

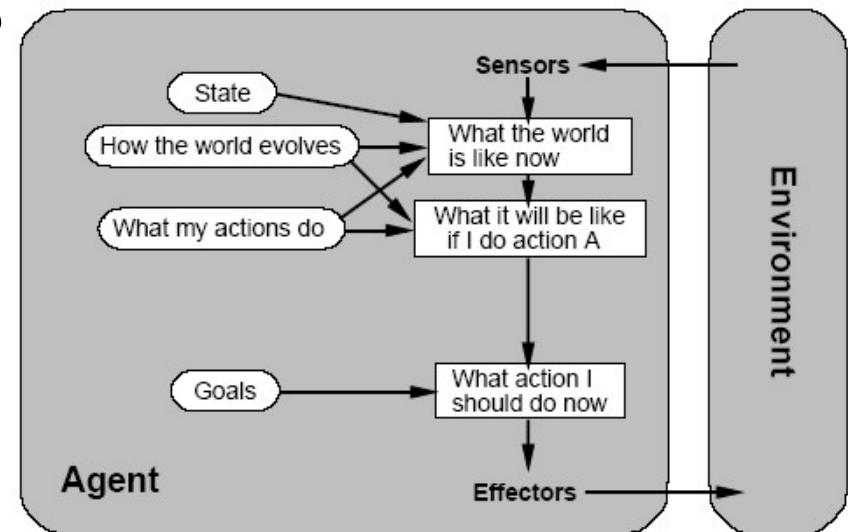
Implementing a Deliberative Agent

Intelligent Agents Course

(Credits to Radu Jurca
Michael Schumacher)

Deliberative Agents

- Also named ***mental*** or ***rational*** Agents
- The agent has an **explicit model of the world** in which it lives.
- Seeks to perform **goals**
- A **planner reasons** on the world model and decides which actions to realize by producing a plan, in order to achieve its goals.



Vehicles - Deliberative Agents

- > The vehicle computes an **optimal plan** using a state-based search algorithm
 - > Rewards on goal states: an optimal plan to **minimize the cost**
-
- States of the world are known with certainty
 - State transitions are deterministic

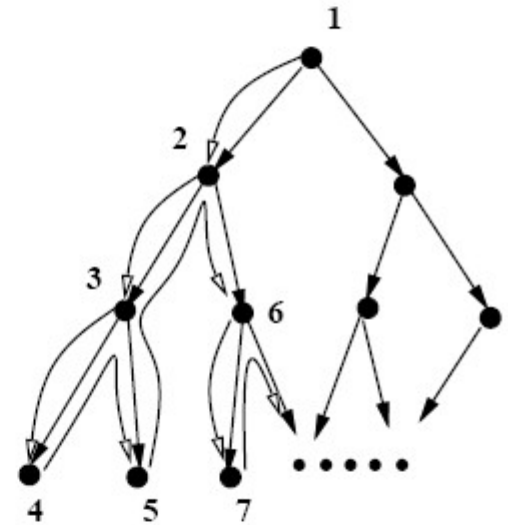
The Cost of a Plan

- Plan
 - Sequence of actions such that all tasks are delivered
 - **Vehicle can carry multiple tasks**
 - E.g. (T1,T2) ->
 - (move to pickup T1, pickup T1, move to delivery T1, deliver T1, move to pickup T2, pickup T2, move to deliver T2, deliver T2)
- Simplest cost form: distance x cost/km

Depth-first Search Algorithm

Search with cycle detection

```
1.  $Q \leftarrow$  initial node
2.  $C \leftarrow$  empty
3. repeat
4.   if  $Q$  is empty, return failure
5.    $n \leftarrow$  first element of  $Q$  ,  $Q \leftarrow \text{rest}(Q)$ 
6.   si  $n$  is a final node, return  $n$ 
7.   si  $n$  is not a member of  $C$  then
8.     add  $n$  to  $C$ 
9.      $S \leftarrow \text{succ}(n)$ 
10.     $Q \leftarrow \text{append}(S, Q)$ 
11.   endif
12. end
```



A* Search Algorithm

Algorithm A* (best-first)

```
1.Q ← initial node
2.C ← empty
3.repeat
4.  if Q is empty, return failure
5.  n ← first element of Q, Q ← rest(Q)
6.  if n is a final node, return n
7.  if n ∉ C, or has lower cost than its copy in C then
8.    add n to C
9.    S ← succ(n)
10.   S ← sort(S,f)
11.   Q ← merge(Q,S,f)
    (Q is ordered in increasing order of  $f(n) = g(n) + h(n)$ )
12. endif
13.end
```

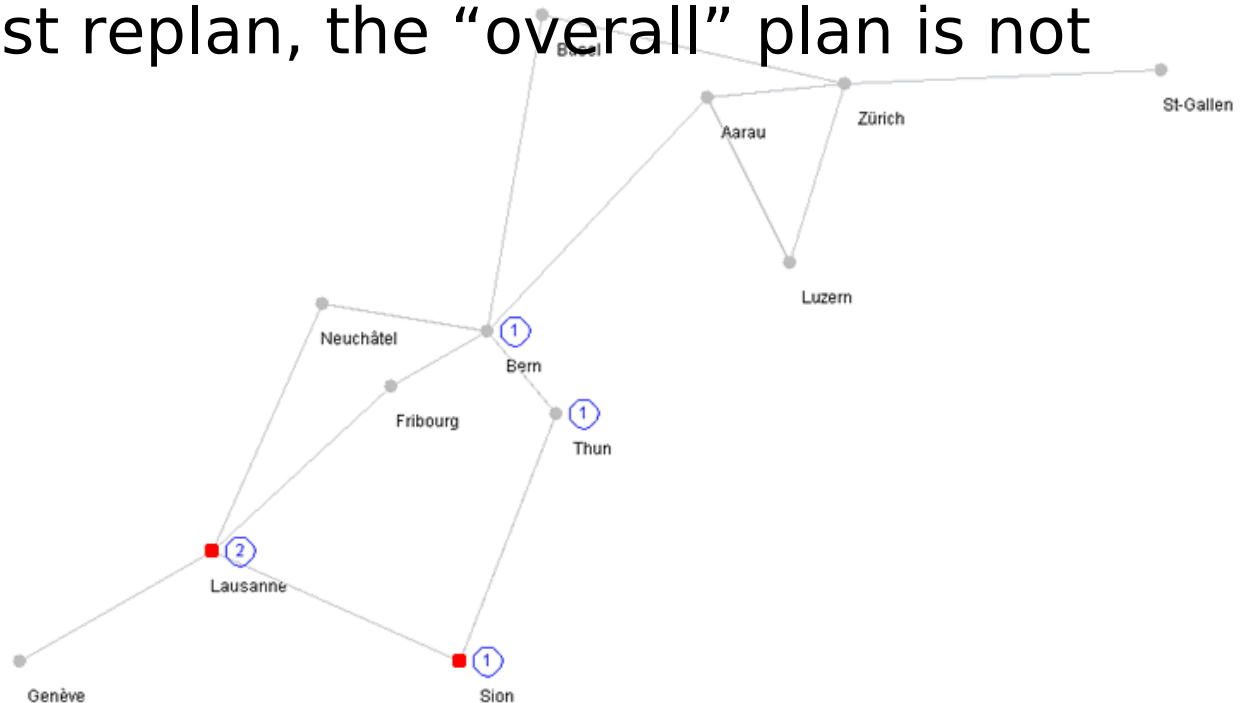
Implementing Vehicles

- As soon as a delivery city is reached:
task is delivered
- Internal planning process:

```
if (current plan is not applicable anymore) {  
  then recompute plan  
}  
execute the plan's next action
```
- > recomputing a plan may be
necessary ...

Delivery Example

- 1 vehicle is able to carry out the intended plan
- 2 vehicles: => interference
 - they must replan, the “overall” plan is not optimal



TO DO

- **Representation for the states, transitions and goals (or final states)**
- Implement breadth-first search and a **state-based A* search algorithm**
- Implement the **deliberative agent**
- Simulate **1** and **2** deliberative agents

Deliverable

- Check Moodle for deadlines
- 100 points

