Behaviors II – *Thinking*

Instructors: Prof. Manuela Veloso & Dr. Paul E. Rybski
TAs: Sonia Chernova & Nidhi Kalra
15-491, Fall 2004
http://www.andrew.cmu.edu/course/15-491
Computer Science Department
Carnegie Mellon University
“Thinking”... Selecting Actions

- Sensory data as input
- “Behaviors” as processing of input to select actions
- Actuators perform the actions
Perception, Cognition, Action

- Deliberative – three levels
  - Sense:
    - Construct *world state* from sensory data
  - Plan
    - Find a *sequence of actions* that lead from world state to a *goal*
  - Act
    - Transform planned actions for *execution by actuators*
Environment/Domain Issues

- Sensory accuracy
  - How reliable is the sensory input?
- Time
  - How much time is there “to think, plan”?
- Goal
  - How defined are the goals?
- Dynamics
  - How “worth” is to think ahead?
Advantages/Disadvantages of Deliberative Approach

- A plan.
- Let’s brainstorm together.
Perception, Cognition, Action

- Reactive – two levels
  - Sense:
    - Process sensory data into sensory information
  - No plan
  - Act
    - Sensory information is the direct input for execution by actuators
Advantages/Disadvantages of a Reactive Approach

Advantages

- Very responsive to changes in environment
- Smooth control changes in response to smooth changes in sensor values

Disadvantages

- Can’t perform different actions from the same state
- Can get stuck
- Don’t scale well to complex tasks
Memory in Reactive Approach?

- What does it mean to have *memory*?
- Brainstorm:
  - State, Markov, single observation, history of observations
Combining Reactive Behaviors

- **Blending**
  - Efficient if sensor values can be mapped into activation values easy to combine as “forces”
  - Problem: equal but opposing forces can cancel each other out

- **Competition**
  - Similar to blending, but introduces “winner” behavior
  - Problem: possible oscillations

- **Subsumption**
  - Provide a strict hierarchical priority ordering for the behaviors
  - Problem: very dependent on definition of hierarchy

- **Sequencing**
  - Run a single reactive behavior and change with state – FSM
  - Problem: state granularity
Blending: Motor Schemas

Goal vector
Avoidance vector
Resulting vector
Competition

Goal vector
Avoidance vector
Subsumption

Goal vector

Wall follow vector
Example of Behavior/FSM

Decompositional  ...........  Sequential

LIFTED STRAIGHT BEHAVIOR

AIBO lifted

AIBO tilted

AIBO back on ground

AIBO lifted

ON GROUND BEHAVIOR

AIBO back on ground

AIBO tilted

Tilted left

Tilted right

SET LED-MIDDLE-LEFT

SET LED-MIDDLE-RIGHT

WALK

TROT

RUN

15-491 CMRoboBits
Three-Tiered Architectures

Planning

Deals with goals and resource interactions

Task decomposition; Task synchronization; Monitoring; Exception handling; Resource management

Executive

Behavioral Control

Deals with sensors and actuators
Path Planning

- Existence of a goal
  - *Goto* some goal point
- ERRT
  - Path planning
  - Smoothing
  - Memory
Replanning with Advice

Probability $p$: Extend closest node in tree towards goal
Probability $r$: Extend closest node in tree towards random cache point
Probability $1-p-r$: Extend closest node towards a random point
Path Planning and Replanning
Behaviors: FSMs

- **Score**
  - not see ball
  - next to ball
  - not next to ball

- **Recover**
  - see ball
  - not see ball
  - timeout

- **Search**
  - see ball

- **Approach**
  - see ball
Hysteresis
Handling Uncertainty

**Score**
- High Fidelity
  - $x, y > 30 \text{ cm}$
  - $\theta > 30$
- Medium Fidelity
  - $\theta > 30$
  - $x, y > 60 \text{ cm}$
- Low Fidelity

**Recover**
- Low Fidelity

**Search**
- Low Fidelity

**Approach**
- High Fidelity
  - $\theta > 30$ or
  - $x, y > 60 \text{ cm}$
- Low Fidelity

- not see ball
- see ball
- next to ball
- not next to ball
- timeout
- timeout and
  - $\theta > 30$
  - $x, y > 60 \text{ cm}$
- Localization
Hierarchy – Adding Scale

- In order to scale to large behaviors, we can reuse collections of lower-level behaviors
  - Libraries of lower-level behaviors form the building blocks for all AIBO behaviors
  - Each state of FSM can be either a single reactive behavior, or another FSM with its own behaviors (or FSMs)
STP Behavior Architecture

- **Skills** – single robot low-level atomic behaviors
- **Tactics** – multi-skill combination for single robot goal achievement
- **Plays** – multi-robot coordination by planned sequence of tactics
Adaptive Playbook Strategy Engine

- **Plays**
  - A multi-robot plan represented as a temporal sequence of parameterized tactics

- **Playbook language**
  - Human understandable
  - Easy to add new plays like a real coach

---

**Role 0**
- Dribble to $P_1$
- Pass to $R_2$
- Wait for loose ball

**Role 1**
- Wait for Pass at $P_2$
- Receive Pass
- Shoot

15-491 CMRoboBits
Play Selection–Opponent Learning

- Each play $p_j$ has an associated weight $w_j$
- Stochastic play selection: from the set $P_A$ of applicable plays with probability determined by $w$'s

$$P(p^* = p_j) = \frac{w_j}{\sum_{k \in P_A} w_k}, \forall j \in P_A$$

- ABORT, SUCCESS, FAILURE conditions
- Weights are adapted online – details later

Playbook adapts to the opponent without explicit modeling.
Replay Real Log – Tactic (< 1s)
Replay – Deflection Play (<1s)
Behaviors-II Conclusion

- Choice of behavior representation depends on conditions of the domain
- Ideally, planning ahead would be great
- Ideally, fast response to sensor changes

- Think well about which approach to take
- Homework: MasterMind.