Artificial Intelligence 1

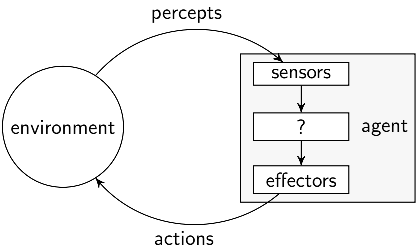
Prof. Dr. Frank Hopfgartner
Dr. Matthias Horbach

Institute for Web Science and Technologies (WeST)
University of Koblenz

Overview

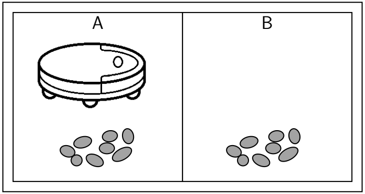
The agent paradigm

An agent is an entity that can *perceive* its environment through *sensors* and *act* through *effectors*.



- The agent paradigm is an appropriate abstract to describe intelligent and autonomously acting software- and hardware systems (e. g. robots)
- It can also be used to describe biological entities

Example: the vacuum cleaner world



- An autonomous robotic vacuum cleaner has to maintain a clean apartment
- Environment: two rooms A and B
- Perceptions: pairs (l, s) with
  - l: current location ($l = A$ or $l = B$)
  - s: status of the current location ($s =$ "clean" or $s =$ "dirty")
- Actions of the robot: `goLeft`, `goRight`, `clean`, `idle`

Rational agents

Definition

An agent is called *rational* if it always selects actions in order to maximise the expected value of some *performance measure* wrt. its current perceptions and its knowledge.

- Rational decision-making is not the same as omniscience
  - the assessment is always wrt. the subjective knowledge (=beliefs)
  - the effects of an action may not necessarily be certain
- Rational decision-making does not necessarily imply successful acting
  - the agent maximises the *expected* value of a performance measure
  - Example:
    - The expected (monetary) value of a casino bet is usually negative; it is rational not to play
    - However, the actual result (if the bet is won) may be significantly greater

Rational agents: PEAS

- The definition of "rational decision-making" is domain-dependent
- In general, this depends on four factors:
  1. *Performance*: what is the performance measure?
  2. *Environment*: how does the environment look like?
  3. *Actions*: what actions can the agent perform?
  4. *Sensors*: how is the environment perceived?
- Example: autonomous taxi
  1. *P*: safety, profit, comfort, legality, destination, ...
  2. *E*: streets, other cars, pedestrians, weather, ...
  3. *A*: steerings, accelerating, decelerating, honking, speakers, ...
  4. *S*: cameras, sonar, keyboard, GPS, ...

Properties of environments

- completely observable vs. partially observable
- deterministic vs. non-deterministic
- episodic vs. sequential
- static vs. dynamic
- discrete vs. continuous
- single agent vs. multi agents

Examples:

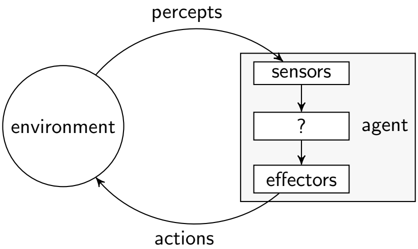
- chess computer: completely observable, deterministic, sequential, static, discrete, multi agents
- taxi: partially observable, non-deterministic, sequential, dynamic, continuous, multi agents

Agents, automatic planning and knowledge representation

- Automatic planning is an important component of an agent
- It is not the only component though
- Knowledge representation plays a role as well
  - How to formalise the general rules of the environment?
  - How to update the agent's beliefs upon new perceptions?
- The agent paradigm combines many disciplines of AI
- This is formalised through *agent models*

Overview

The agent paradigm, revisited

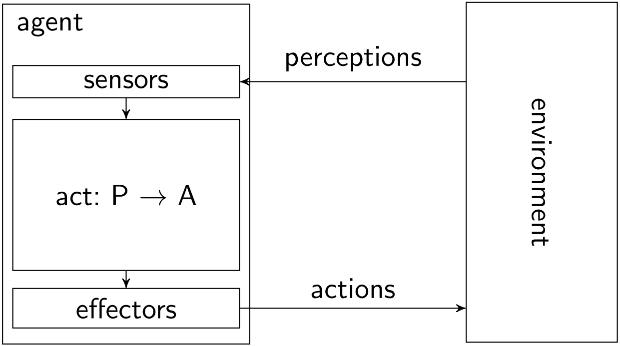


Question: how to proceed from perception to action selection?

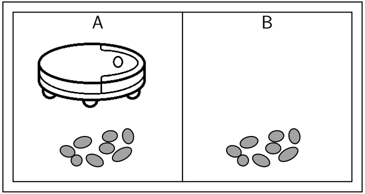
A simple agent model

A simple agent model can be formalised as follows

- Let P be the set of all possible perceptions
- Let A be the set of all possible actions
- An agent can be implemented by a single function $act : P \rightarrow A$



Example: the vacuum cleaner world



- A simple strategy for the robot: "If the current location is dirty, clean it; otherwise go into the other room"
- As agent function act_{vac}:

$$act_{vac}(l, s) = \begin{cases} \texttt{goLeft} & \text{if } l = B \text{ and } s = \text{“clean”} \\ \texttt{goRight} & \text{if } l = A \text{ and } s = \text{“clean”} \\ \texttt{clean} & \text{if } s = \text{“dirty”} \end{cases}$$

Algorithmic formalisation

```
Algorithmus simpleAgent
Input:
   Perception p
Output:
   Action a

a = act(p);

return a;
```

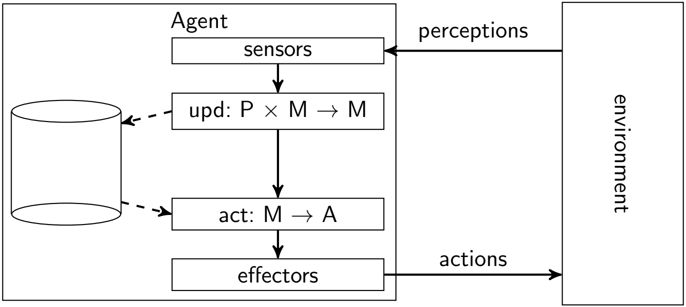
Types of agents

- Agents as the one from before are called *reflex-based agents*
- Advantages:
  - Easy program development
  - Efficient implementation
  - Sufficient for many simple agents
- Disadvantages:
  - no memory, no world view
  - no planning
  - rationality of decision-making is only dependent on the program
- Other agent models:
  - Model-baed agents
  - BDI agents
  - ...

Model-based agents

Idea: keep a model of the world up to date

- Let P be the set of all possible perceptions
- Let A be the set of all possible actions
- Let M be the set of all possible world models
- Then an agent can be implement via the functions
  - $upd : P \times M \to M$
  - $act : M \to A$



Algorithmic formalisation

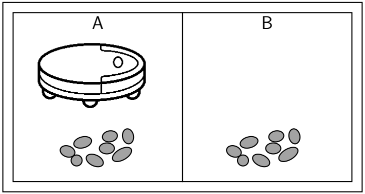
```
Algorithm modelAgent
Input:
   Perception p
   Model m (statisch)
Output:
   Action a

m = upd(p,m);
a = act(m);

return a;
```

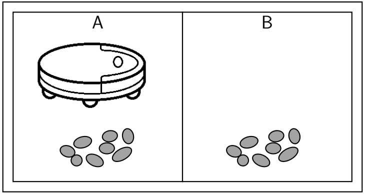
- Perceptions are not just interpreted but integrated into a model of the world
- Perceptions can therefore be exploited for later action selections

Example: vacuum cleaner world 1/2



- Assume that the system is closed and there are no random changes in the environment (rooms get not "dirty")
- World model is a triple (pos, sa, sb) with
  - pos = position of the robot
  - sa = status of room A
  - sb = status of room B
- $M = \{(pos, sa, sb) \mid pos \in \{A, B\}, sa, sb \in \{\text{"dirty"}, \text{"clean"}, \text{"unknown"}\}\}$

Example: vacuum cleaner world 2/2



Implementation of a model-based agent through functions upd_{vac2} and act_{vac2}:

$$upd_{vac2}((pos, sa, sb), (l, s)) = \begin{cases} (l, sa, s) & \text{if } l = B \\ (l, s, sb) & \text{if } l = A \end{cases}$$

$$act_{vac2}((pos, sa, sb))$$
$$= \begin{cases} \texttt{clean} & \text{if } pos = A \wedge sa = \text{“dirty”} \\ \texttt{clean} & \text{if } pos = B \wedge sb = \text{“dirty”} \\ \texttt{goLeft} & \text{if } pos = B \wedge sb = \text{“clean”} \wedge sa \in \{\text{“dirty”}, \text{“unknown”}\} \\ \texttt{goRight} & \text{if } pos = A \wedge sa = \text{“clean”} \wedge sb \in \{\text{“dirty”}, \text{“unknown”}\} \\ \texttt{idle} & \text{otherwise} \end{cases}$$

The state of an agent

- In the previous agent model the state of an agent only consisted of the world model
- But this is not the only factor that influences action selection
- Other factors:
  - Goals: What does the agent desire to achieve?
  - Motivation: What drives the agent?
  - Intentions: What is the current plan to achieve the goals?
  - Know-how: How can the agent plan actions to fulfil intentions?
  - ...
- Let us consider only
  - *Beliefs*: What knows the agent (about the world, itself)?
  - *Desires*: What does the agent wish to achieve?
  - *Intentions*: How can goals be decomposed into actions?

$\rightarrow$ BDI model

Deliberation and means-end reasoning

- Human decision-making consists of two activities:
  1. *deliberation*:
     - Decide *what* exactly has to be achieved
     - Result are intentions
  2. *means-end reasoning* (or *planning*):
     - Decide *how* to achieve intentions
     - Result are plans

- How does deliberation work?
  1. First, one deliberates about the options (possible goals = desires)
  2. Select an option (or multiple ones) and commit yourself to them
- Selected options are intentions

A simple BDI model: formalisation 1/3

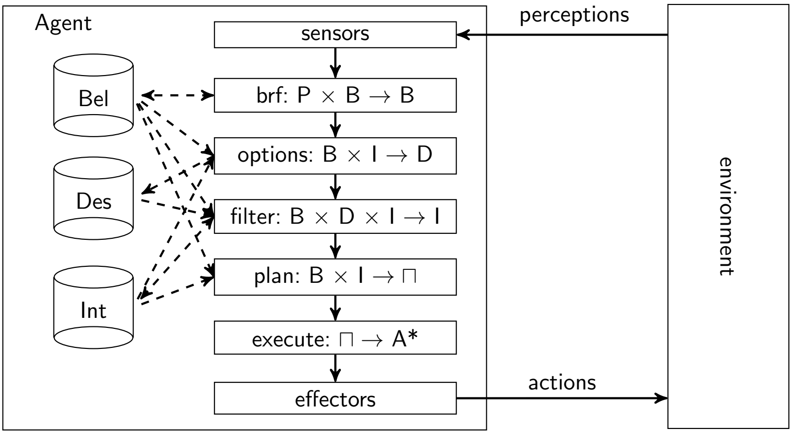
- The following sets of objects are relevant:
  - P: the set of all possible perceptions
  - A: the set of all possible actions
  - B: the set of all possible belief bases
  - I: the set of all possible sets of intentions
  - D: the set of all possible sets of desires
  - Π: the set of all possible plans
- Then an agent can be implement via the functions
  - $brf : P \times B \to B$
    Maps new perceptions and the old beliefs onto new beliefs (belief revision)
  - $options : B \times I \to D$
    What are the possible goals wrt. the current beliefs and the current intentions?
  - $filter : B \times D \times I \to I$
    Which concrete goals should be selected in the current situation?

A simple BDI model: formalisation 2/3

- Then an agent can be implement via the functions
  - $brf : P \times B \rightarrow B$
  - $options : B \times I \rightarrow D$
  - $filter : B \times D \times I \rightarrow I$
  - $plan : B \times I \rightarrow \Pi$
    Computes a concrete plan that fulfils the intentions
  - $execute : \Pi \rightarrow A^*$
    Realises a plan through a sequence of actions in the environment

A simple BDI model: formalisation 3/3

Let $Bel \in B$, $Des \in D$, $Int \in I$ be the current state of an agent.



Algorithmic formalisation

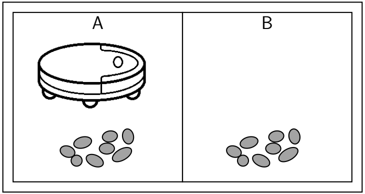
```
Algorithm bdiAgent
Input:
   Perception p
   Beliefs Bel (static)
   Desires Des (static)
   Intentions Int (static)

Output:
   Action sequence π=(a_1,...,a_n)

Bel = brf(Bel, p);
Des = options(Bel, Int);
Int = filter(Bel, Des, Int);
π = plan(Bel,Int);

return π;
```

Example: vacuum cleaner worldt 1/4



- Let us extend the scenario a bit:
  - The robotic vacuum cleaner now has a battery that drains over time
  - Every perception of the agent contains the current state of the battery
  - The new action `recharge` allows the agent to completely recharge his battery
  - The agent now as two goals
    - It must keep all rooms clean
    - Its battery level shall not be below 5%

Example: vacuum cleaner world 2/4

We formalise the scenario as follows

- Perceptions are of the form (location, status, battery level)
- Beliefs are of the form (location, status room A, status room B, battery level)
- Desires are *cleanRooms* and *chargeBattery*
- Intentions are *cleanRooms*, *chargeBattery*, as well as *inspectRoomA* und *inspectRoomB* (the last two are concrete goals that are used to fulfil the goal *cleanRooms*)
- Actions are `goLeft`, `goRight`, `clean`, `recharge`, and `idle`

Example: vacuum cleaner world 3/4

We simulate an exemplary run through the BDI model:

1. The initial state of the agent is given through
   - $Bel = (A, \text{“unknown”}, \text{“unknown”}, \text{“100\%”})$
   - $Des = \emptyset$
   - $Int = \emptyset$
2. Now the agent perceives $(A, \text{“dirty”}, \text{“95\%”})$
3. The agent update his beliefs using *brf*
   - $Bel = (A, \text{“dirty”}, \text{“unknown”}, \text{“95\%”})$
4. The agent computes (using *options*) the current options
   - The belief of the agent that room A is dirty generates the desire *cleanRooms*
   - Furthermore, *chargeBattery* is a desirable goal as well
   - $Des = \{cleanRooms, chargeBattery\}$

Example: vacuum cleaner world 4/4

5. The two options *cleanRooms* and *chargeBattery* are now weighted against each other (using *filter*)
   - As the battery level is quite alright, priority is given to the goal *cleanRooms*
   - $\mathit{Int} = \{\mathit{cleanRooms}\}$
6. Now a plan is generated for the selected intention *cleanRooms* (using *plan*), hier only consisting of one action
   - $\pi = (\texttt{clean})$
7. This plan will now be executed

Remarks

- The presented BDI model is still a very abstract description of an agent
- For example, a drawback in the current version is that a plan is generated and then completely executed without taking new perceptions (during execution) into account
- Moreover: the functions *brf*, *options*, *filter* and *plan* are defined in a general fashio
- Concrete implementations of these functions should satisfy certain requirements
  - *brf* should satisfy requirements similar to the AGM postulates
  - *plan* should generate executable plans
  - *options* should not generate goals that are already satisfied
  - *filter* should not abandon current intentions without reason and create new ones without reasons
- We will have a closer look a the last point

Self-commitment for intentions 1/3

Some time in the not-so-distant future, you are having trouble with your new household robot. You say "Willie, bring me a beer." The robot replies "OK boss." Twenty minutes later, you screech "Willie, why didn't you bring me that beer?" It answers "Well, I intended to get you the beer, but I decided to do something else." Miffed, you send the wise guy back to the manufacturer, complaining about a lack of commitment.

- Current intentions should not be (unnecessarily) abandoned by the *filter* function (no *under-commitment*)

Self-commitment for intentions 2/3

After retrofitting, Willie is returned, marked "Model C: The Committed Assistant." Again, you ask Willie to bring you a beer. Again, it accedes, replying "Sure thing." Then you ask: "What kind of beer did you buy?" It answers: "Bitburger." You say "Nevermind, I do not want that one." One minute later, Willie trundles over with a Bitburger in its gripper.

- If intentions are no longer viable in the current situation (e. g., because of new information) they should be abandoned (no *over-commitment*)

Self-commitment for intentions 3/3

After still more tinkering, the manufacturer sends Willie back, promising no more problems with its commitments. So, being a somewhat trusting customer, you accept the rascal back into your household, but as a test, you ask it to bring you your last beer. The robot gets the beer and starts towards you. As it approaches, it lifts its arm, wheels around, deliberately smashes the bottle, and trundles off. Back at the plant, when interrogated by customer service as to why it had abandoned its commitments, the robot replies that according to its specifications, it kept its commitments as long as required—commitments must be dropped when fulfilled or impossible to achieve. By smashing the bottle, the commitment became unachievable.

- Action planning (in *plan*) must be oriented towards satisfaction of intentions

Self-commitment strategies

- *blind commitment* (or *fanatical commitment*)
  - An agent with blind commitment will pursue an intention as long as it thinks that the intention is not fulfilled.
- *single-minded commitment*
  - An agent with single-minded commitment will pursue an intention until it either thinks that is has been fulfilled or when it thinks that it cannot be fulfilled anymore.
- *commitment* always addresses two aspects
  - Commitment towards the intention (the selected goal)
  - Commitment towards the plan

What about our current model?

- Blind commitment towards both the goal and the plan
- First modification: re-plan if necessary

Algorithmic formalisation

```
Algorithm bdiAgent2
Input: Perception p
   Beliefs Bel (static)
   Desires Des (static)
   Intentions Int (static)
   Current plan π (static)

Output: Action a

Bel = brf(Bel, p);

if π is empty
  Des = options(Bel, Int);
  Int = filter(Bel, Des, Int);
  π = plan(Bel,Int);
else
   if not sound(π, Int, Bel)
     π = plan(Bel,Int);

a = head(π);
π = tail(π);

return a
```

Analysis

- With this modification the agent is no longer blindly committed towards the plan
  - plans are executed step by step
  - After every action the agent perceives the environment and updates its beliefs
  - If the current plan becomes impossible, the agent re-plans
- The agent is still blindly committed towards its intention
  - If a plan fails, a new one will be created
  - It is not checked whether the current intention should be maintained after all
- Another modification: always check whether the intention is satisfied and, if not, whether it's still satisfiable ($\rightarrow$ single-minded commitment)

Algorithmic formalisation

```
Algorithm bdiAgent3
Input/Output: <as before>

Bel = brf(Bel, p);

if π is empty
     or succeeded(Int, Bel)
     or impossible(Int, Bel)
  Des = options(Bel, Int);
  Int = filter(Bel, Des, Int);
  π = plan(Bel,Int);
else
   if not sound(π, Int, Bel)
     π = plan(Bel,Int);

a = head(π);
π = tail(π);

return a
```

Analysis and propects

- With this modification the agents is single-mindedly committed towards both intention and plan
- An intention is pursued for as long as it is either fulfilled or becomes unfulfillable
- However, this behaviour is sometimes also not appropriate

  An agent has the goal "to own one million dollars". Its plan to achieve this goal is to buy one lottery ticket per day.

  Assuming the agent has enough money for the daily lottery ticket, achieving the goal via this plan is possible and the agent would never abandon the plan nor the intention.
- Nevertheless, it may be advisable to overthink one's intentions from time to time

Chapter 5.1: Agent models

Summary

- A simple agent model of a reflex-based agent is realisiert through a function $act : P \to A$
- Model-based agent with functions

$$upd : P \times M \to M \qquad\qquad act : M \to A$$

- The BDI model: *Beliefs*, *Desires*, *Intentions*
  - Deliberation and means-end reasoning
  - Realised through the functions *brf*, *options*, *filter*, *plan*, *execute*
  - Self-commitment towards intentions and plans