Model and Graphical Tool to Formalize Human-Robot Interaction Based on Automated Planning

Alba Gragera Joint work with Ángel García-Olaya and Fernando Fernández

uc3m Universidad Carlos III de Madrid

1. Motivation

Social Robotics working in real scenarios must act according to the environment and show flexible and robust behaviors, useful in dynamic and changing situations.

Dealing with the **uncertainty of the real world** makes the development of use cases in Social Autonomous Robotics a complex and time consuming task, becoming a bottleneck for robotics progress.

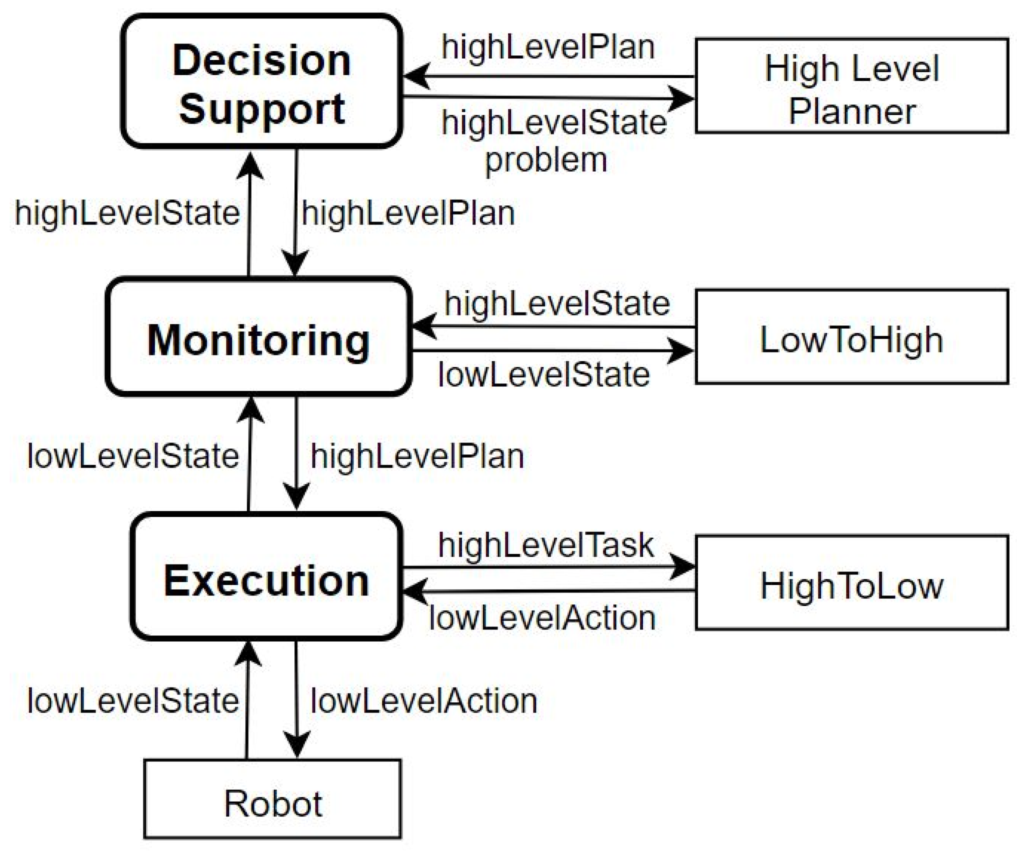
Automated Planning (AP) approaches allow the deployment of robotic platforms through control architectures based monitoring and execution.

2. Objectives

1. Bridge the gap between the domain expert definition of the use case and its formalization
2. Model description of the use case through state transition diagrams
3. Automatic translation from the diagram to the formal PDDL files

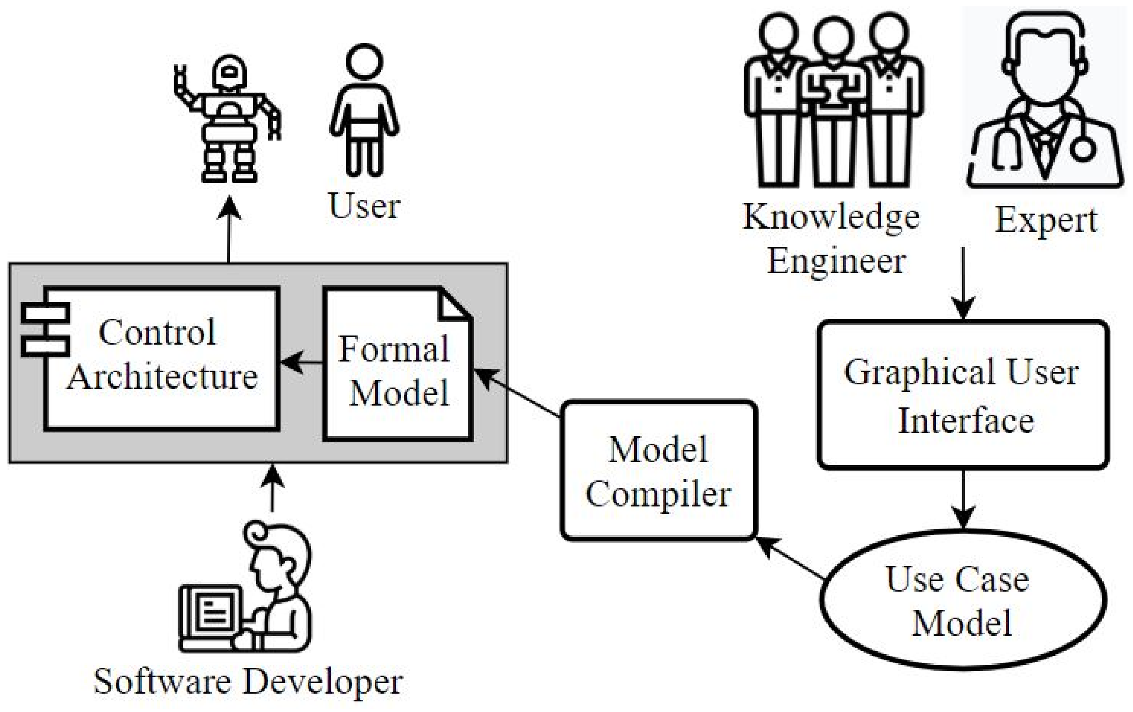
3. Control Architectures

Planning, execution and monitoring architectures are essential for implementation of AP in real Social Robotics. These systems typically involve a planner and a formal planning model to generate the sequence of actions the robot must perform. Once generated, they are executed using the appropriate robot commands, while monitoring the correct execution of the plan.



4. Use Case Definition

Similar to AP, we define a robotic use case as a tuple $P = \{S, A, I, G\}$, where S is the set of possible states, A is the set of actions that the robot can carry out to transit between states, $I \subset S$ defines the initial state, and $G \subset S$ specifies the goals that the use case must meet. These elements have to be specified by the domain expert in collaboration with knowledge engineers and modelled by them in the graphical interface. No further help from software developers will be need to formalize the use case.

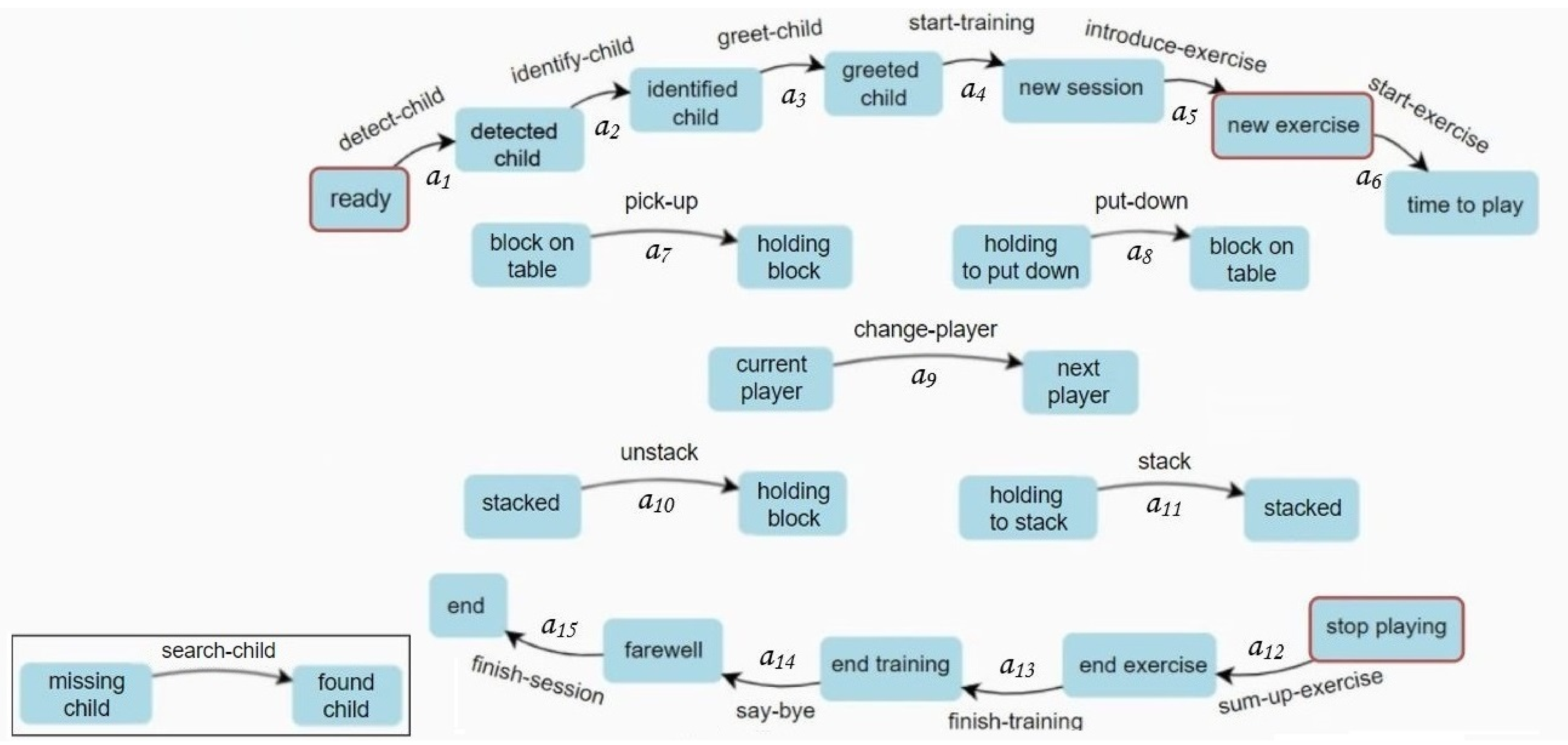


5.Use Case Definition using the concept of *options*

Options are composed by a state where it starts, the policy followed throughout the option, and the state where it terminates. To be able to combine them we also define *partial options*: non-empty sequences of states and actions which may be incomplete, being open to interleave any number of actions to meet intermediate states.

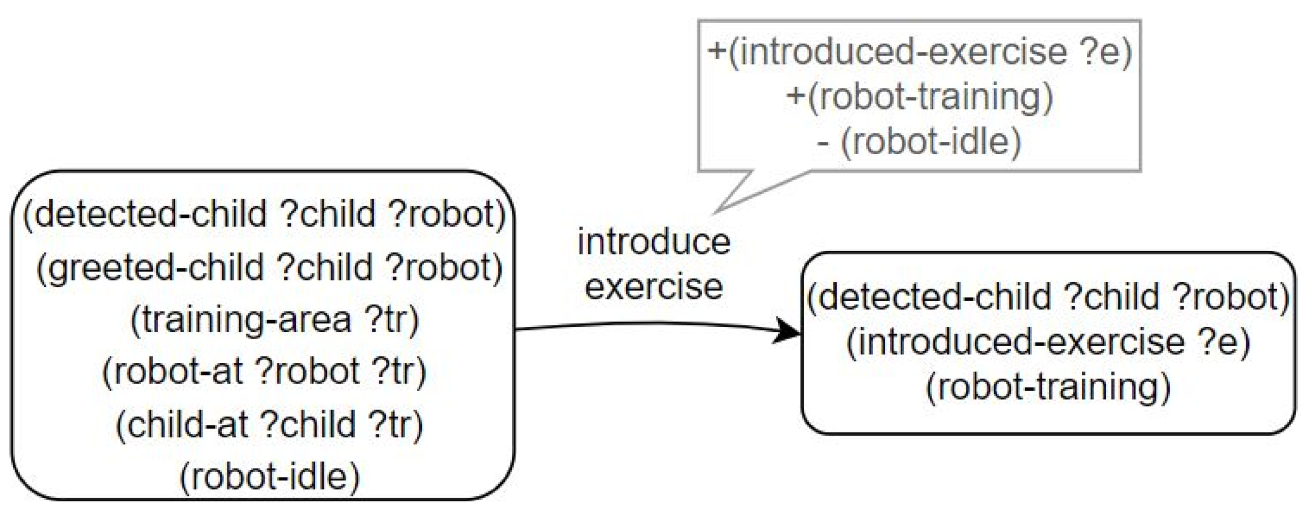
Recovery actions are tasks defined by the expert to deal with unexpected situations (*search-child* action). They will be included by replanning only after such cases.

Checkpoints (highlighted in red) work as restoration points, defined by the expert as places where the execution is desired to return after correcting an exogenous situation, instead of the point where the interruption took place.



6. Use Case Formalization

Every state where an action starts defines the facts that the environment must hold to execute the action, i.e., its preconditions. Formalizations for **sequential connections**, **loops** and **error recovery** are considered in this work.




```
(:action introduce-exercise
  :parameters(?r-robot ?c-child ?tr-location ?e-exercise)
  :precondition (and (detected-child ?c ?r)
                     (greeted-child  ?c ?r)
                     (training-area ?tr) (robot-idle)
                     (robot-at ?r ?tr) (child-at ?c ?tr))
   :effect (and (introduced-exercise ?e) (robot-training)
                (not (robot-idle)))
)
```

7. Conclusions

1. A workflow knowledge representation is a suitable way for modelling tipical Social Robotics problems
2. Automated Planning allows the implementation of Social Robotics use cases by adding extra concepts to achieve better control of the interaction
3. A user interface provided as a support tool to build Social Robotics use cases, which automatically generates the PDDL formalization of the given model.