BLG456E
Robotics
Robot Software Architectures & Planning

- Temporal Decomposition
- Main Architecture Types
  - Deliberative.
    - Planning Intro
  - Reactive.
  - Behaviour-based.
  - Hybrid.
- Examples
  - Shakey Example.
  - ROS Example.

Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule
Coordination: http://ninova.itu.edu.tr/

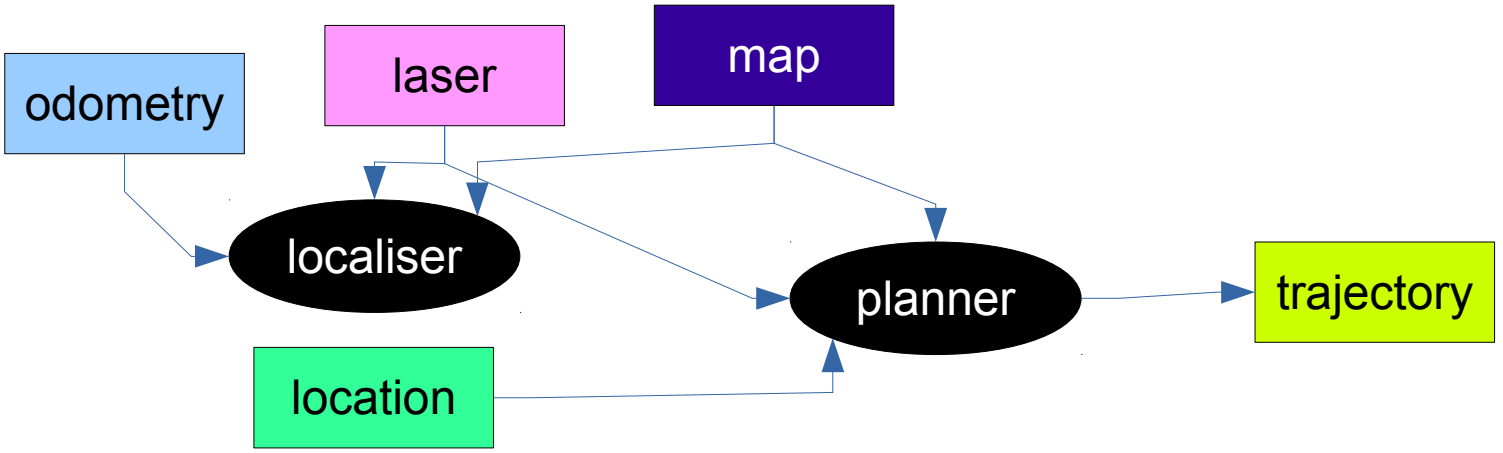
Considerably adapted from work by Dr. Sanem Sarıel-Talay

What is a Robot Architecture

The organisation of a robot (software) control system.

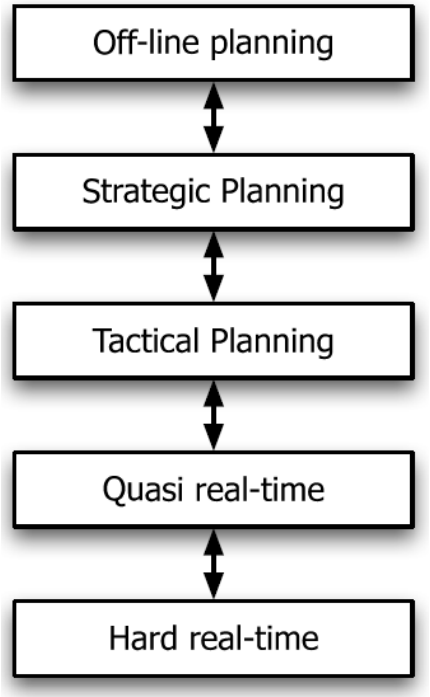
E.g.:

- How components coordinate (e.g. localisation and navigation).
- Conceptual approach to achieving tasks.
- ...

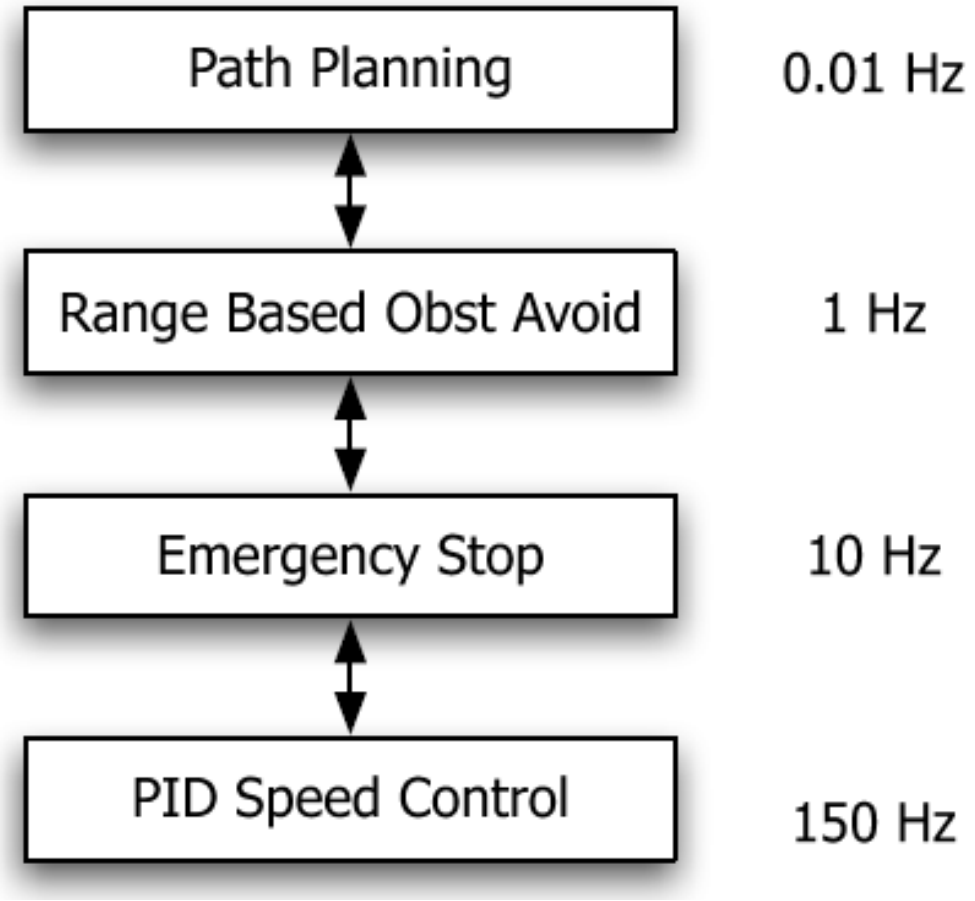


Temporal decomposition of robot architectures

- Coarse division of control
- Layering can be loosely synchronous



Sample Mobile Platform Decomposition



BLG456E
Robotics
Robot Software Architectures & Planning

- Temporal Decomposition
- Main Architecture Types
  - Deliberative.
    - Planning Intro
  - Reactive.
  - Behaviour-based.
  - Hybrid.
- Examples
  - Shakey Example.
  - ROS Example.

Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule
Coordination: http://ninova.itu.edu.tr/

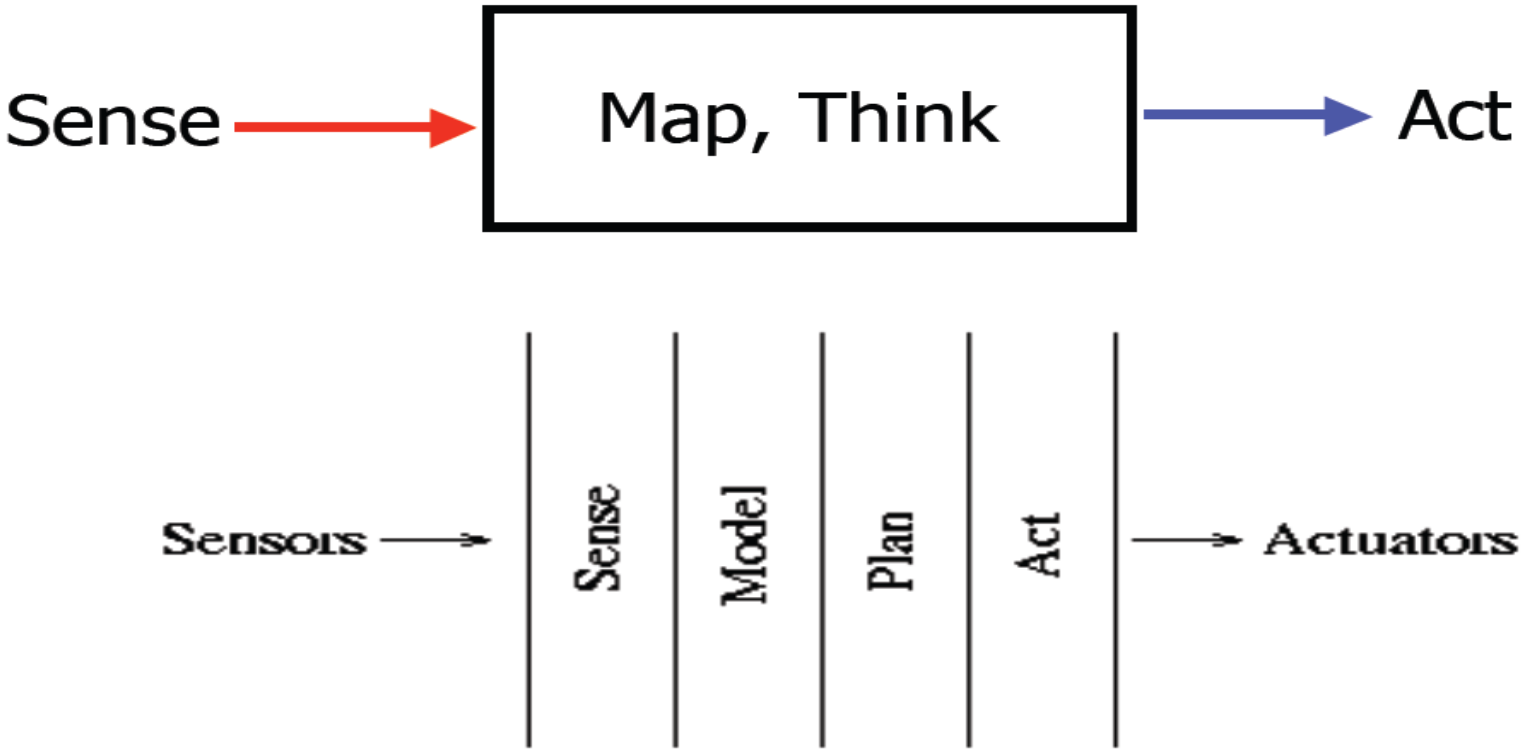
Considerably adapted from work by Dr. Sanem Sarıel-Talay

Control Architecture Types (Intro)

- Deliberative
- Reactive
- Behaviour-based
- Hybrid

Deliberative Architecture

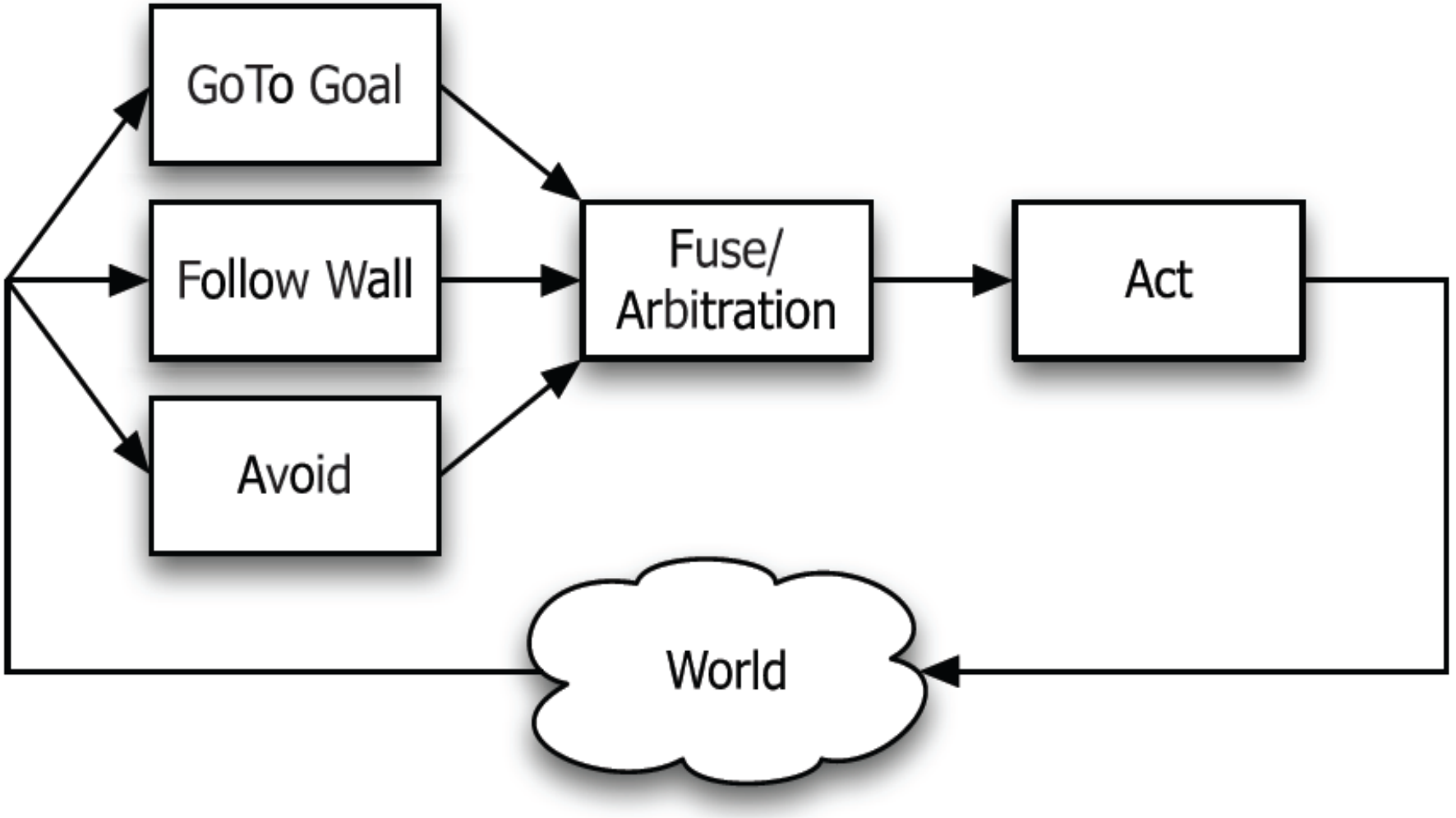
- Maps, lots of state
- Look-ahead



Reactive Architecture

- No maps, no state
- No look ahead
- Could be implemented by a look-up table

Parallel Decomposition - Behavioural



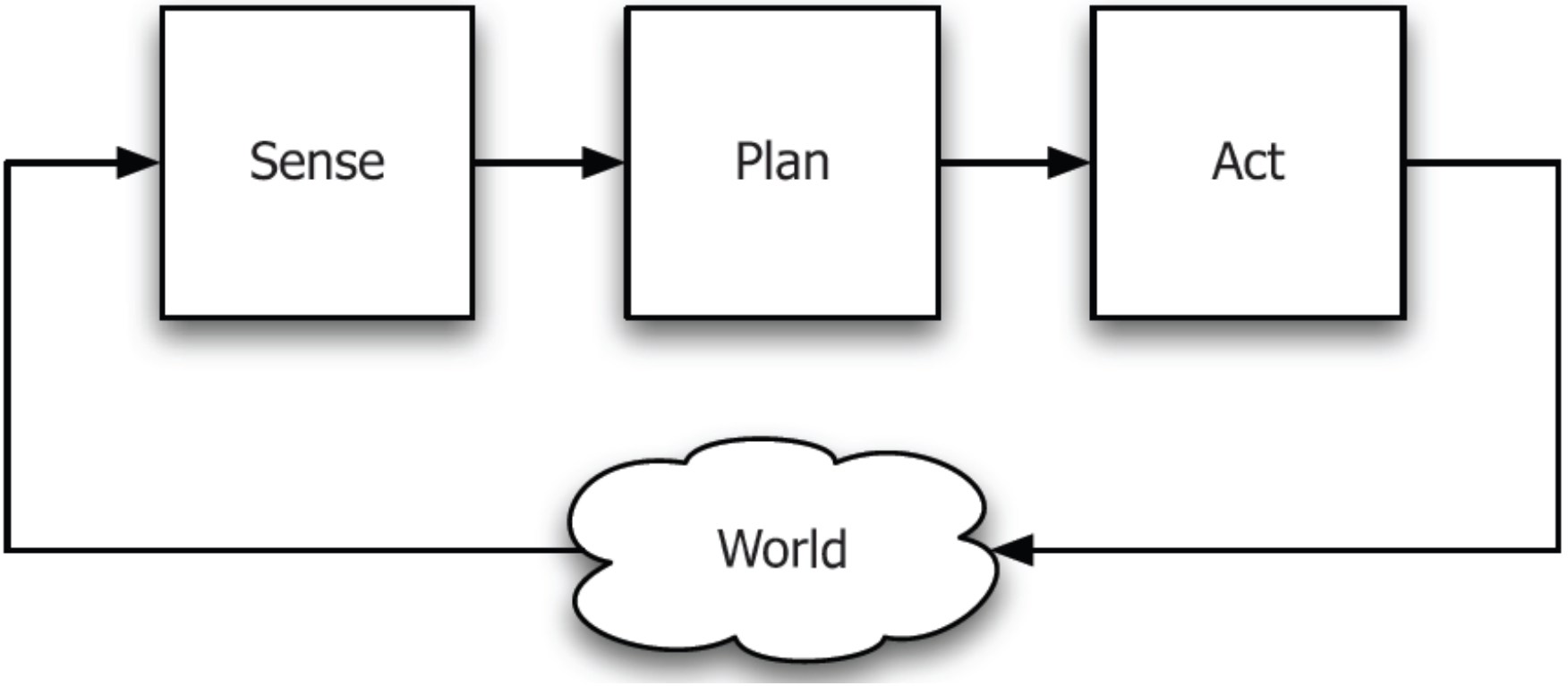
BLG456E
Robotics
Robot Software Architectures & Planning

- Temporal Decomposition
- Main Architecture Types
  - Deliberative.
    - Planning Intro
  - Reactive.
  - Behaviour-based.
  - Hybrid.
- Examples
  - Shakey Example.
  - ROS Example.

Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule
Coordination: http://ninova.itu.edu.tr/

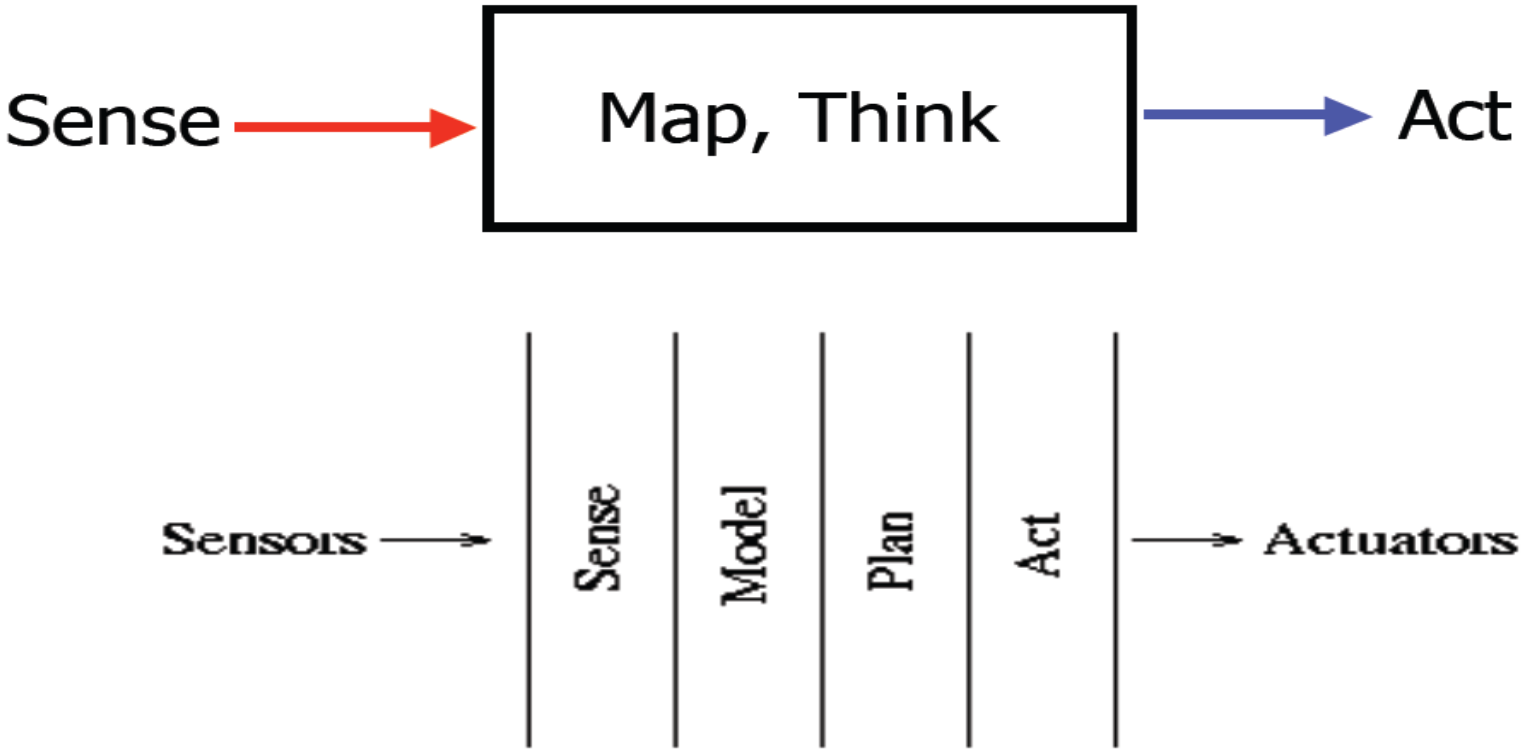
Considerably adapted from work by Dr. Sanem Sarıel-Talay

Control Decomposition: Sense-Plan-Act

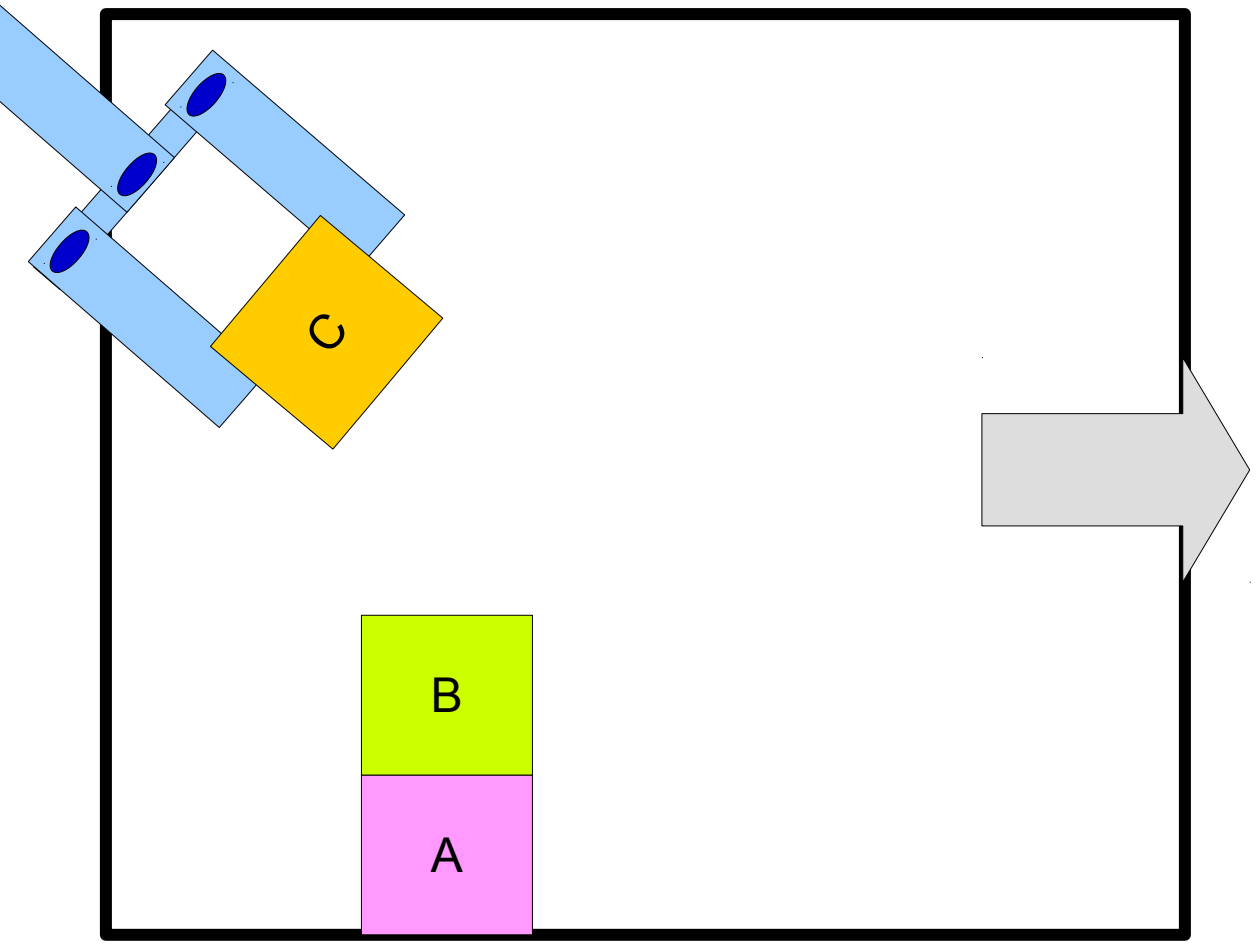


Deliberative Architecture

- Maps, lots of state
- Look-ahead



World Modelling from Perception



on(B,A)
on(A,Table)
gripped(C)

High-level planning: Planning problems

Example high-level planning problem:

Initial state:

`on(B,A),on(A,Table),gripped(C)`

Goal state description:

`on(A,B),on(B,C)`

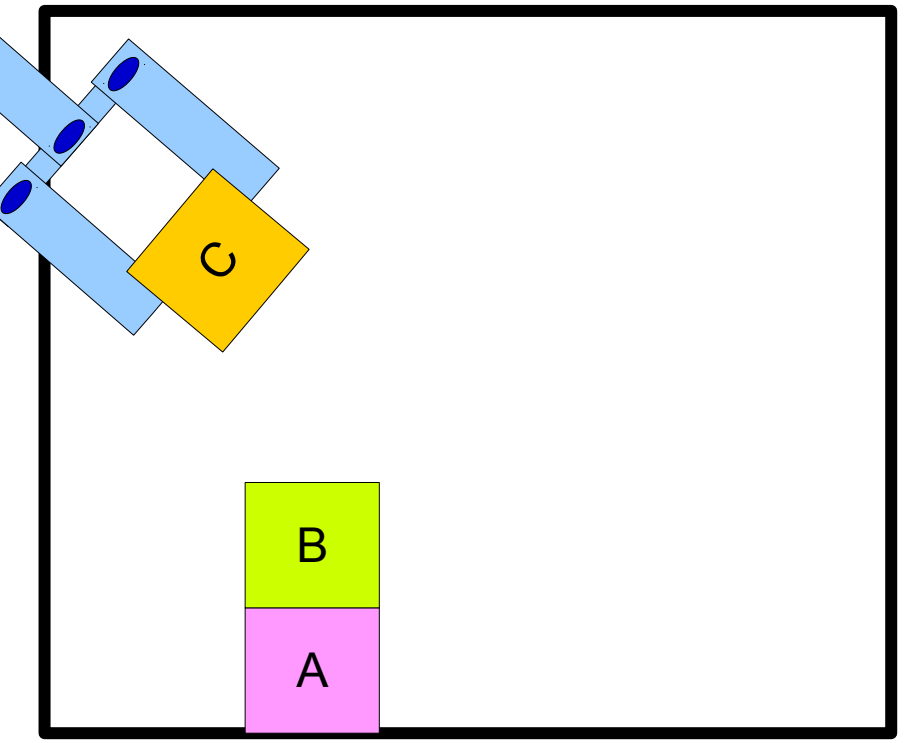
Also need a domain description with:

- Possible actions.
  - Action preconditions.
  - Action effects.

Example plan

```
1 STATE: on(B,A),on(A,Table),gripped(C)
  ACTION: put(C,Table)

2 STATE: on(B,A),on(A,Table),on(C,Table)
  ACTION: take(B,A)

3 STATE: on(A,Table),on(C,Table),gripped(B)
  ACTION: put(B,C)

4 STATE: on(A,Table),on(C,Table),on(B,C)
  ACTION: take(A,Table)

5 STATE: on(C,Table),on(B,C),gripped(A)
  ACTION: put(A,B)

6 STATE: on(C,Table),on(B,C),on(A,B)
  DONE
```

Domain Modelling for Action

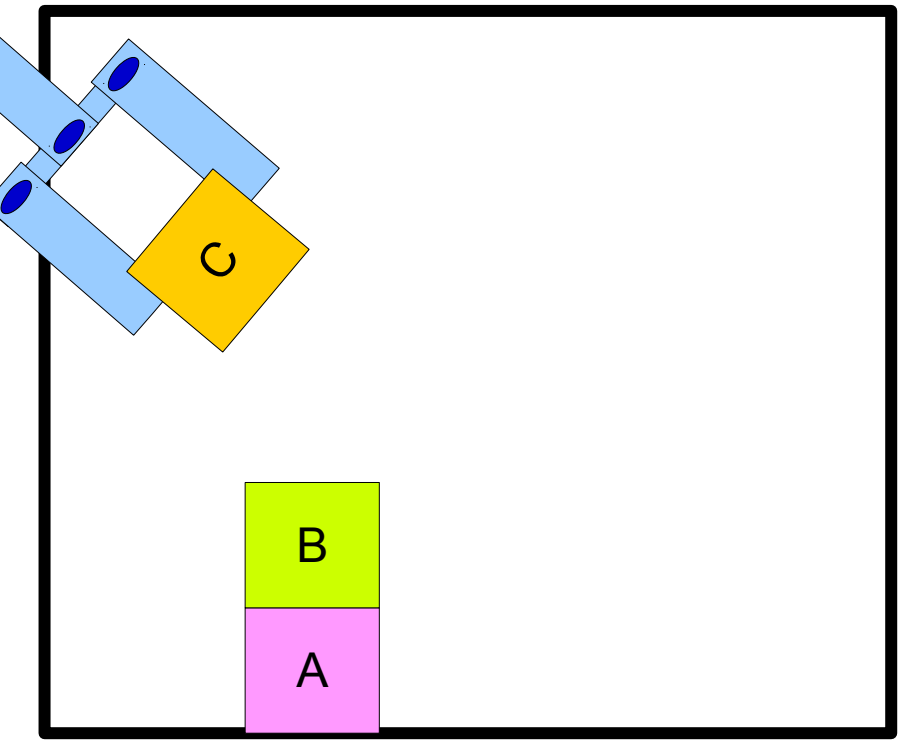


put(X,Y):

preconditions:
- -on(Z,Y)
- gripped(X)

effects:
- -gripped(X)
- on(X,Y)

Domain Modelling for Action



take(X,Y):

preconditions:
- on(X,Y)
- -on(Z,X)
- -gripped(V)

effects:
- gripped(X)
- -on(X,Y)

Finding plans

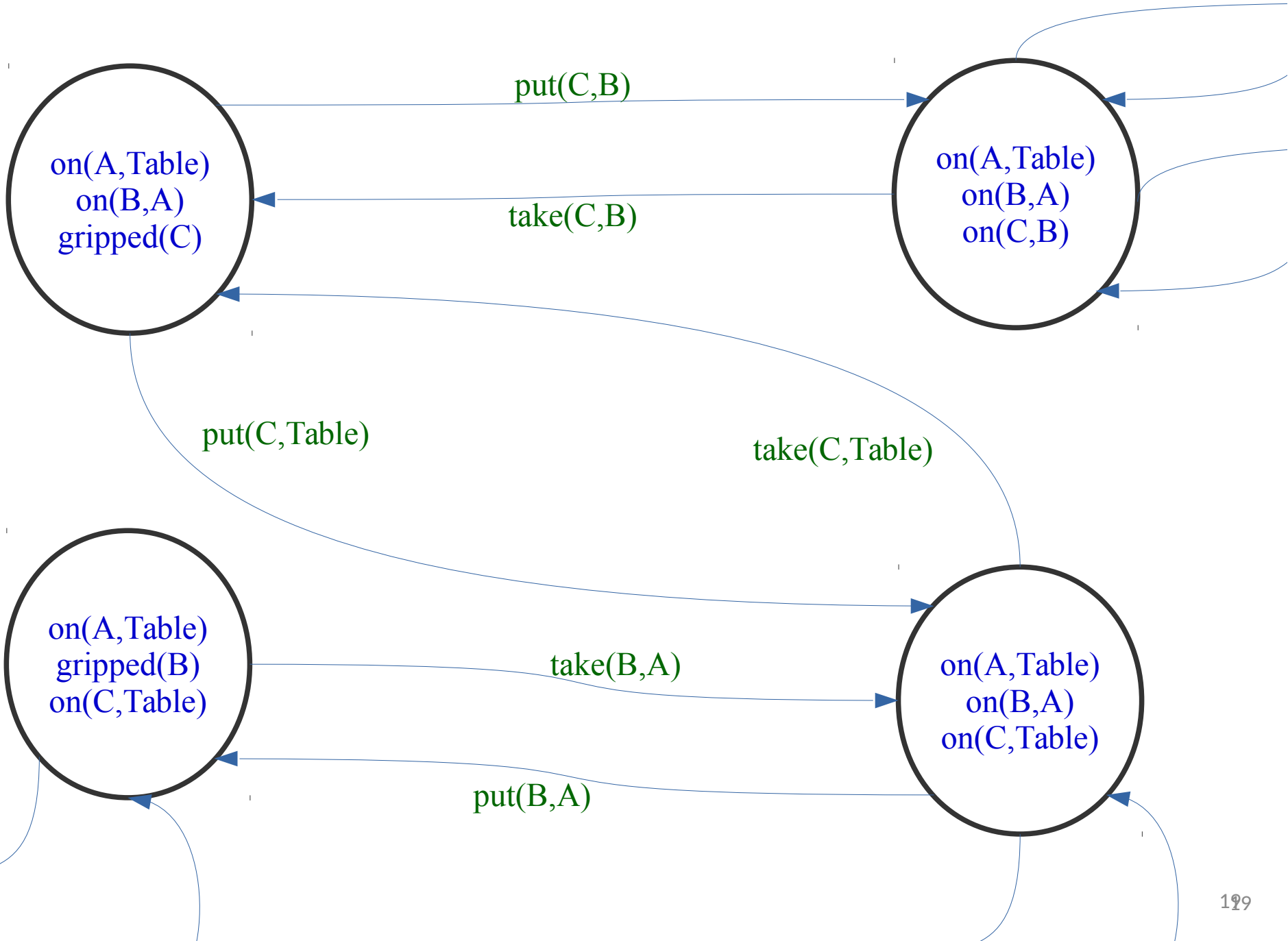
- Given:
  - Domain description (available actions).
  - Goal description and initial state.
- Find:
  - Sequence of actions.

- Sample Approaches:
  - First-order logic & theorem-proving.

    → SEARCH
  - General purpose planners.

    → SEARCH **(like A*)**

Example state subgraph



Making planning work (I)

- Connect with lower-levels.
- Supervisory mechanisms.

Making planning work (II)

- Quick planning.
  - → Online planning.
- Respond to changes.
  - → Online planning.
  - → Plan repair.
- Low level detail.
  - → Human effort.
  - → Semantic attachment?
  - → Learning?

Making planning work (III)

- Implement the actions.
  - → Human effort.
  - → Learning?
- Implement the perception.
  - → Human effort.
  - → Learning?
- Match model to world.
  - → Human effort.
  - → Learning!!

BLG456E
Robotics
Robot Software Architectures & Planning

- Temporal Decomposition
- Main Architecture Types
  - Deliberative.
    - Planning Intro
  - Reactive.
  - Behaviour-based.
  - Hybrid.
- Examples
  - Shakey Example.
  - ROS Example.

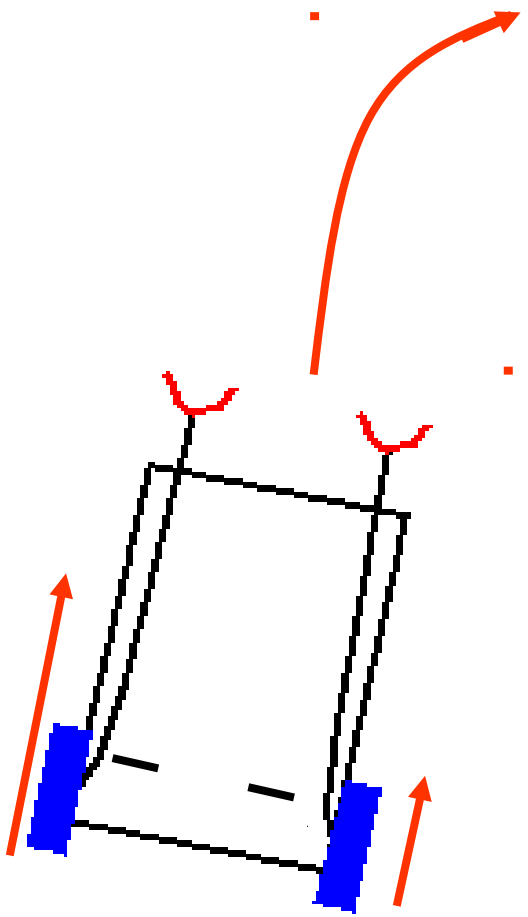
Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule
Coordination: http://ninova.itu.edu.tr/

Considerably adapted from work by Dr. Sanem Sarıel-Talay

Reactive Architecture

- No maps, no state
- No look ahead
- Could be implemented by a look-up table

Braitenberg Car



- By default, motors turn.
- Sensor inhibits same-side motor.

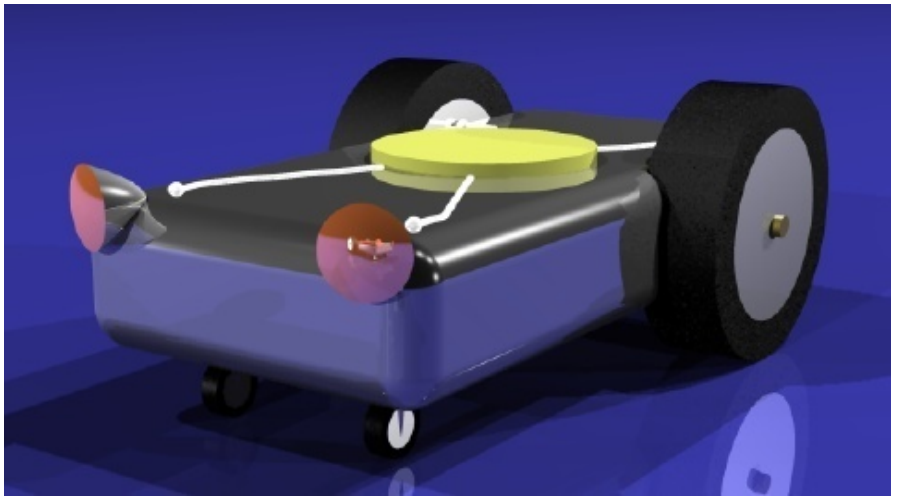
- Goes toward light.
- Speeds up in dark.
- Slows down in light.

↓

- Spends more time in light.

Braitenberg Car

- Appears to have goals
- No states (current velocities).
- "loves light".



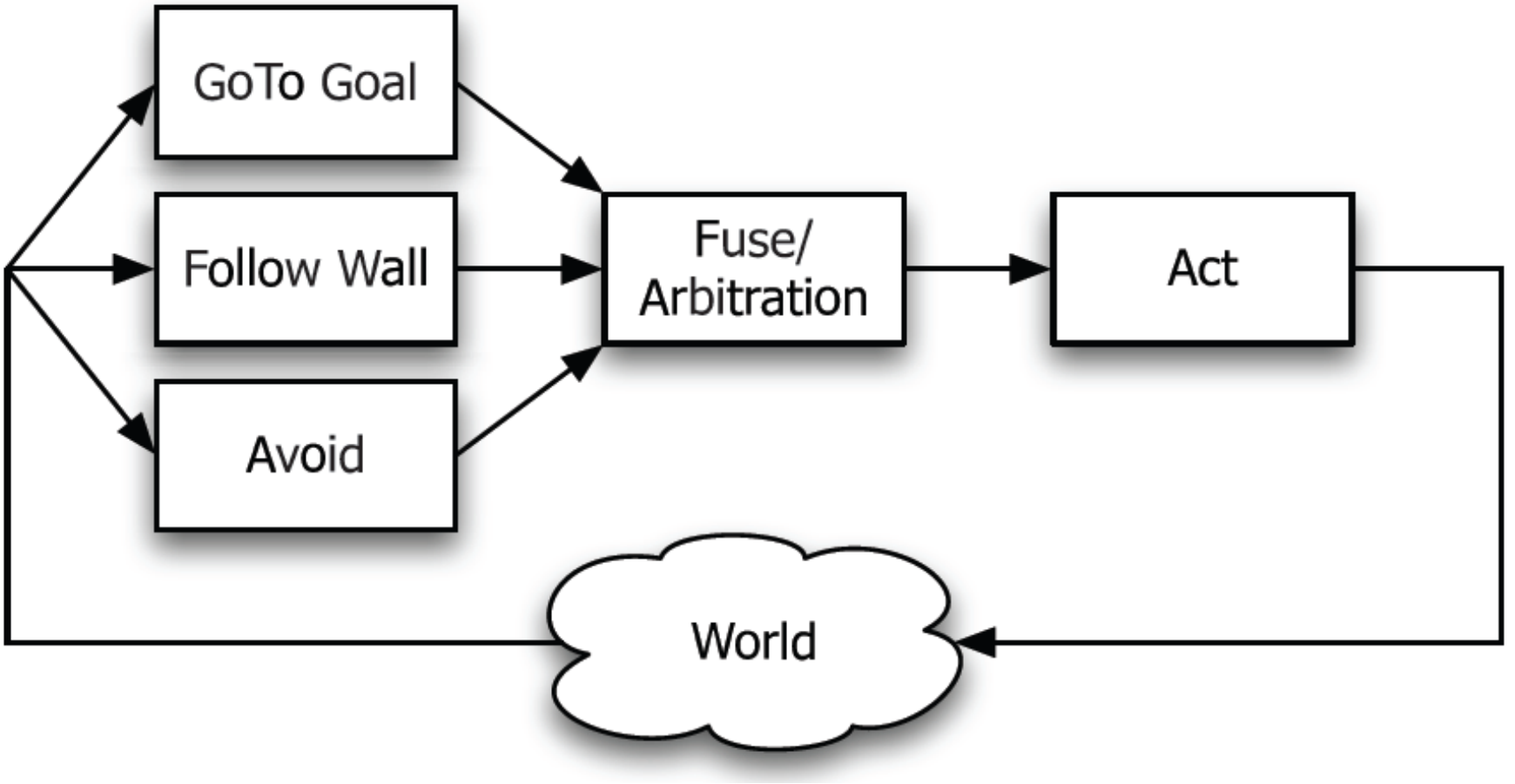
BLG456E
Robotics
Robot Software Architectures & Planning

- Temporal Decomposition
- Main Architecture Types
  - Deliberative.
    - Planning Intro
  - Reactive.
  - Behaviour-based.
  - Hybrid.
- Examples
  - Shakey Example.
  - ROS Example.

Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule
Coordination: http://ninova.itu.edu.tr/

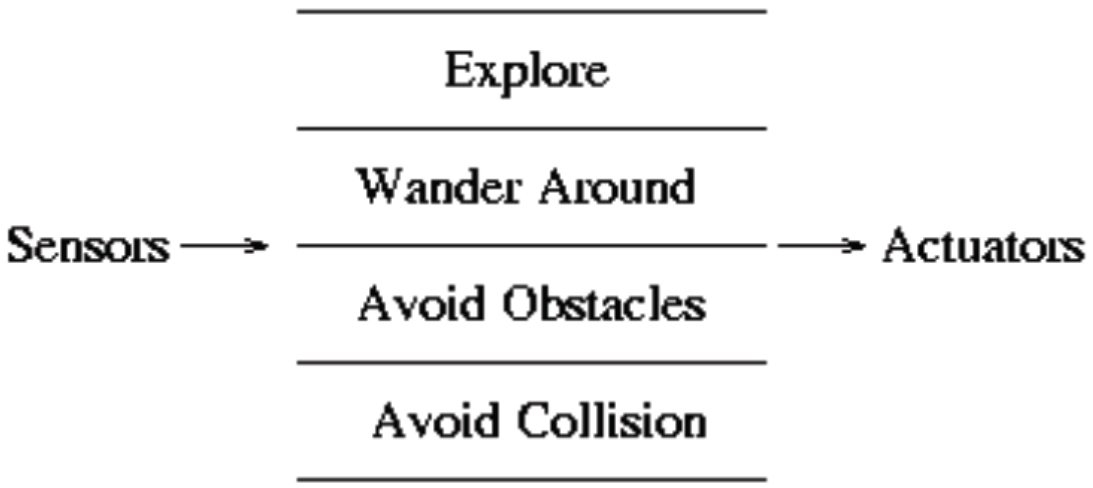
Considerably adapted from work by Dr. Sanem Sarıel-Talay

Parallel Decomposition - Behavioural



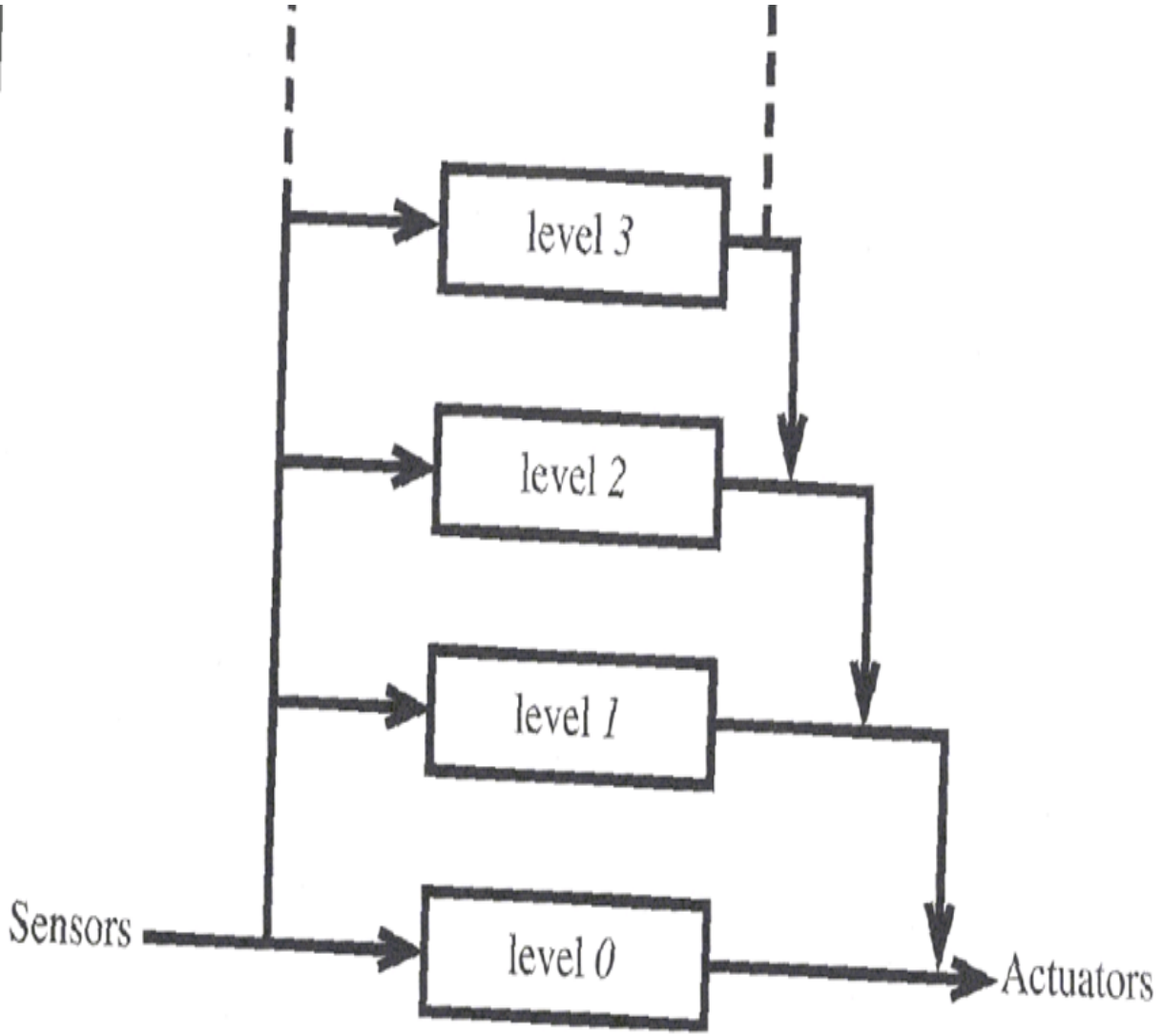
Behaviour-based Architecture

- Combine reactive "behavioural" modules.
- Behavioural modules can have internal state.
- Behavioural modules might deliberate independently.



Combining modules does not give any guarantees of correctness (not easy to formally model)

Layered (incremental) control: Subsumption architecture



- Higher level layers subsume roles of lower layers (take control).
- Layers below any level form an operational control system.

Mobile robot example

- Level 0: Avoid objects.
- Level 1: Wander aimlessly (without hitting).
- Level 2: “Explore”: head towards distant places.

------ ------ ------ ------ ------ ------ ------

- Level 3: Build map, plan routes.
- Level 4: Notice changes in map.
- Level 5: Reason about objects and tasks.
- Level 6: Formulate plans.
- Level 7: Reason about object dynamics.

Mobile robot example

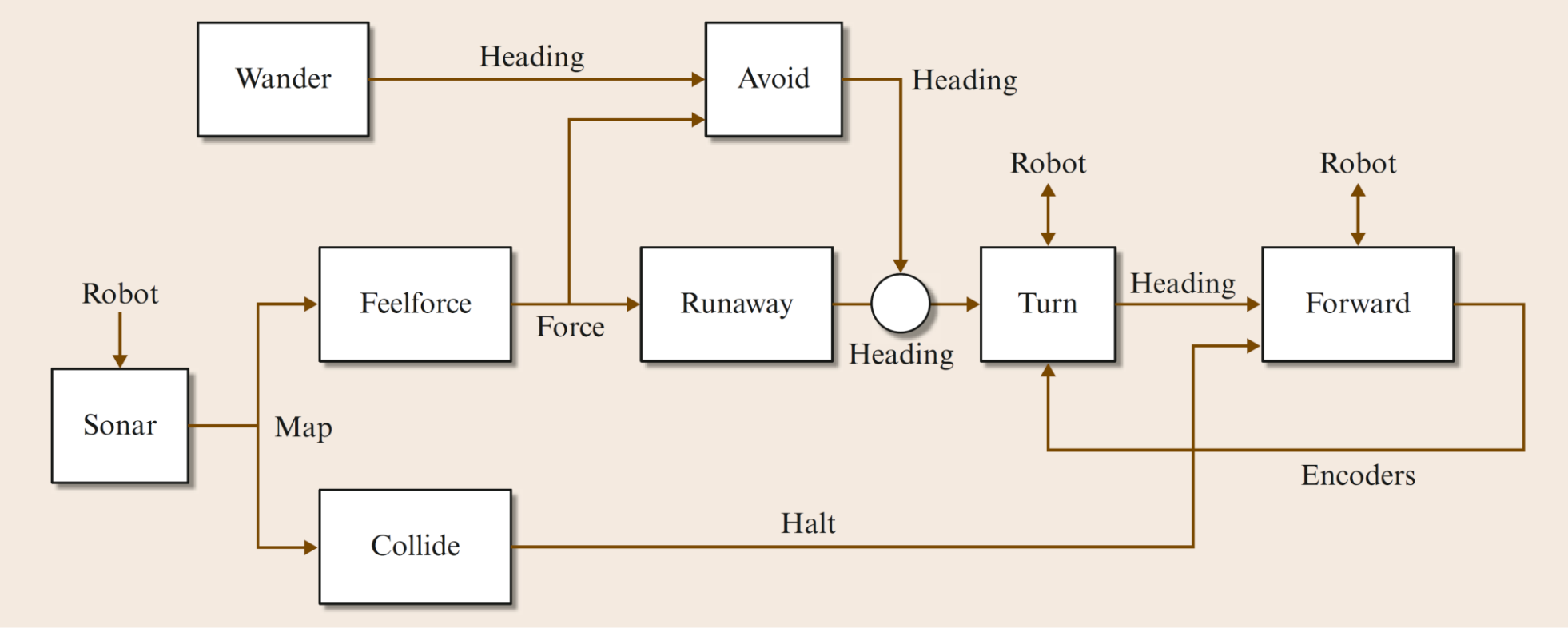
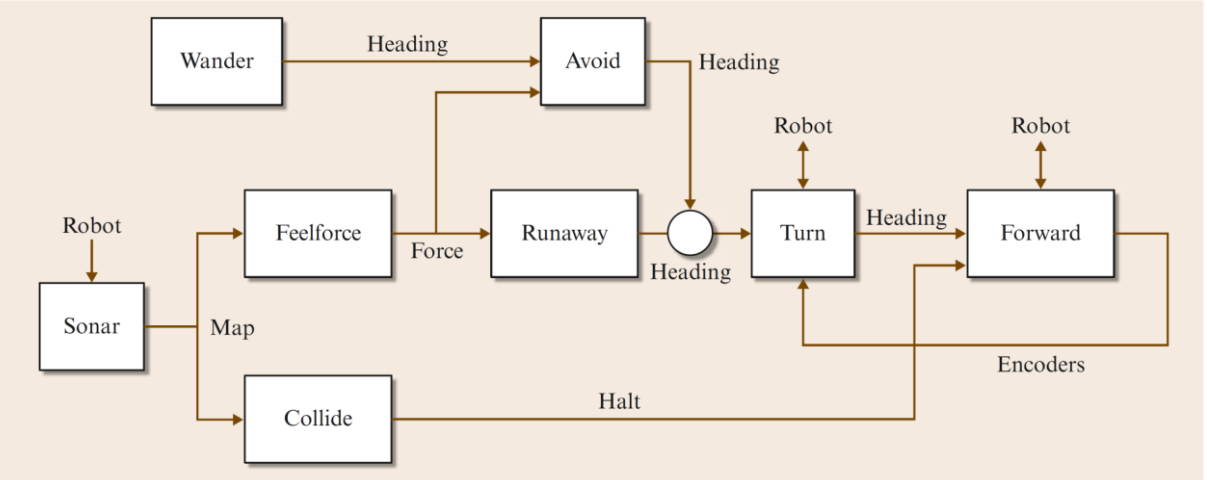


**Figure courtesy of Handbook of Robotics, Springer, 2008*

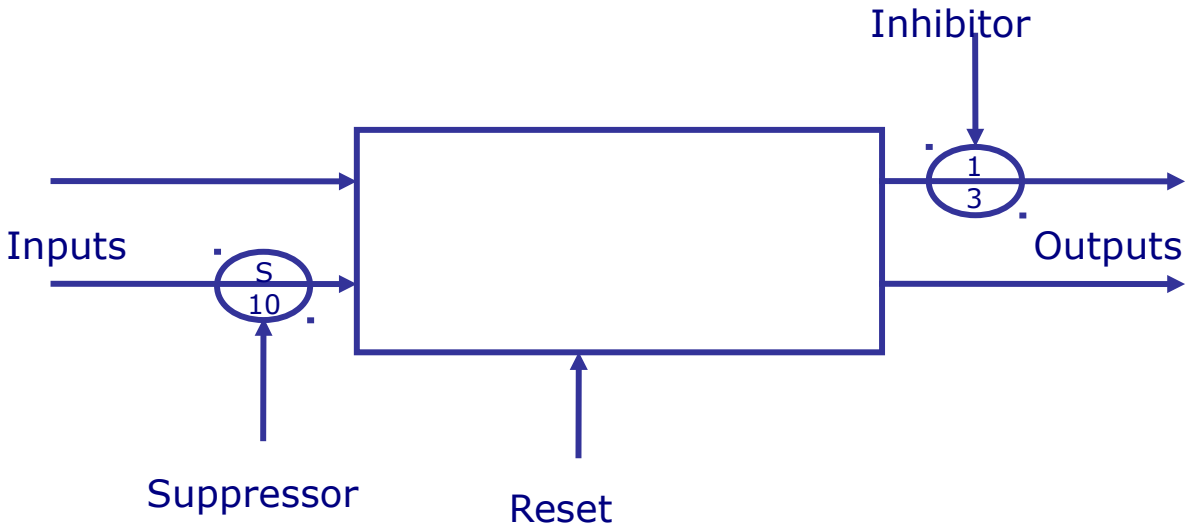
Mobile robot example

- sonar: Vector of sonar readings → robot-centered map of obstacles.
- collide: Map → detects objects ahead → halt signal.
- feelforce: Obstacle map → repulsive force.
- runaway: Repulsive force → heading.
- turn and forward: Feedback control from encoders.
- wander: Random movement generation.
- avoid: Reactive avoidance: Combine feelforce and wander.



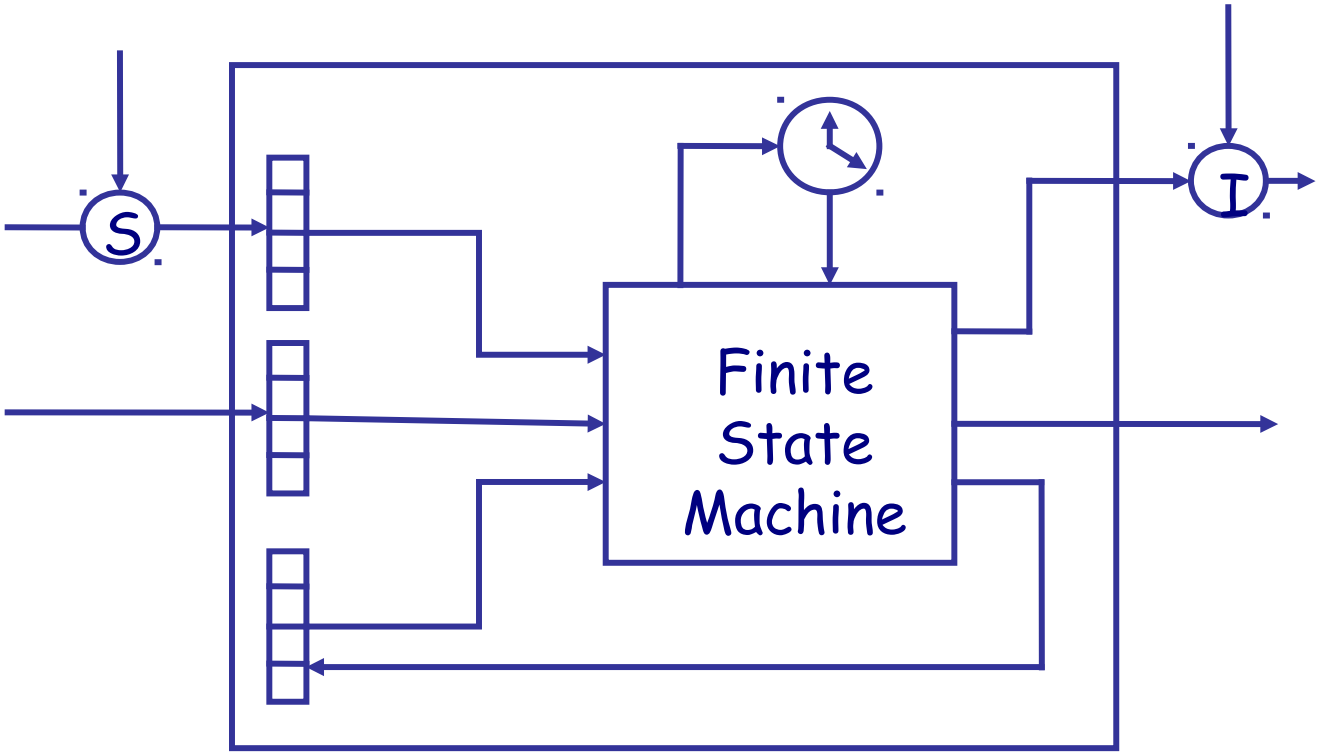
Subsumption Modules

- Input and output lines: “wires”.
- 1-message buffer.
- Suppression of inputs or inhibition of outputs.
- Can be reset.



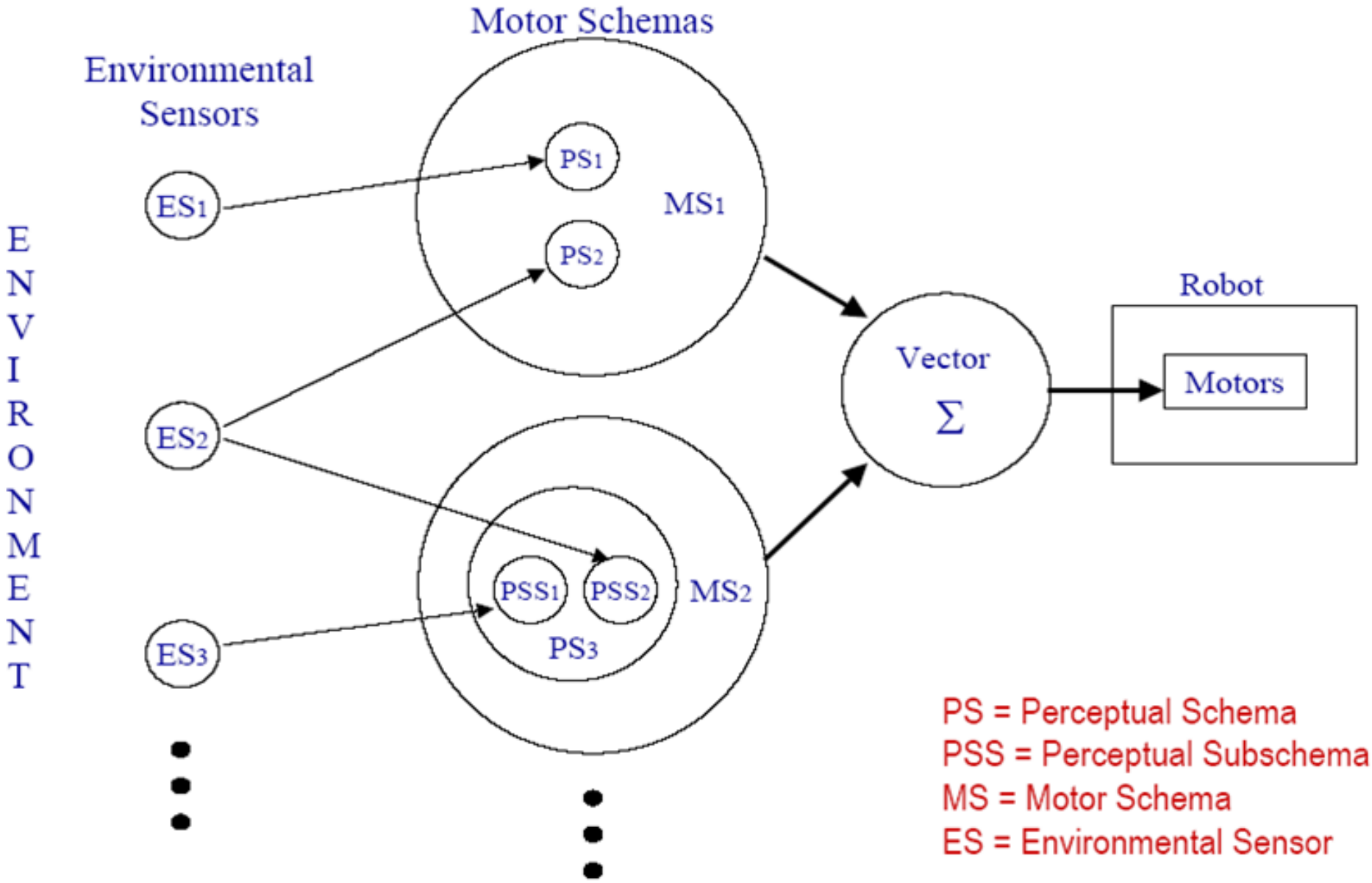
Augmented Finite State Machines

Message arrival/timer expiration → trigger a change in the state of FSM.



Alternative behavioural approach:

Robot Schemas



BLG456E
Robotics
Robot Software Architectures & Planning

- Temporal Decomposition
- Main Architecture Types
  - Deliberative.
    - Planning Intro
  - Reactive.
  - Behaviour-based.
  - Hybrid.
- Examples
  - Shakey Example.
  - ROS Example.

Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule
Coordination: http://ninova.itu.edu.tr/

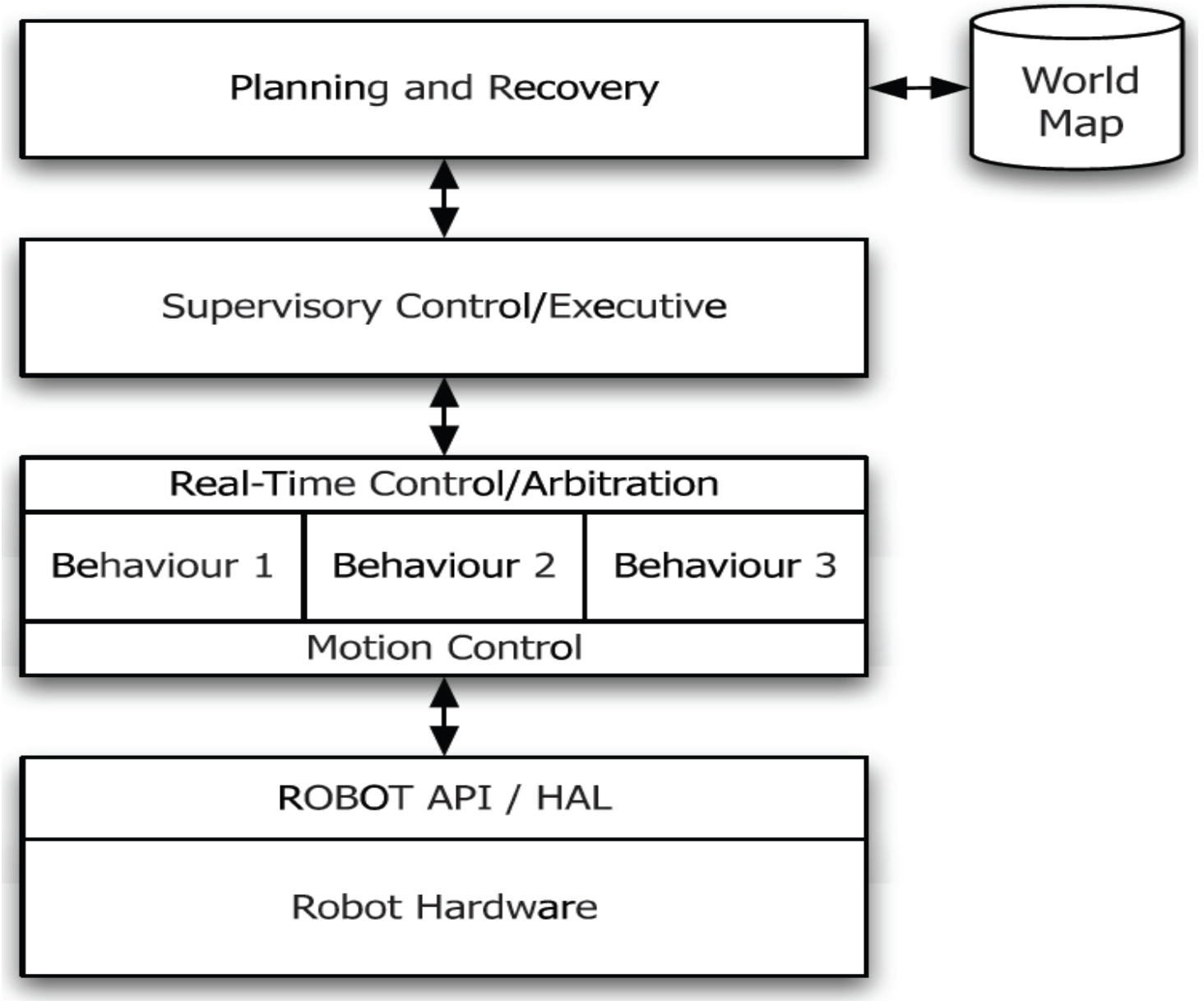
Considerably adapted from work by Dr. Sanem Sarıel-Talay

Hybrid Architectures

- State.
- Look ahead but react.
- Combine long and short time-scales.

- Interleaves deliberation (planning) with reactive control – e.g. moving obstacle avoidance.
- Very common nowadays.

Hybrid Architectures



BLG456E
Robotics
Robot Software Architectures & Planning

- Temporal Decomposition
- Main Architecture Types
  - Deliberative.
    - Planning Intro
  - Reactive.
  - Behaviour-based.
  - Hybrid.
- Examples
  - Shakey Example.
  - ROS Example.

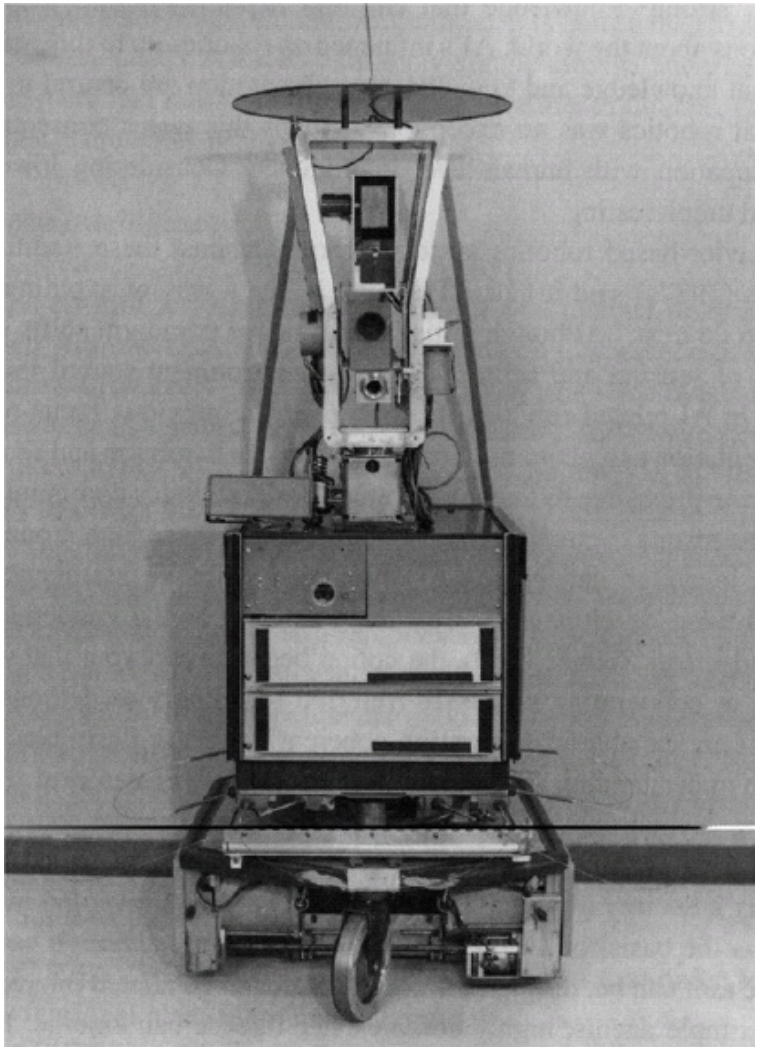
Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule
Coordination: http://ninova.itu.edu.tr/

Considerably adapted from work by Dr. Sanem Sarıel-Talay

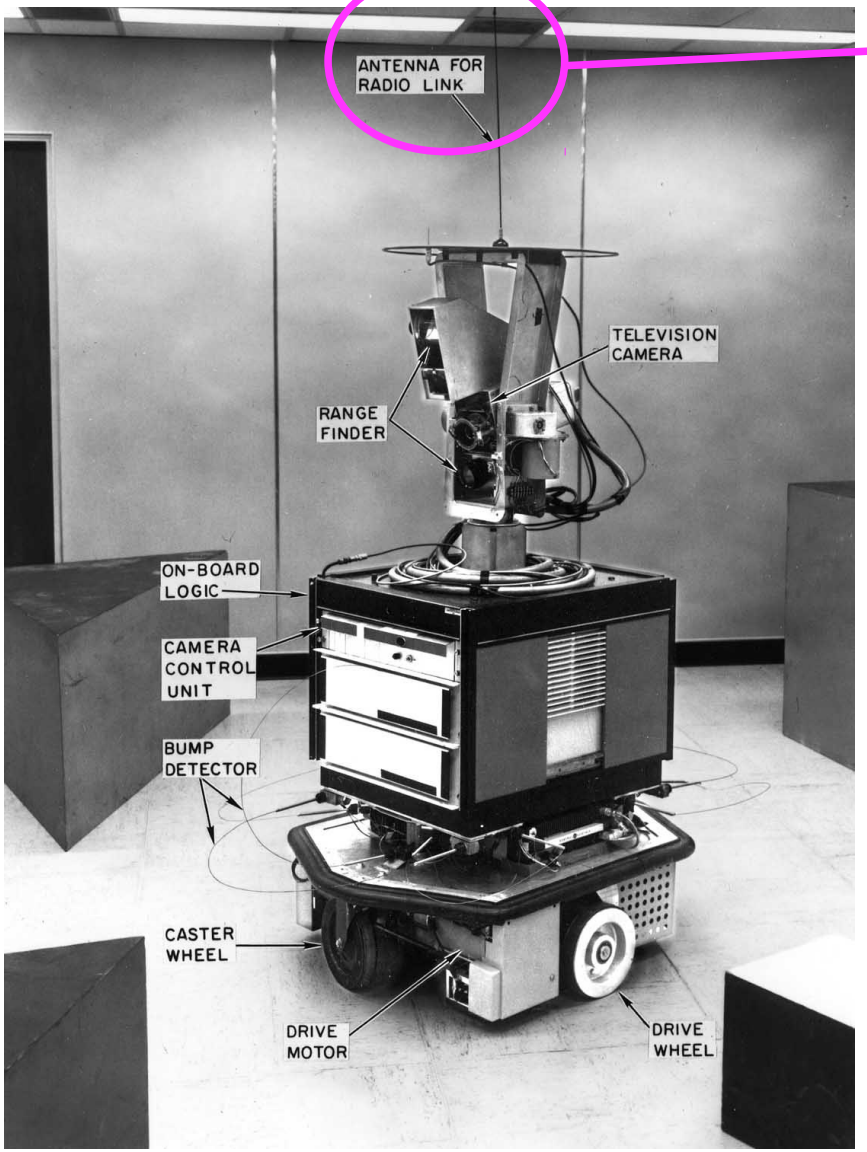
Early robots - Shakey

Shakey (SRI), 1960's

- Sensors:
  - VidiconTV camera
  - Optical range finder
  - Whisker bump sensors
- Environment: Office with specially colored and shaped objects
- STRIPS planner: developed for this system

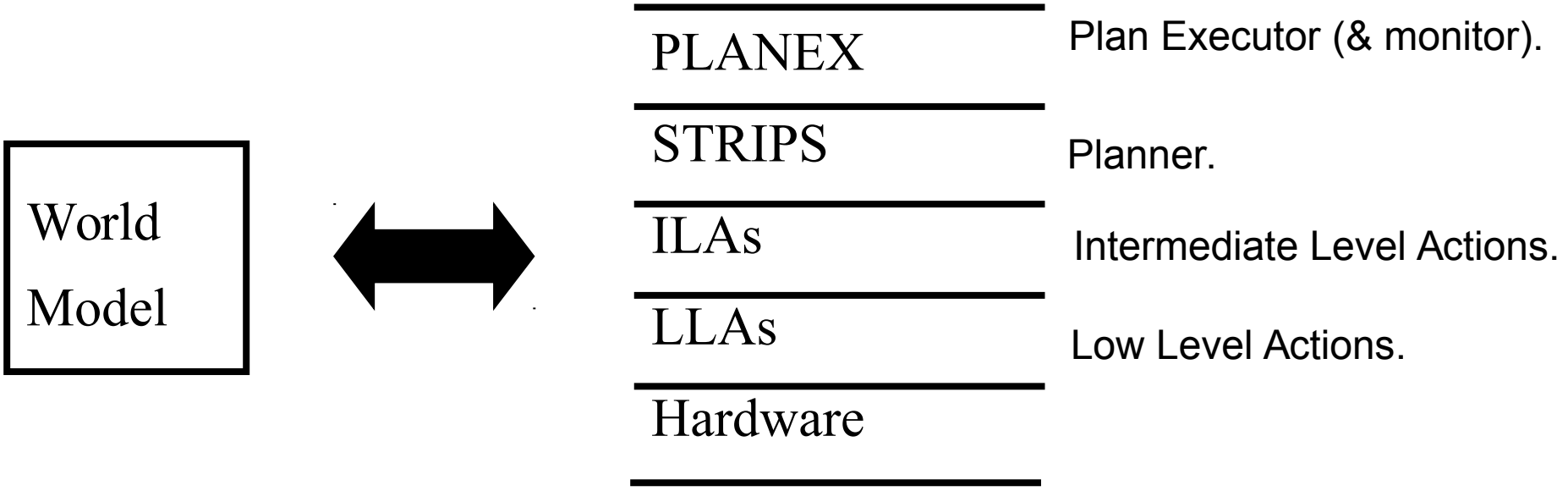


Shakey the robot



To controlling computer

Shakey outline



- Central representation (world model)
- Logic based representation.
- Components communicate via world model.
- Error recovery at several levels (e.g. plan executor, low level actions).

(strictly speaking, a hybrid approach)