Abstract agent architectures : 'standard agent'

- S = {s_1, s_2 , …} environment states
- A = {a_1, a_2, …} actions
- Abstractly, an agent can be viewed as a function:

action: S* → A

mapping sequences of environments to actions

188

Abstract agent architectures

- Nondeterministic behavior environment:

env: S × A → P(S)

- History: interaction agent and environment:

h: s_0 → s_1 → s_2 → …

189

Several types of agents

- Behavior of a *purely reactive agent*:

action: S → A

- Agent with *perception*:

see: S → P

action: P* → A

190

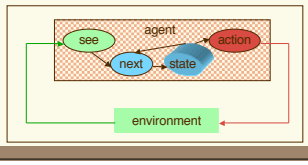
Several types of agents (2)

- Agents with *a(n internal) state*:

see: S → P

action: I → A

next: I × P → I



191

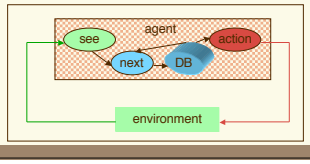
Several types of agents (3)

- Deliberative agent

see: S → P

action: DB → A

next: DB × P → DB



192

Agent architectures

- Deliberative systems
  - Based on *physical symbol system hypoth.:*
  - *'GOFAI approach'*
- Reactive systems
  - Based on *physical grounding hypothesis*
- Hybrid systems
  - *Synthesis* of deliberative and reactive systems

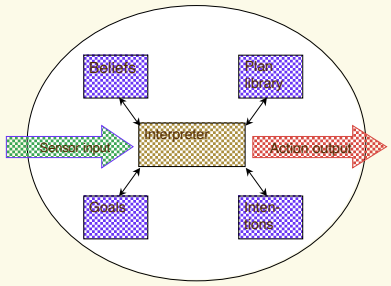
193

Deliberative systems

- Uses *symbolic* representations, e.g.:
  - Maintains models, e.g. world model
  - Planning (e.g. STRIPS)
  - (Logical) Reasoning
  - Sense-*plan/reason*-act cycle
  - Belief, desires, intentions (BDI)
  - Internal ('mental') state (belief base, …)

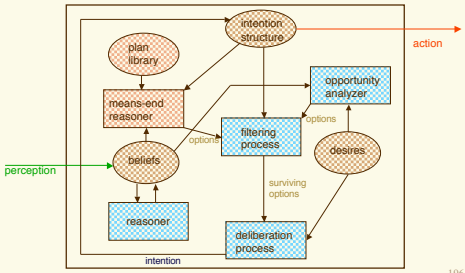
194

BDI Architecture / PRS



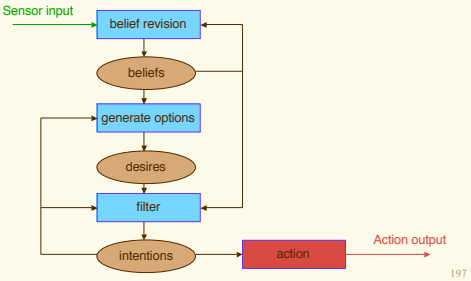
195

IRMA (Bratman et al. 1987)



196

BDI architecture



197

Abstract BDI Interpreter (Rao & Georgeff 1991)

- initialize-state();
- **do**
  options := option-generator(event-queue,B,G,I);
  selected-options := deliberate(options,B,G,I);
  update-intentions(selected-options,I);
  execute(I);
  get-new-external-events();
  drop-successful-attitudes(B,G,I);
  drop-impossible-attitudes(B,G,I);
- **until** quit.

198

Problems with deliberative systems

- Representation & symbol grounding problem
  - How to *translate* the real world into an accurate & adequate symbolic description
  - How do symbols *relate* to real-world entities and processes?
- Reasoning is *computation-intensive*
  - How to get agents to reason with real world information *in time* for the results to be useful

199

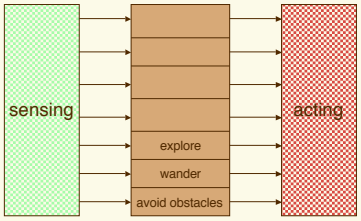
Reactive systems

Problems with deliberative systems

- `Reaction': *reactive systems*
- *Stimulus-response* approach
- No symbolic reasoning, no search
- No internal, symbolic representations
- *Layered / hierarchical* architecture

200

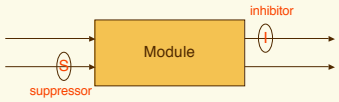
Subsumption architecture (Brooks 1991)



201

Subsumption architecture: (hard-wired) control by suppression and inhibition

- S: signal suppresses (and replaces) input into module for some pre-determined time
- I: signal inhibits output from module for some pre-determined time



202

Hybrid systems

- Trying to *synthesize* both reactive and deliberative systems
- Typically (levels) *layered*: either *vertically* or *horizontally*

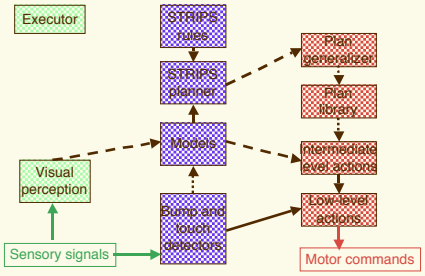
203

Levels in hybrid systems

- Levels, e.g. *three-level* architecture:
  - Low-level: *'reflexes'* (e.g. servo control)
  - Intermediate level: *combines* low-level actions into more complex behaviors
  - High-level: *planning* (e.g. STRIPS planner)
- 'Classical' example: Shakey

204

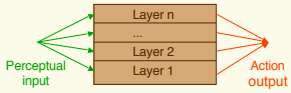
Shakey architecture (Nilsson 1984)



205

Layered hybrid systems

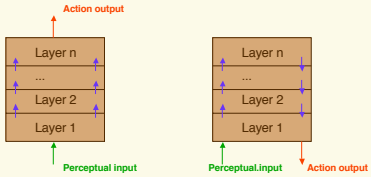
- *horizontally* or *vertically* layered



Horizontal layering

206

Layered hybrid systems (ctd)

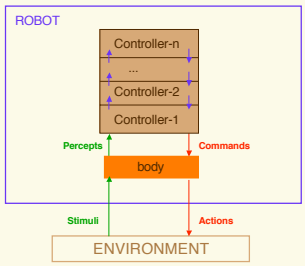


Vertical layering
(1 pass control)

Vertical layering
(2 pass control)

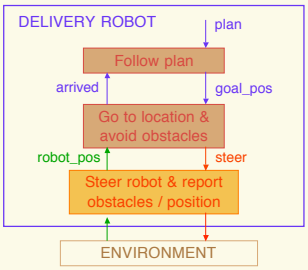
207

Hierarchical Robotic System Architecture (vertically layered)



208

Example: Delivery Robot



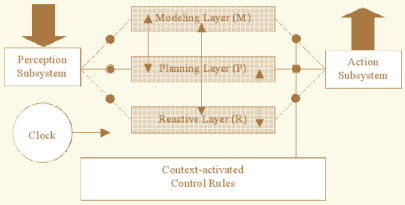
209

(Concrete) hybrid architectures

- TouringMachine (horizontally layered)
- InteRRaP (vertically layered)

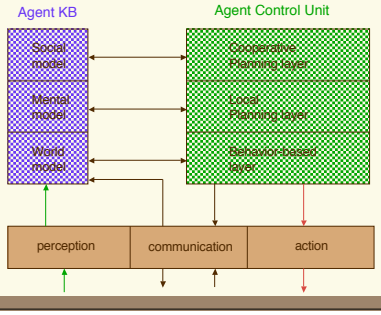
210

TouringMachine (Ferguson 1992)



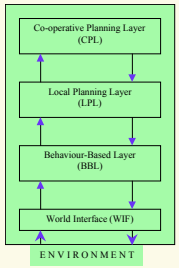
211

InteRRaP (Müller 1996)



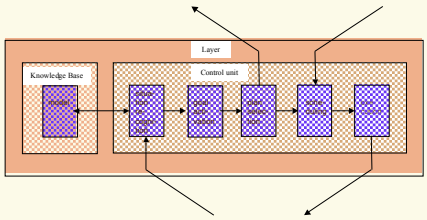
212

InteRRaP: agent control unit



213

InteRRaP generic layer



214

Kinds of hybrid systems (Arkin)

- Role of deliberation/planning in system:
  - Selecting lower-level actions/behaviors
  - Adapting lower-level actions/behaviors (smaller but more frequent modifications)
  - Advising lower-level actions (planner output competes with the lower-level actions)
  - Postponing reactivity by using the planner itself for generating reactions

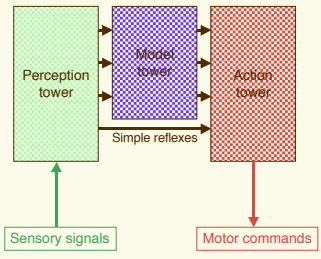
216

'Towers' in hybrid systems

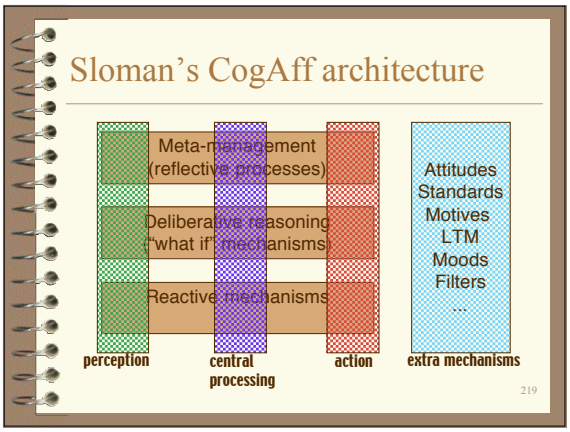
- In hybrid systems there are *hierarchies* both on the *perception* and the *action* side (called *towers*, Albus)
- Nilsson observed that there is also a hierarchy in the *model* part: a third tower, thus obtaining the so-called *Triple Tower Architecture*

217

Triple-Tower Architecture



218


Sloman's CogAff architecture
Meta-management
(reflective processes)
Deliberative reasoning
("what if" mechanisms)
Reactive mechanisms
Attitudes
Standards
Motives
LTM
Moods
Filters
...
perception
central processing
action
extra mechanisms
219