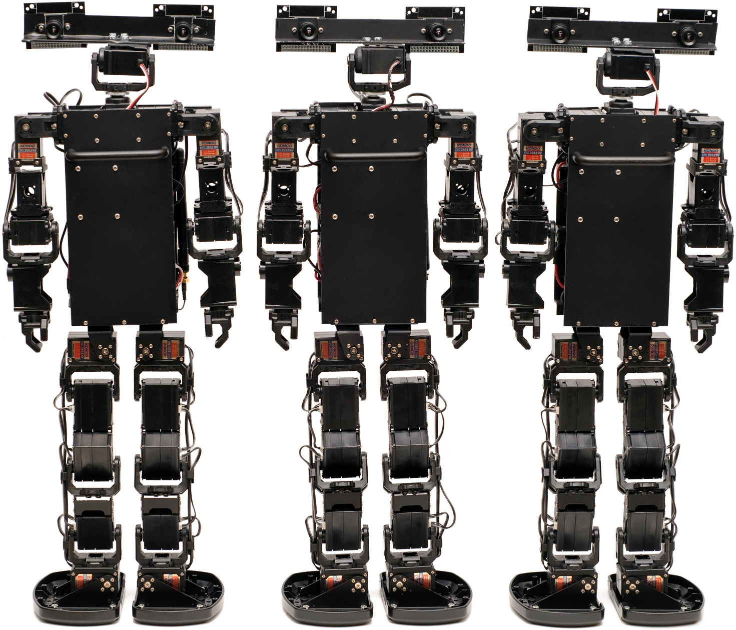
Multiagent Supervised Training with Agent Hierarchies and Manual Behavior Decomposition

Keith Sullivan Sean Luke

Department of Computer Science, George Mason University
Fairfax, VA 22030 USA

RoboCup Motivation



Motivation for Training

- Programming agent behaviors is tedious
  - Code, test, debug cycles
- Changing of agent behavior is desirable
  - Non-programmers (consumers, animators, etc.)
  - Future tasks, possibly greatly different from original task
- Learning from Demonstration (LfD)
  - Iteratively builds policy from examples (state/action pairs)
  - Supervised learning

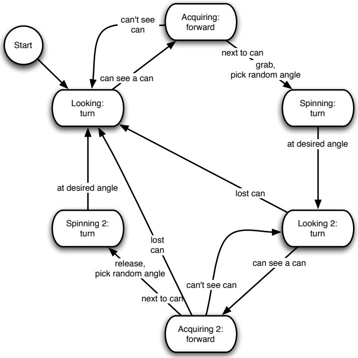
Hierarchical Training of Agent Behaviors (HiTAB)

- Motivation: Rapidly train complex behaviors with very few examples
- Behaviors are automata
- Expandable behavior library
  - Start with atomic behaviors
  - Iteratively build more complex behaviors via scaffolding
- Features describe internal and world conditions
  - Continuous, torodial, categorical (boolean)
- Behaviors and features are parameterizable

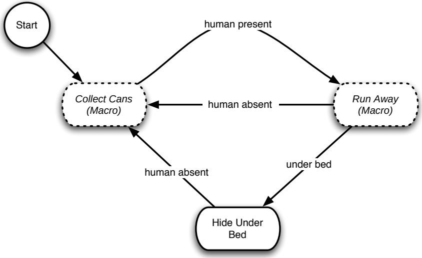
HiTAB (cont.)

- Gathering examples is expensive
  - Each example is an experiment conducted in real-time
- Admission: close to programming by example and far away from machine learning
- Limited number of samples, but high dimensional problem!
  - Behavior decomposition via hierarchical finite automata (HFA)
  - Per-behavior feature reduction
- Learn transition functions $\rightarrow$ Supervised classification task
  - C4.5 with probabilistic leaf nodes
  - Different types of features

Example Behavior



(a) Moore Machine



(b) HFA

Formal Model

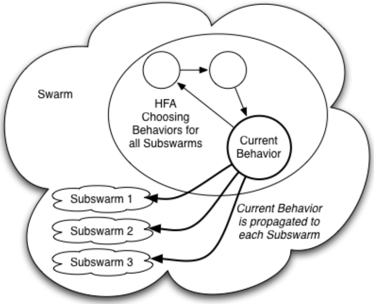
- $S = \{S_1, ..., S_n\}$ is the set of *states* in the automaton. Among other states, there is one *start state* S_1 and zero or more *flag states*.
- $B = \{B_1, ..., B_k\}$ is the set of *basic* (hard-coded) behaviors.
- $F = \{F_1, ..., F_m\}$ is the set of observable *features* in the environment.
- $T = F_1 \times ... \times F_m \times S \to S$ is the *transition function* which maps the current state S_t and the current feature vector $\vec{f_t}$ to a new state S_{t+1}.
- We generalize the model with free variables (parameters) $G_1, \ldots, G_n$ for basic behaviors and features.

Using HiTAB

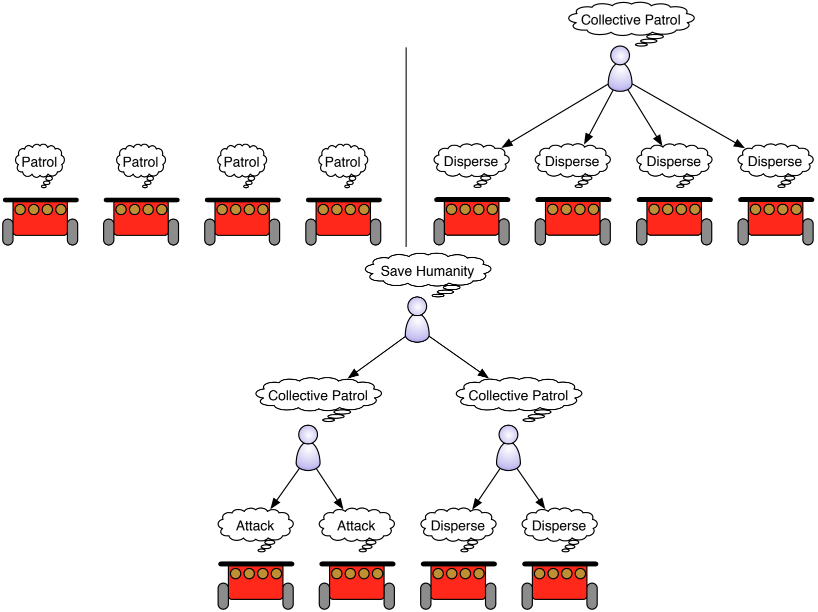
- Running HiTAB
  - Begin in *start* state
  - Query transition function, transition, perform associated behavior
- Training with HiTAB
  - Alternate *training mode* and *testing mode*
  - Build example database, adding corrections as needed
  - Trim unused behaviors and features for saving

Homogeneous Agent Hierarchy

- Problem
  - Size of learning space grows → number of samples grows
  - Inverse problem between micro- and macro-level behaviors
- Agent hierarchy: tree with *coordinator agents* as non-leaves and regular agents as leaves
- Coordinator agent features: statistical information about subsidiary agents a
- Agents at same level run same HFA, but might be in different states
- Train agents bottom-up



Notions of Homogeneity

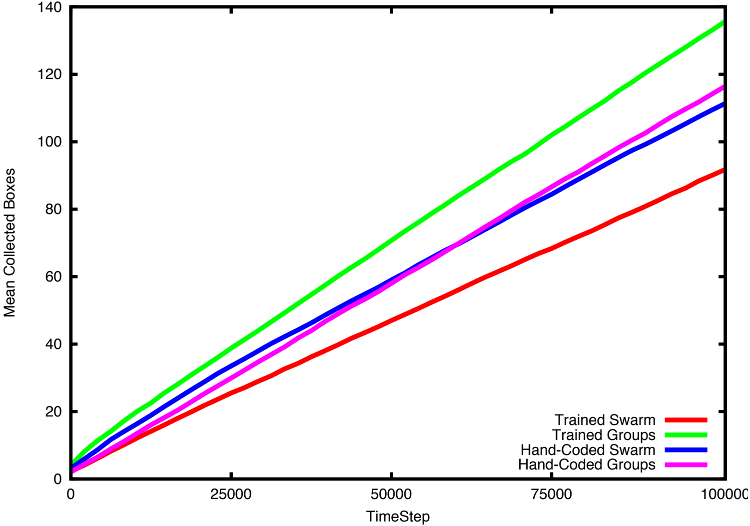


Experiments

- Simulated box Foraging
  - Known deposit location
  - Randomly placed boxes
  - 10 boxes in all experiments
- 50 agents: two levels of hierarchy
  - Teams of 5 agents
  - Grouped these teams into groups of 5
- Boxes require either 5 or 25 agents to pull back
- 100 iterations of 100,000 timesteps each

Simulation

Results



Mean Collected Boxes
TimeStep
0
20
40
60
80
100
120
140
0
25000
50000
75000
100000
Trained Swarm
Trained Groups
Hand-Coded Swarm
Hand-Coded Groups

Preliminary Multirobot Work

Future Work

- Training Multiple Agents
  - Behavior Bootstrapping
- Heterogeneous Groups
  - Behavior **and** Capability
- Dynamic Hierarchies
- Correction of Demonstrator Error