A graphical tool to generate behaviors

Margaux Lhommet* and Domitile Lourdeaux

Laboratoire Heudiasyc, Université de Technologie de Compiègne (UTC), France

31st Soar Workshop, June 13-17 2011, Ann Arbor, Michigan

8 June 2011

SIMADVF

Objective, partners and funds

- Train nannies to everyday life dangers (risk prevention and response in case of an accident).
- Heudiasyc UMR CNRS 6599 UTC, Virtuofacto, AFPA.
- Funded by DGCIS

Virtuofacto©

Graphical tool to specify behavior

What we have : Visual Hawai

- Formalism created with and used by ergonomists to model real human activity in work situations.
- Specify preconditions and postconditions of actions.
- Specify links between tasks in a tree task.
- Used by the computer programmer to program the behaviors.

What I need

- Automatically generate the behavioral rules (in Soar !).
- Specify the objects of the simulation.
- Specify the interactions.

Cognitive paragigm : Enactivism

Definition

The humans organize themselves by interacting with their environment.
Enactivism is related to situated cognition and embodied cognition.

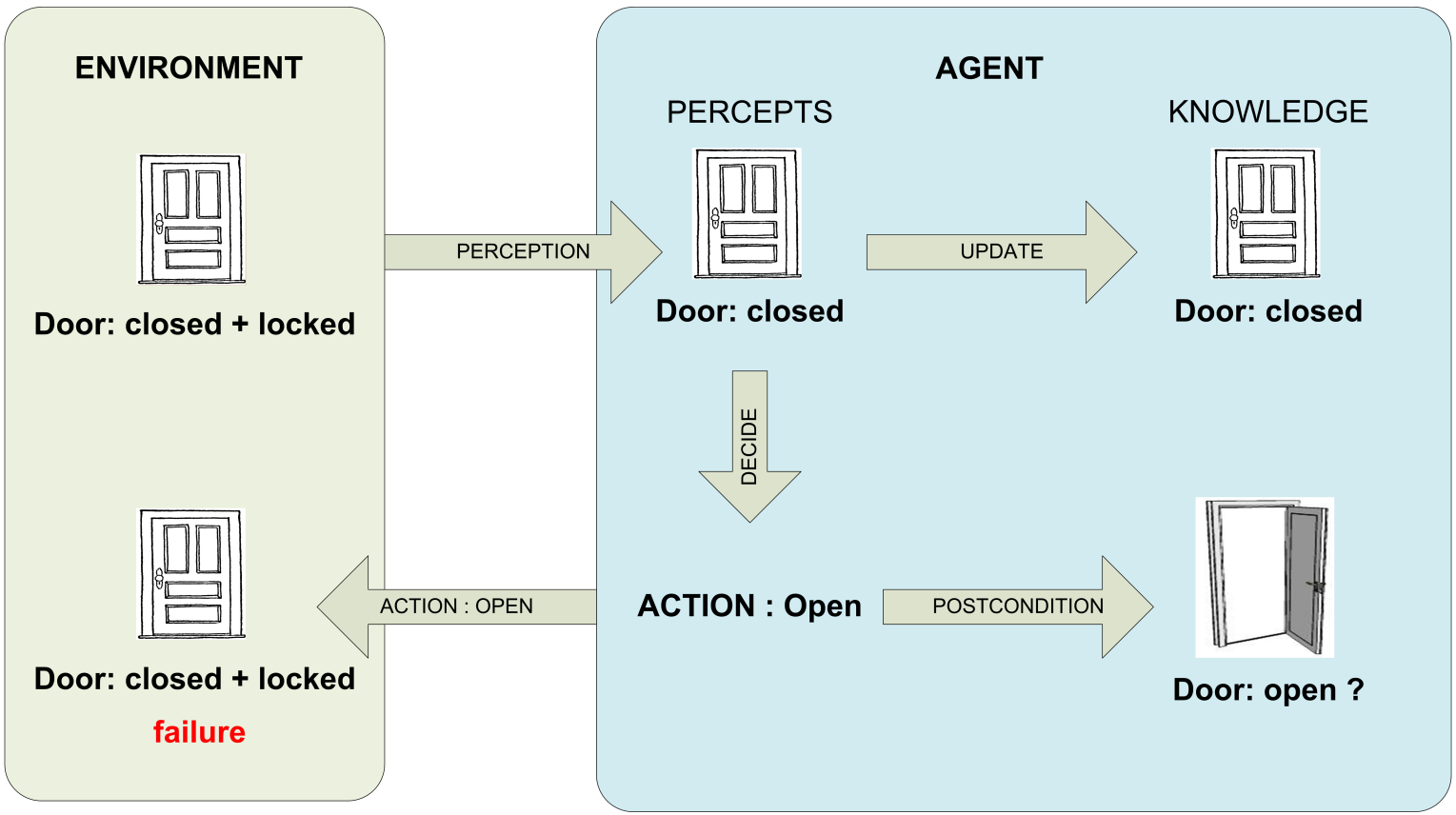
Proposal

- Affordance : what we can do is written in the object.
- Knowledge : the knowledge about the world is created by agents themselves when they interact, and not transmitted by the perception module.

A concrete example

Inappropriate action

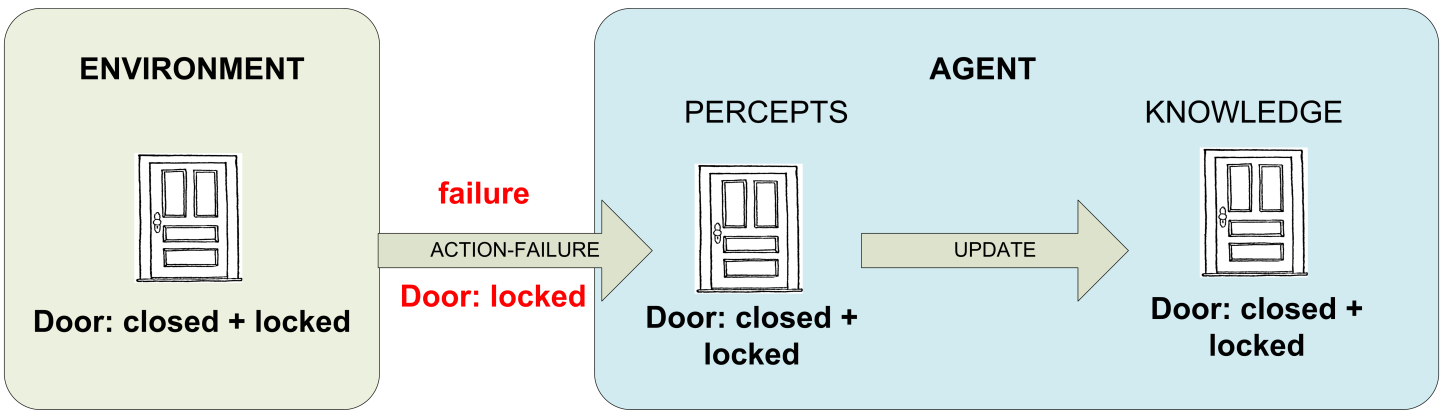
Uncomplete knowledge about the world leads to inappropriate actions. How does the system react ?



The classical approach

What happens

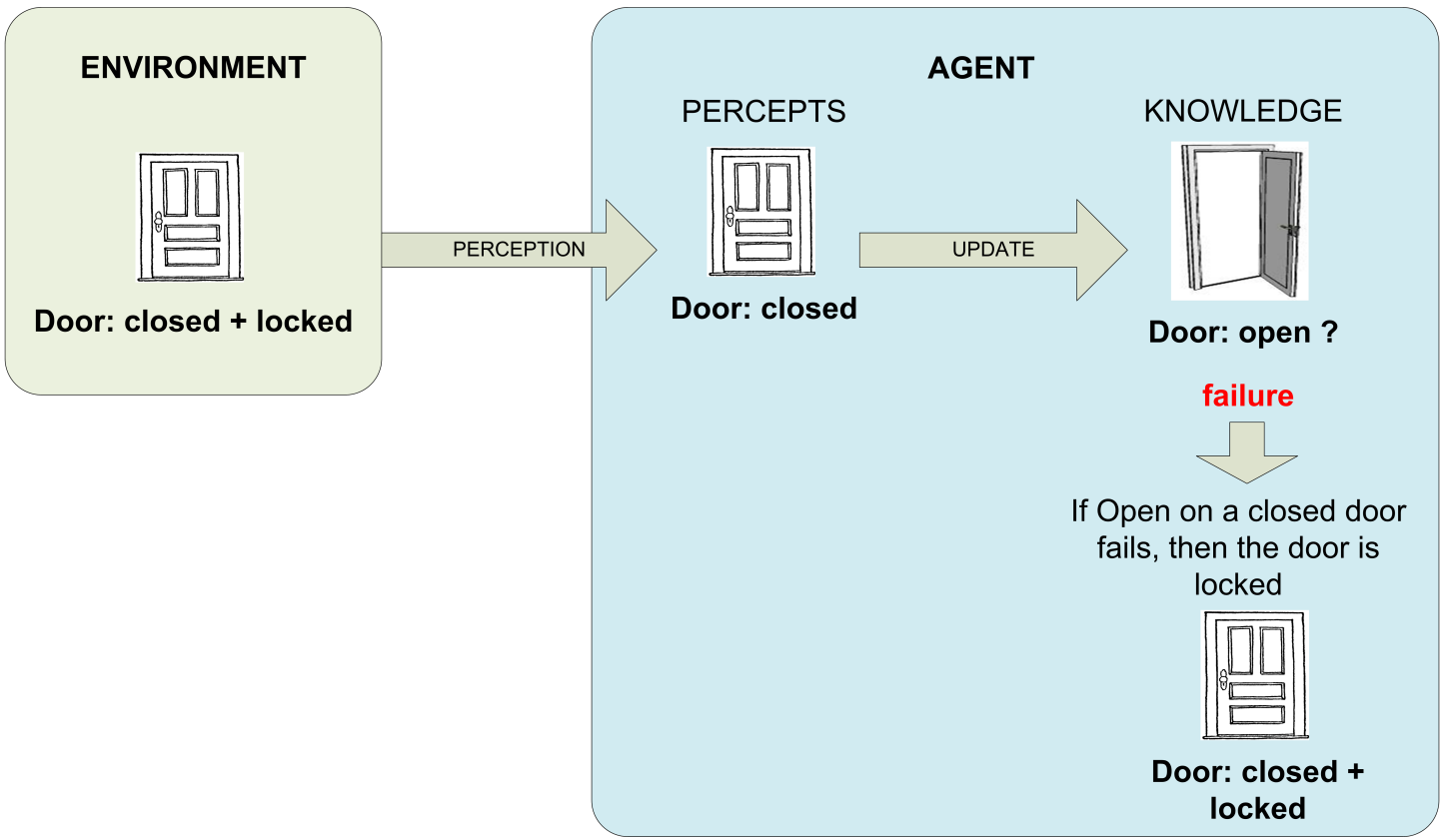
The environment explains to the agent the reasons why it did not work.



The enactive approach

What happens ?

The door is not in its supposed state. The agent uses its knowledge to imagine what went wrong.



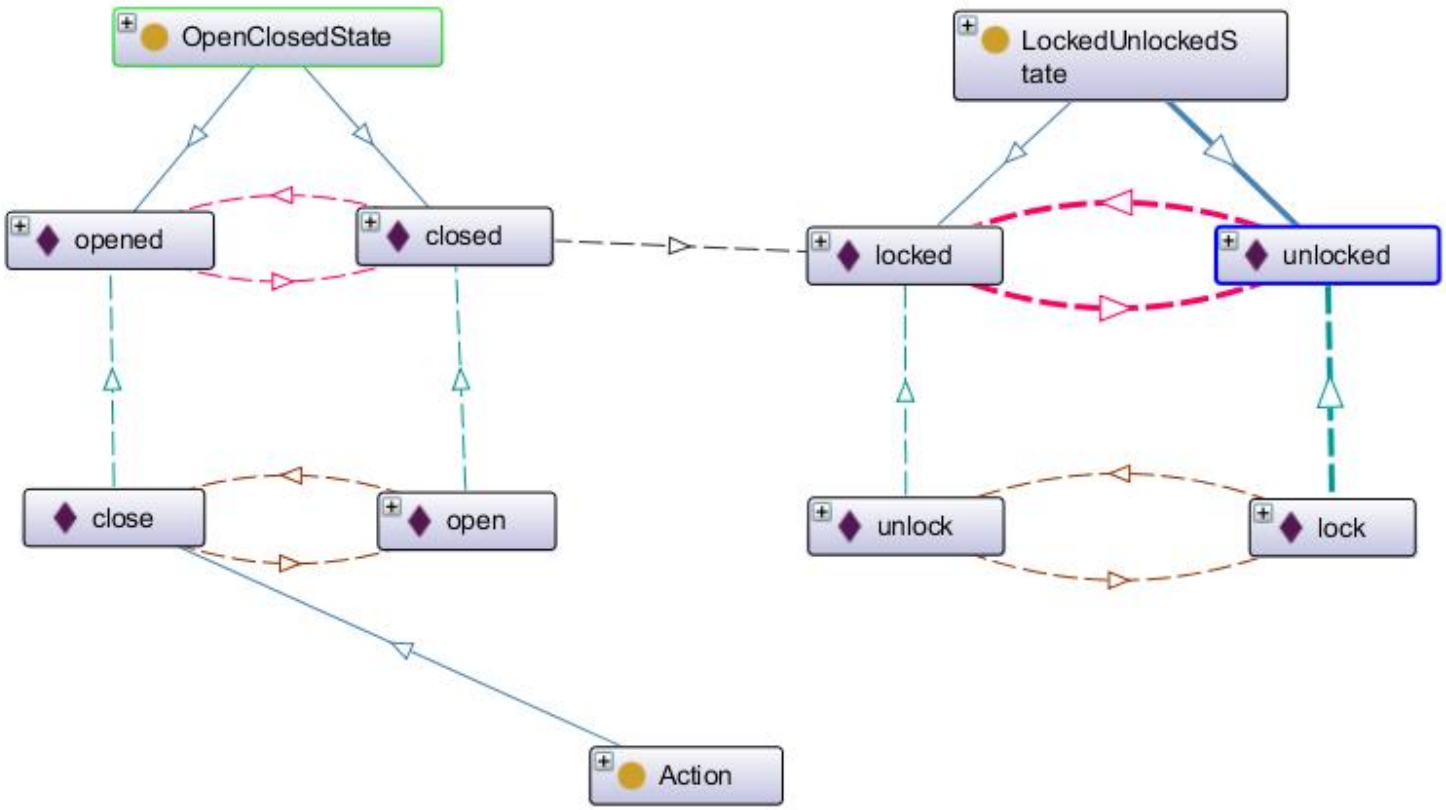
Ontology contents

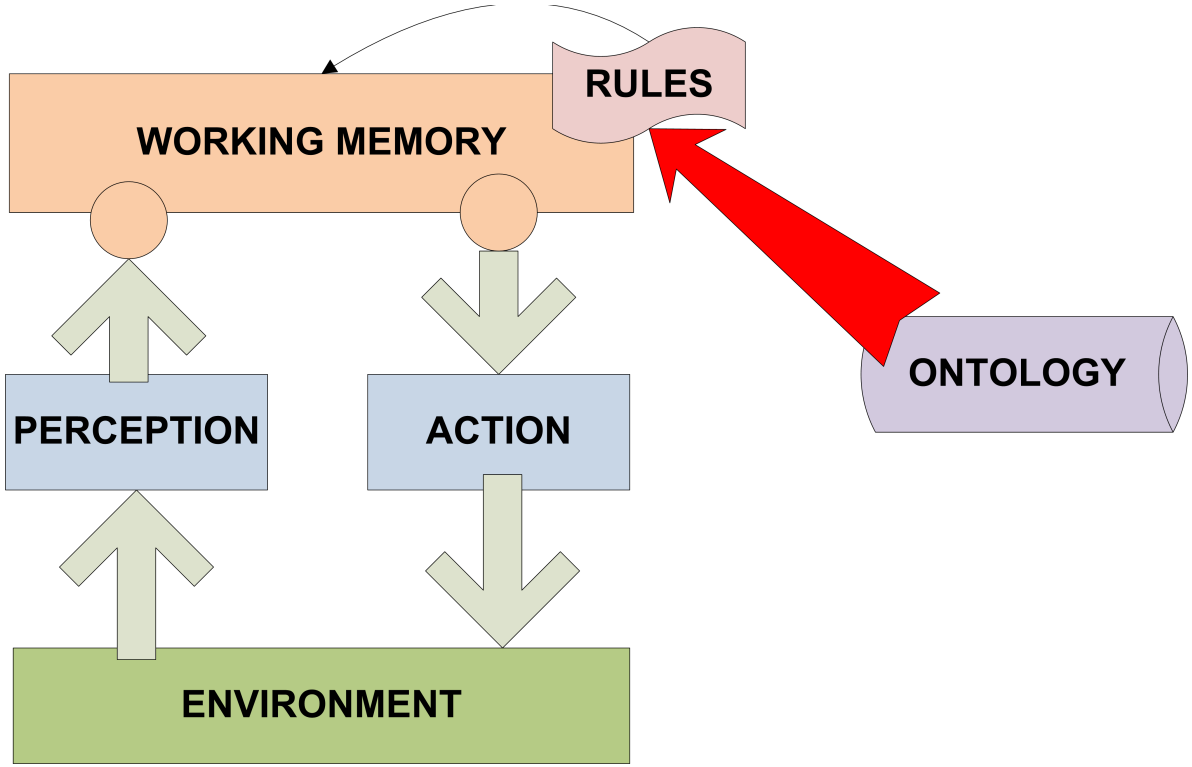
- Objects
- States
- Actions

Ontology relations

- Objects have states
- Actions change the state of a target
- States have failure states to infer knowledge when the related action fails.

Example


OpenClosedState
LockedUnlockedS
tate
opened
closed
locked
unlocked
close
open
unlock
lock
Action



What is done

Automatic generation of Soar rules for binary-states (open/close) ; actions with resources (unlock a door with a key) ; many changed states in post-condition (push a button to turn on a light).

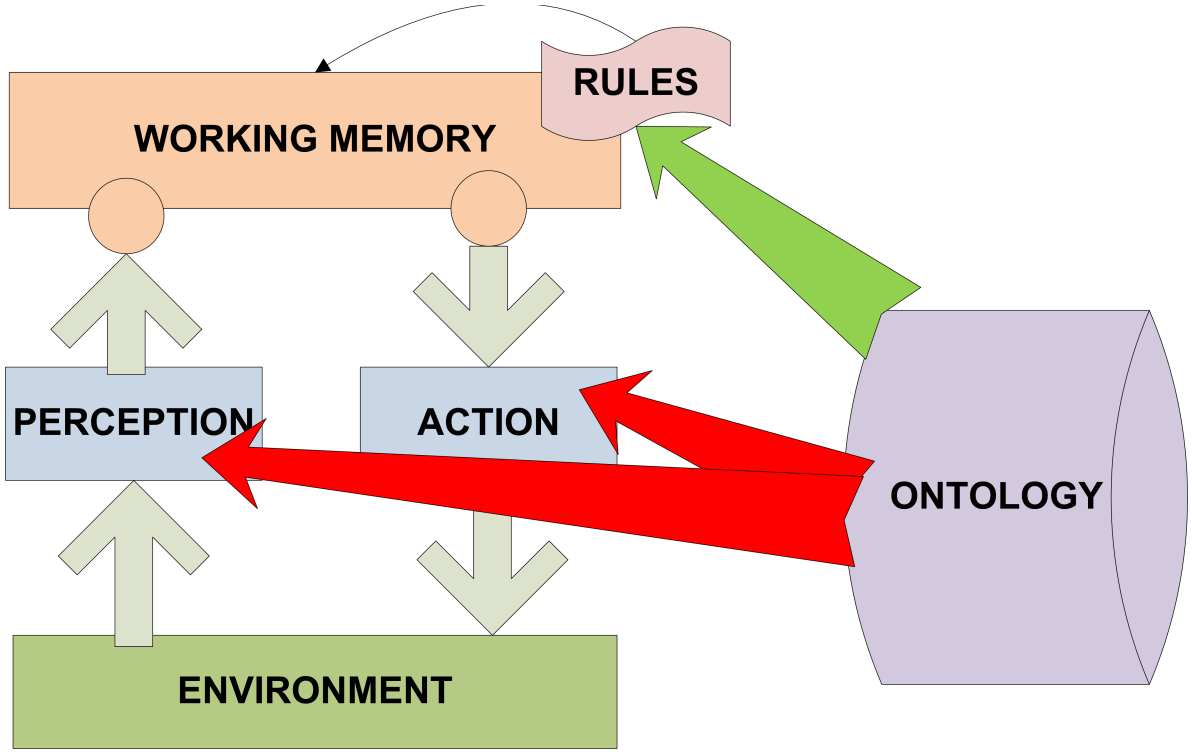
Soar Rules

Set of generic rules for binary-actions

1. propose
2. apply : record the supposed state
3. verify-result : check that the perceived state is the supposed state and update knowledge if not.

Generated knowledge

Every knowledge about objects, states and actions is stored in the agent's state during its initialization.
This is the only "domain-dependant" part.
This file is generated from the ontology.



What is to be done

Automatic generation of code to write in the IL and generate the OL handlers.

Conclusion

Nuggets

- Enactive proposal to model knowledge appraisal in Soar
- Graphical way to write Soar rules
- Binary-actions (open door)
- Actions with resources (unlock a door with a key)
- Non directly-related action and target (push a button to turn on light)

Coal

- Non-binary actions (pull a triger) ?
- Planning ?
- Preferences ?
- Still a lot to do !