Hybrid architectures

IAR Lecture 6

Barbara Webb

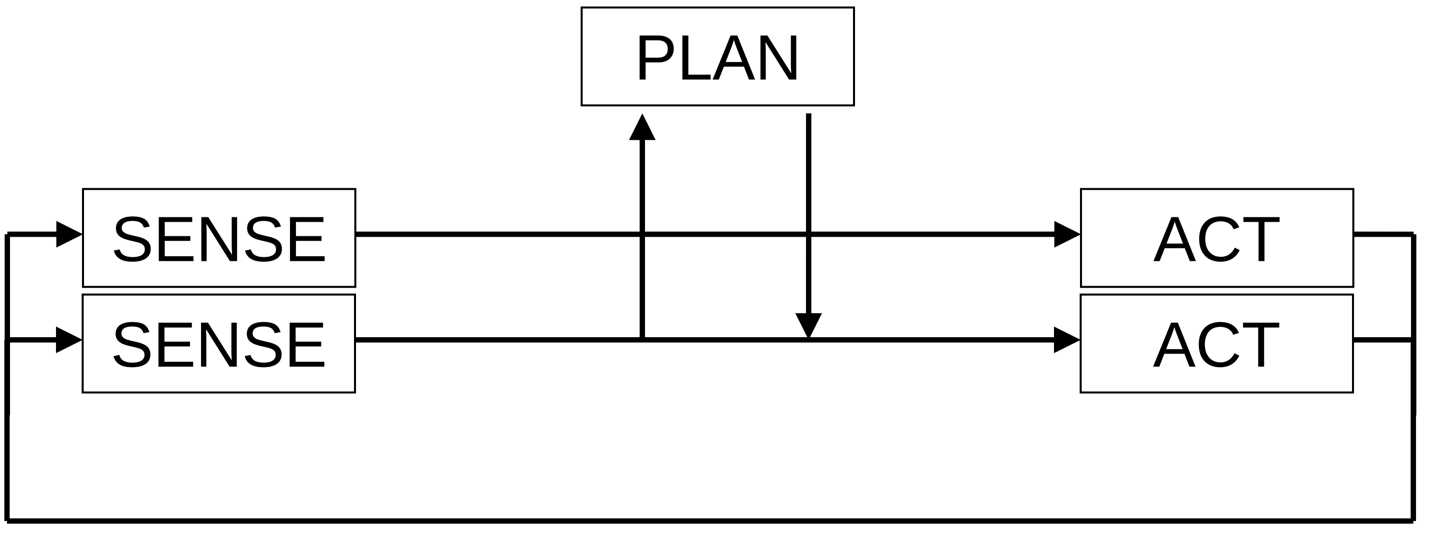
Behaviour Based: Conclusions

- But arbitrary and difficult to design emergent behaviour for a given task.
  - Architectures do not impose strong constraints
- Options?
  - Build up toolbox of techniques
  - Use learning or evolutionary methods
  - Copy existing systems (i.e. biology)
  - Formalise interactions as dynamical systems
- Difficult to do some traditional (and useful) tasks.
  - Increasingly common to adopt ‘hybrid’ approach, e.g. classical planner operating on top of basic behaviours

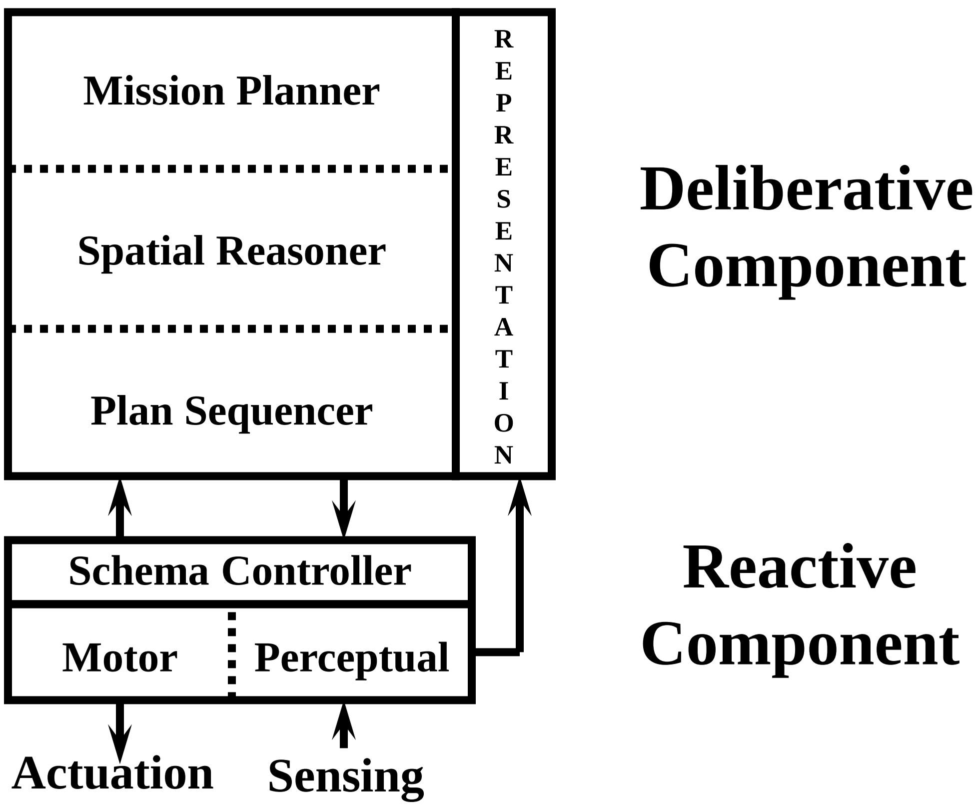
Can we get the best of both worlds?

- Behaviour-based architectures, using combinations of reactive controllers, rather than reasoning over world models, seem to be robust, real-time, modular – good for robot control.
- But may still want world models and traditional reasoning for path planning, map making, performance monitoring, problem solving…
- And have introduced new problem of how to compose behaviour interactions
- **Solution:** have the robot itself intelligently select, sequence or assemble behaviours to achieve a task

Introduce high-level capabilities without losing low-level advantages



AuRA: Autonomous Robot Architecture (Arkin, 1997)



AuRA: Autonomous Robot Architecture (Arkin, 1997)

- **Mission planner:** determines current goal
- **Spatial reasoner:** produces sequence of path legs to traverse
- **Plan sequencer:** translates each path leg into a set of behaviours – linking sensor and motor schemas
- **Schema manager:** starts, stops and monitors schema execution

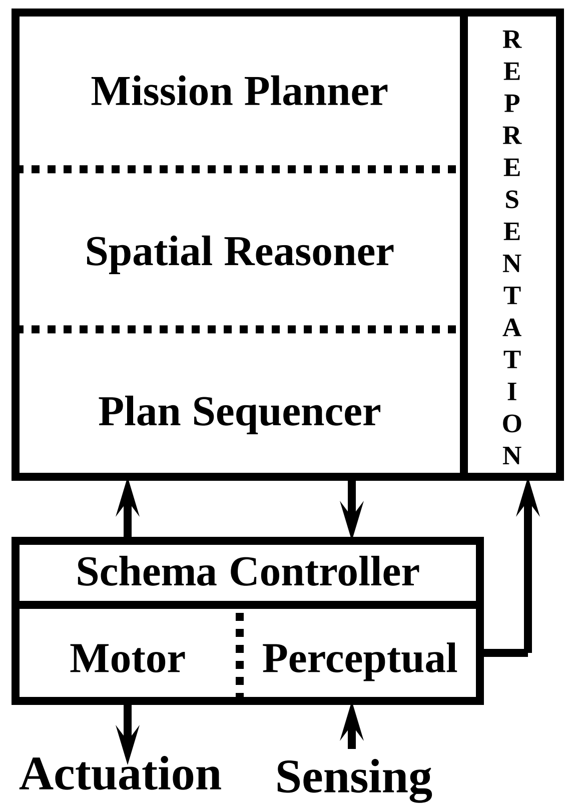
As for the basic schema architecture (see lecture 5), the robot's action is the weighted sum of motor schema vector outputs

AuRA: Autonomous Robot Architecture (Arkin, 1997)

- Once reactive execution begins, deliberation is de-activated, unless detect failure.
- Failure recovery works up the hierarchy:
  - Revise plan sequence based on input so far
  - Have spatial reasoner devise new route
  - Have mission planner revise goals

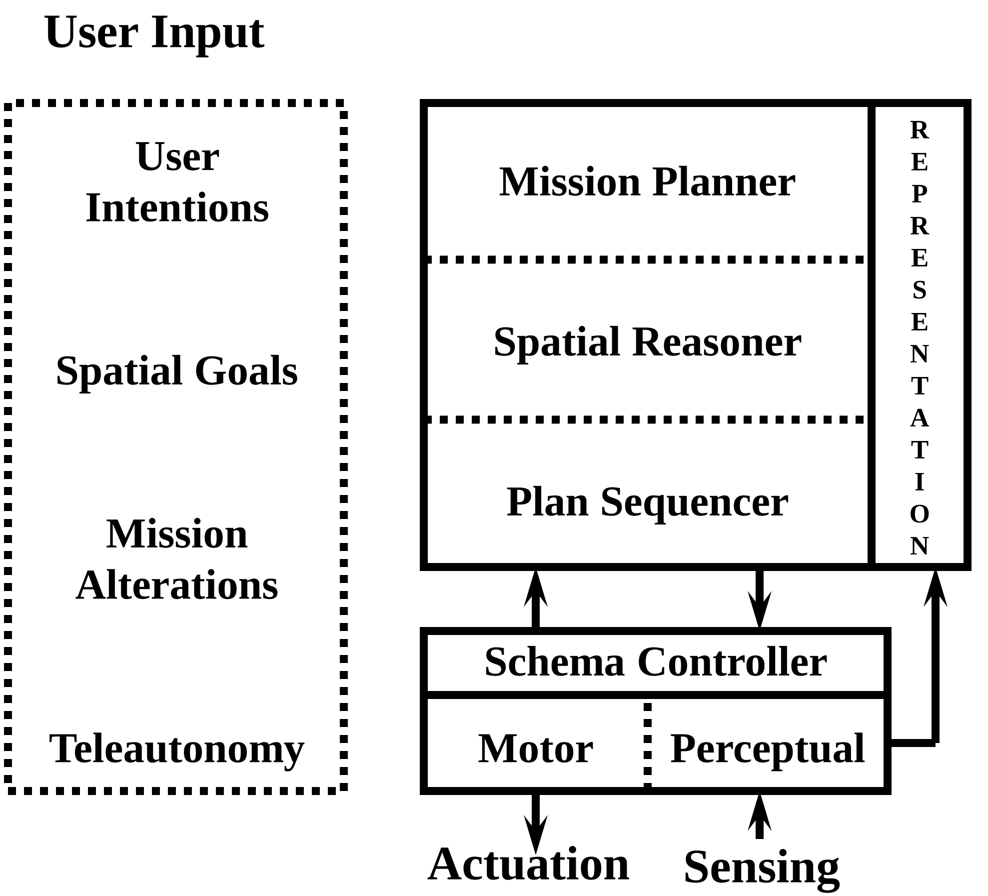
AuRA: Autonomous Robot Architecture (Arkin, 1997)

Potential to substitute improved methods at each level independently



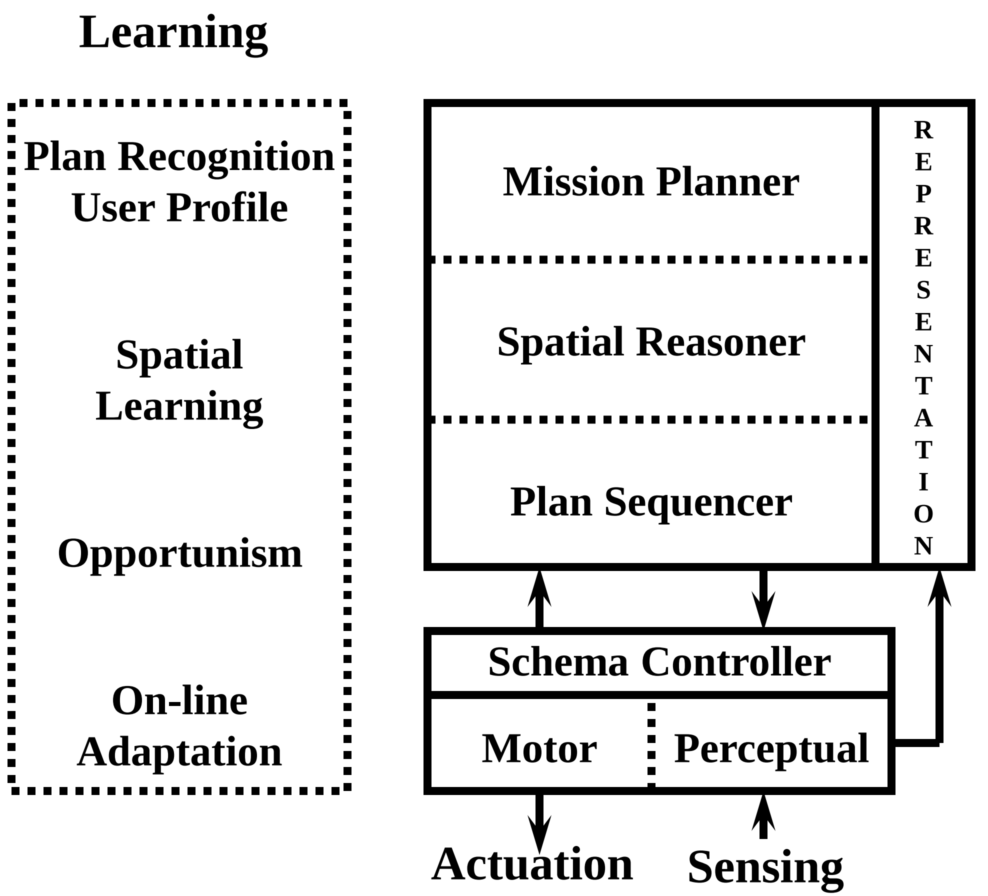
AuRA: Autonomous Robot Architecture (Arkin, 1997)

Rich options for user interaction

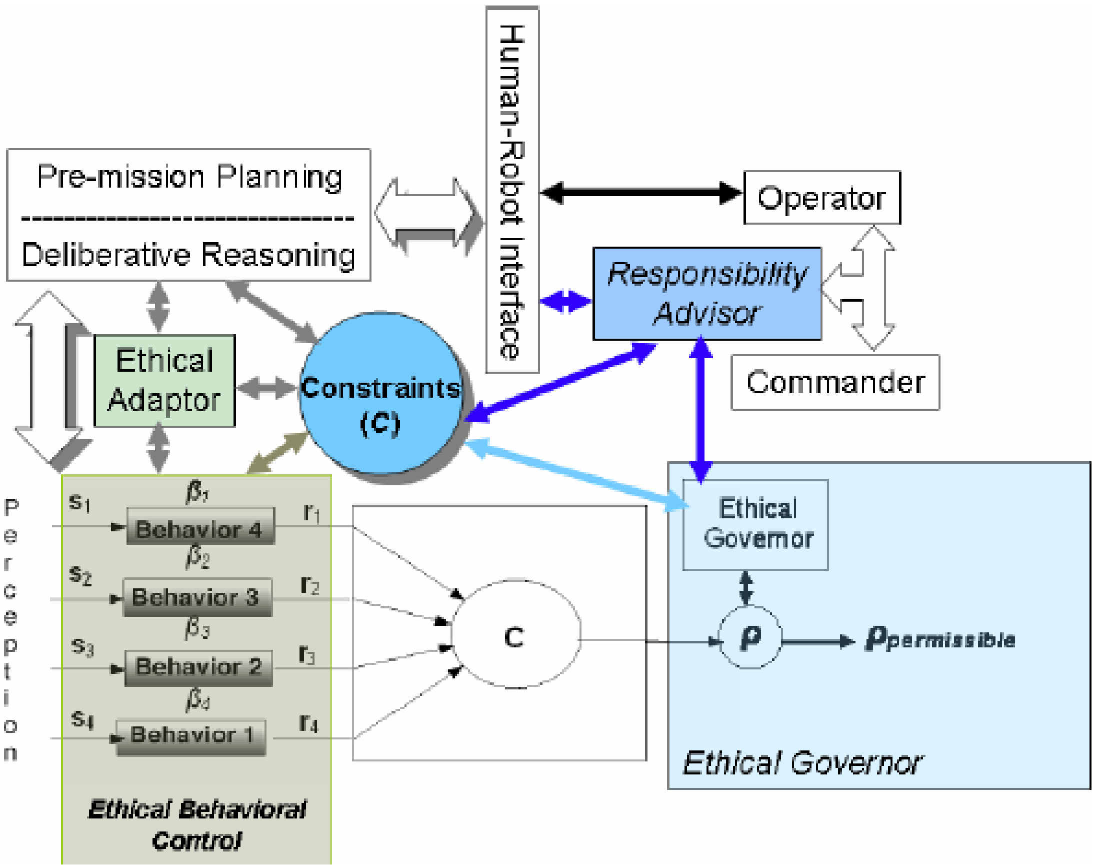


AuRA: Autonomous Robot Architecture (Arkin, 1997)

Can introduce learning mechanisms at different levels

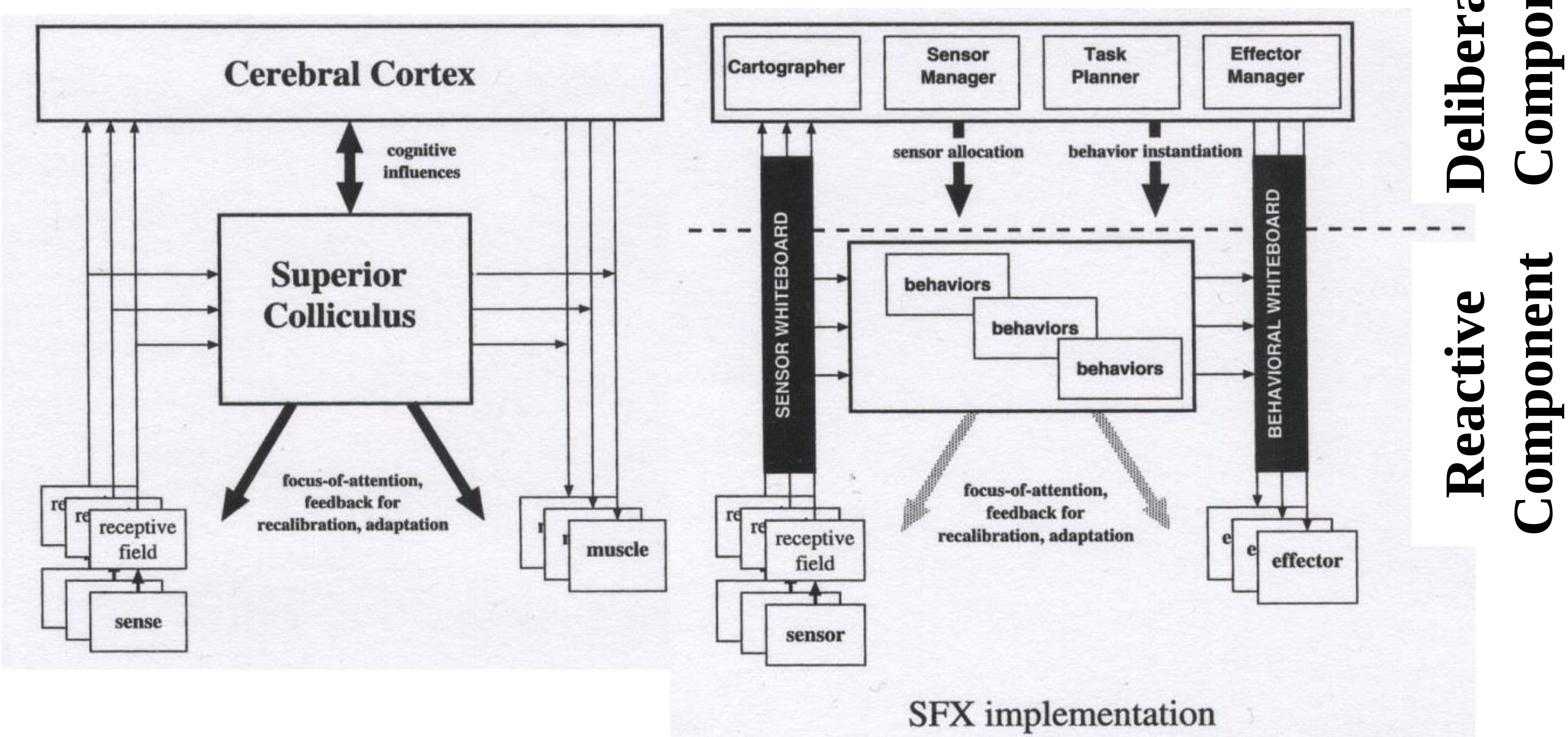


Embedding ethical control (Arkin, 2009)



SFX: sensor fusion effects (Murphy, 2000)

Biomimetic inspiration for hybrid architecture

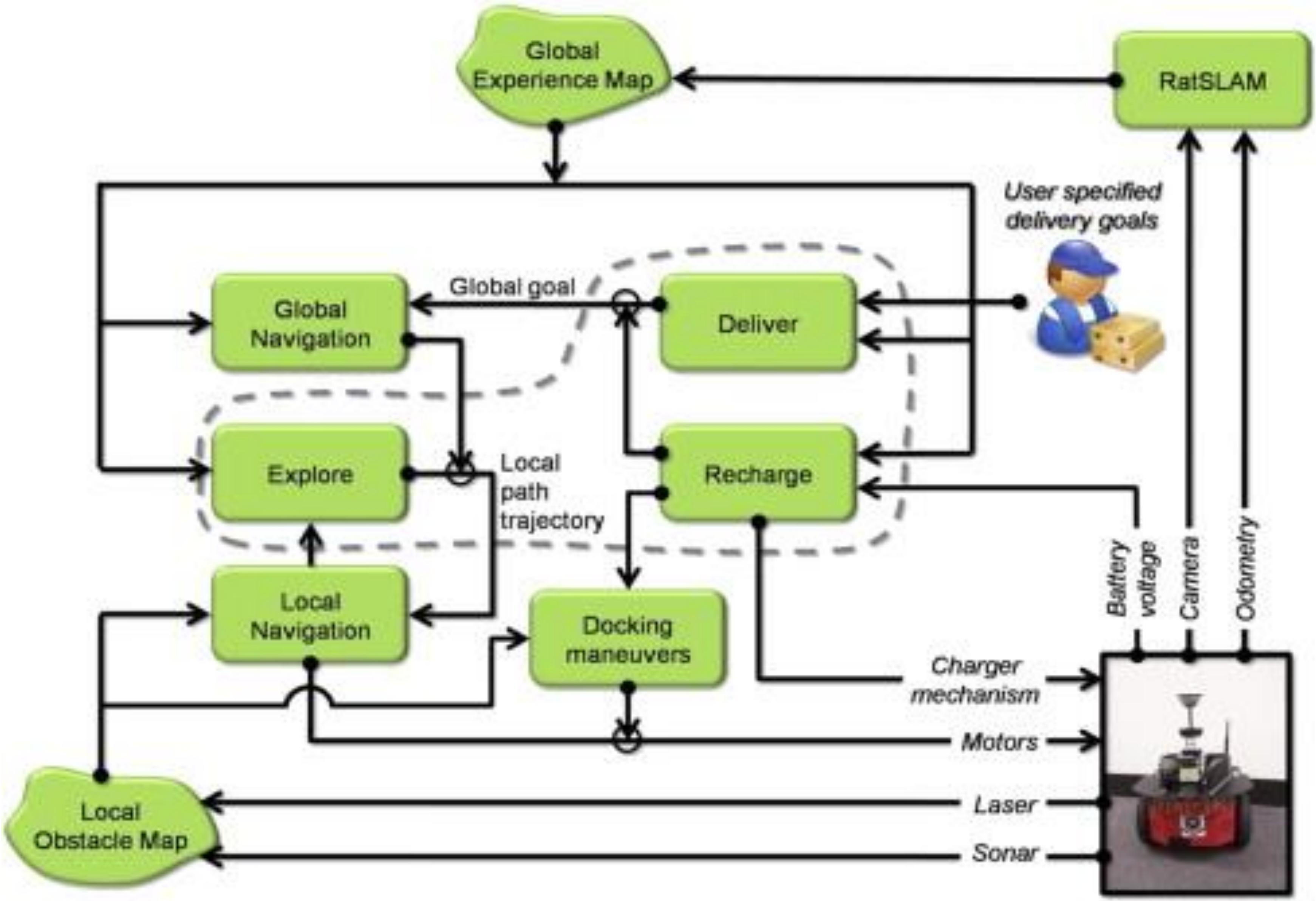


Low level behaviours can support abstractions that simplify planning

- Planning is difficult because of uncertainty
  - Precise location of robot and things around it
  - Accuracy/repeatability of robot actions
  - Adequacy/reliability of sensor data
- **Solution:** If can off-load uncertainty handling to low level behaviours, planning becomes much more tractable

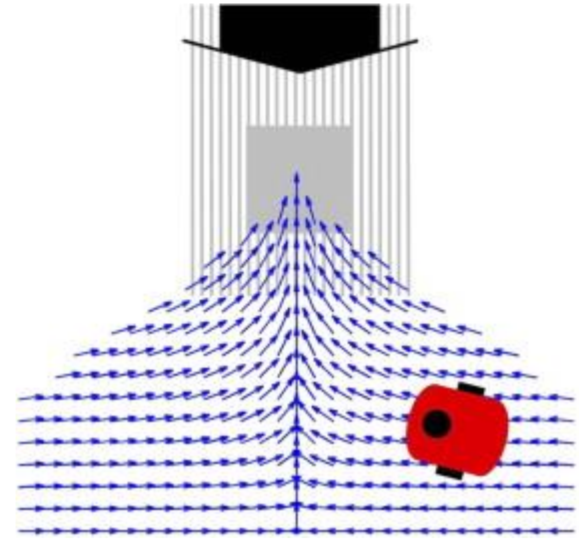
Hybrid control for navigation (Milford & Wyeth 2010)

- Use of low-level immediate sensing to build high-level map.
- Effective if same control system is used in both creating and using this world representation.
- Robot placed in new office building environment, learns map (including its charger location) then makes deliveries – continuously operating over two weeks in dynamic environment.

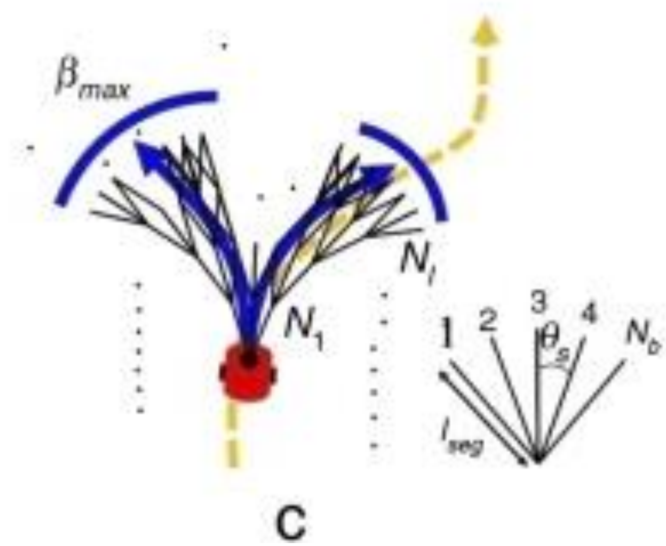

Global Experience Map
RatSLAM
User specified delivery goals
Global goal
Global Navigation
Deliver
Explore
Local path trajectory
Recharge
Local Navigation
Docking maneuvers
Battery voltage
Camera
Odometry
Charger mechanism
Motors
Laser
Sonar
Local Obstacle Map

Milford & Wyeth (2010)

- Reactive docking procedure (potential field method) relies on assumption that recharge and global navigation have brought robot to suitable starting position



- Local obstacle avoidance generates safe trajectories, clusters these and takes midpoints. Comparison to trajectory desired by explore or global navigation determines choice of robot action



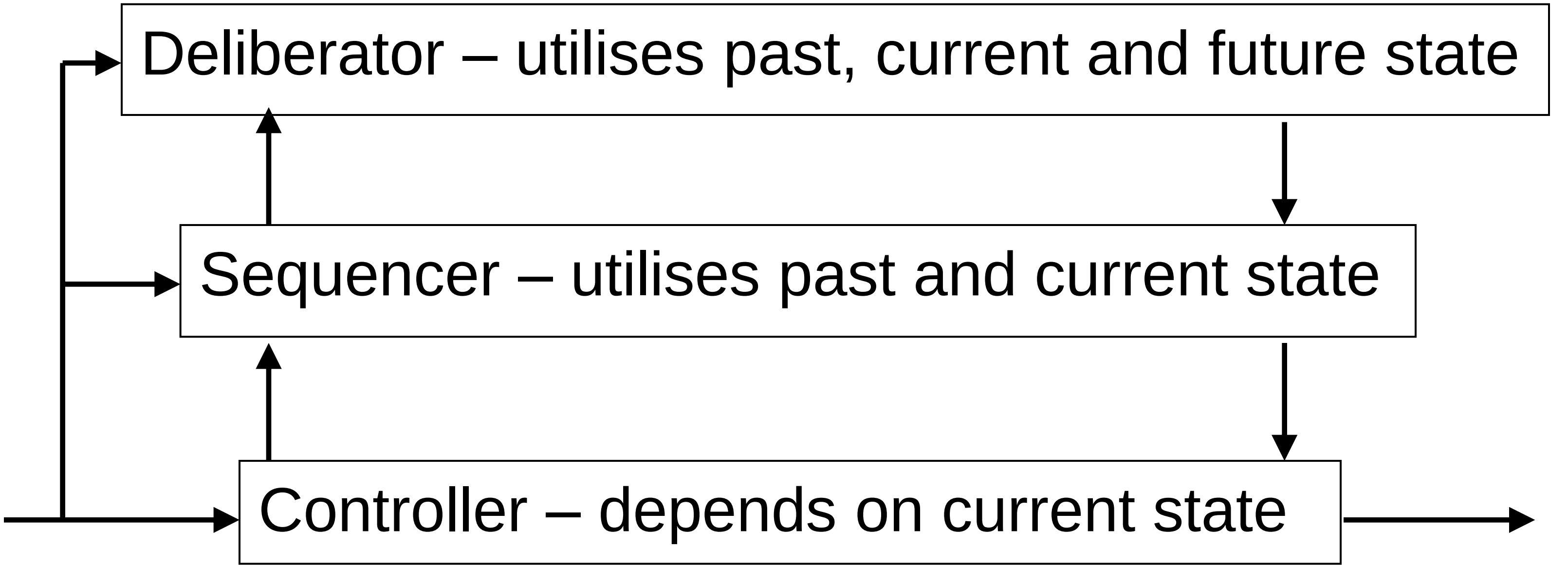
Temporal task decomposition

- One basic problem with the traditional sense-plan-act was that planning is slow, and the internal state gets out of synchrony with reality
- Reactive solution of minimal internal state increases risk of mistakes due to sensor error
- **Solution:** Use parallel layers to maintain internal states appropriate to the different speeds of processing

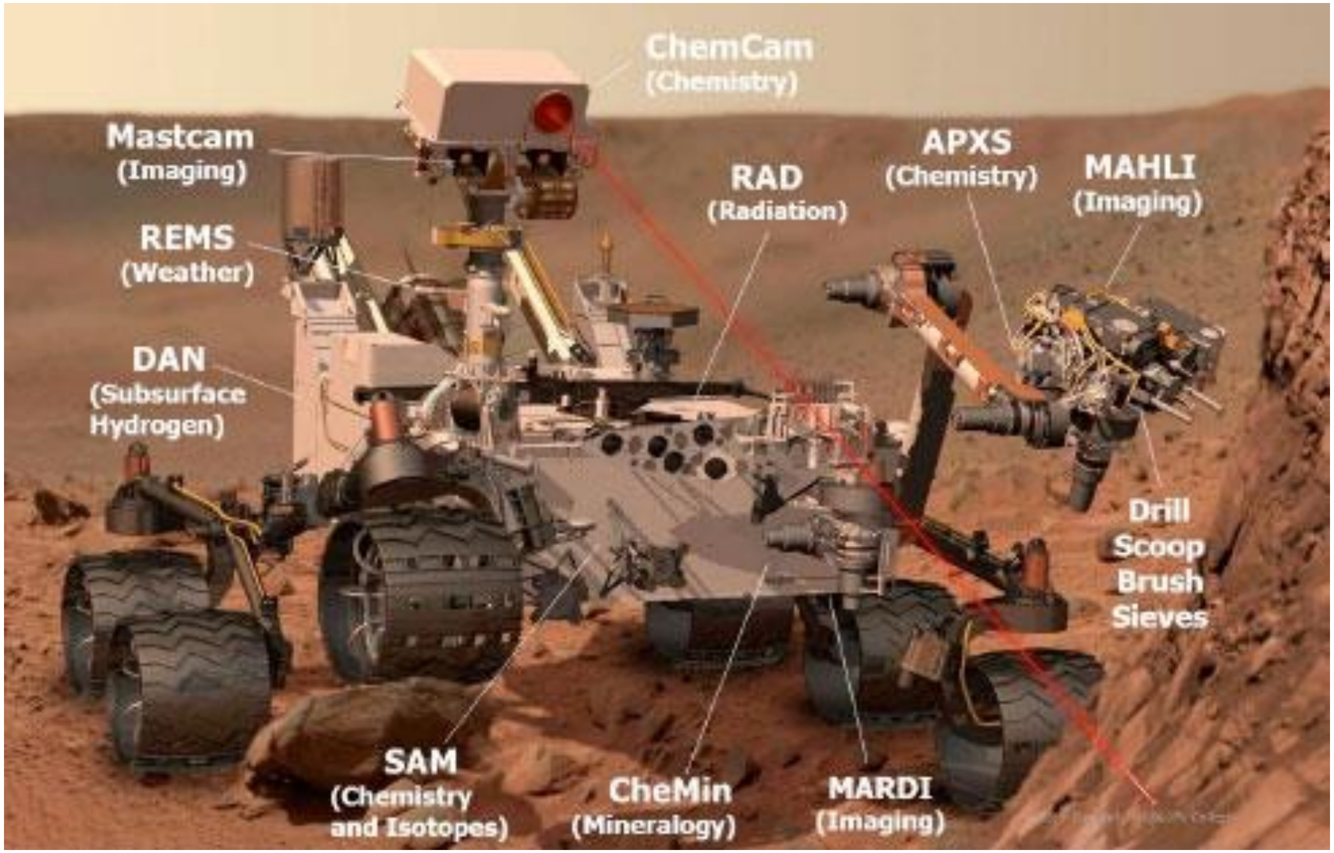
Temporal task decomposition

Offline/episodic reasoning	Map-based path planning	0.01Hz
Strategic decisions	Range-sensor based obstacle avoidance	1 Hz
Real time control	PID speed	100Hz

Three-layer architecture (Gat 1998)



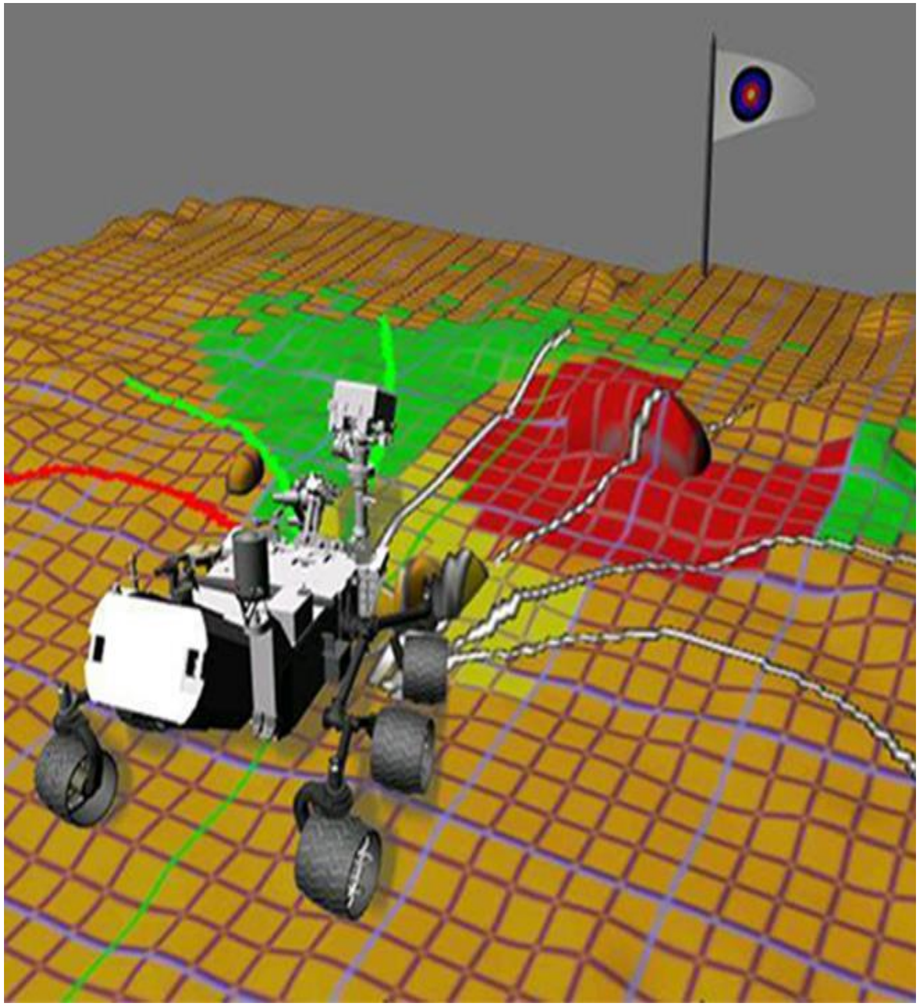
Also called 'Three Tier' or 3T architecture; used by NASA



Three-layer architecture (Gat 1998)

Controller:

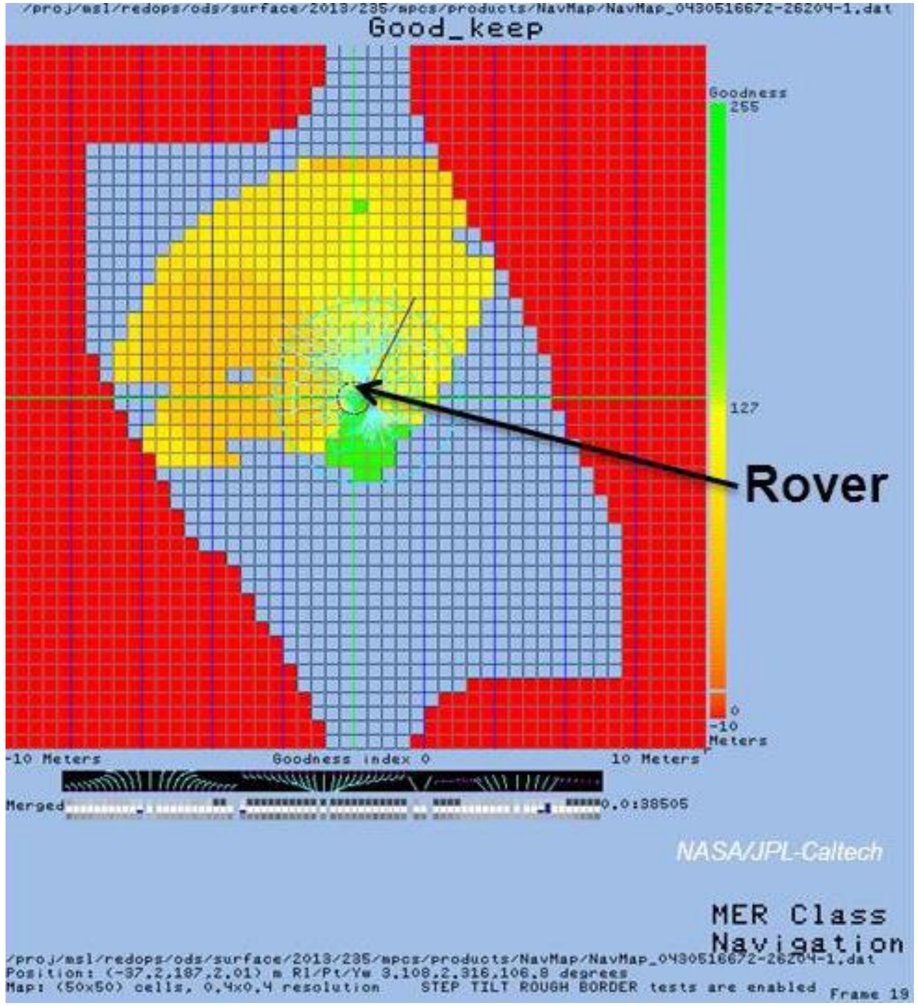
- Behaviour library of handcrafted sensorimotor transfer functions (e.g. wall follow – with appropriately oriented sonar, servo on distance to wall)
- Must be fast enough for stable closed loop control
- Should avoid internal states, or use ephemeral states (e.g. filter sonar by rejecting sudden increases in distance)
- Internal state should not introduce behavioural discontinuities (these should be new behaviours)



Three-layer architecture (Gat 1998)

Sequencer/Executive:

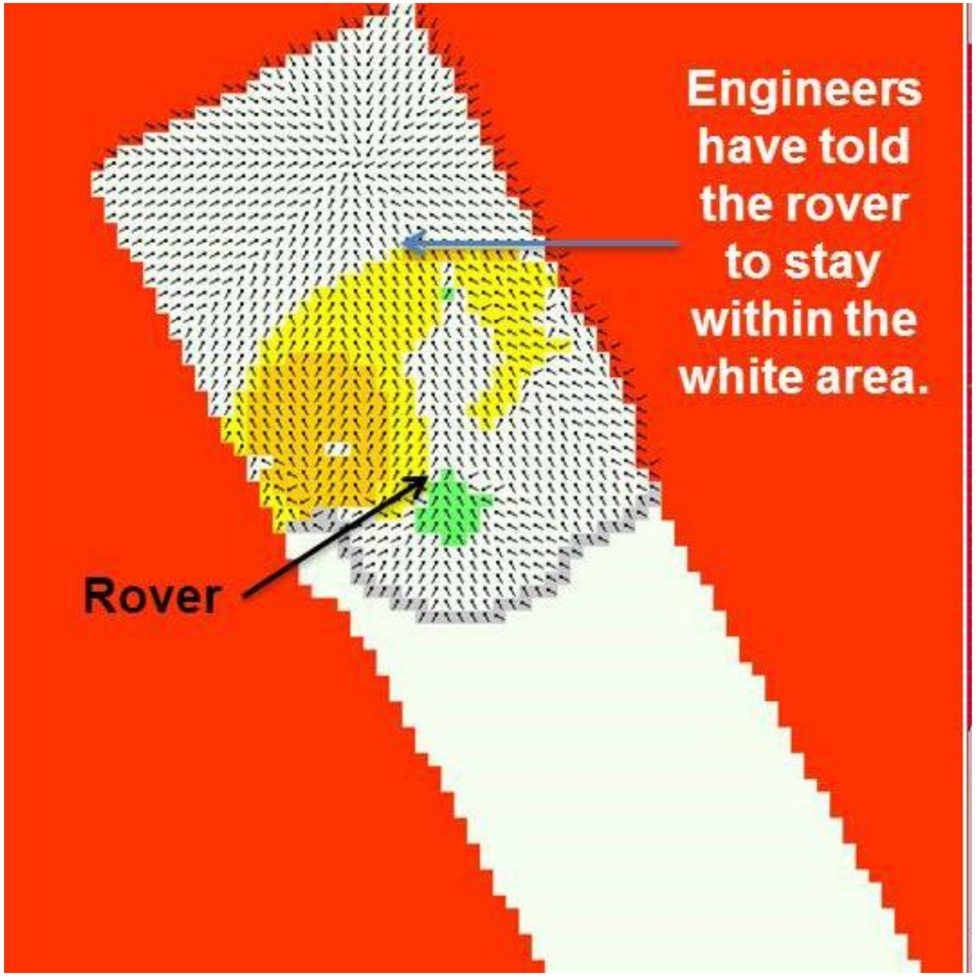
- Selects which behaviours are active
- Not just following fixed sequence but responds to situation in interpreting any plan
- Does not search into future to decide action
- Should not take a long time, relative to the environment and the behaviours
- E.g. in maze, first find wall; then follow it to confirm is wall not obstacle; then remember sequence of turns


/proj/msl/redops/ods/surface/2013/235/mpcs/products/NavMap/NavMap_0430516672-26204-1.dat
Good_keep
Goodness
255
127
0
-10
Meters
Rover
-10 Meters
Goodness index 0
10 Meters
Merged
0.0:38505
NASA/JPL-Caltech
MER Class
Navigation
/proj/msl/redops/ods/surface/2013/235/mpcs/products/NavMap/NavMap_0430516672-26204-1.dat
Position: (-37.2,187.2,2.01) m Rl/Pt/Yw 3.108,2.316,106.8 degrees
Map: (50x50) cells, 0.4x0.4 resolution STEP TILT ROUGH BORDER tests are enabled
Frame 19

Three-layer architecture (Gat 1998)

Deliberator:

- Does any time consuming computations
  - Typically plans
  - May include complex sensor processing (e.g. vision)
- Running as separate thread/on separate processor
- Can either produce a plan for the sequencer, or respond to sequencer requests for deliberation
- E.g. in maze, compare sequence of turns to stored map; once recognise the location, plan moves to reach exit


Engineers have told the rover to stay within the white area.
Rover

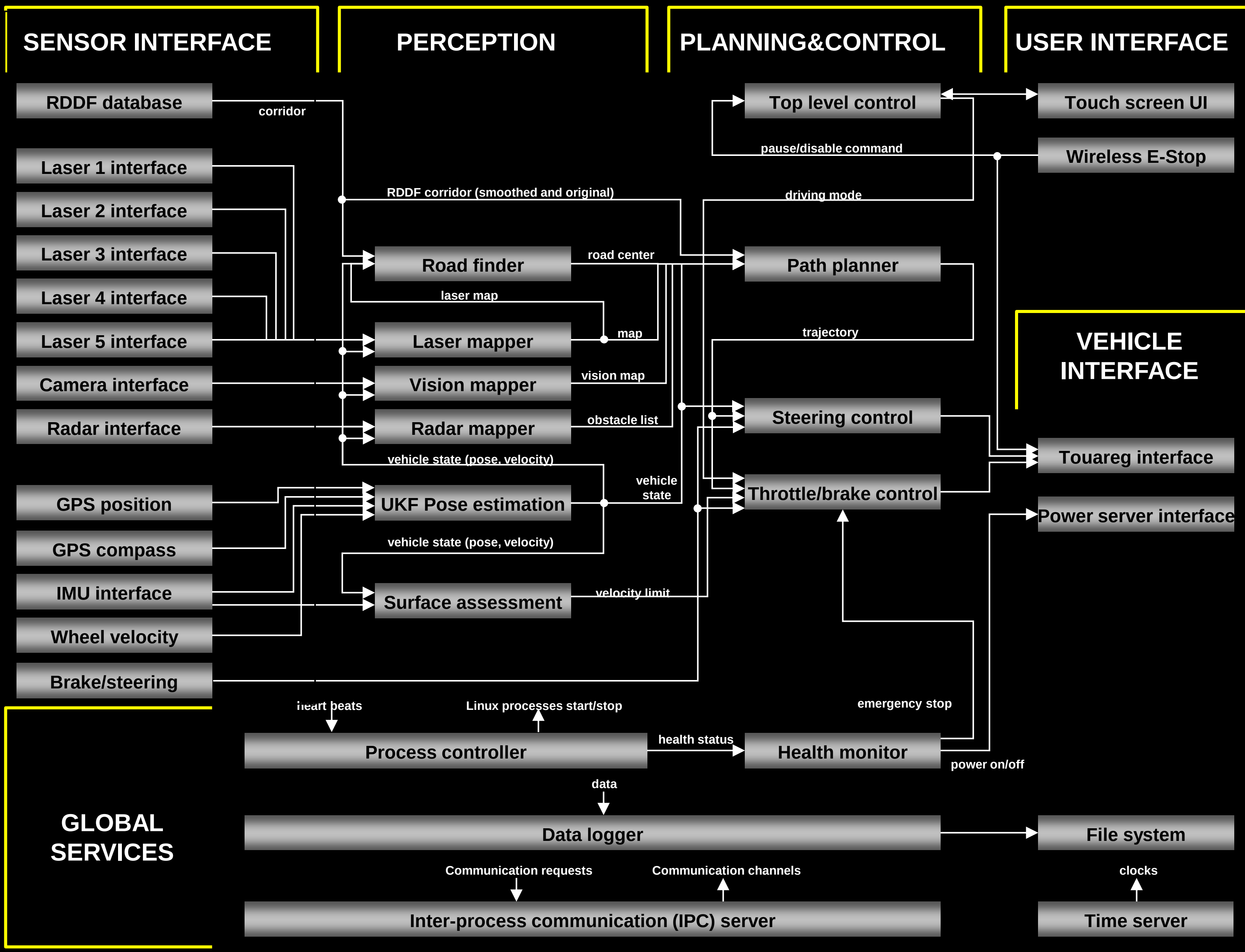
Three-layer architecture (Gat 1998)

- "*lines between the components of the three layer architecture can be blurred to accommodate reality*"
- "*If, as seems likely, there is no One True Architecture, and intelligence relies on a hodge-podge of techniques, then the three layer architecture offers itself as a way to help organise the mess*"

Hybrid architectures

- Most robotic approaches today use some form of hybrid architecture, combining low-level reactivity with higher level reflection, in multiple parallel modules
- Many variants, and tendency towards ‘hodge-podge’ solutions in real applications
- Substantial speed up in planning methods has made it more viable to include within real time control loops
- Also improvements in sensor algorithms & state estimation methods, so internal models more reliable
- Example: Stanley, winner of 2005 DARPA Grand Challenge (Thrun et al 2006)

Stanley Software Architecture



References:

Arkin, R.C. & Balch, T.R. (1997) AuRA: principles and practice in review. J. Exp. Theor. Artif. Intell. 9(2-3): 175-189

Arkin, R.C. (2009) Governing lethal behavior: embedding ethics in a hybrid deliberative/reactive control architecture. Tech. Report GIT-GVU-07-11 www.cc.gatech.edu/ai/robot-lab/online-publications/formalizationv35.pdf

Murphy, R.R. (2000) Introduction to AI Robotics MIT Press

Gat, E. (1997) On three-layer architectures. In D. Kortenkamp, R. P. Bonnasso, and R. Murphy, editors, Artificial Intelligence and Mobile Robots. MIT/AAAI Press, 1997. http://citeseer.ist.psu.edu/gat97threelayer.html

Milford, M. & Wyatt, G. (2010) Hybrid robot control and SLAM for persistent navigation and mapping. Robotics and Autonomous Systems 58:1096-1104.

Thrun, S. et al (2006) Winning the DARPA Grand Challenge, Journal of Field Robotics, 23(9):661-692