TouringMachines

Credits go to Prof. Brian Logan

TouringMachines

theconversation.com

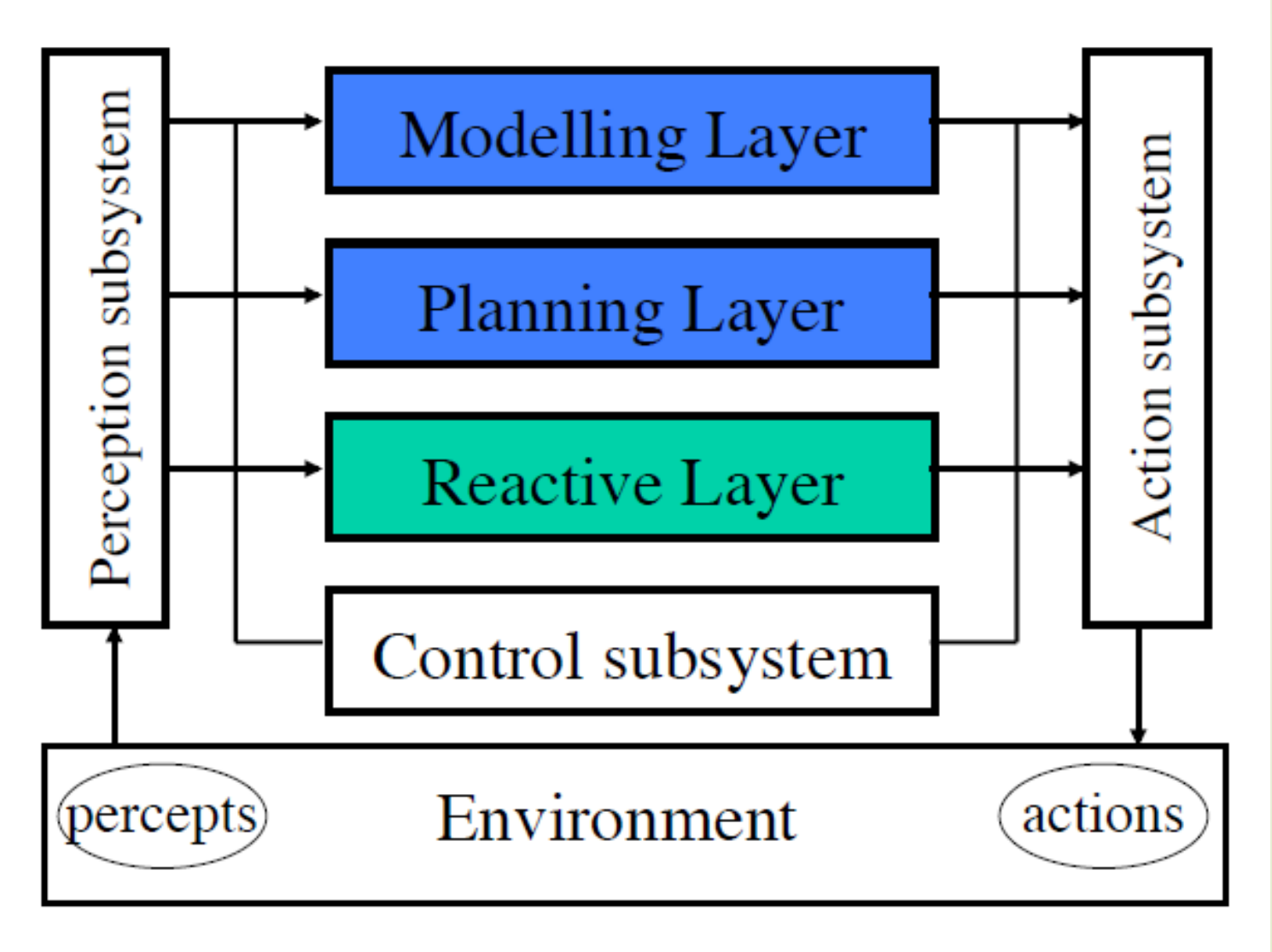
TouringMachines

- TouringMachines are autonomous (simulated) vehicles which drive along streets in the TouringWorld.

- the environment contains obstacles, traffic lights, rain (which affects braking distance) and fog (which changes theTouringMachines' visual field and range)

- the TouringMachines goals are to reach a given location by a specified time, avoiding collisions with obstacles and other TouringMachines and obeying the traffic regulations

TouringMachines Architecture



Reactive layer

- responsible for producing an immediate response to changes in the environment
- e.g., obstacle avoidance
- implemented as a set of condition action rules which map percepts directly to actions
- the rules can only refer to the agent’s current state and they cannot do any explicit reasoning about the world

Planning layer

- responsible for achieving simple goals, e.g., moving from place to place
- implemented as library of predefined plan schemas which are elaborated at run time
- to achieve a goal, the planning layer attempts to find a schema that matches that goal
- if a schema contains subgoals, the planning layer attempts to find schemas in the plan library that match each sub-goal

Modelling layer

- responsible for representing other entities (agents) in the world, including the agent itself
- predicts conflicts between agents and (autonomously) generates new goals to resolve these conflicts
- these goals are passed to the planning layer which plans to achieve them in the normal way

Control subsystem

- responsible for deciding which of the reactive, planning and modelling layers should have control of the agent at any given time

- implemented as a set of control rules which can either
  - suppress percepts output by the perceptual subsystem or
  - censor actions generated by the control layers

- e.g., a control rule may prevent the reactive layer from ever knowing that a particular obstacle has been perceived, if another layer is more appropriate for dealing with this type of obstacle

Integration

- the TouringMachines architecture can be viewed as hierarchical in the sense that there is a single subsystem (layer) which effectively makes all the control decisions

- in this sort of architecture, the designer must potentially consider all possible interactions between the layers

- if there are *n* layers and each layer can suggest *k* actions, this means that there are *k^n* interactions to be considered