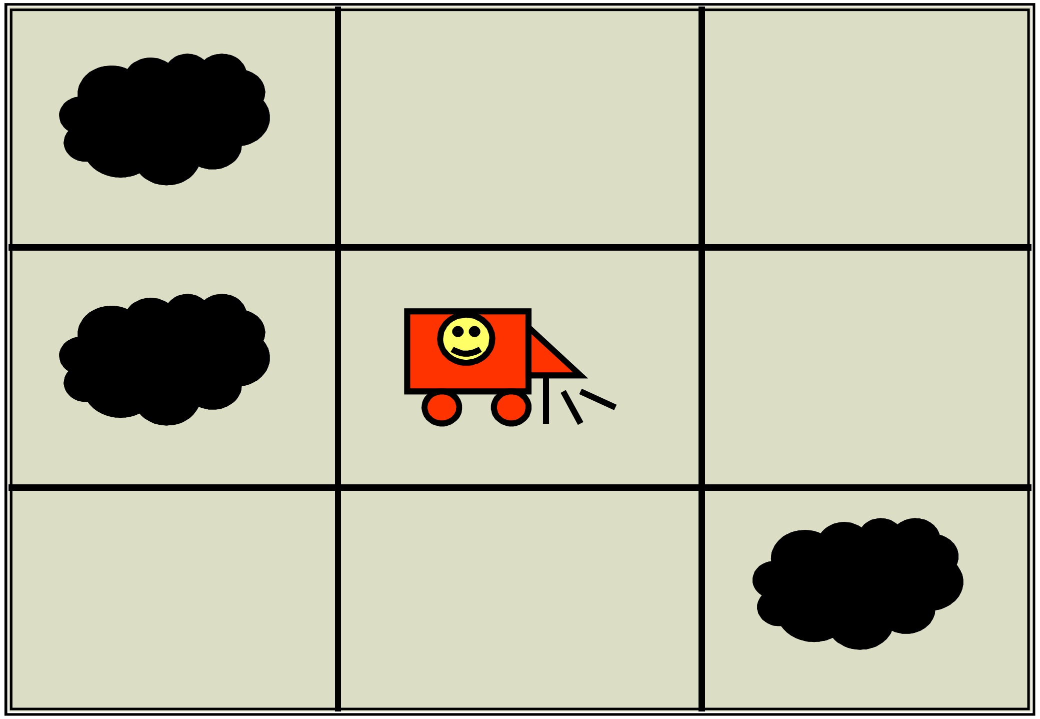
Logic-based architectures: example

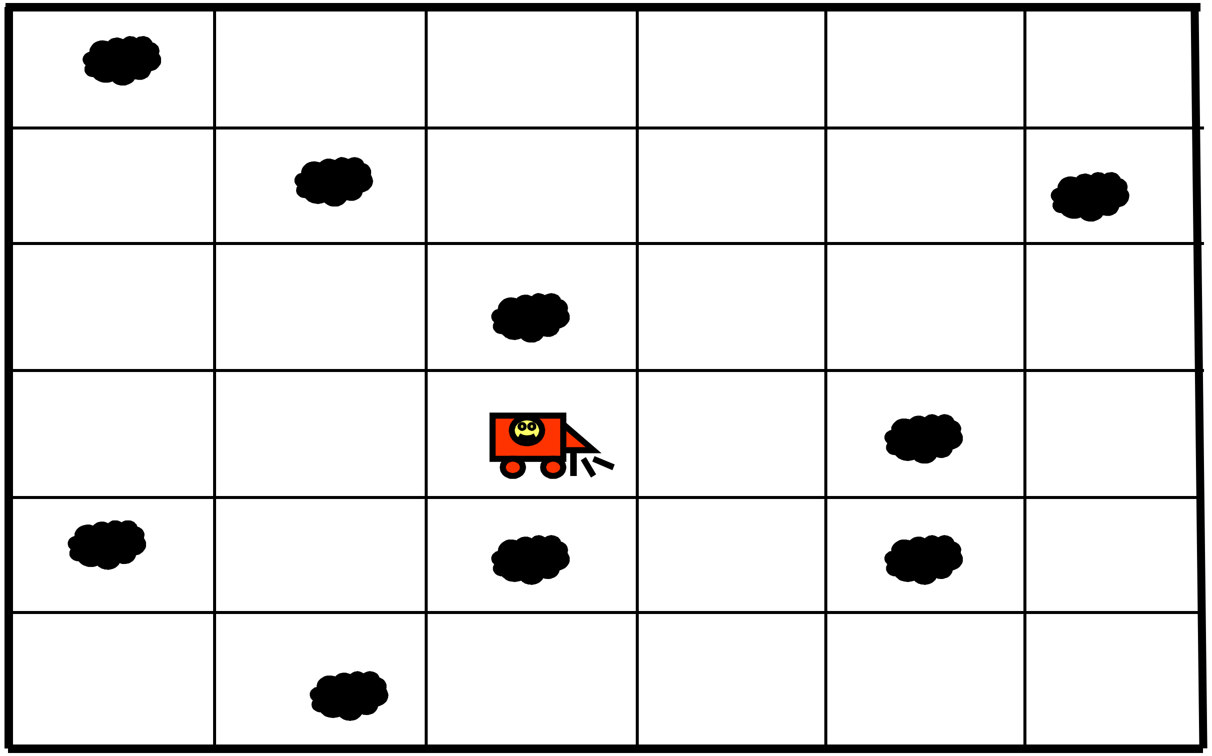
❑ A cleaning robot



- ***In(x,y)*** *agent is at (x,y)*
- ***Dirt(x,y)*** *there is a dirt at (x,y)*
- ***Facing(d)*** *the agent is facing direction d*
- ***∀x,y (¬ Dirt(x,y)) – goal***
- ***Actions:***
  - ***change_direction***
  - ***move_one_step***
  - ***suck***

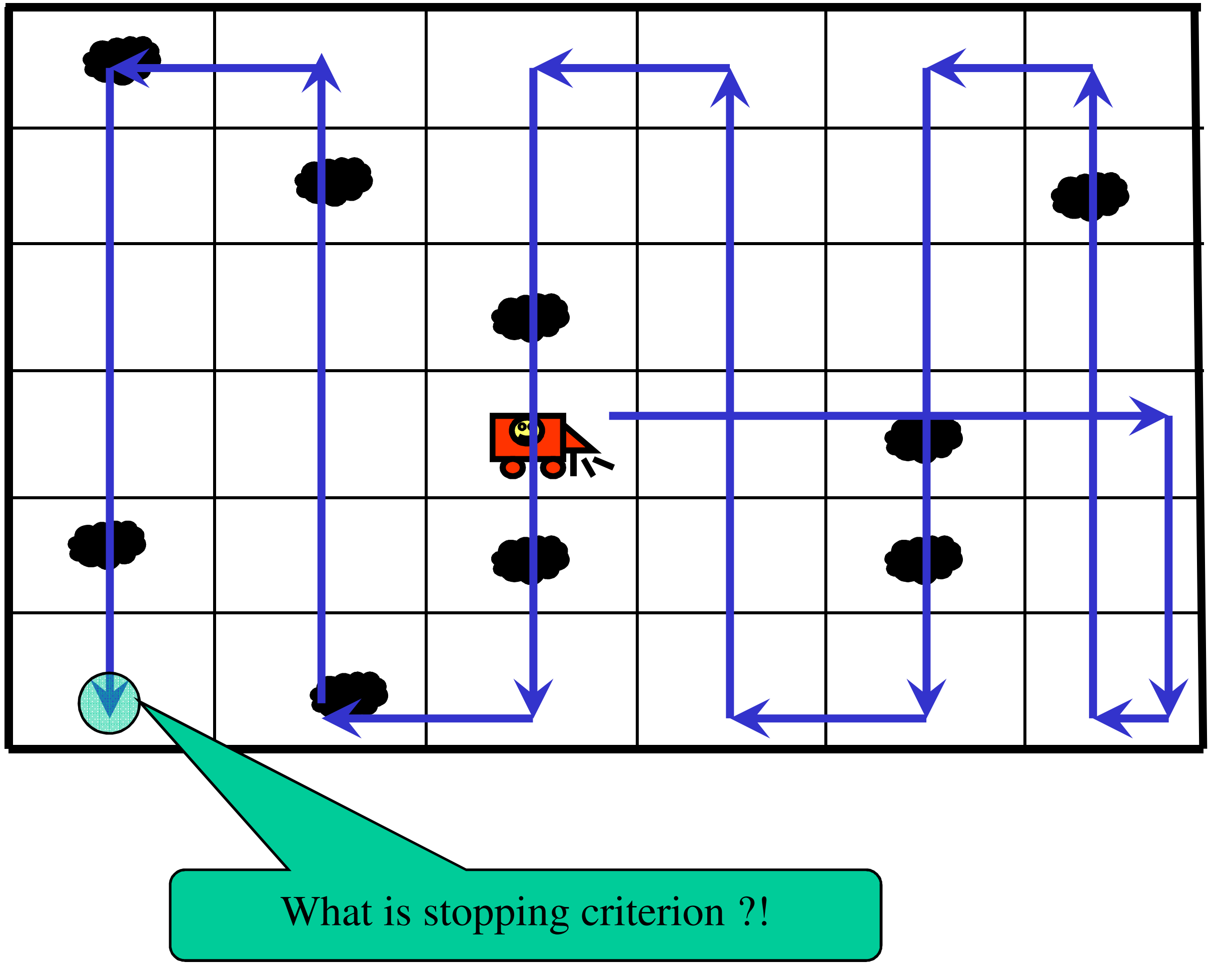
Logic-based architectures: example

- What to do ?



Logic-based architectures: example

- ❑ Solution

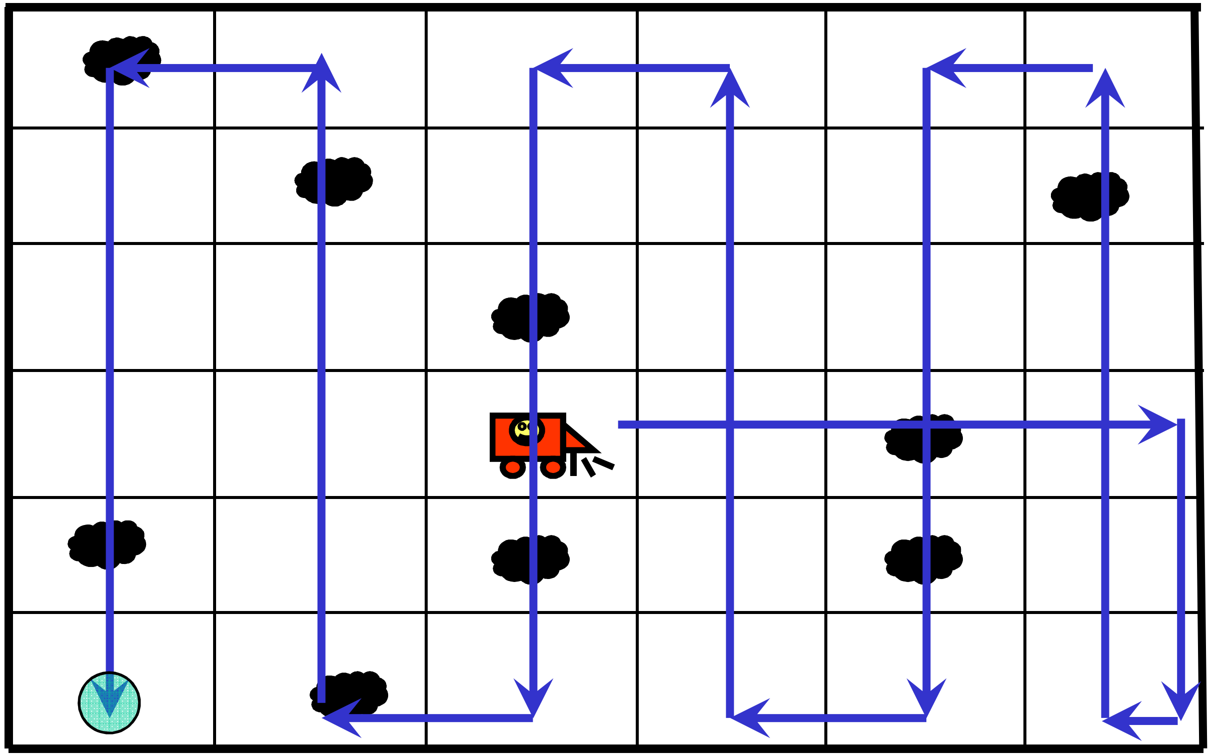


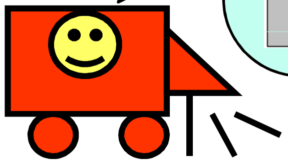
What is stopping criterion ?!

```
start
        // finding corner
continue while fail  { do move_one_step}
do change_direction
continue while fail  {do move_one_step}
do change_direction
         finding corner //
         // cleaning
continue  {
      remember In(x,y) to Mem
      do change_direction
      continue while fail      {
           if Dirt(In(x,y)) then suck
           do move_one_step   }
      do change_direction
      do change_direction
      do change_direction
      continue while fail      {
          if Dirt(In(x,y)) then suck
        do move_one_step   }
      if In(x,y) equal Mem then      stop
               }
       cleaning //
```

Logic-based architectures: example

- ❑ is that intelligent?



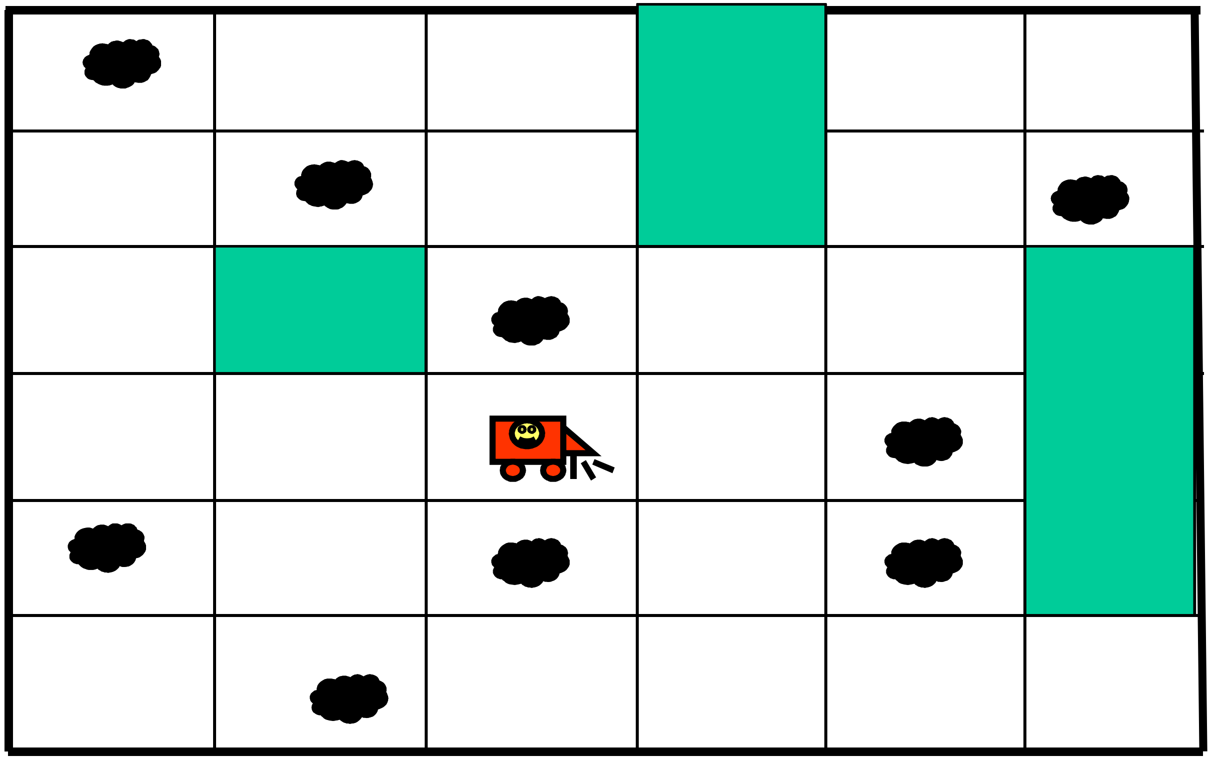


```
start
        // finding corner
continue while fail { do move_one_step}
do change_direction
continue while fail {do move_one_step}
do change_direction
        finding corner //
        // cleaning
continue {
    remember In(x,y) to Mem
    do change_direction
    continue while fail    {
        if Dirt(In(x,y)) then suck
        do move_one_step  }
    do change_direction
    do change_direction
    do change_direction
    continue while fail    {
        if Dirt(In(x,y)) then suck
        do move_one_step  }
    if In(x,y) equal Mem then       stop
            }
    cleaning //
```

How to make our agent capable to “invent” (derive) such a solution (plan) autonomously by itself ?!

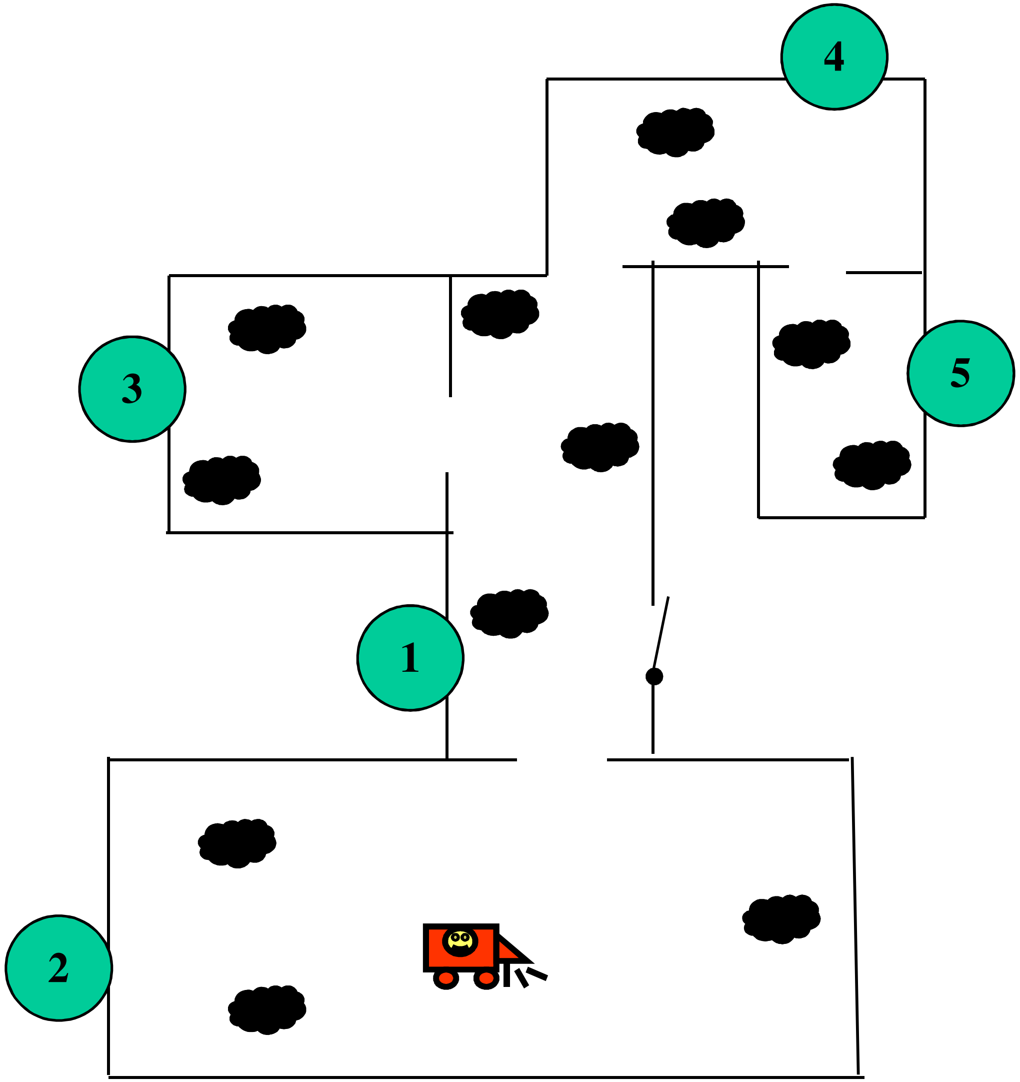
Logic-based architectures: example

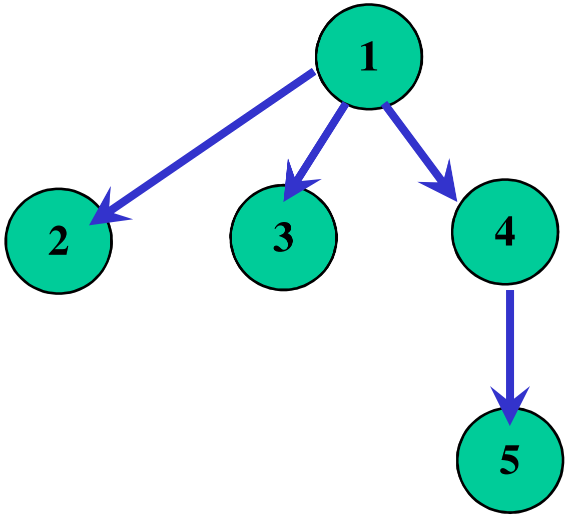
- ❑ Looks like previous solution will not work here. What to do ?



Logic-based architectures: example

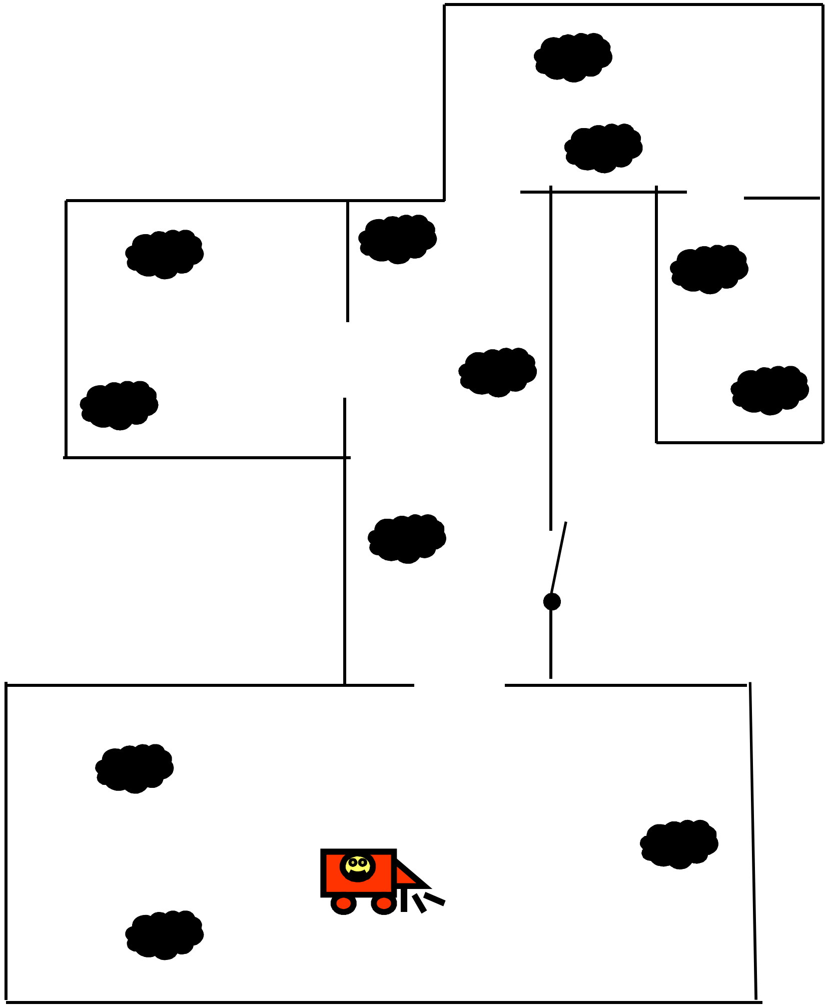
- ❑ What to do now??





Restriction: a flat has a tree-like structure of rectangle rooms !

ATTENSION: Course Assignment !



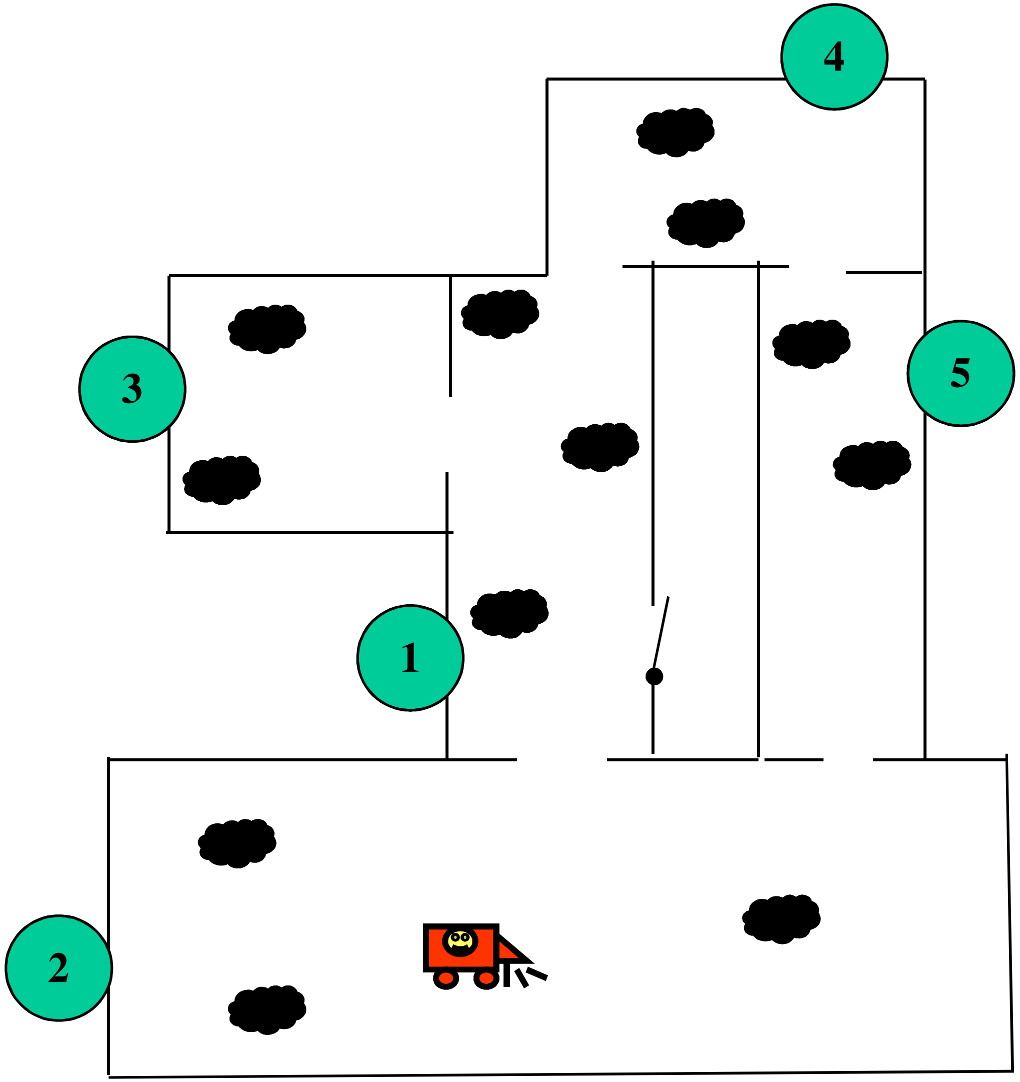
- To get 5 ECTS and the grade for the TIES-453 course you are expected to write 5-10 pages of a free text **ASSIGNMENT** describing how you see a possible approach to the problem, example of which is shown on the picture: (requirements to the agent architecture and capabilities (as economic as possible); view on agent’s strategy (or/and plan) to reach the goal of cleaning free shape environments); conclusions

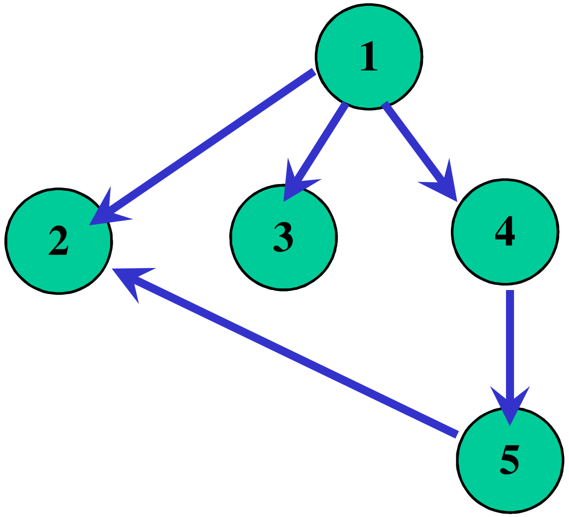
Assignment: Format, Submission and Deadlines

- ❑ Format: Word (or PDF) document;
- ❑ Deadline - 15 March of this year (24:00);
- ❑ Files with the assignment should be sent by e-mail to Vagan Terziyan (vagan@jyu.fi);
- ❑ Notification of evaluation - until 30 March;
- ❑ You will get 5 credits for the course;
- ❑ Your course grade will be given based on originality and quality of this assignment;
- ❑ **The quality of the solution will be considered much higher if you will be able to provide it in the context of the Open World Assumption and agent capability to create a plan!**

Logic-based architectures: example

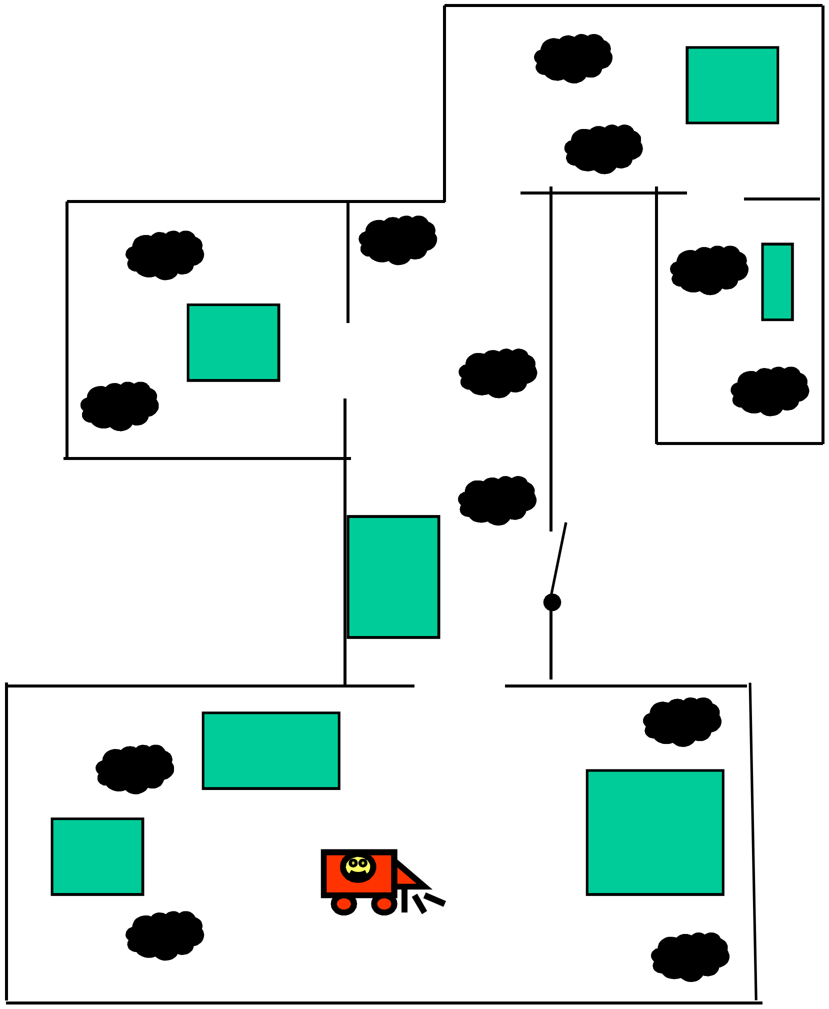
- ❑ What to do now??





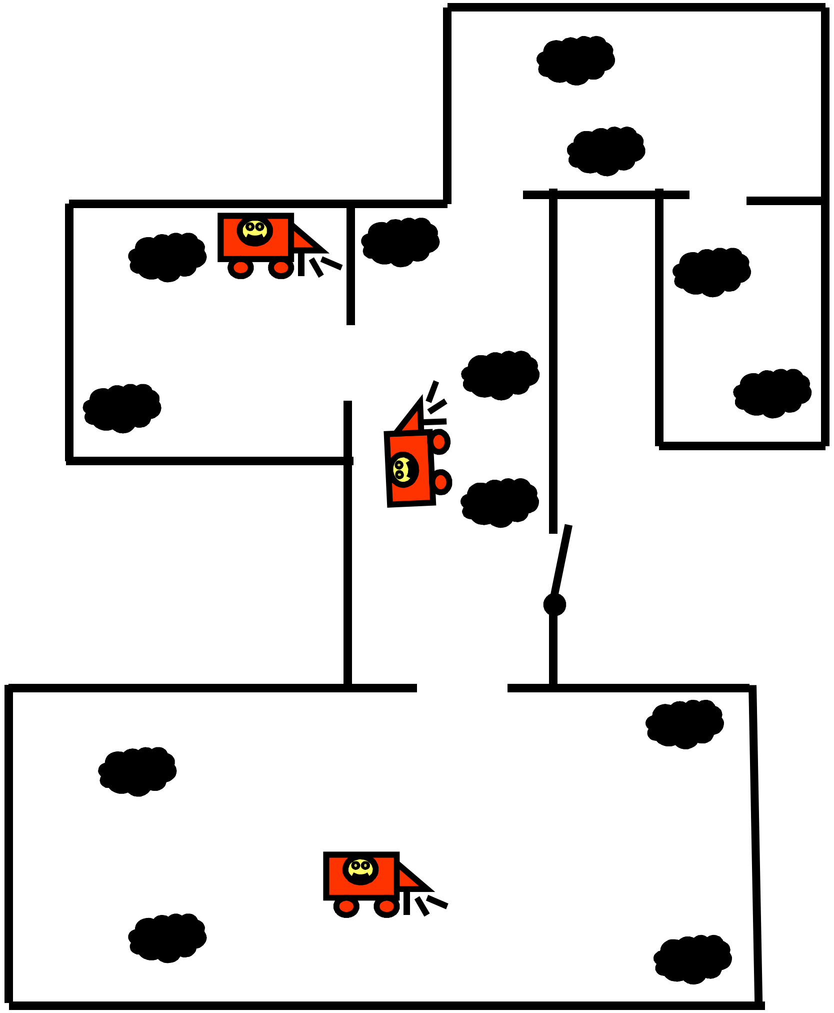
Logic-based architectures: example

- What now???



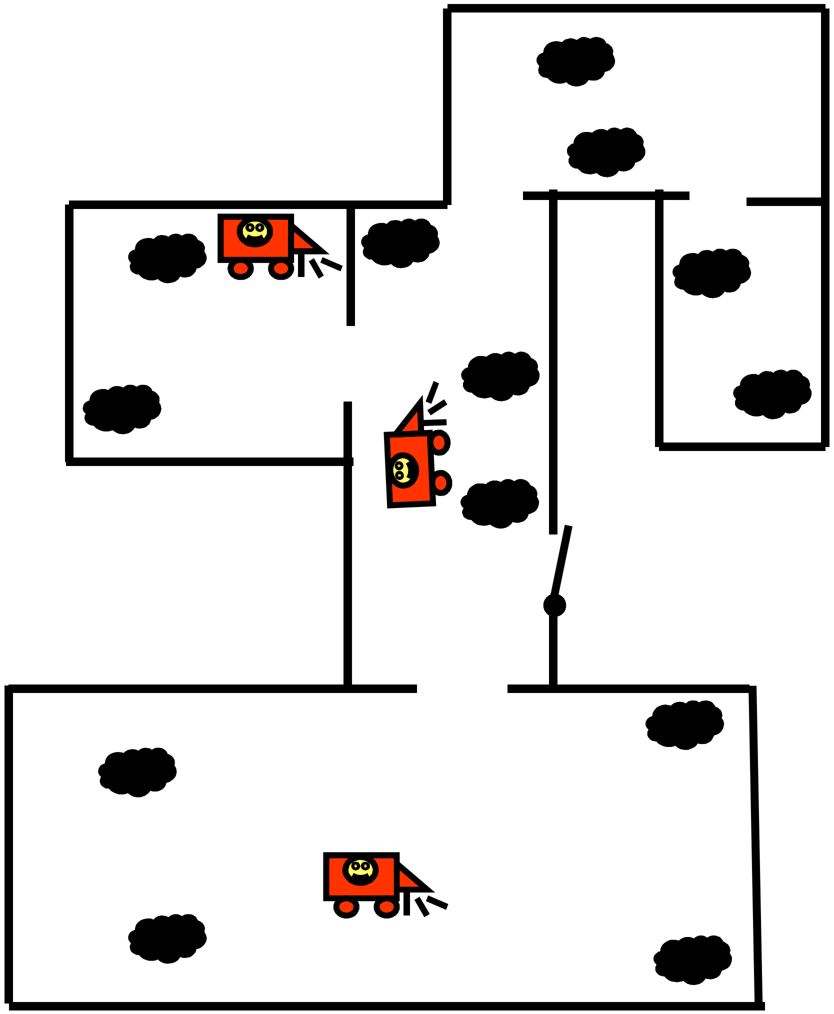
Logic-based architectures: example

❑ or now … ??!



Logic-based architectures: example

2 extra credits !

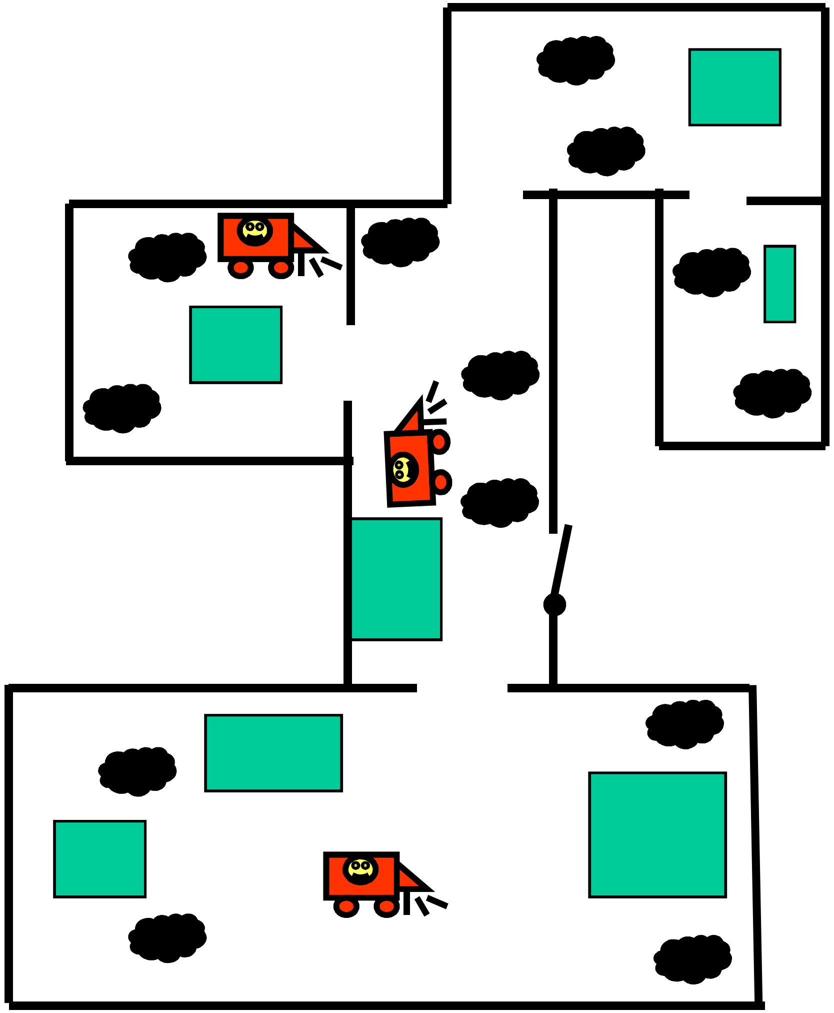


To get 2 ECTS more in addition to 5 ECTS and get altogether **7 ECTS** for the TIES-453 course you are expected to write extra 2-5 pages within your **ASSIGNMENT** describing how you see a possible approach to the problem, example of which is shown on the picture: (requirements to the agent architecture and capabilities (as economic as possible); view on agents' collaborative strategy (or/and plan) to reach the goal of collaborating cleaning free shape environments); conclusions.

IMPORTANT ! This option of 2 extra credits is applied only to those who registered only to this TIES-453 course and not registered to TIES-454 course

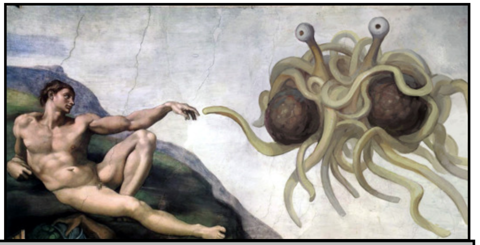
Logic-based architectures: example

❑ or now … ??!

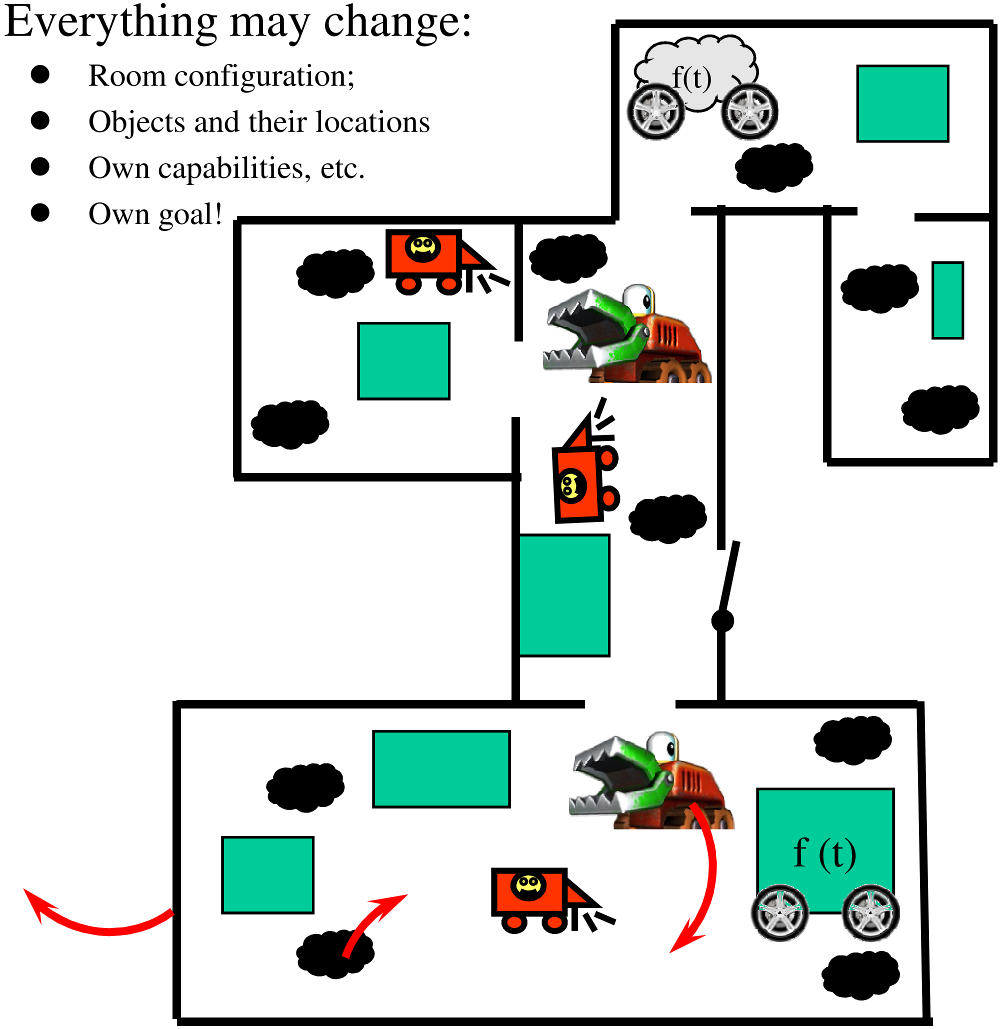


Logic-based architectures: example

Open World Assumption

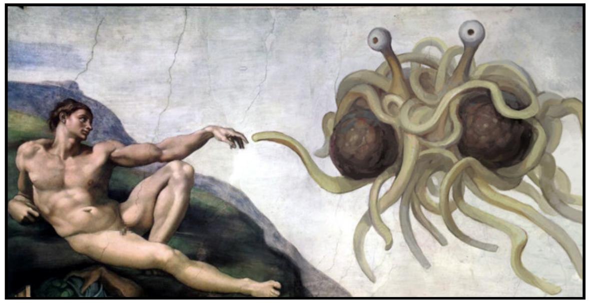


- ❑ Now … ???!!!!!!
- ❑ Everything may change:
  - ● Room configuration;
  - ● Objects and their locations
  - ● Own capabilities, etc.
  - ● Own goal!

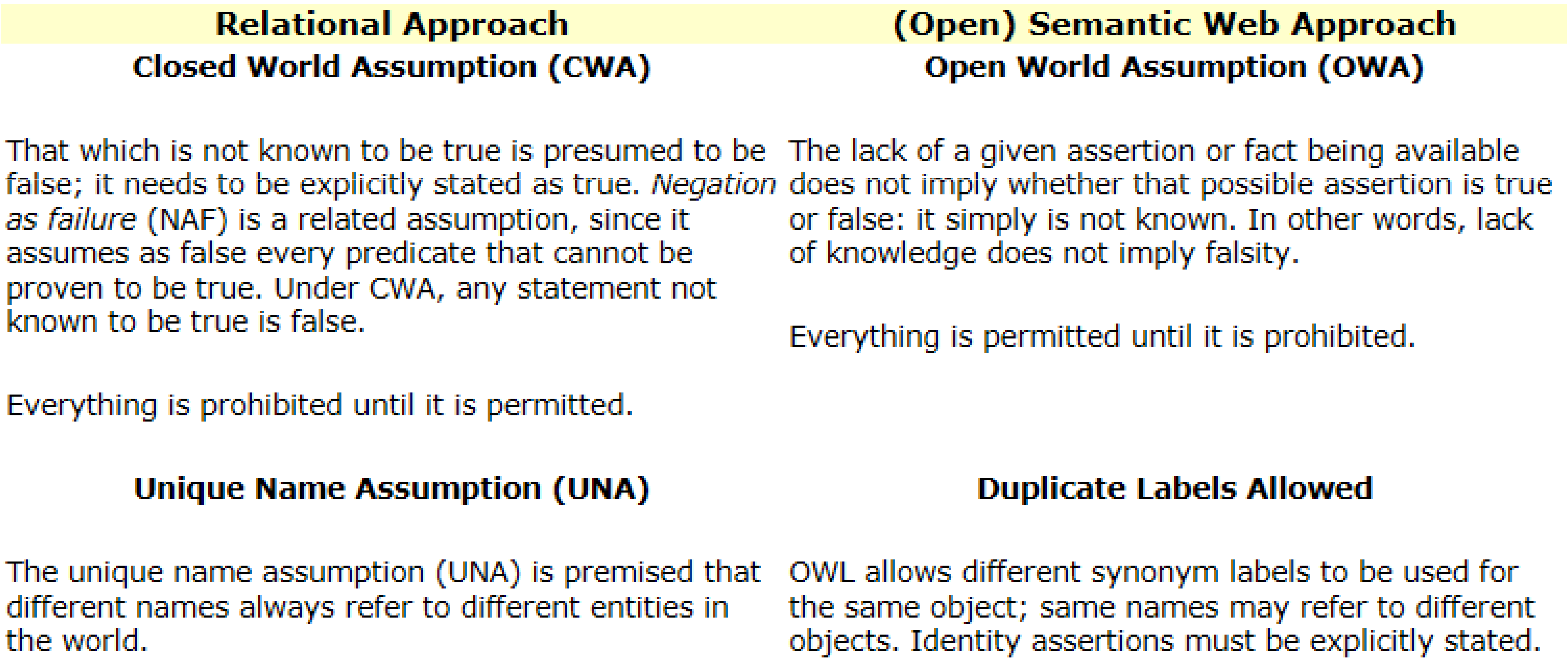


❑ When you will be capable to design such a system, this means that you have learned more than everything you need from the course "Design of Agent-Based Systems"

The Open World Assumption (1)



The Open World Assumption (OWA): a lack of information does not imply the missing information to be false.



Relational Approach	(Open) Semantic Web Approach
Closed World Assumption (CWA)	**Open World Assumption (OWA)**
That which is not known to be true is presumed to be false; it needs to be explicitly stated as true. *Negation as failure* (NAF) is a related assumption, since it assumes as false every predicate that cannot be proven to be true. Under CWA, any statement not known to be true is false.	The lack of a given assertion or fact being available does not imply whether that possible assertion is true or false: it simply is not known. In other words, lack of knowledge does not imply falsity.
Everything is prohibited until it is permitted.	Everything is permitted until it is prohibited.
Unique Name Assumption (UNA)	**Duplicate Labels Allowed**
The unique name assumption (UNA) is premised that different names always refer to different entities in the world.	OWL allows different synonym labels to be used for the same object; same names may refer to different objects. Identity assertions must be explicitly stated.

http://www.mkbergman.com/852/