Abstract agent architectures: ‘standard agent’
- $S = \{s_1, s_2, \ldots \}$ environment states
- $A = \{a_1, a_2, \ldots \}$ actions
- Abstractly, an agent can be viewed as a function:
  
  $\text{action: } S^* \rightarrow A$

  mapping sequences of environments to actions

Nondeterministic behavior environment:
- $\text{env: } S \times A \rightarrow \mathbb{P}(S)$

History: interaction agent and environment:
- $h: s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$

Several types of agents
- Behavior of a purely reactive agent:
  
  $\text{action: } S \times A$

- Agent with perception:
  
  $\text{see: } S \times P$

  $\text{action: } P^* \times A$

Several types of agents (2)
- Agents with an internal state:
  
  $\text{see: } S \times P$

  $\text{action: } I \times A$

  $\text{next: } I \times P \times I$

Several types of agents (3)
- Deliberative agent
  
  $\text{see: } S \times P$

  $\text{action: } DB \times A$

  $\text{next: } DB \times P \times DB$

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
Deliberative systems

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

BDI Architecture / PRS

IRMA (Bratman et al. 1987)

Abstract BDI Interpreter (Rao & Georgeff 1991)

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

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.
Reactive systems

Problems with deliberative systems

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

Subsumption architecture

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

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

Hybrid systems

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

Shakey architecture

Shakey architecture (Nilsson 1984)
Layered hybrid systems

- **horizontally** or **vertically** layered

Horizontal layering:

- Perceptual input
- Layer 1
- Layer 2
- Layer n...
- Action output

Vertical layering:

- Perceptual input
- Layer 1
- Layer 2
- Layer n...
- Action output

**Layered hybrid systems (ctd)**

Vertical layering (1 pass control):

- Perceptual input
- Layer 1
- Layer 2
- Layer n...
- Action output

Vertical layering (2 pass control):

- Perceptual input
- Layer 1
- Layer 2
- Layer n...
- Action output

**Hierarchical Robotic System Architecture (vertically layered)**

- Stimuli
- Controller-1
- Controller-2
- Controller-n...
- Actions
- body
- Commands
- Percepts
- Controls
- Actions
- ENVIRONMENT

**Example: Delivery Robot**

- DELIVERY ROBOT
- PLAN
- Follow plan
- plan
- arrived
- goal_pos
- Go to location & avoid obstacles
- robot_pos
- Slow robot & report obstacles / position
- Steer
- ENVIRONMENT

**(Concrete) hybrid architectures**

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

**TouringMachine** (Ferguson 1992)

- Action
- Control
- Perception
- Subsystem
- Context awareness
- Control feedback
InteRRaP (Müller 1996)

InteRRaP: agent control unit

InteRRaP generic layer

Kinds of hybrid systems (Arkin)

‘Towers’ in hybrid systems

Triple-Tower Architecture
Sloman’s CogAff architecture

- Meta-management (reflective processes)
- Deliberative reasoning ("what if" mechanisms)
- Reactive mechanisms

Perception  Central processing  Action

- Attitudes
- Standards
- Motives
- LTM
- Moods
- Filters

Extra mechanisms