Active Execution Monitoring Using Planning and Semantic Knowledge

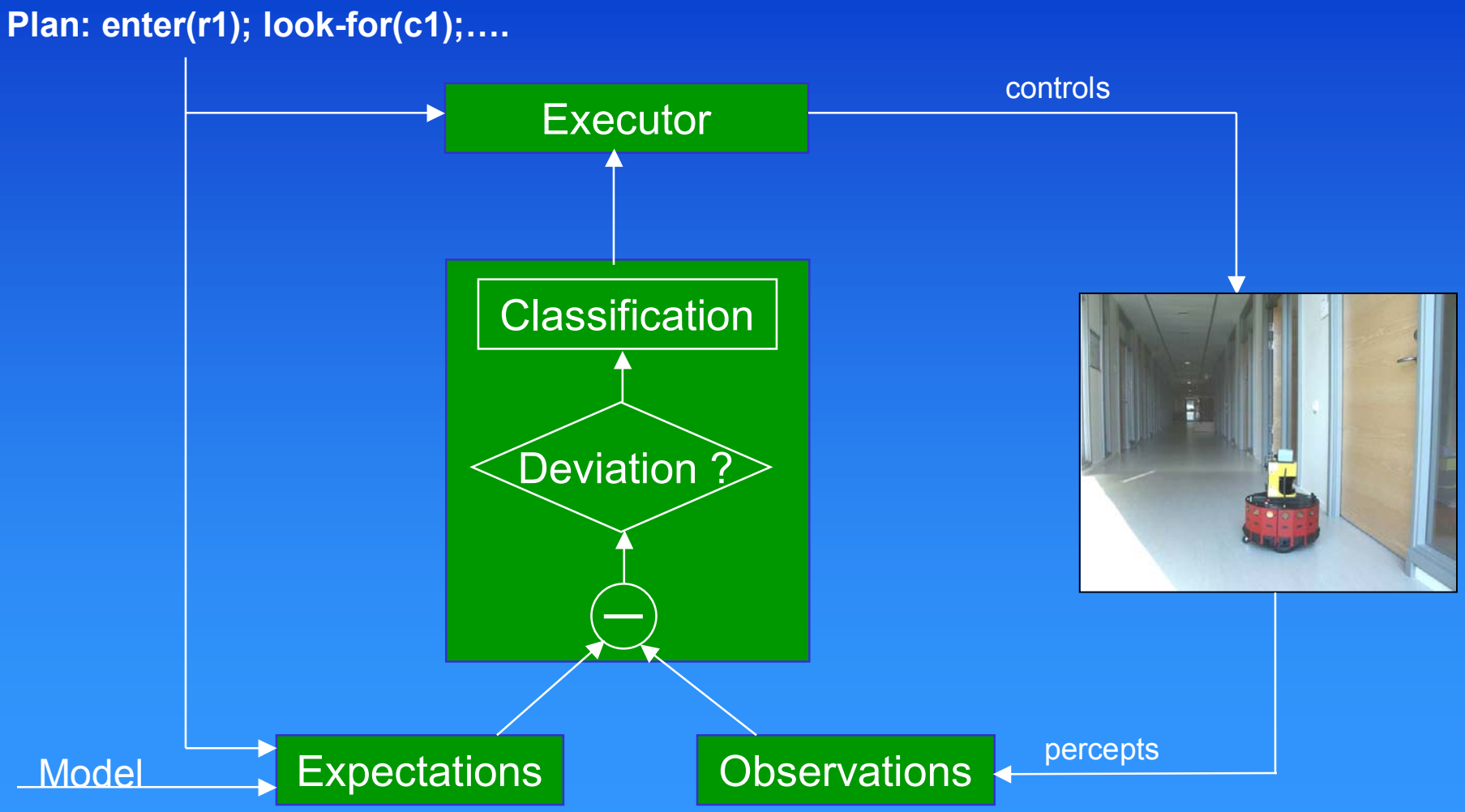
A. Bouguerra & **L. Karlsson** & A. Saffiotti
The Mobile Robotics Lab @ AASS
Örebro University
Sweden

ÖREBRO UNIVERSITY
DEPARTMENT OF TECHNOLOGY

Outline

- Overview of Execution Monitoring
- Semantic Knowledge
- The Proposed Framework
- Active Monitoring
- Test Scenarios
- Conclusions

Overview of Execution Monitoring



Overview of Execution Monitoring

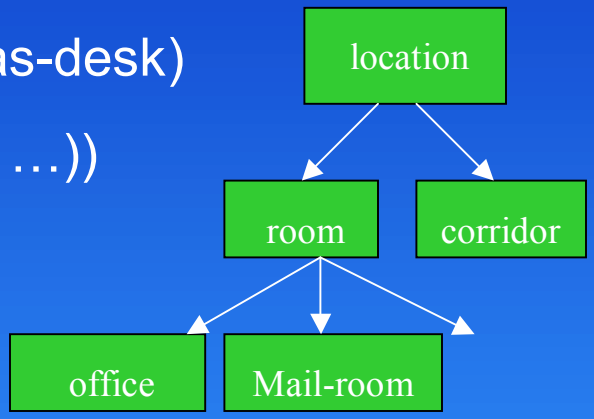
- Expectations to monitor: from action model in planning domain
  action: (enter ?r)
  precond: (and (robot-in in ?r2) (connected ?r2 ?r))
  effect: (robot-in ?r)
- There might be a lot more information that is useful for monitoring, but we don't want it to burden our planning domain
- Use semantic knowledge!

Semantic Knowledge

- Description Logics: A traditional AI tool
- Knowledge expressed as:

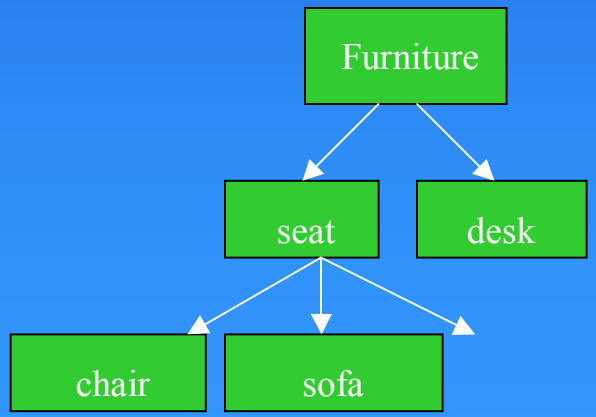
  (defconcept office :is (:and room (:at-least 1 has-desk)
  (:at-most 0 sink) …))

  (defrelation has-desk :domain location
  :range desk)



- Real world description
  - (tell (room r1))
  - (tell (has-desk r1 dsk1)),…
- Inferring implicit knowledge

  (room r1), (has-desk r1 dsk1),… ⇒ (office r1)



The Proposed Framework

- Identify the assertions to monitor from action model
  - (enter room1) $\Rightarrow$ (robot-at room1)
- Derive implicit expectations from semantic knowledge
  - (living-room room1) $\Rightarrow$ (room room1) & (at-least 1 has-desk room1) &...
- Check implicit expectations using available perceptual Information
  - If all expectations are verified, then success
  - If at least one expectation is violated, then failure
  - **If there are expectations with unknown truth values, then more information is needed**

The Proposed Framework

- If more information is needed, then plan to gather it (active monitoring)
- Sensor-based planning problem
- Actions for checking various forms of expectations as well as for movement, etc.
- Active monitoring steps:
  - generate current belief state and goal
  - generate an information gathering plan
  - execute the information gathering plan

Active Monitoring: actions

- Actions for checking number constraints
  - Observe and count objects and evaluate the corresponding constraint, e.g.,
    (eval-at-least 2 bed r1)
- Actions for checking observable concepts
  - Check whether an object is of a certain type, e.g.,
    (check-corridor c1)
- Actions for movement and manipulation
  - Change position to gather more information, e.g.,
    (move c1-2)

Active Monitoring: state and goal

- The initial belief state contains different hypotheses about the unknown truth values of the constraints, e.g.,
  h1: (at-least 1 bed r1)
  h2: (not (at-least 1 bed r1))
  It also contains topological info etc
- The goal formula specifies that the truth value of the constraints must be known, e.g.,
  (known (at-least 1 bed r1))

Active Monitoring: planning

- PTLPLAN a forward chaining planner
- Search in belief state space
- Plans have the structure of an “if-then-else” program
- Plan actions are executed by translating them into sensori-motoric-cognitive processes
- Branching Conditions are evaluated in the execution-time belief state

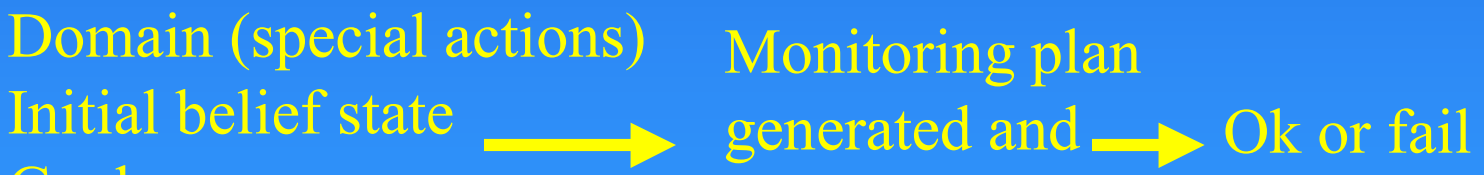
Active monitoring: summary

1) All verified $\Rightarrow$ ok

2) Some violated $\Rightarrow$ fail

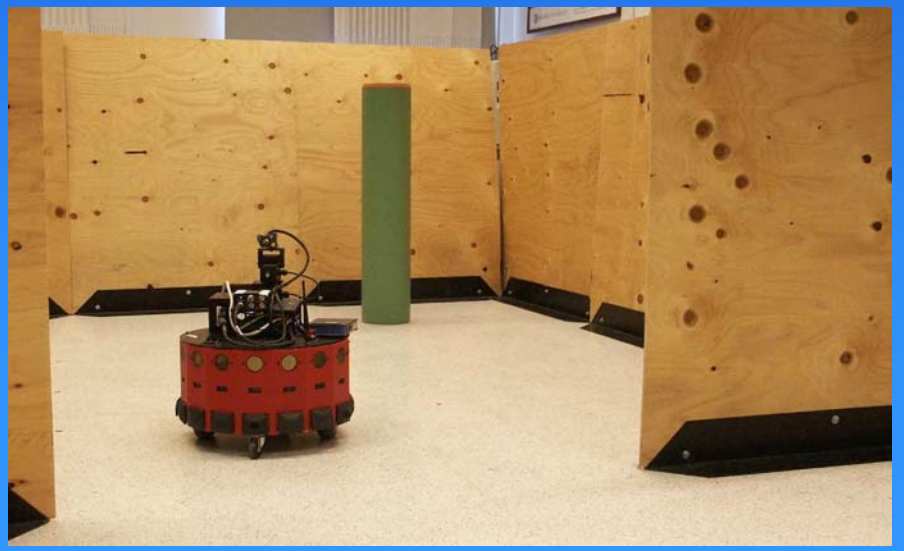
3) Some unknown $\Rightarrow$

Active monitoring



Test Scenarios

- Loom for semantic knowledge
- Navigation Tasks
- Indoor Environment
  - House with rooms
- Magellan Pro mobile robot
- Simplified Vision
  - Simple objects for furniture items
- Crisp approach



Test Scenarios

- Task: clean the living-room
- Plan: (enter r3)(clean r3)
  - Execution of (enter r3)
    ⇒ check expectations
    - explicit: (robot-at = r3)
    - implicit:
      (room cr), (at-least 1 sofa cr),(exactly 0 sink cr), (exactly 1 TV cr)
    - robot observes *room* but cannot determine the truth values of the other concepts
      ⇒ Information gathering needed

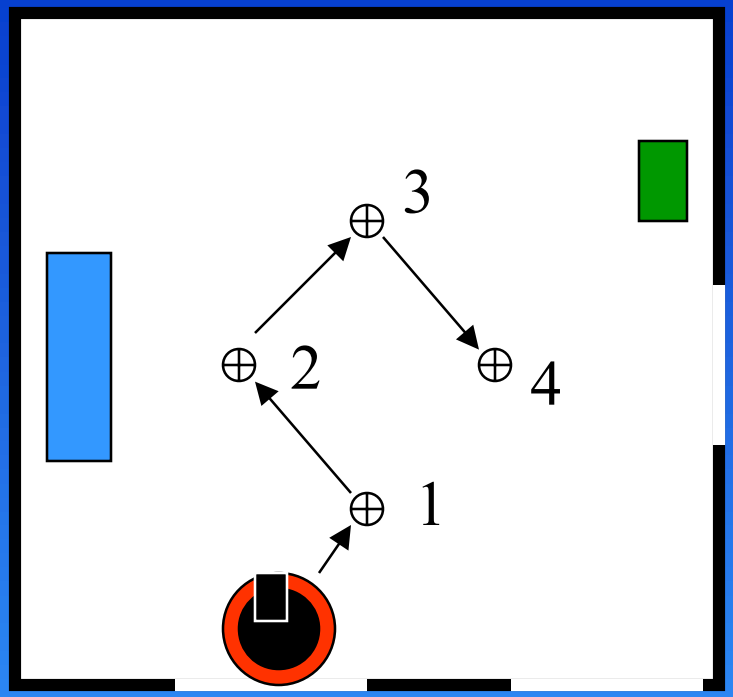
```
(defconcept living-room :is
  (and  room
        (at-least 1 has-sofa)
        (exactly 1 has-tv)
        (exactly 0 has-sink) )

(tell  (living-room r3))
```

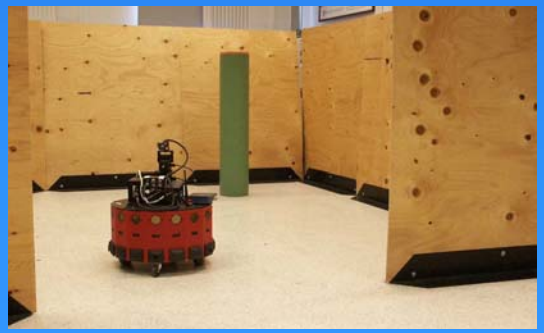
Test Scenarios

- Information Gathering
  - Initial belief state:
    - h1: (at-least 1 sofa cr),(exactly 0 sink cr),(exactly 1 TV cr)
    - h2: (not (at-least 1 sofa cr)), ...
    - h3: (at-least 1 sofa cr),(not (exactly 0 sink cr)),…
    - etc.

    Topological information etc is also included
  - Goal:
    - (known (and (at-least 1 sofa cr)(exactly 0 sink cr)(exactly 1 TV cr))
  - Plan:
    - ((move loc1)(check-exactly 0 sink cr)
      (cond ((not (exactly 0 sink cr)) (:fail))
            ((exactly 0 sink cr) (check-exactly 1 TV cr) …)))

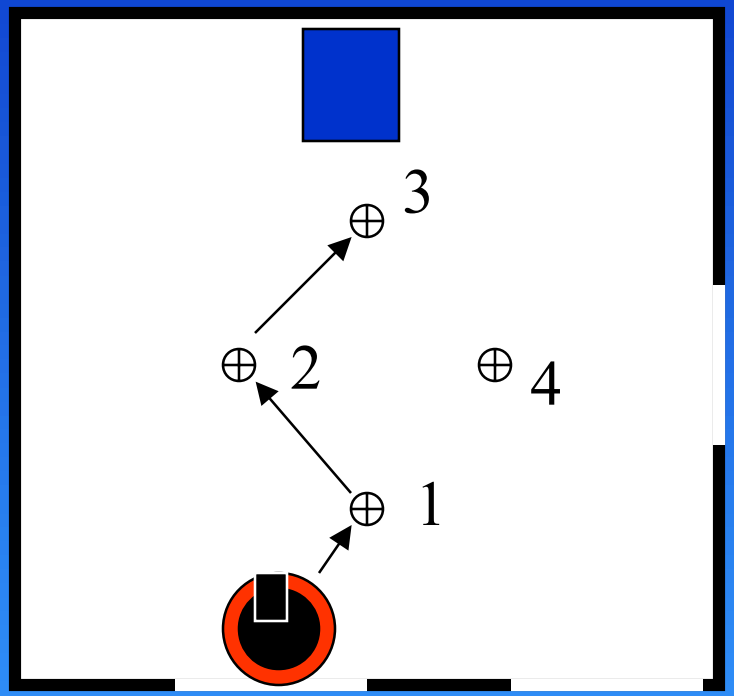
Test scenarios





- Executing the plan: setup #1
  - At point 2: sofa seen, and (at-least 1 has-sofa) verified
  - At point 3: TV seen and counted
  - At point 4: explored entire room, (exactly 1 has-tv) verified, (exactly 0 has-sink) verified
  - Conclusion: correct room

Test scenarios



- Executing the plan: setup #2
  - At point 3: (exactly 0 has-sink) violated
  - Conclusion: wrong room

Conclusions

- Use of semantic knowledge for execution monitoring
- Active execution monitoring: planning for information gathering
- Implemented and tested onboard a mobile robot

Conclusions

- Issues to be addressed:
  - Handling uncertainty (forthcoming paper at IROS-07)
  - Which expectations to check?
  - How to make sure one can continue the original plan (currently: constrained to stay in present room)
  - Deeper experimental evaluation (scalability, etc.)