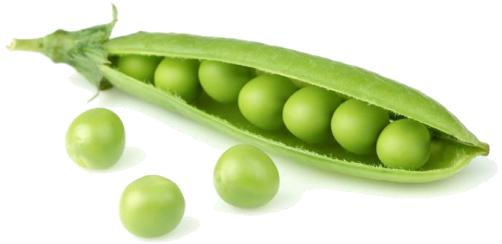
Example: the task of designing an automated taxi

- **Performance measure?** □ safety, destination, profits, legality, comfort
- **Environment?** □ streets/freeways, traffic, pedestrians, weather
- **Actuators?** → steering, accelerator, brake, horn, speaker/display
- **Sensors?** □ video, accelerometers, gauges, engine sensors, keyboard, GPS



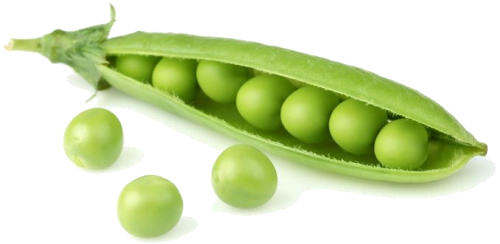
PEAS for an internet shopping agent

- **Performance measure?**
- **Environment?**
- **Actuators?**
- **Sensors?**



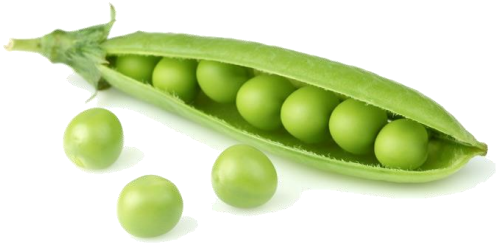
PEAS for an internet shopping agent

- **Performance measure?** price, quality, appropriateness, efficiency
- **Environment?**
- **Actuators?**
- **Sensors?**



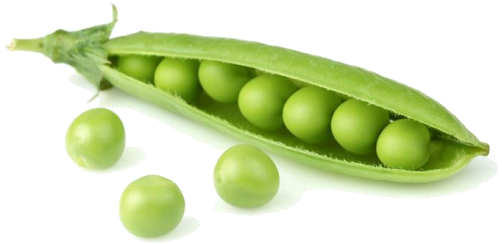
PEAS for an internet shopping agent

- **Performance measure?** □ price, quality, appropriateness, efficiency
- **Environment?** □ current and future WWW sites, vendors, shippers
- **Actuators?**
- **Sensors?**



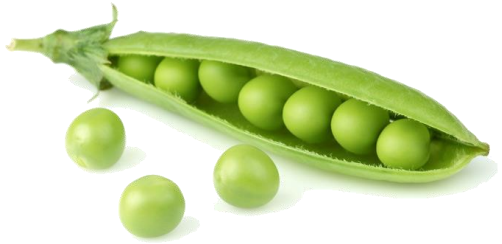
PEAS for an internet shopping agent

- **Performance measure?** □ price, quality, appropriateness, efficiency
- **Environment?** □ current and future WWW sites, vendors, shippers
- **Actuators?** □ display to user, follow URL, fill in form
- **Sensors?**



PEAS for an internet shopping agent

- **Performance measure?** □ price, quality, appropriateness, efficiency
- **Environment?** □ current and future WWW sites, vendors, shippers
- **Actuators?** □ display to user, follow URL, fill in form
- **Sensors?** □ HTML pages (text, graphics, scripts)



Environment properties

The *internal* features of an environment

- Fully observable vs. partial observable vs. unobservable
  → Do sensors give full access to the *relevant* state of the environment?
- Single agent vs. multiple agents → competitive vs. (partially) cooperative
  → Do others optimize a performance measure dependent on our agent?
- Deterministic vs. stochastic vs. nondeterministic
  → Do actions have certain consequences, or with certain probabilities (other's actions don't count)?
- Episodic vs. sequential
  → Do current actions influence future decisions (probably not in classification settings)?
- Static vs. dynamic
  → Does the world keep turning while our agent decides what to do?
- Discrete vs. continuous
  → Regarding states, time, percepts and actions

- Known vs. unknown
  → Are the rules/laws governing the environment known to the agent (not strictly a property of the env.)?

Zurich University
of Applied Sciences

Examples: Environments and their properties

	Solitaire	Poker	Image analysis	Internet shopping	Taxi
Observable?	(x)	(x)	x	-	-
Single-agent?	x	-	x	x (except auctions)	-
Deterministic?	x	- (stochastic)	x	(x)	-
Episodic?	-	-	x	-	-
Static?	x	x	(x)	(x)	-
Discrete?	x	x	-	x	-

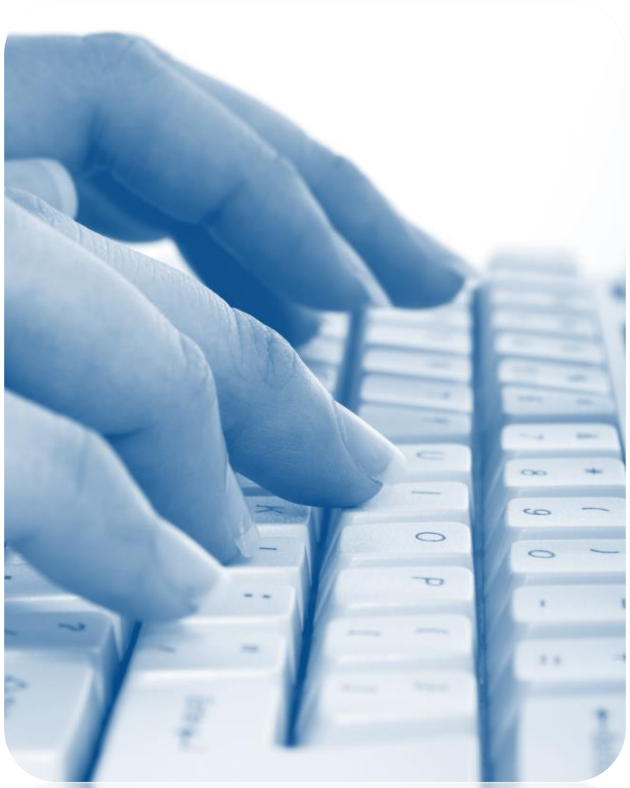
The **environment properties** largely **determine** the **agent design**

- The real world is (of course) partially observable, multi-agent, stochastic, sequential, dynamic, continuous

Exercise: Examine OpenAI Gym Retro

- Go to https://blog.openai.com/retro-contest/ for a list of gaming environments assembled specifically to build AI agents
- Why did OpenAI create Gym Retro and deprecate their much broader «Universe» archive (see https://github.com/openai/universe)?
- Select one environment of your liking: How would you classify it according to the properties of the last slide?

OpenAI



Zurich University
of Applied Sciences

3. AGENT TYPES

Four basic agent types

Agent := architecture + program

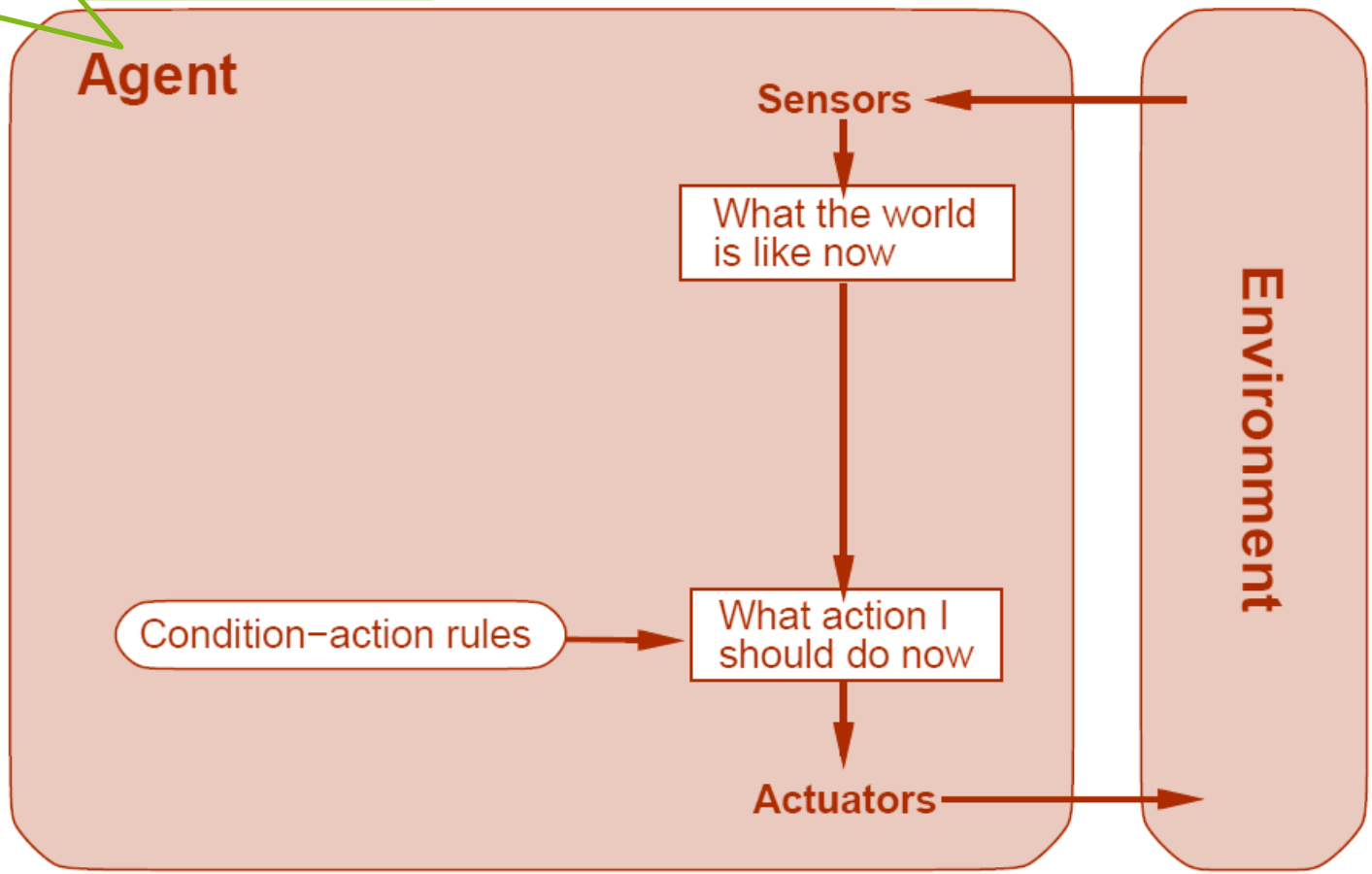
In order of increasing generality

- simple **reflex agents**: select action based on last percept
- **reflex** agents **with state**: regards history
- **goal-based** agents
- **utility-based** agents

All these can be turned into **learning agents**

Simple reflex agents

No explicit goal: the agent **just maps from last percept to next action**, any implicit goal is reached eventually

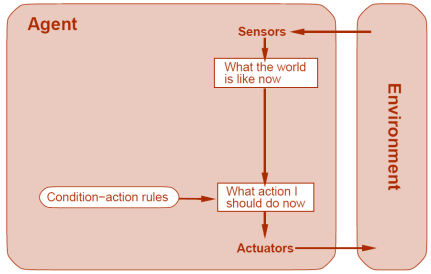


Example of simple reflex agent

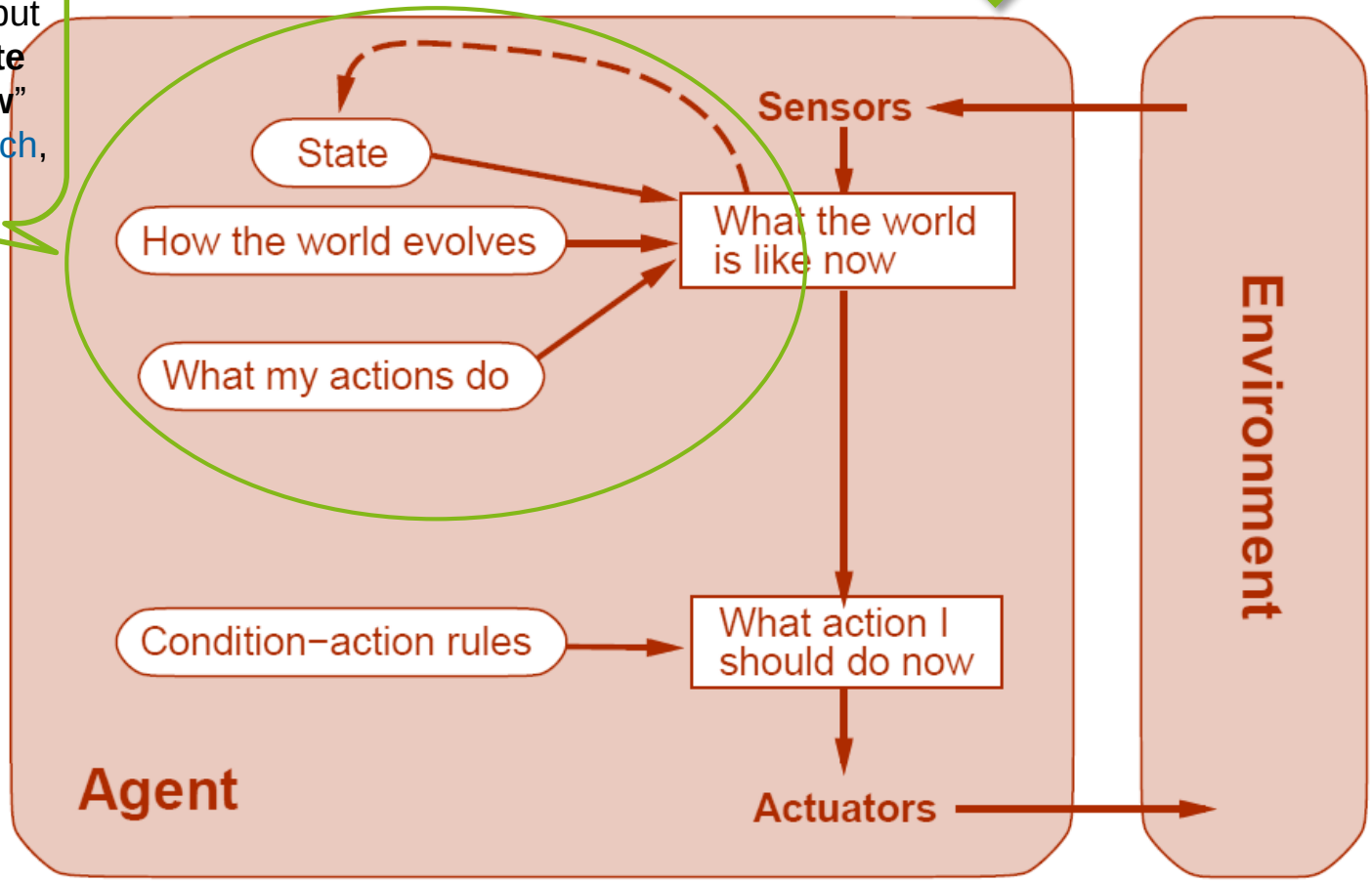
```
function Reflex-Vacuum-Agent([location, status]) returns an action
  if status = Dirty then return Suck
  else if location = A then return Right
  else if location = B then return Left
```

Reflex agents with state

A.k.a. model-based reflex agents



Still no goal, still no planning ahead; but a **more complete view of** the "**now**"
➔ e.g., local search, Bayes net

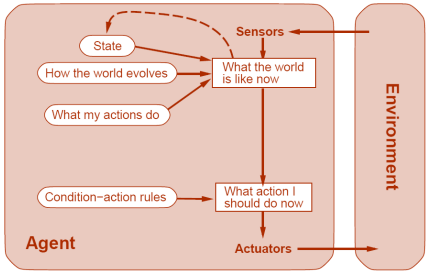


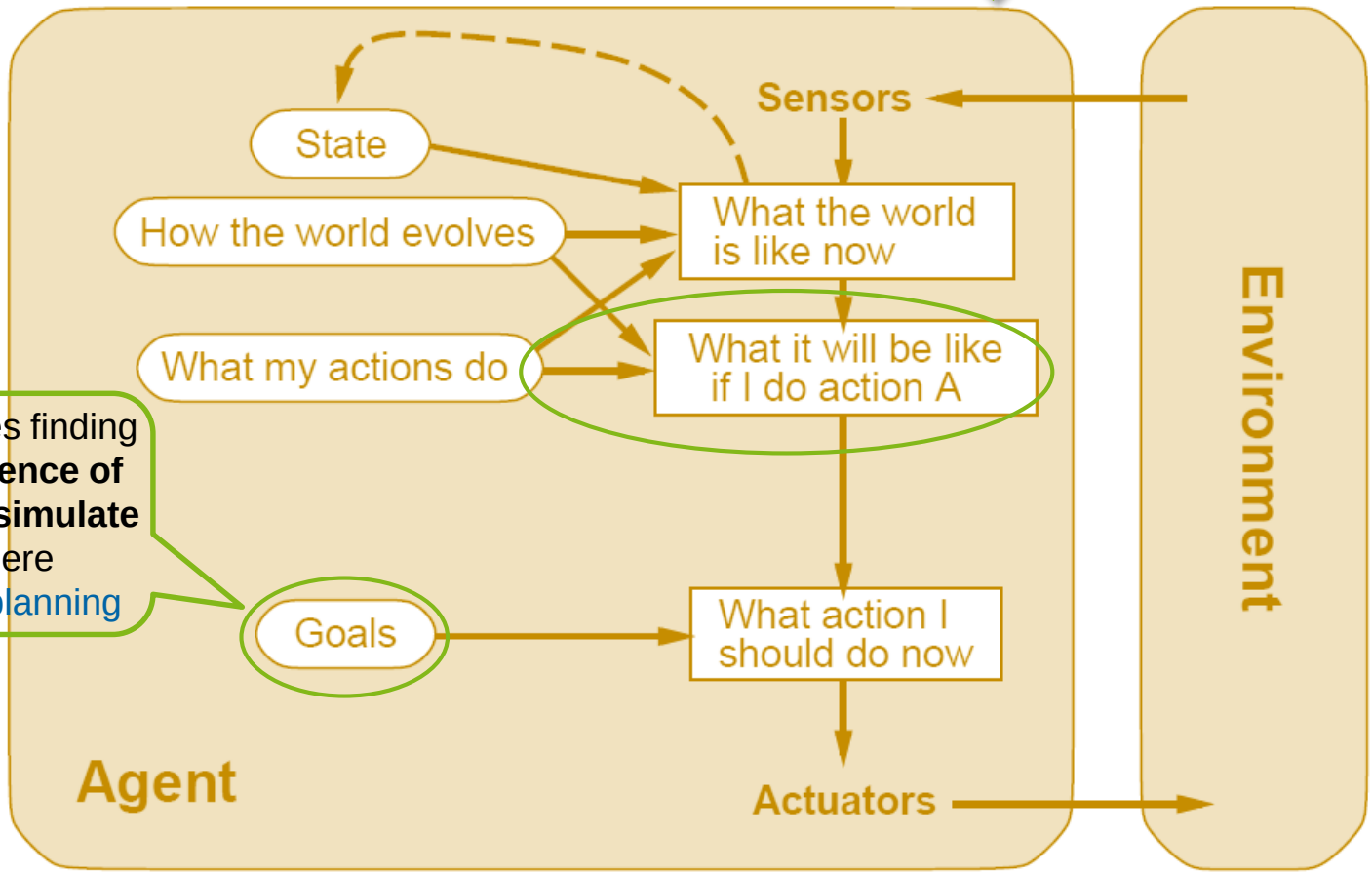
Example of stateful reflex agent

```
function Reflex-Vacuum-Agent([location, status]) returns an action
  static: last_A, last_B, numbers, initially INF
  if status = Dirty then …
```

Goal-based agents

A.k.a. model-based agents



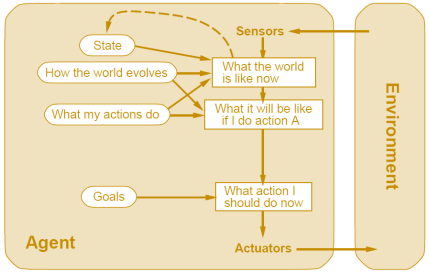


Explicit goal guides finding **congruent sequence of steps**; agent can **simulate** how to get there

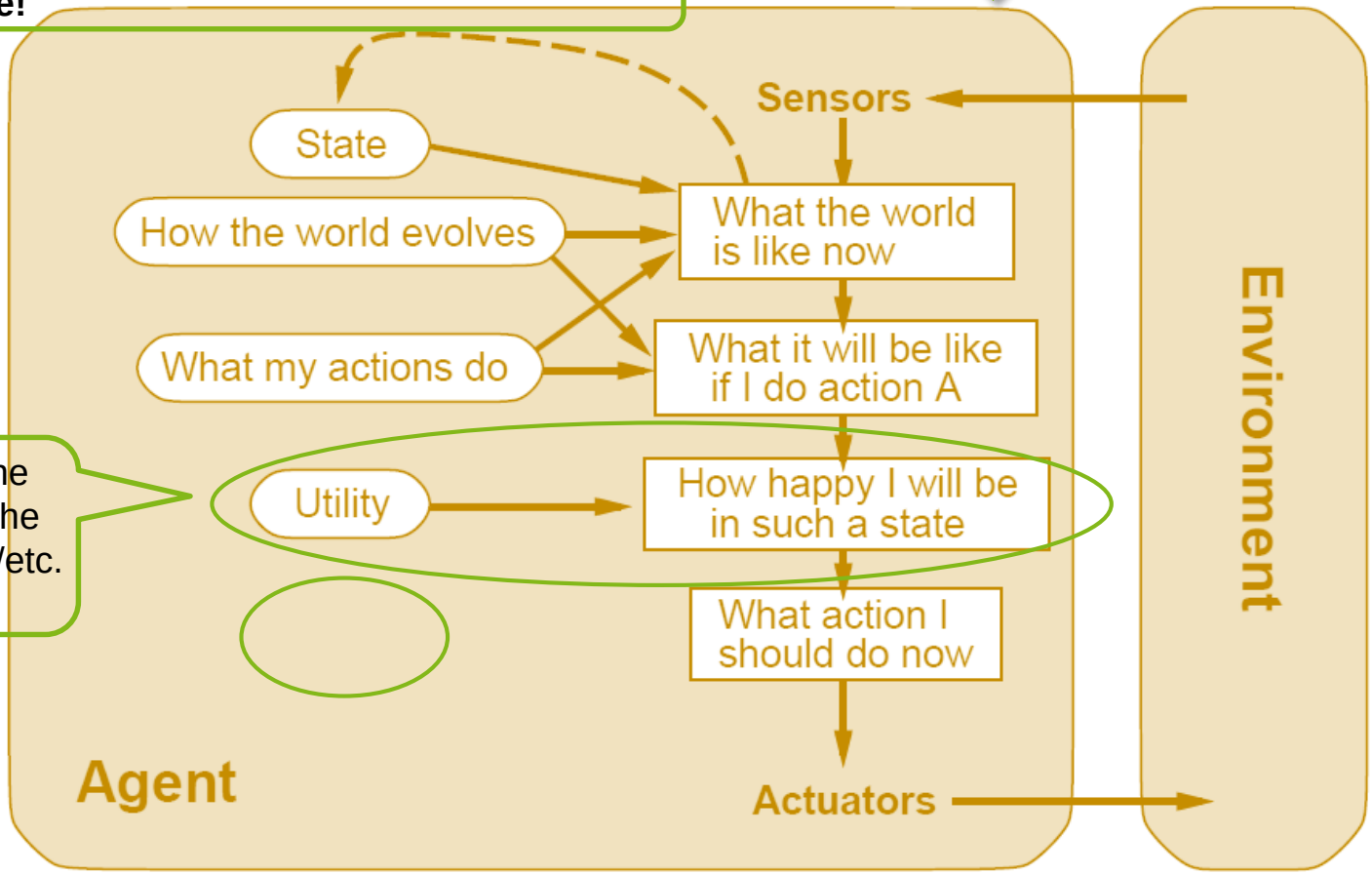
➔ e.g., search, planning

Utility-based agents

Finally reaching intelligence – or not?



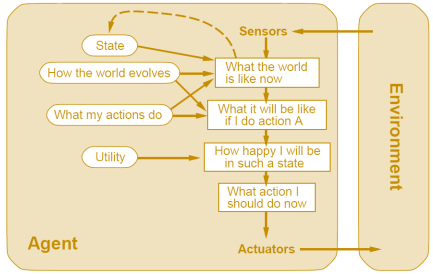
Just maximizing (expected) utility will create rational agents; but **it is not simple!**



Not just reach the goal; reach it in the **fastest/cheapest/**etc. **way possible**

Learning agents

Learning is applicable to each agent type



Zurich University of Applied Sciences

zh aw

Performance standard

At least a **performance measure** (e.g., error, recall/precision); ideally an **active learning** component (most often the ML engineer himself)

Critic

Sensors

feedback

changes

Learning element

Performance element

knowledge

The machine ***learning*** **algorithm** per se

learning goals

Problem generator

Balancing **exploration** / exploitation tradeoff

Agent

Actuators

Environment

The **trained model** (of a certain **representation**) in action ("testing")

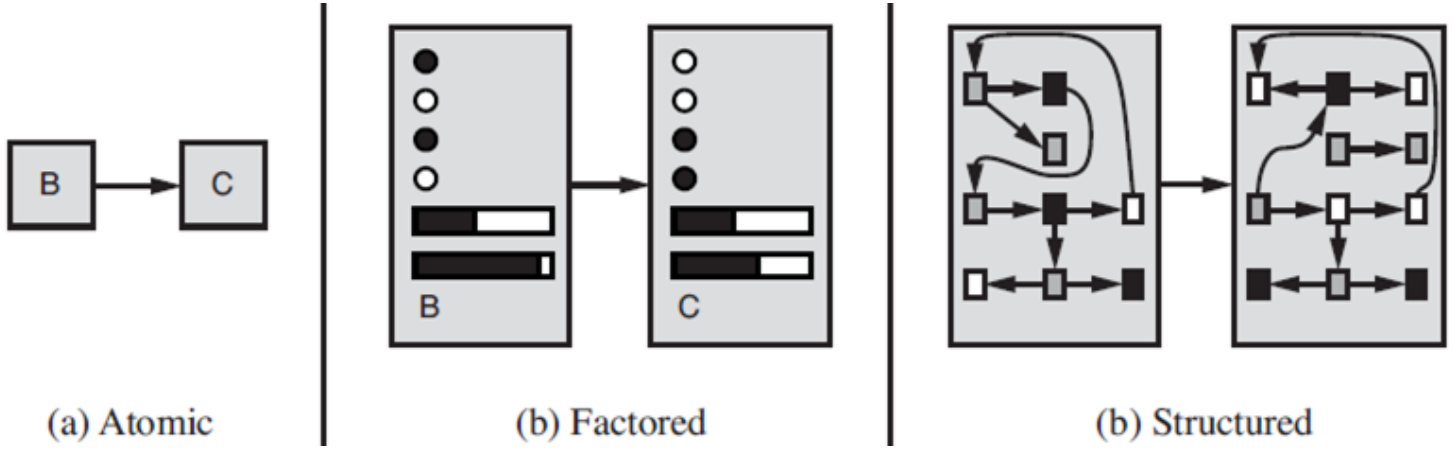
An representation taxonomy

Consider the representation of any building block, e.g. *«What my actions do»*

- **Atomic**: states are “just different” from each other
  ➔ **search**, **game-playing,** hidden Markov models, Markov decision processes

- **Factored**: states described by vectors (attributes), allowing for overlap and uncertainty
  ➔ **constrained satisfaction**, propositional logic, **planning**, Bayesian networks, **machine learning**

- **Structured**: states as entities and their relationships with each other
  ➔ **first-order logic**, first-order probability models, knowledge-based learning, NLP



Expressiveness revisited

Why a more capable agent is not always better

- Atomic – factored – structured is ordered by increasing expressiveness

- A mixed blessing:
  - More expressive → **captures more**, often **much more concise**
  - More expressive → **learning**/reasoning becomes **much harder**

➔ Intelligent systems may need to operate at several points on the axis (task-dependent)

A model of practical AI

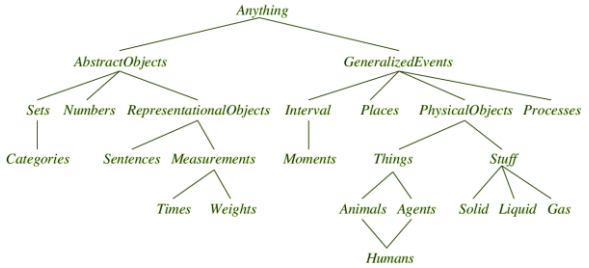
Inspired by E. Mogenet @ Zurich ML Meetup #31

Zurich University of Applied Sciences

zh aw

AI Knowledge engineering (symbolic):

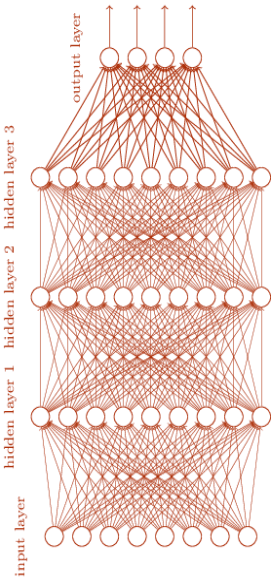
- ↓ Ontologies
- ↓ Logical inference



Gap to be filled by**: common sense DB**, **NLP**

Machine Learning (sub-symbolic):

- ↑ Hierarchical unsupervised learning
- ↑ Solid computer vision stack
- ↑ Images of the world



Review

- **Agents** interact with **environments** through **actuators** and **sensors**

- The **agent function** describes what the agent does in all circumstances
- The **performance measure** evaluates the environment sequence
- A **perfectly rational** agent **maximizes expected performance**
- **Agent programs** implement (some) agent functions
- **PEAS** descriptions **define** (specify) task **environment**s

- **Environment**s are categorized along several **dimensions**
  - observable? single-agent? deterministic?
  - episodic? static? discrete?

- Several basic **agent architectures** exist (all also learnable)
  - reflex, reflex with state,
  - goal-based, utility-based