AI Robotics

Outline

What is a robot?

- Recall the definition of a robot

Definition

A physical system capable of acting autonomously, with the capacity to sense its environment, and act accordingly to achieve a goal.

Robotic Challenges

- Robots in real environments are subject to many challenges
- What is my location?
- What is contained in my surroundings?
- Are my surroundings static or dynamic?
- What is my goal?
- How should I navigate in or manipulate my environment?
- How quickly do I need to respond to changes in the environment?

Control primitives

- Based on the definition of a robot control must encompass the following primitives in order to meet such challenges
- Sense
- Plan
- Act
- Different control paradigms are based on these concepts

Control primitives

- Based on the definition of a robot control must encompass the following primitives in order to meet such challenges
- Sense
- Plan
- Act
- Different control paradigms are based on these concepts

Control primitives

- Based on the definition of a robot control must encompass the following primitives in order to meet such challenges
- Sense
- Plan
- Act
- Different control paradigms are based on these concepts

Control primitives

- Based on the definition of a robot control must encompass the following primitives in order to meet such challenges
- Sense
- Plan
- Act
- Different control paradigms are based on these concepts

Control primitives

- Based on the definition of a robot control must encompass the following primitives in order to meet such challenges
- Sense
- Plan
- Act
- Different control paradigms are based on these concepts

Fundamental Architectures

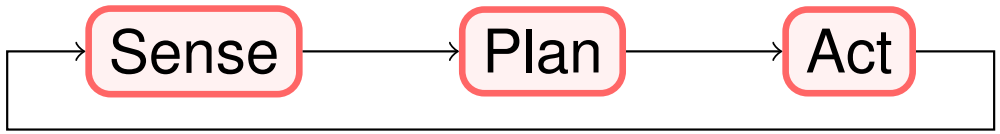
- There are three core robot architectures
- Deliberative

- Reactive

- Hybrid

Fundamental Architectures

- There are three core robot architectures
- Deliberative

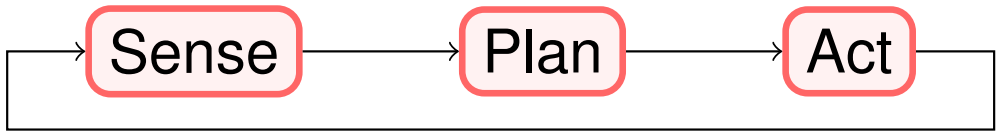


- Reactive

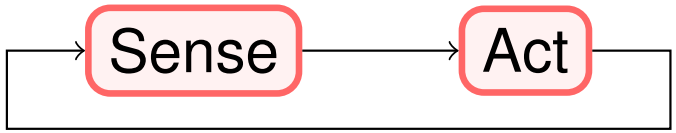
- Hybrid

Fundamental Architectures

- There are three core robot architectures
- Deliberative



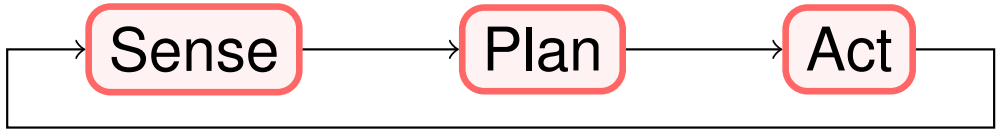
- Reactive



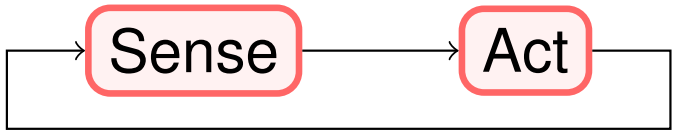
- Hybrid

Fundamental Architectures

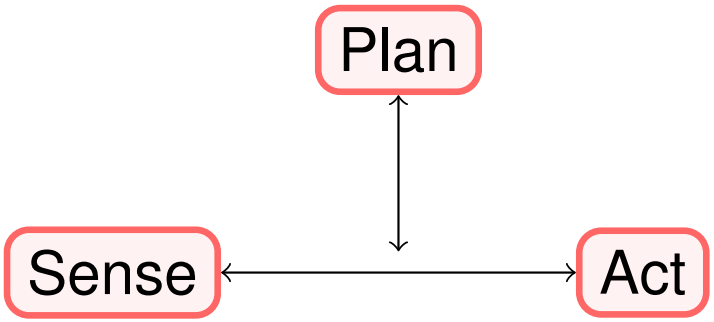
- There are three core robot architectures
- Deliberative



- Reactive

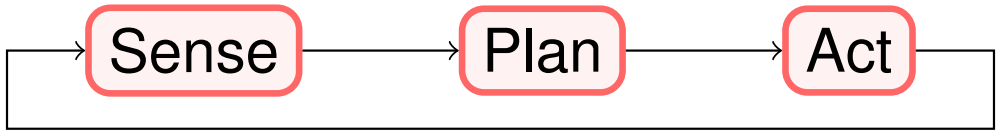


- Hybrid



Outline

The deliberative paradigm



- Introduced 1967
- In this paradigm the agent first senses the environment and produce a plan on how to act
- Requires the formulation and maintenance of an accurate world model built from sensor measurement

Question: In what environments would purely deliberative control be viable?

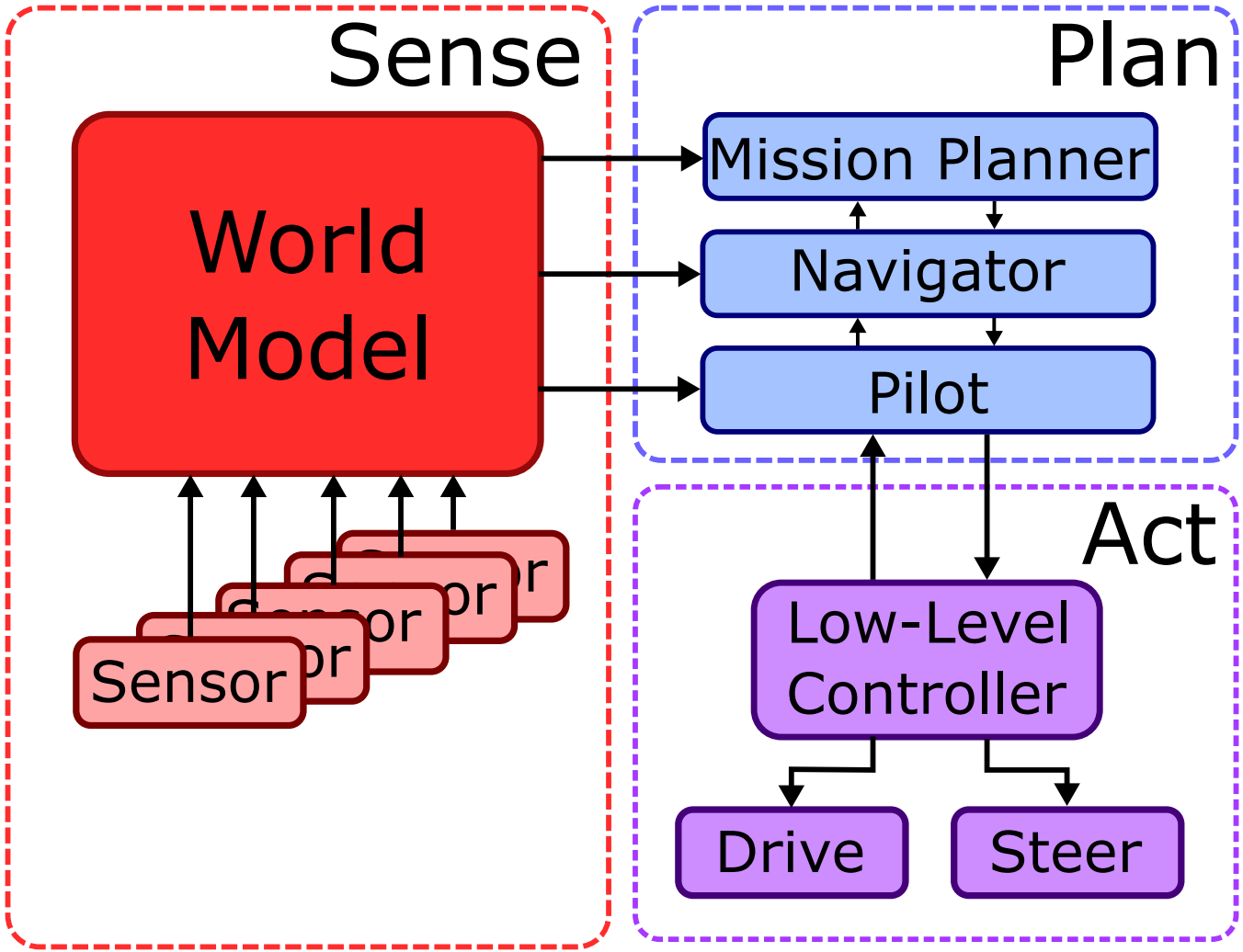
Examples of deliberative control

- Shakey was one of the first mobile robots with a deliberative control paradigm
- Shakey's controller was the Stanford Research Institute Problem Solver (STRIPS)

Implementations of deliberative control

- Two core hierarchical architectures
  - Nested Hierarchical Controller(NHC)
  - NIST Realtime Control System(RCS)

Nested Hierarchical Controller(NHC)



NHC Sensing

- External world model is built by measurement of the environment
- World Model is centralised and needs to be kept up-to-date

NHC Planning

- NHC decomposes planning into three sub-modules: Mission Planner, Navigator, Pilot
- Sub-modules can communicate between each other, but directives are top down
- Each higher level contains greater intelligence

NHC Planning

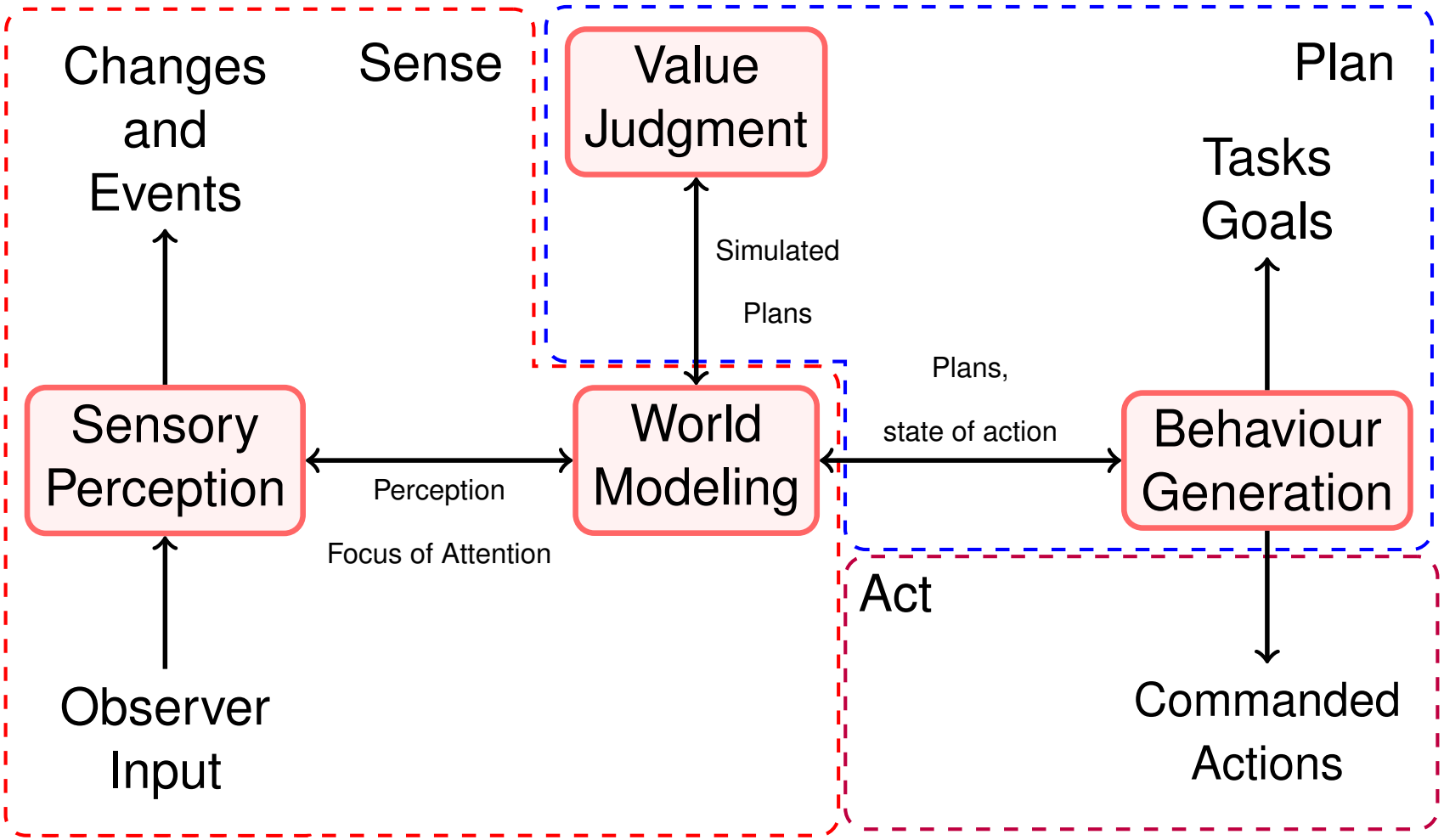
- Mission Planner Receives mission from external source or determines mission internally Determines location and sets the goal
- Navigator Recieves goal and current location from Mission planner and Plans the path to the goal
- Pilot Receives line segments or way points from the Navigator and Generates actions to be sent to actuators to navigate to goal

NHC Act

- The NHC is designed for navigation tasks therefore acting is composed of steering and driving
- The Low-Level controller receives instruction from the Pilot sub-module and translates pilot instructions into the required motion
- If the controller experiences difficulties it can communicate back up the chain to the pilot module

- Iterates between planning and action sequences
- Hierarchical breakdown of NHC is only applicable to navigation

NIST Realtime Control System(RCS)



- Developed as a standard for industries building intelligent robots
- Developed by James Albus in the 1980s
- Builds on NHC includes sensory processing/perception, behavior generation, value judgment
- Behaviour generation component generates action commands based on received plan

NIST RCS

- Sensory perception includes preprocessing such as feature extraction, or focus of attention.
- Value judgment module plans and the simulates the plans
- Finalised plans are provided to the Behaviour Generator which translates the plan into actions based on the Task/Goal

Disadvantages of deliberative control

- Information
  - The planner assumes the world model and other representations are accurate
- Time Scale
  - The sensor inputs, internal representations for a very large state space. Which can be slow to search
- Space
  - Storing the full state space of the robot is memory intensive
- Enacting plans
  - The determined plan can only be enacted if the environment has not significantly changed between sensing and planning
  - The robot has sufficient knowledge of the state of the environment and the plan at all time steps
  - The effectors are accurate enough to carry out the plan

Disadvantages of deliberative control

- Information
  - The planner assumes the world model and other representations are accurate
- Time Scale
  - The sensor inputs, internal representations for a very large state space. Which can be slow to search
- Space
  - Storing the full state space of the robot is memory intensive
- Enacting plans
  - The determined plan can only be enacted if the environment has not significantly changed between sensing and planning
  - The robot has sufficient knowledge of the state of the environment and the plan at all time steps
  - The effectors are accurate enough to carry out the plan

Disadvantages of deliberative control

- Information
  - The planner assumes the world model and other representations are accurate
- Time Scale
  - The sensor inputs, internal representations for a very large state space. Which can be slow to search
- Space
  - Storing the full state space of the robot is memory intensive
- Enacting plans
  - The determined plan can only be enacted if the environment has not significantly changed between sensing and planning
  - The robot has sufficient knowledge of the state of the environment and the plan at all time steps
  - The effectors are accurate enough to carry out the plan

Disadvantages of deliberative control

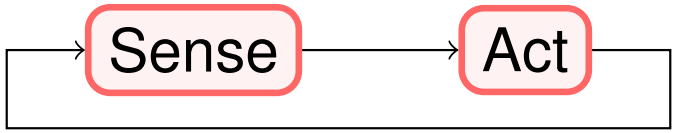
- Information
  - The planner assumes the world model and other representations are accurate
- Time Scale
  - The sensor inputs, internal representations for a very large state space. Which can be slow to search
- Space
  - Storing the full state space of the robot is memory intensive
- Enacting plans
  - The determined plan can only be enacted if the environment has not significantly changed between sensing and planning
  - The robot has sufficient knowledge of the state of the environment and the plan at all time steps
  - The effectors are accurate enough to carry out the plan

Summary of deliberative control

- Defines a rigid relationship between sensing, planning and acting
- Two well known implementation of this are the Nested Hierarchical Controller (NHC) and NIST Real-Time Controller
- NHC is no longer used, but the NIST RTC standard can still be found in use
- Generally fully deliberative control has fallen out of favour to Reactive and Hybrid approaches

Outline

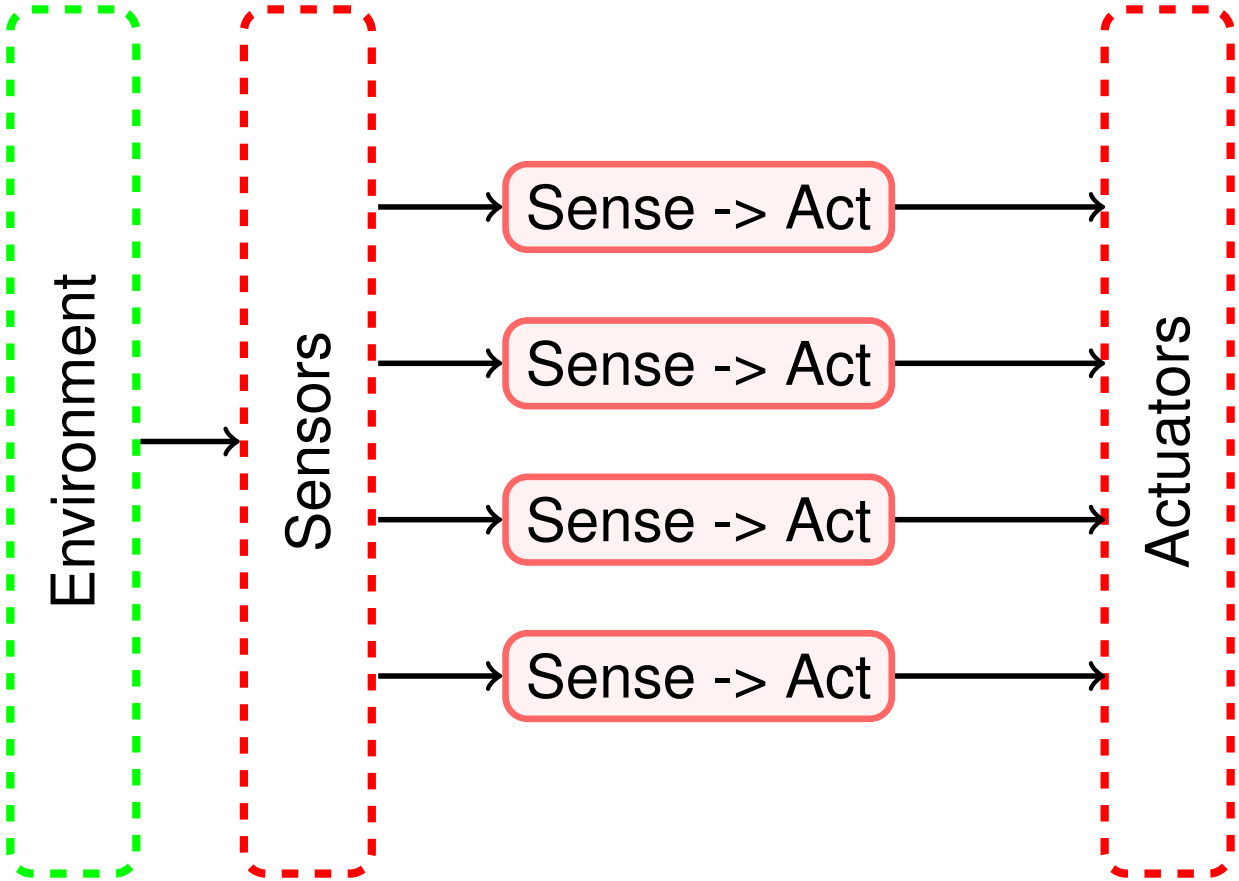
The Reactive paradigm



- The reactive paradigm removes the planning component from the control loop
- Reactive control is biologically inspired from animal reflexive behaviour

Reactive behaviours

- Reactive systems consist of many rules/behaviours
- Each behaviour is triggered by an external stimulus



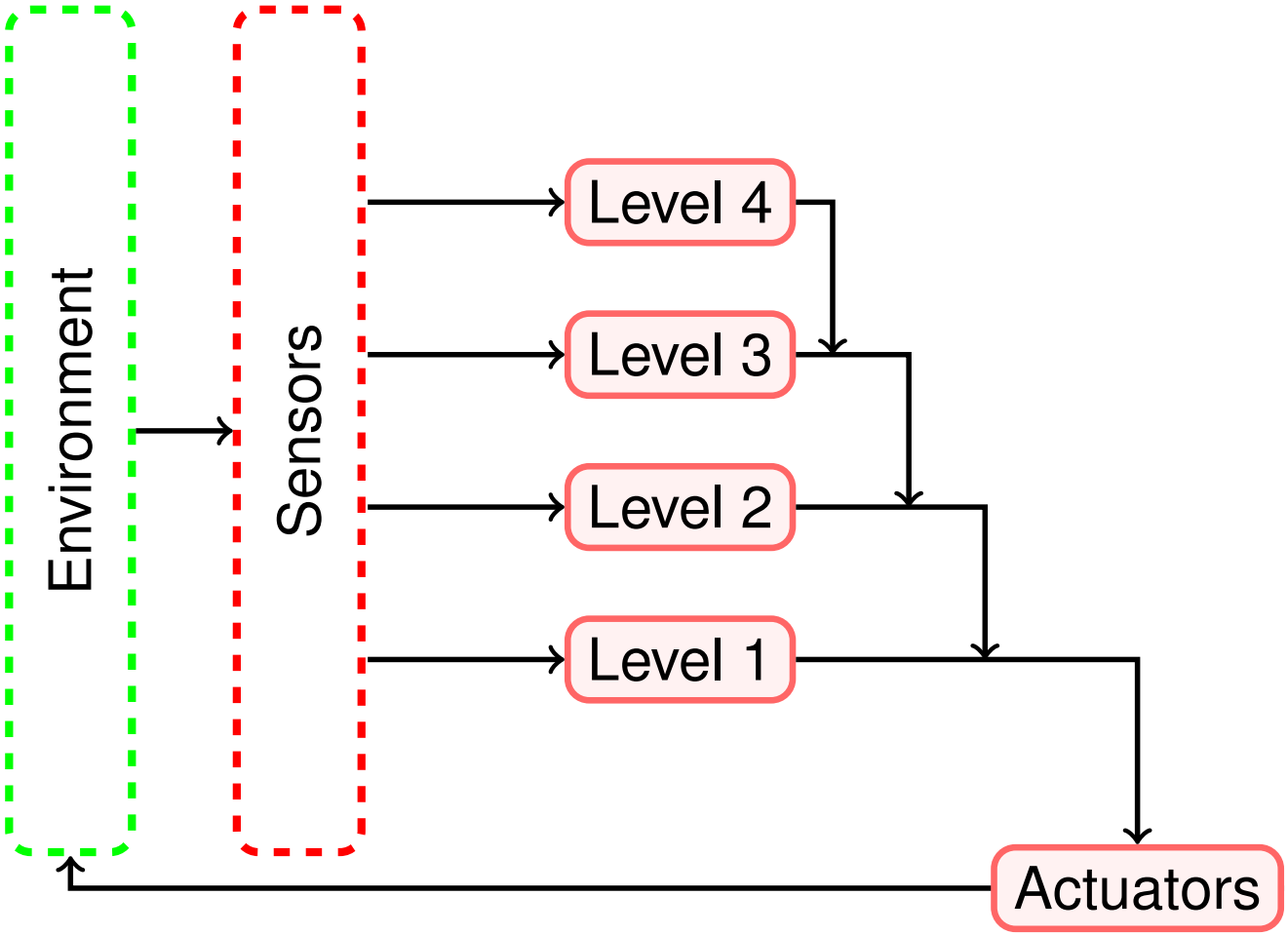
Reactive behaviours

- To reduce system complexity behaviours are treated as being mutually exclusive
- Each detected state from a sensor input can only trigger a single behaviour

Examples of reactive control

- The most well known reactive control is the Subsumption architecture
- Created by Prof. Rodney Brook in 1985
- Aims to build complex systems incrementally, from simple to complex.
- System is built on a collection of layers, where each layer has a single task

The Subsumption Architecture



Advantages of the Subsumption Architecture

- Simpler debugging. Each layer achieves a single task which can be well specified and tested new layers don't effect lower layers behaviour
- Robust. If higher layers fail, lower levels continue unaffected.

Action Selection

- As Sensory input maps in a 1-1 fashion with behaviour, how are actuators instructed in complex environments?
- Command arbitration is when a single action is chosen based on some priority
  - Fixed priority
  - Dynamic priority
- Action fusion aims to combine the actions in a meaningful way
  - Potential Fields
  - Fuzzy Logic
  - Motor Schemas

Signal Inhibition

- Higher layers may disable lower levels temporarily
- Inputs to a layer can be suppressed leading to no action in the layer and no output
- Outputs of a layer can be inhibited. Sensory input is received and the action determined but no actuation occurs

Subsumption Example:

- Exercise: Design a subsumption based system with light seeking behaviour ala Elmer and Elise

Subsumption Example:

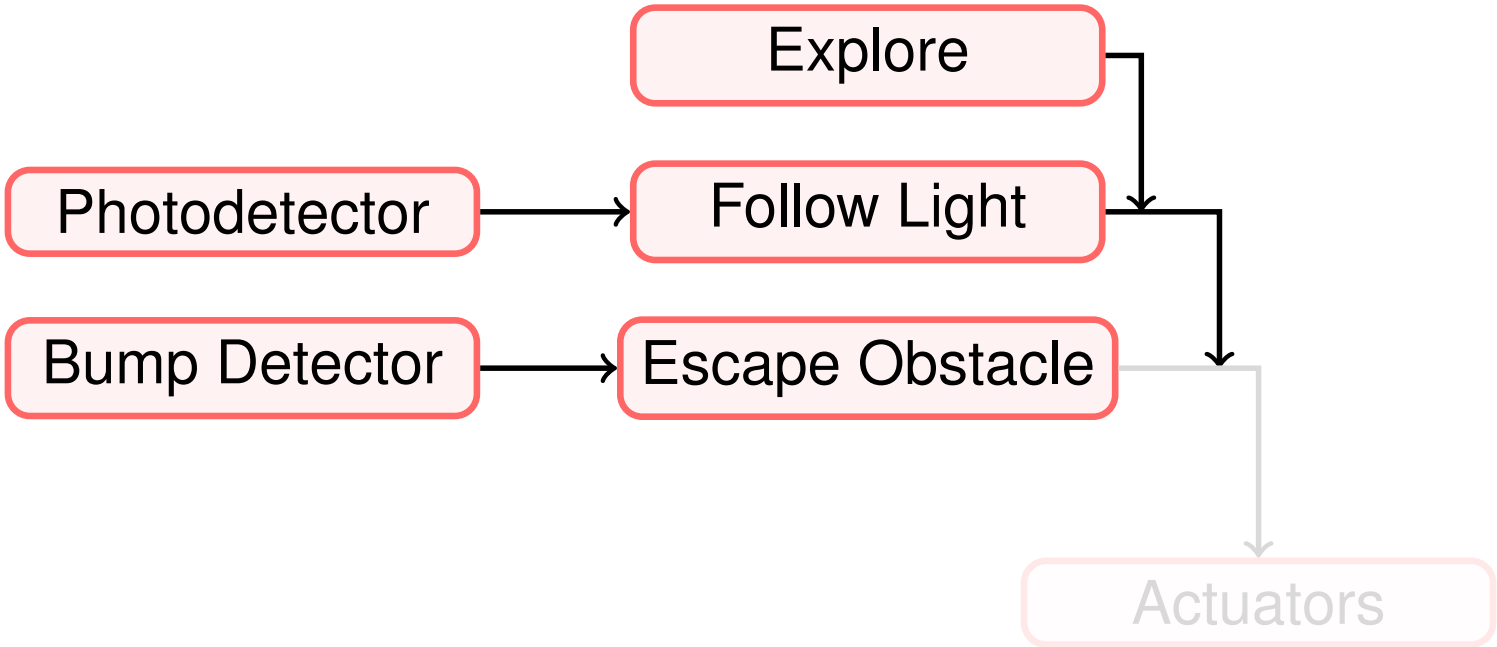
- Design a subsumption based system with light seeking behaviour ala Elmer and Elise

Photodetector

Bump Detector

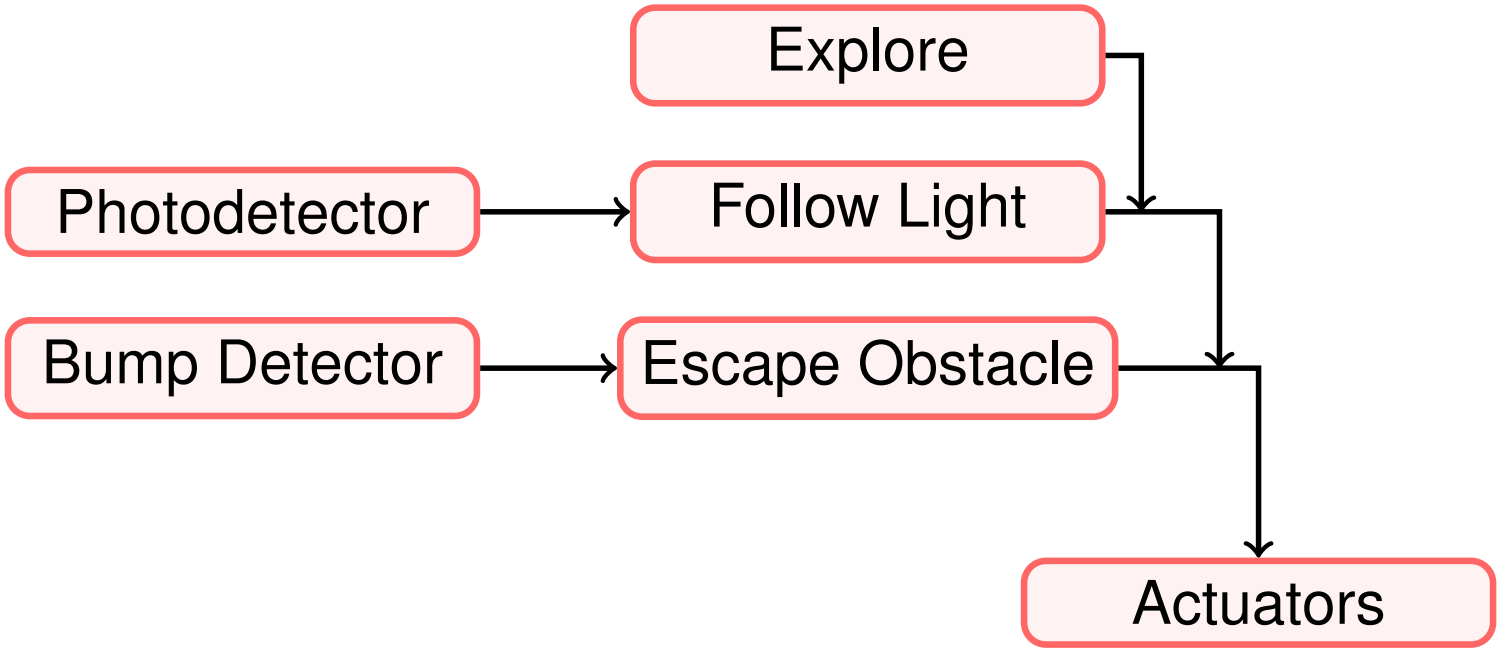
Subsumption Example:

- Design a subsumption based system with light seeking behaviour ala Elmer and Elise



Subsumption Example:

- Design a subsumption based system with light seeking behaviour ala Elmer and Elise



Disadvantages of reactive control

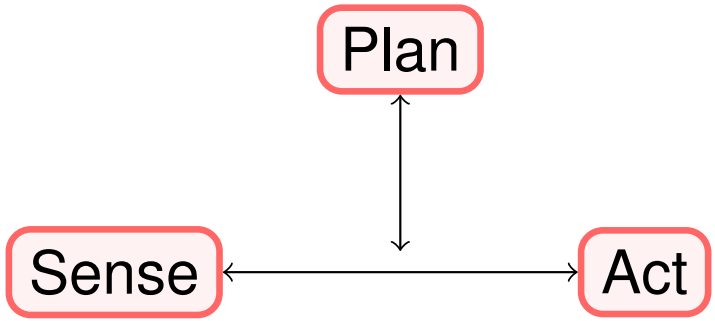
- No memory
- No world representation
- No Learning
- No state

Summary of reactive control

- Eliminates the planning phase and creates reflexive sense-action pairs
- Provides a structured bottom up approach to controller design
- Behaviors need to be coded at design time, complex tasks require coverage of a large state space

Outline

The Hybrid paradigm



- The Hybrid paradigm aims to combine the speed of the reactive paradigm with the intelligence of the deliberative
- These controllers feature three layers: Reactive, Planning and Linking layers
- Combines multiple time-scales: Long term (Planning) and Short term (Reactive)

Implementations of Hybrid Architectures

- Managerial
  - AuRa
  - SFX
- State Hierarchies
  - 3T
- Model-Oriented
  - Saphira, TCA

Implementations of Hybrid Architectures

- Managerial
  - AuRa
  - SFX
- State Hierarchies
  - 3T
- Model-Oriented
  - Saphira, TCA

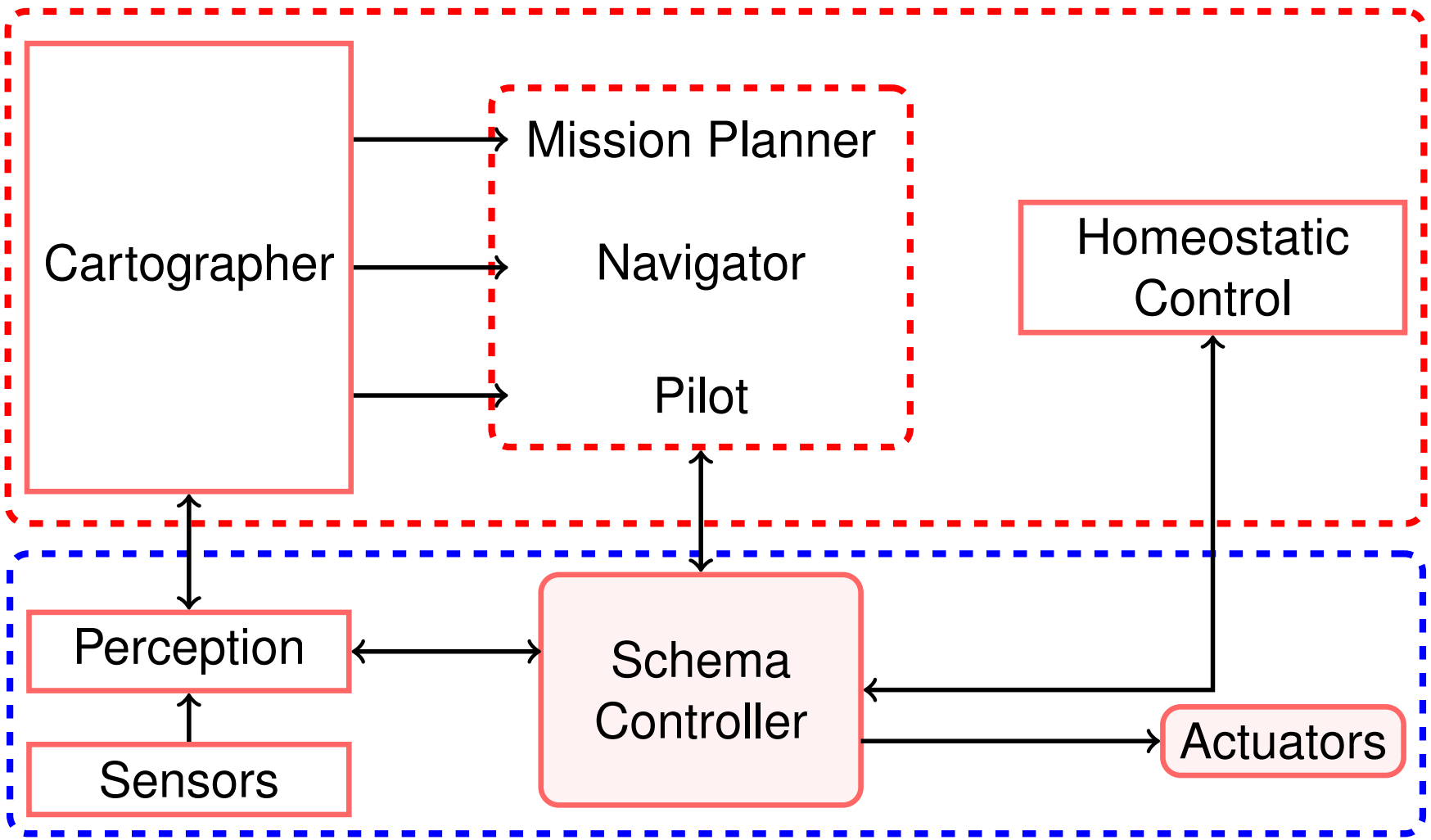
Implementations of Hybrid Architectures

- Managerial
  - AuRa
  - SFX
- State Hierarchies
  - 3T
- Model-Oriented
  - Saphira, TCA

Common Components

- Mission Planner: Receive commands and create plans
- Cartographer: Form and maintain spatial information
- Sequencer: Generate the sequence of behaviours required to solve a task
- Behavioural manager: Designate which resources are available to behaviours
- Performance monitor: Monitor and asses if progress is being made towards goals

AuRA


Cartographer
Mission Planner
Navigator
Pilot
Homeostatic Control
Perception
Sensors
Schema Controller
Actuators

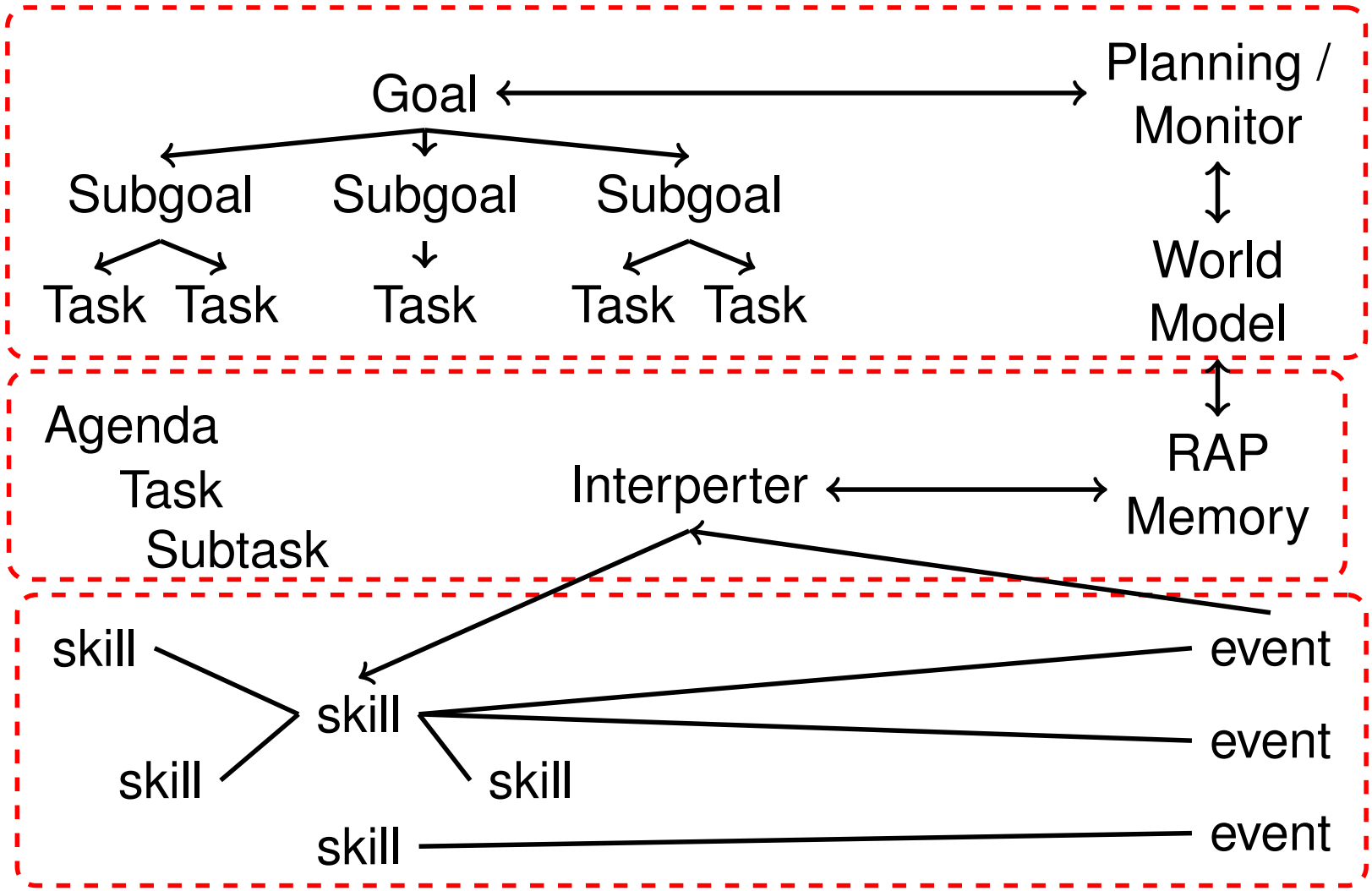
AuRA

- Cartographer: Provides map
- Mission Planner: Plans how to achieve task
- Navigator: Generates the desired path to the goal
- Pilot: Determines the actions required to traverse the path set by the navigator
- Perception: Receives sensor input and processes them to extract relevant features
- Schema Controller: Generates commands to be sent to the actuators and monitors the behaviour of the actuators

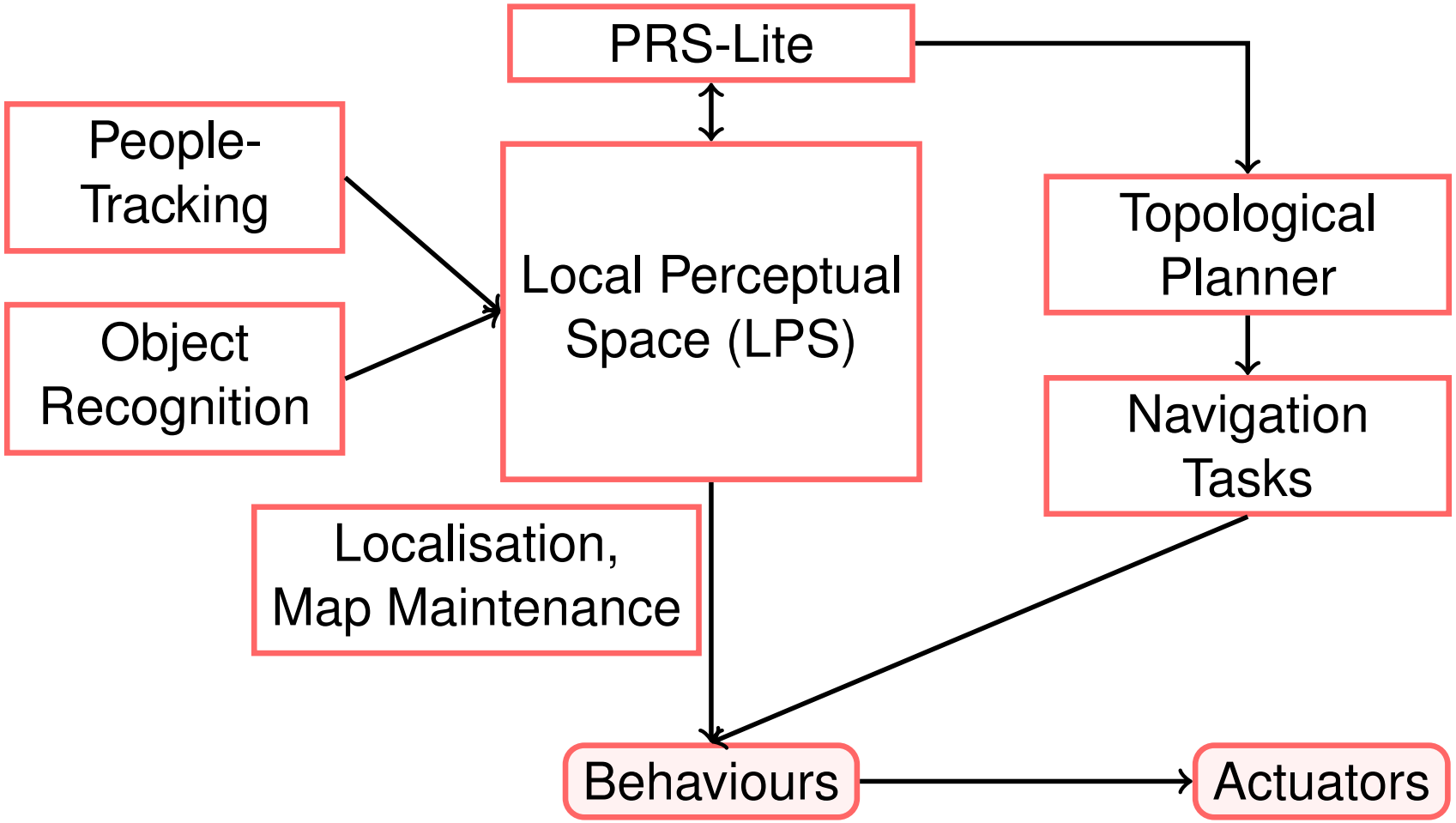
3T

- Developed in by 1997 widely used at NASA
- Is made of three layers: Planner, Sequencer and Skill Manager
- The planner is the deliberative layer which determines goals, constraints, world model and timing
- Sequencer manages the order and required skills which need to be triggered to achieve a specific task
- Skill manager coordinates a set of reactive skills which the robot can perform

3T


Goal
Planning / Monitor
Subgoal
Subgoal
Subgoal
World Model
Task
Task
Task
Task
Task
Agenda
Task
Subtask
Interperter
RAP Memory
skill
skill
skill
skill
event
event
event

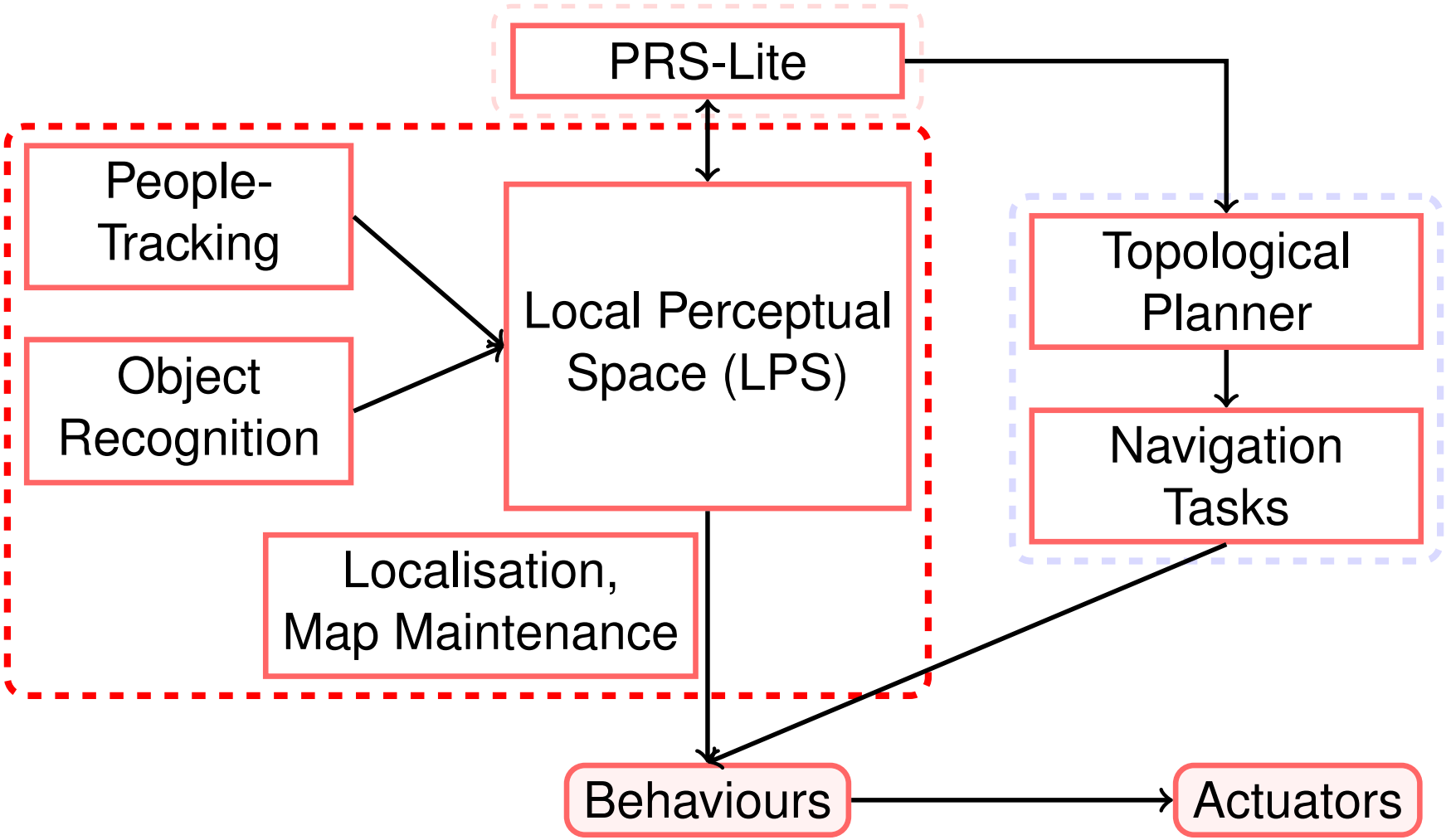
Saphira


PRS-Lite
People-
Tracking
Object
Recognition
Local Perceptual
Space (LPS)
Topological
Planner
Navigation
Tasks
Localisation,
Map Maintenance
Behaviours
Actuators

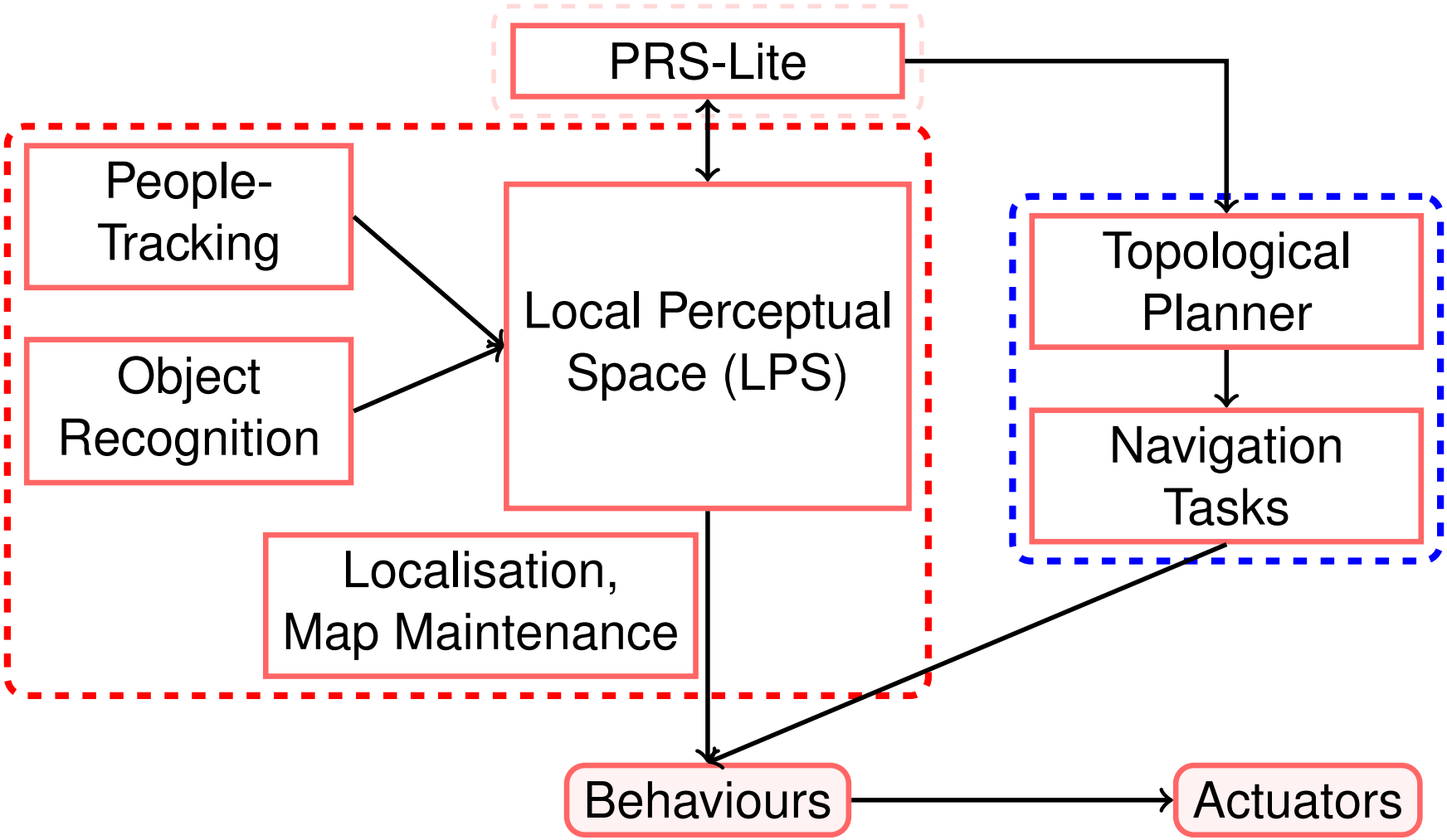
Saphira

- Saphira is an example of a model-based hybrid architecture
- The LPS provides a geometric representation of the space surrounding the robot
- Sensors feed input into defined perceptual routines
- The Procedural Reasoning System (PRS) provide the ability to plan actions, monitor system resources and performance

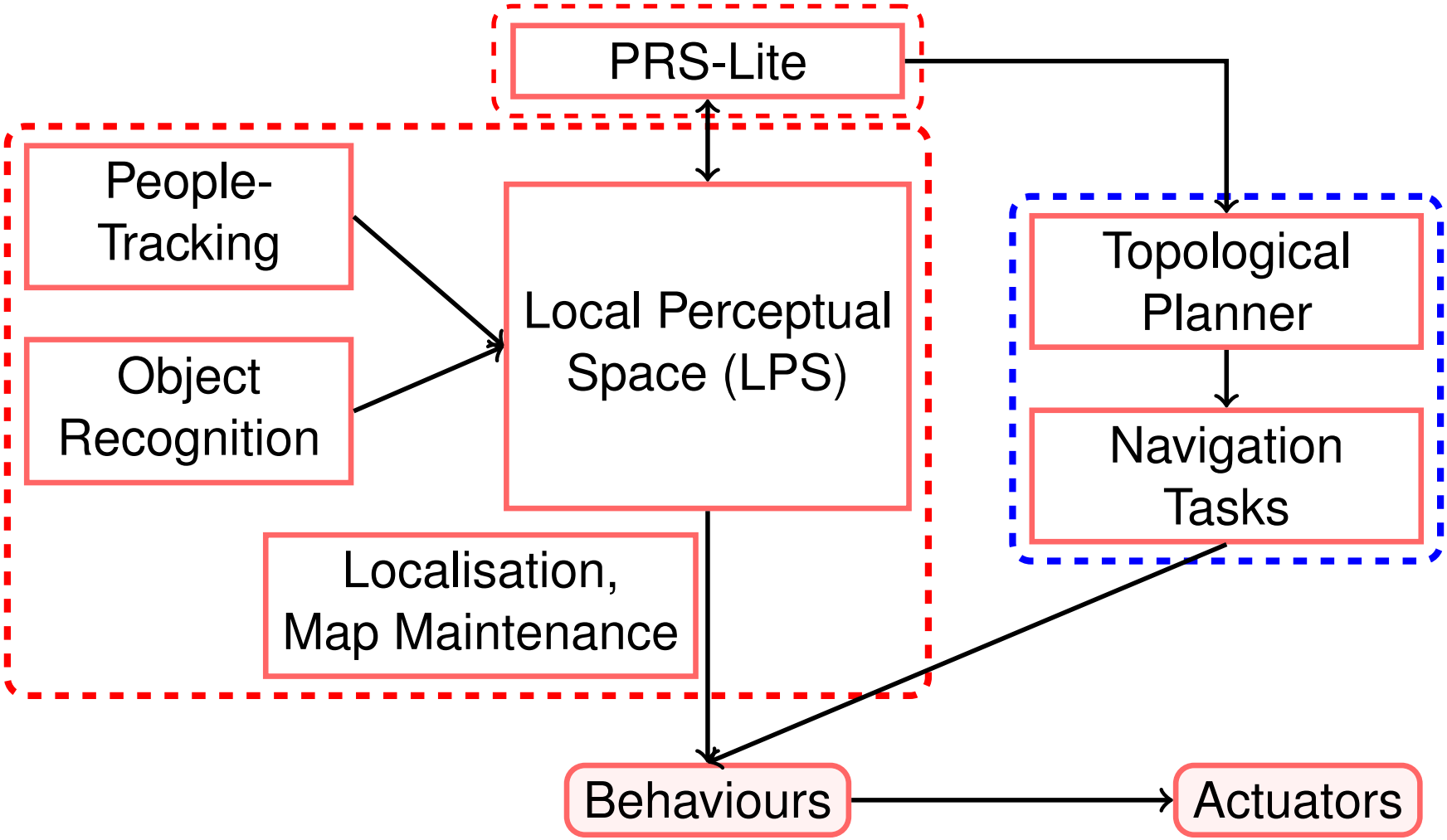
Saphira


PRS-Lite
People-
Tracking
Object
Recognition
Local Perceptual
Space (LPS)
Localisation,
Map Maintenance
Topological
Planner
Navigation
Tasks
Behaviours
Actuators

Saphira


PRS-Lite
People-
Tracking
Object
Recognition
Local Perceptual
Space (LPS)
Localisation,
Map Maintenance
Topological
Planner
Navigation
Tasks
Behaviours
Actuators

Saphira


PRS-Lite
People-
Tracking
Object
Recognition
Local Perceptual
Space (LPS)
Localisation,
Map Maintenance
Topological
Planner
Navigation
Tasks
Behaviours
Actuators

Disadvantages of hybrid control

- Middle layer is difficult to design and hard to generalise
- Can degenerate into a reactive architecture if deliberative layers become too slow
- Difficult to debug

Summary of hybrid control

- Combines the speed of reactive and intelligence of deliberative paradigms
- Different styles of hybrid architecture exist
  - Managerial - Top down scope of control
  - State Hierarchies - Utilises current state to distinguish between deliberative and reactive
  - Model-Oriented - Behaviours have access to world model, fuzzy border between deliberative and reactive

Summary of hybrid control

- Combines the speed of reactive and intelligence of deliberative paradigms
- Different styles of hybrid architecture exist
  - Managerial - Top down scope of control
  - State Hierarchies - Utilises current state to distinguish between deliberative and reactive
  - Model-Oriented - Behaviours have access to world model, fuzzy border between deliberative and reactive

Summary

- Three robot primitives: Sense, Plan, Act. These make up all robot control paradigms
- Three core control paradigms exist: Deliberative, Reactive and Hybrid

- Next Lecture
  - Sensing, Perception and SLAM