Validating Complex Agent Behavior

Scott Wallace
University of Michigan

The Problem of Correctness

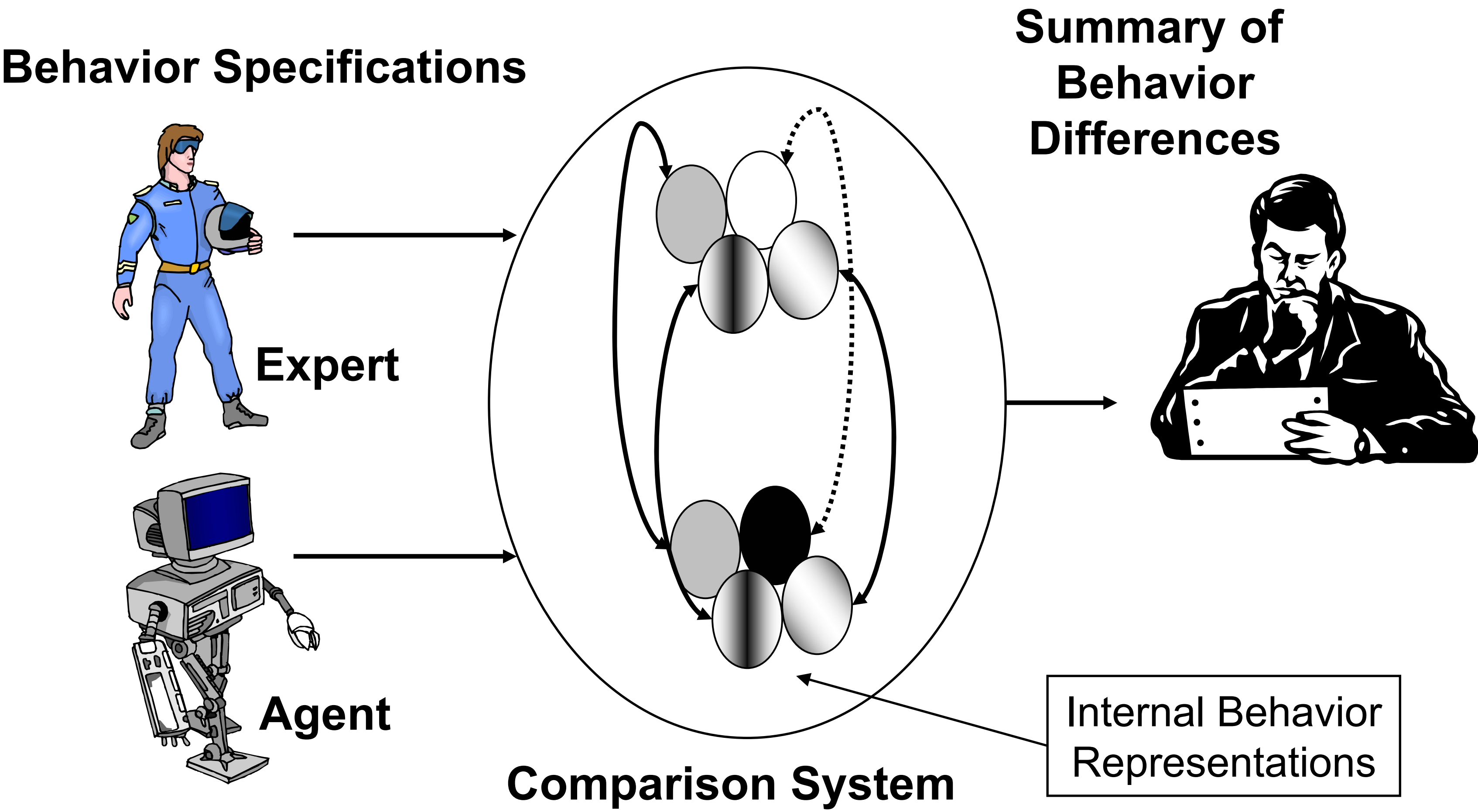
- Agents must have correct, expert-level behavior
- Errors undermine project’s goals
- How can we ensure correctness?

The Validation Bottleneck

- Manual Validation: Expert critiques agent behavior
  - Requires significant human effort
  - Difficult to detect every error
  - Standard approach to obtaining correct behavior

- Challenges for Automated Validation
  - *Difficult to formalize and articulate parameters of correct/incorrect behavior*
  - *"I can't tell you what's incorrect, but I know it when I see it."*
  - *Removing humans from the process creates new opportunities for failure*

Validation Framework Overview



Behavior Specifications

Expert

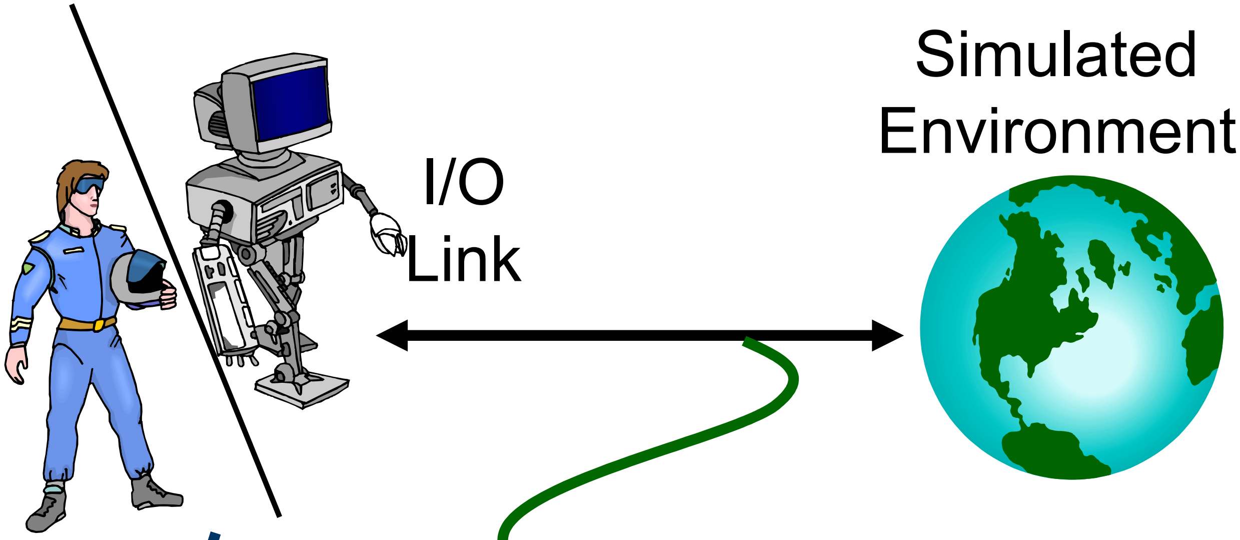
Agent

Comparison System

Summary of Behavior Differences

Internal Behavior Representations

Behavior Specifications



- Actor interacts with simulator
- Simulator provides a clean interface for:
  - Identifying salient state information
  - Identifying relevant actions

Captured Actor I/O: (s,a)

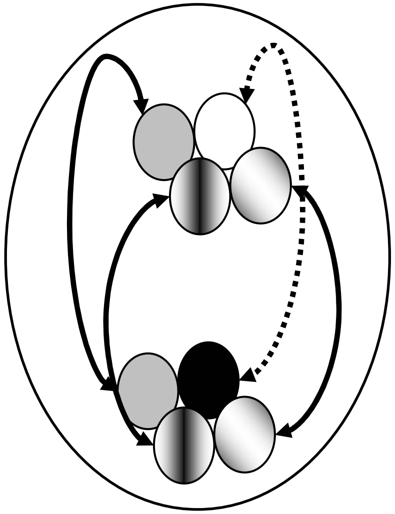
Actor's Goal

Annotations: (s,G)

$(s,G,a)_1$, … , $(s,G,a)_n$

Behavior Trace

Comparison System

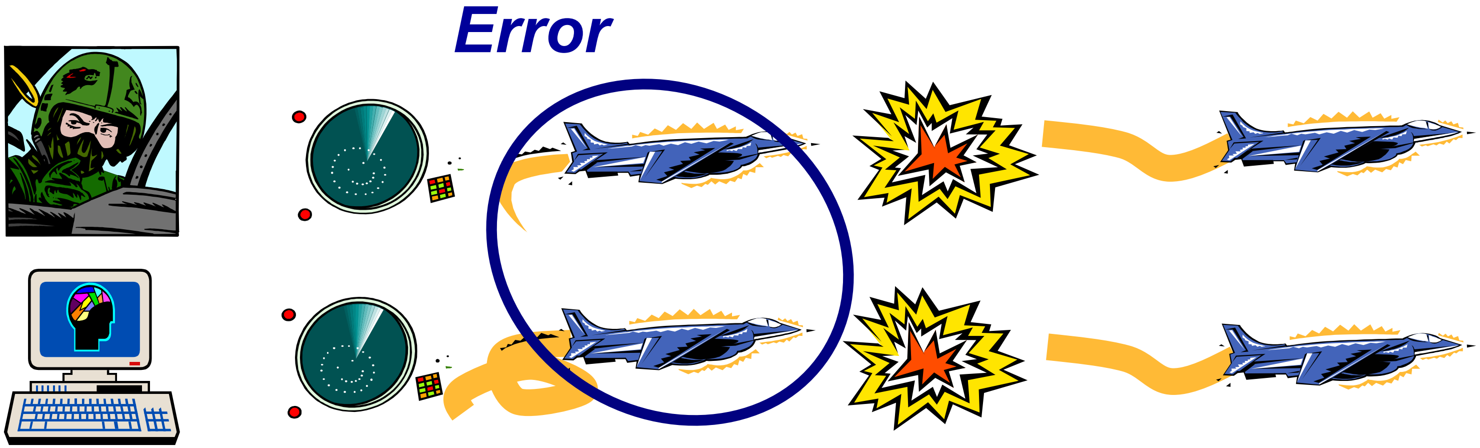


- Desirable attributes:
  - Low Human/Computational Effort
  - Domain Independence
  - Efficacy

- We examine two types of approaches
  - Sequential (actions, goals)
  - Behavior bounding

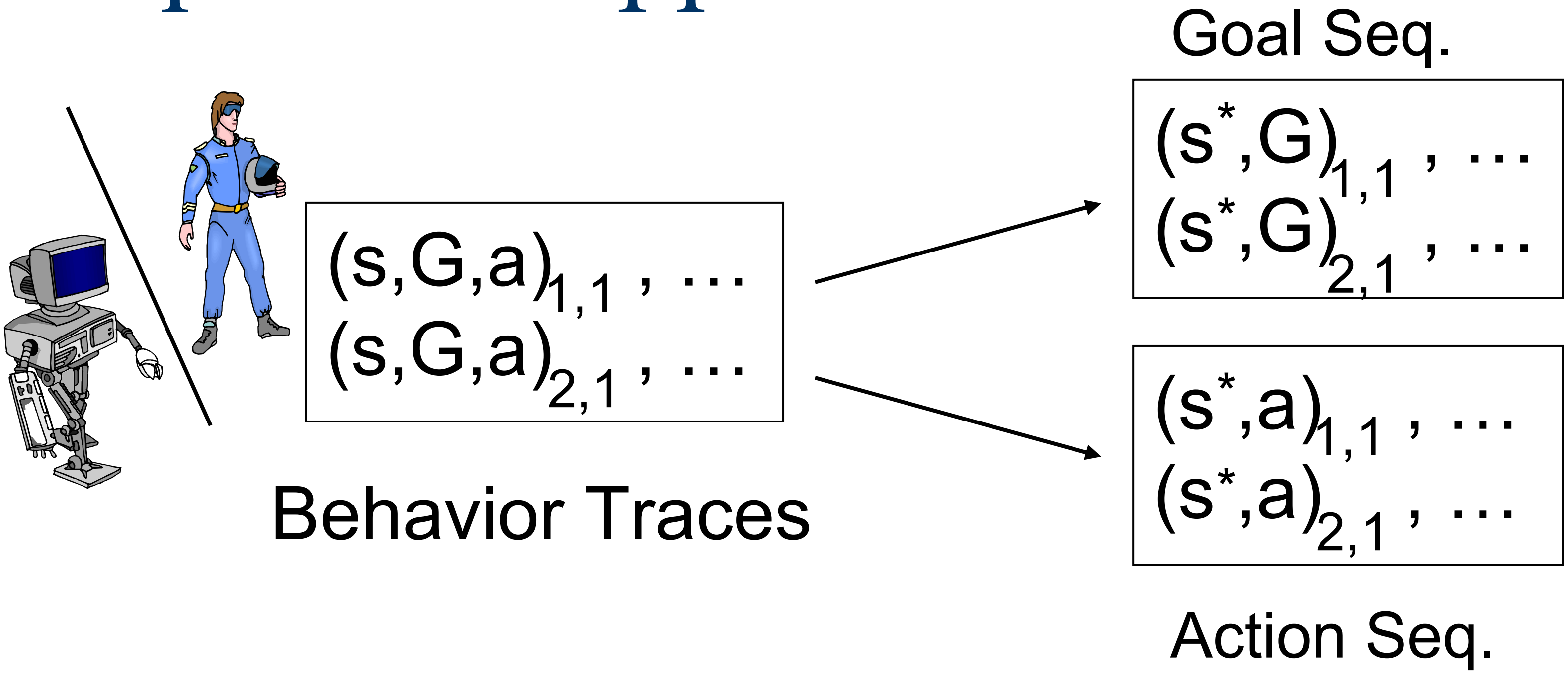
- Quality of the comparison system will be influenced by choice of representation

Overview: Sequential Approach



- Discrepancies between sequences indicate errors

Sequential Approaches



Goal Seq.

$(s^*,G)_{1,1}, \ldots$
$(s^*,G)_{2,1}, \ldots$

$(s,G,a)_{1,1}, \ldots$
$(s,G,a)_{2,1}, \ldots$

Behavior Traces

$(s^*,a)_{1,1}, \ldots$
$(s^*,a)_{2,1}, \ldots$

Action Seq.

- Extract symbols from behavior traces to form sequences (internal behavior representations)
- Compare sequences once aligned to minimize differences

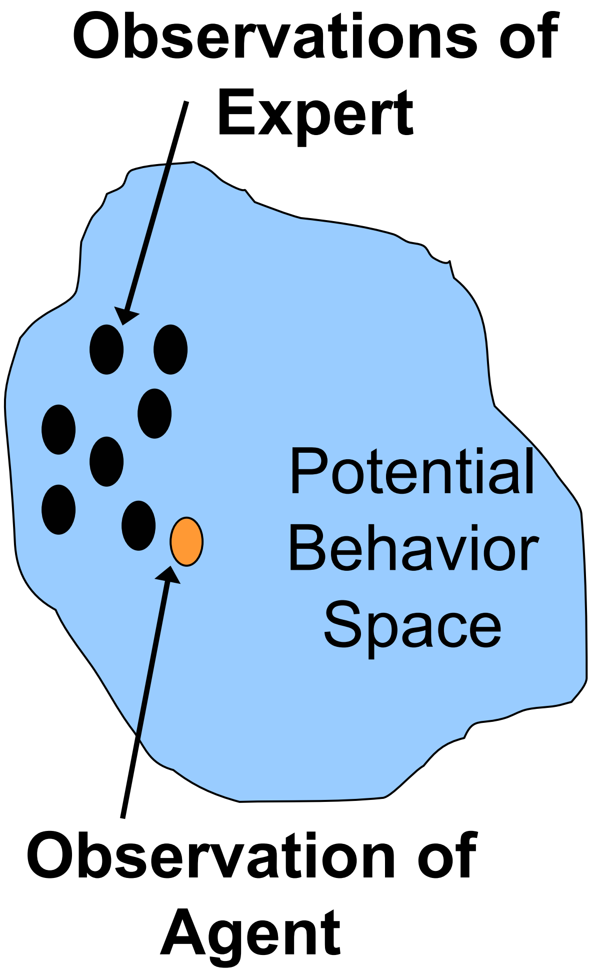
Effort and Domain Independence

- Sequences are very weak generalizations of behavior traces
- Required expert examples grows rapidly with:
  - Complexity of domain/behavior
  - Variability of behavior
- Internal representation grows with number of expert examples
- Computational complexity (time/space) of comparison is a function of representation size

- *But* representation makes few assumptions

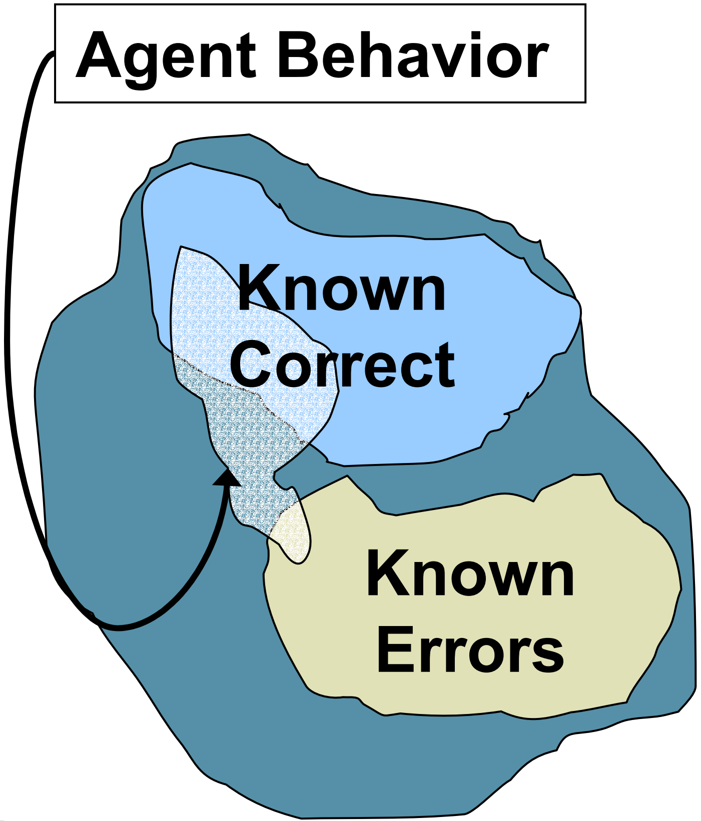
Weakness of Sequential Approach

- Sequences represent instances of behavior
- Instances are points in the behavior space
- Want to represent aggregate behavior

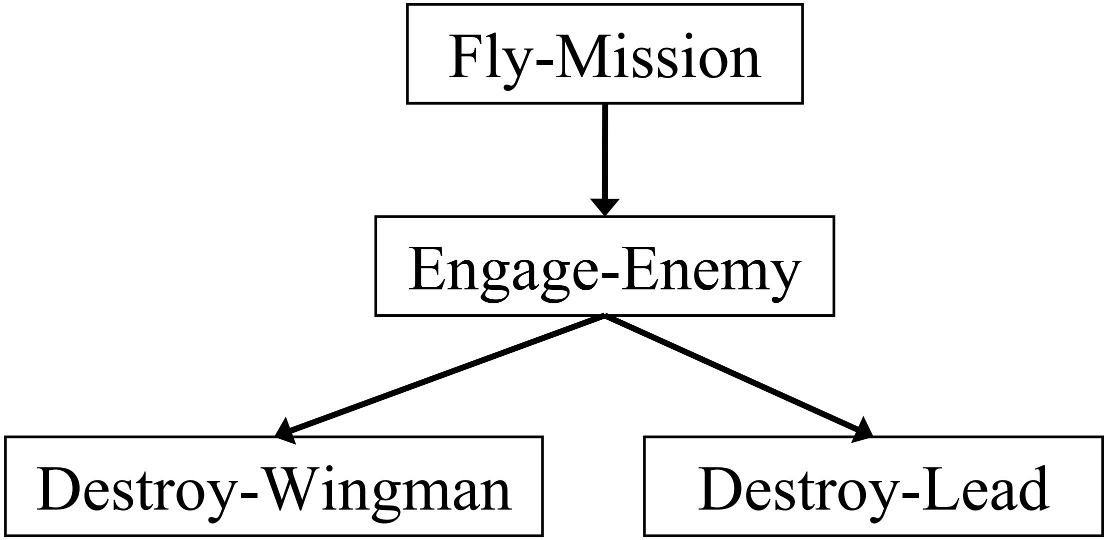


Behavior Bounding

- Define boundaries in the space of potential behavior using:
  - – observations
  - – knowledge of task requirements
- Determine portion of agent behavior in each region

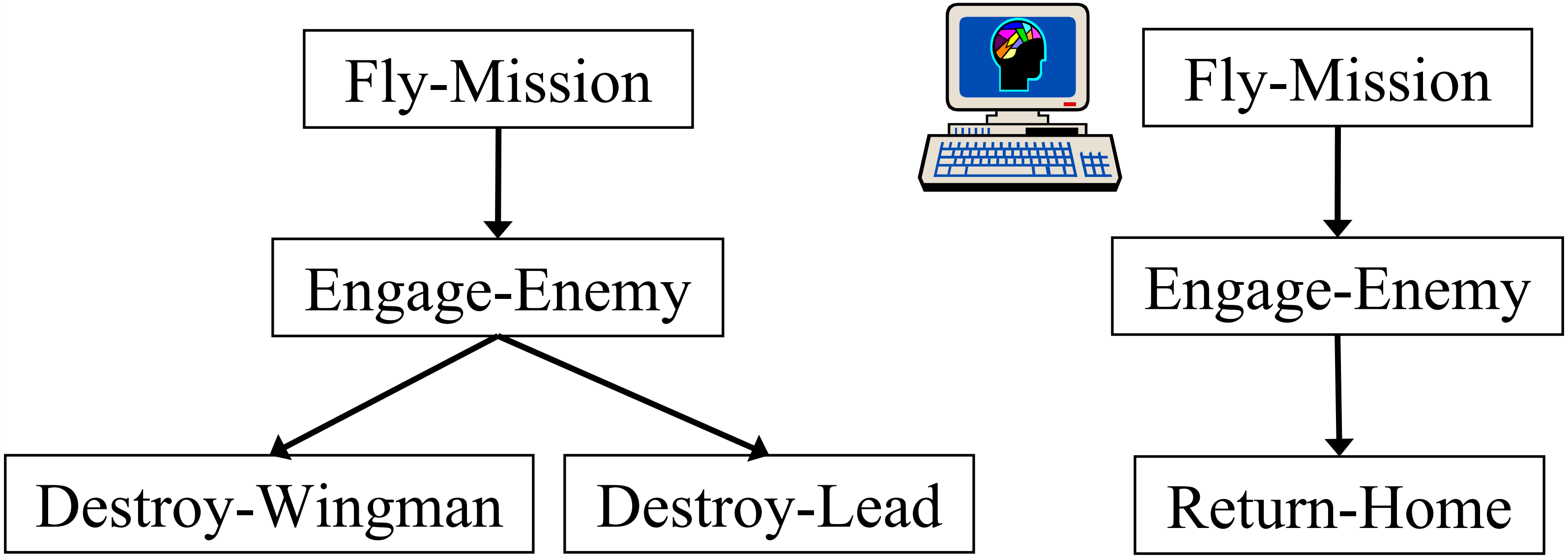


Leveraging the Goal Hierarchy



- A hierarchy compactly represents a subset of the behavior space.
- Agents are often constructed via task-decomposition.
- A hierarchy can be built from behavior traces.

Goal Hierarchies as Classifiers



Fly-Mission

Engage-Enemy

Destroy-Wingman

Destroy-Lead

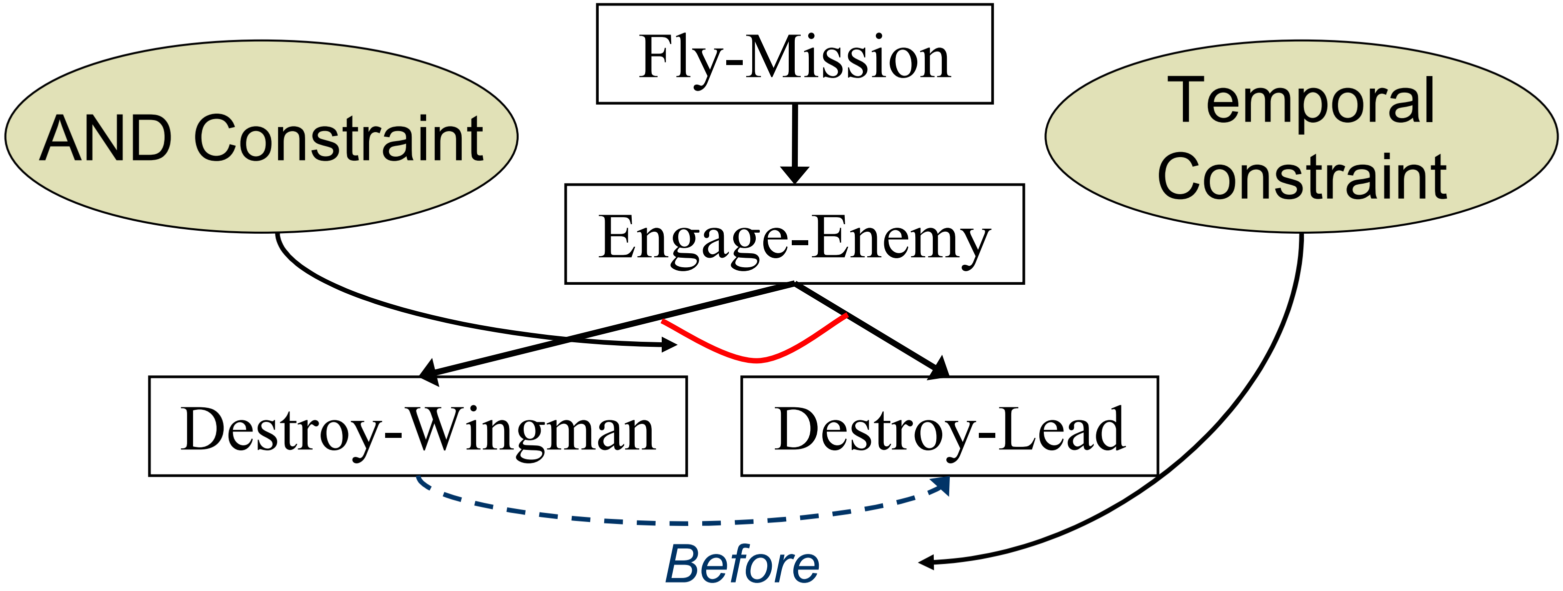
Fly-Mission

Engage-Enemy

Return-Home

- Basic hierarchy identifies differences in topology.

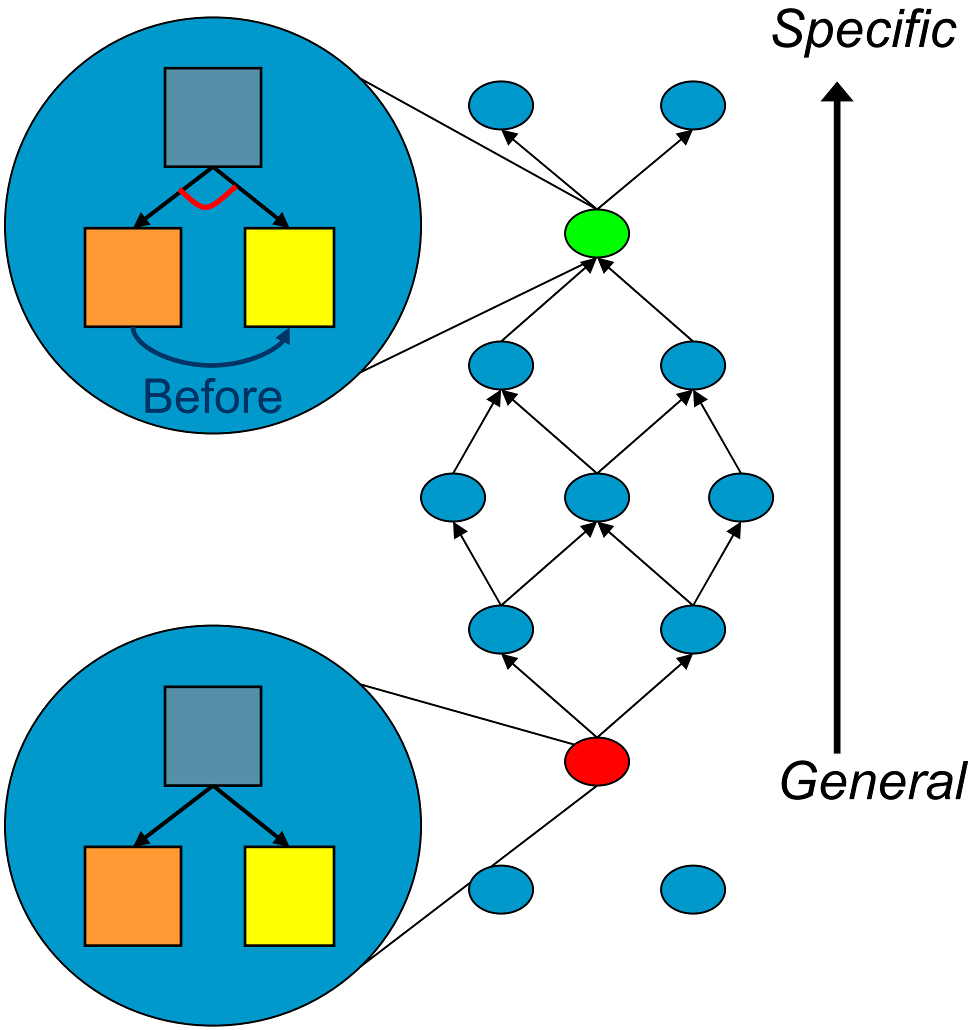
Constrained Goal Hierarchy



- Constraints reduce degrees of freedom
- Create specializations of original hierarchy
- Can also be used to classify behavior

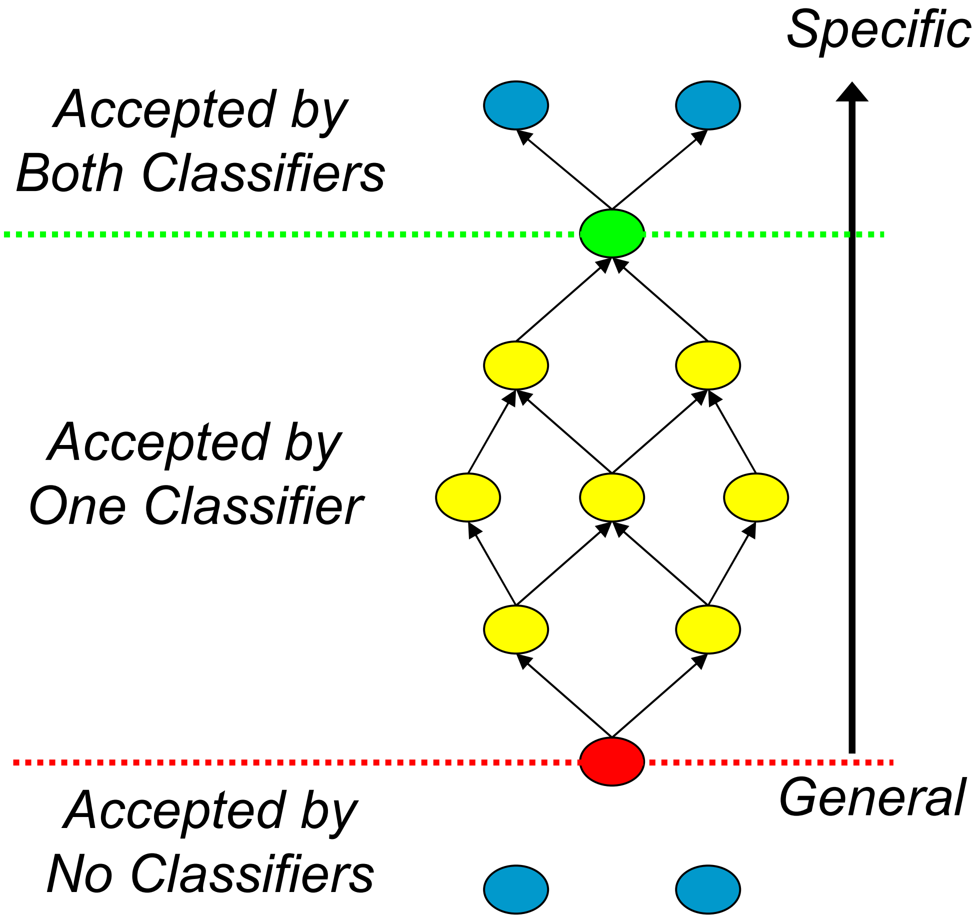
Hierarchies As Partitions

- Constraints impose an ordering on the behavior space



Hierarchies As Partitions

- Boundaries partition space into three regions
- Paves way for efficient error detection



Specific

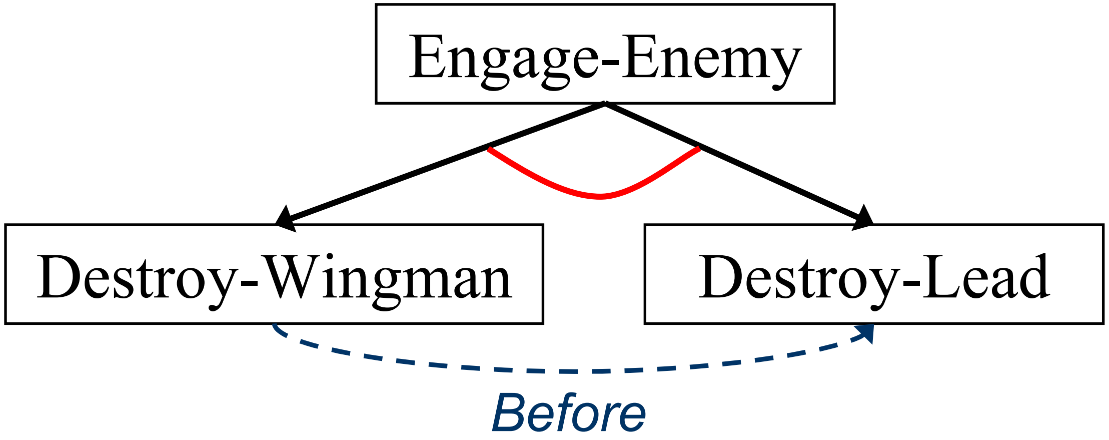
Accepted by Both Classifiers

Accepted by One Classifier

Accepted by No Classifiers

General

Building an Upper Boundary

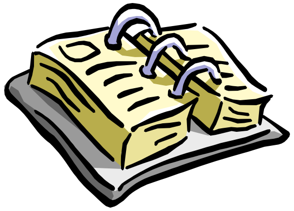


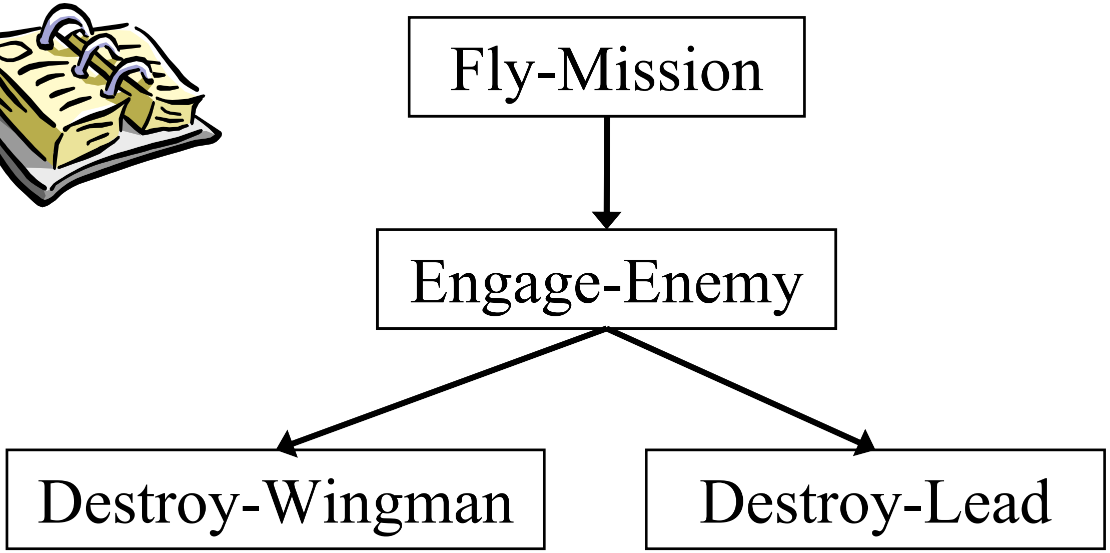
Fly-Mission
Engage-Enemy
Destroy-Wingman

Fly-Mission
Engage-Enemy
Destroy-Lead

- Construct a maximally specific hierarchy covering the observations

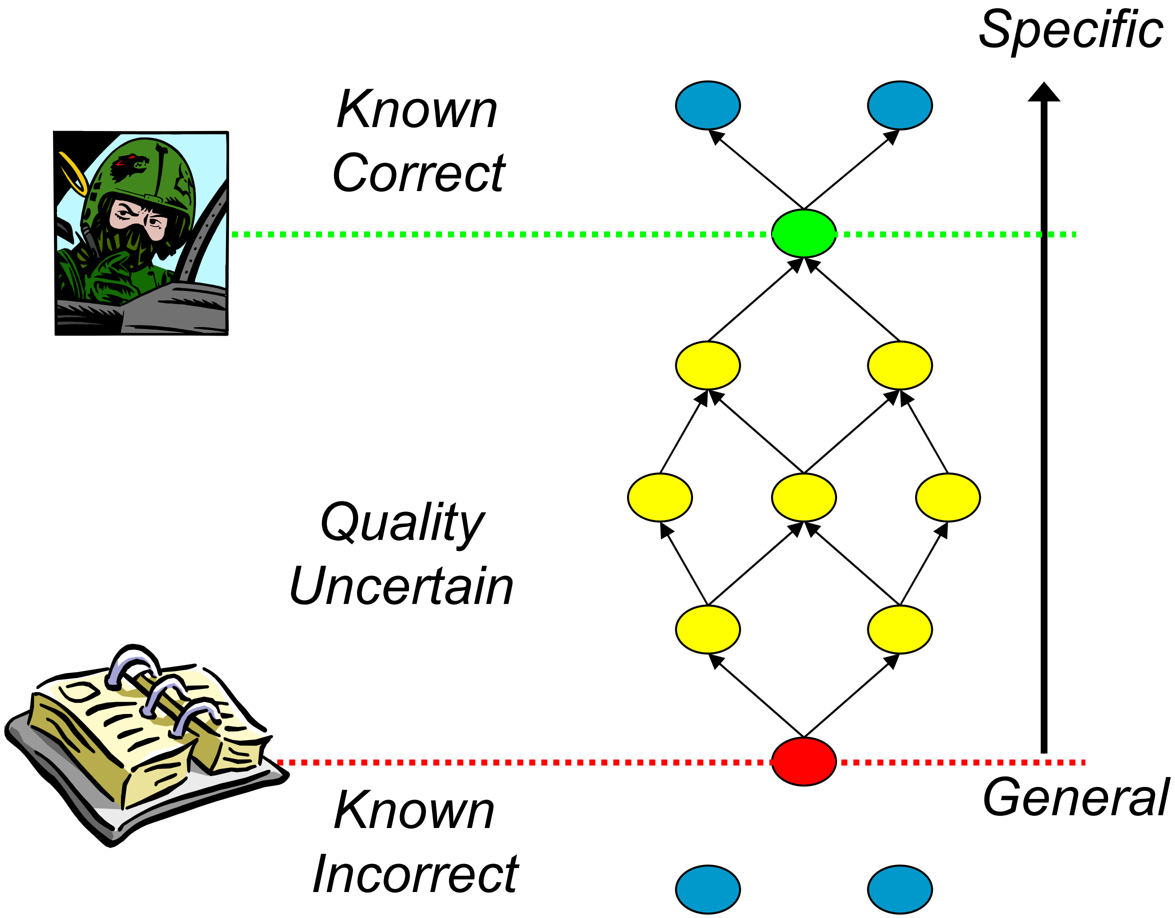
Building a Lower Boundary





- Removing constraints yields lower bound
- Alternatively, lower bound may be generated manually.

Partition Behavior Space



Specific

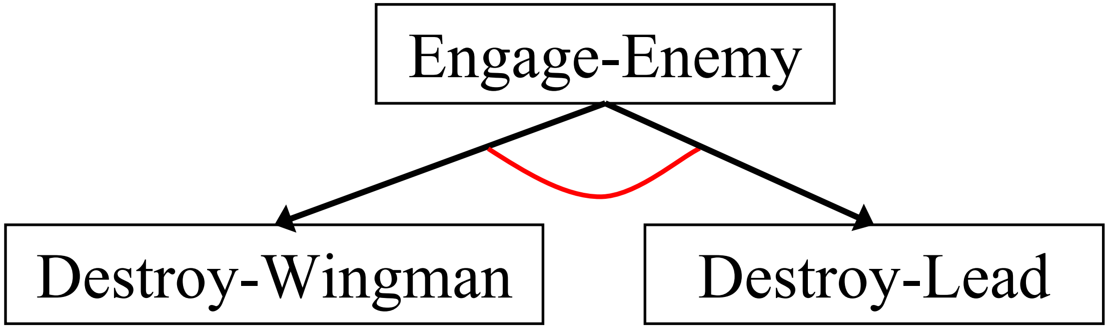
Known Correct

Quality Uncertain

General

Known Incorrect

Observe Agent Behavior



Engage-Enemy
Destroy-Wingman

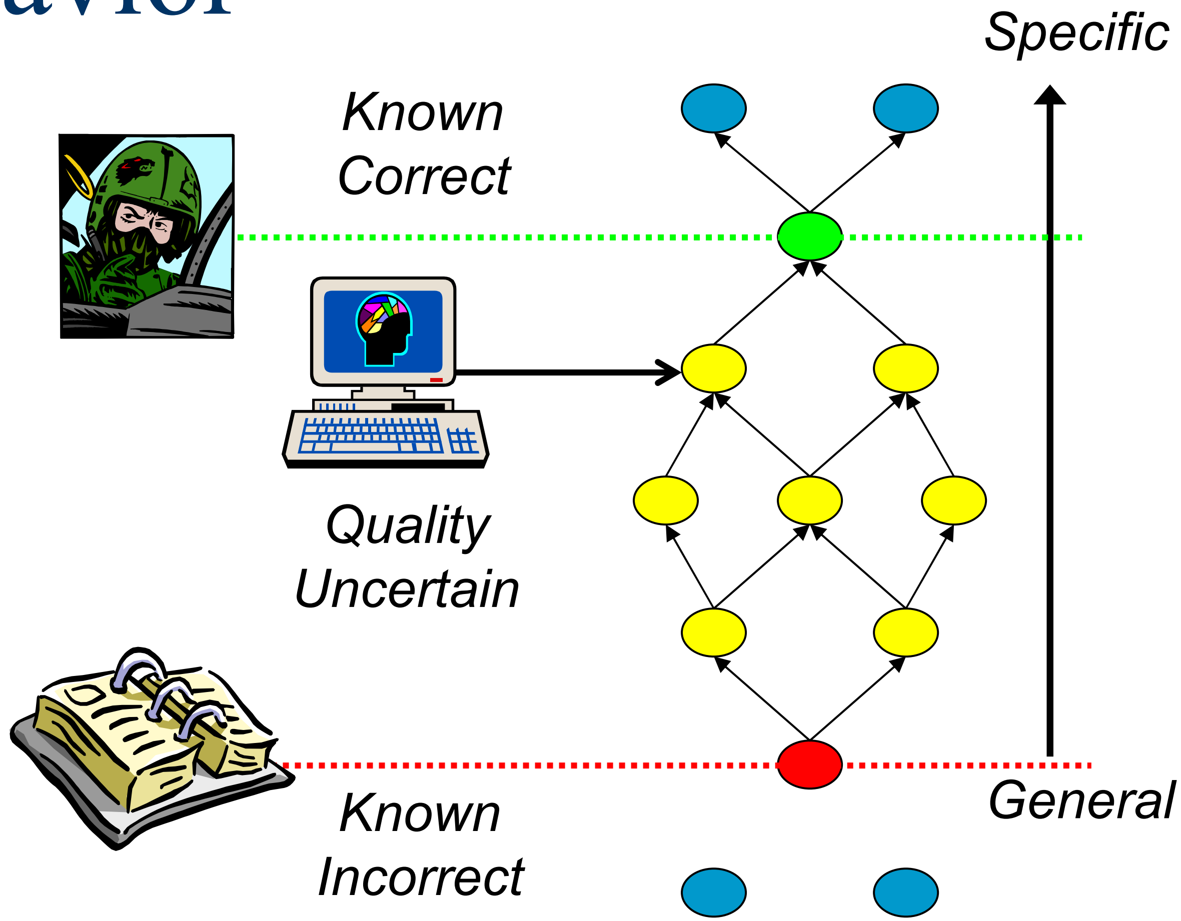
Engage-Enemy
Destroy-Lead

Engage-Enemy
Destroy-Lead

Engage-Enemy
Destroy-Wingman

- Construct a maximally specific hierarchy covering the observations

Identify Quality of Agent Behavior



Specific

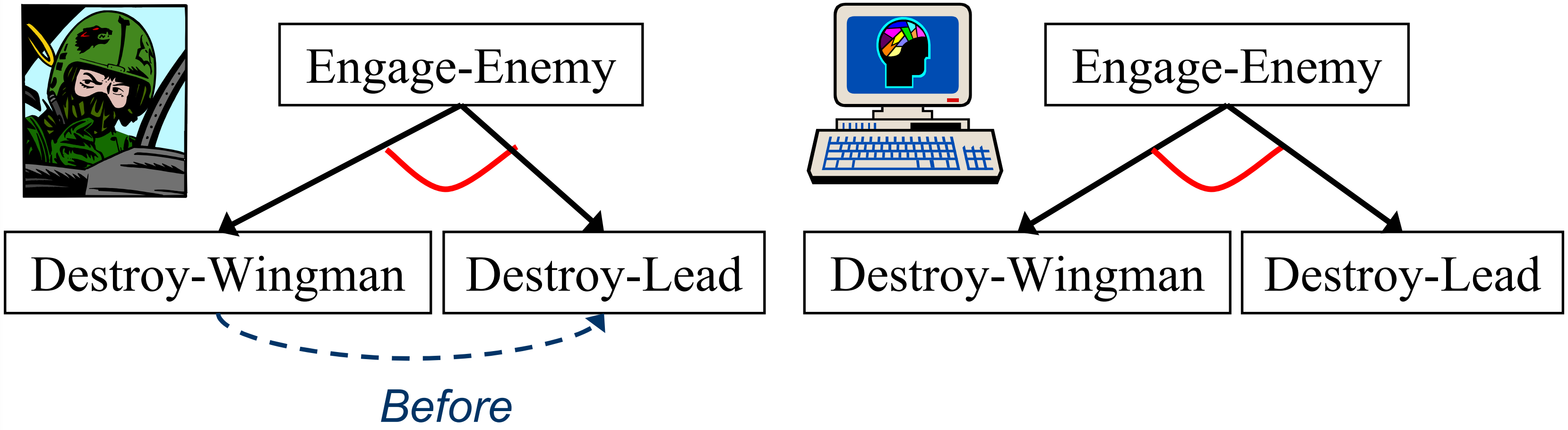
Known Correct

Quality Uncertain

Known Incorrect

General

Identify Quality of Agent Behavior



Engage-Enemy

Destroy-Wingman

Destroy-Lead

Before

Engage-Enemy

Destroy-Wingman

Destroy-Lead

- Agent behavior is not a specialization of Expert behavior
- Looking at behaviors encapsulated by hierarchy gives details of similarities and differences
  - Agent may perform sub-goals in an incorrect order

Effort and Domain Independence

- Hierarchies can be built using relatively few behavior traces
- Computation effort of comparison
  - Independent of number of expert examples
  - Polynomial in size of hierarchy
- *Representation should be compatible with many goal based agents*

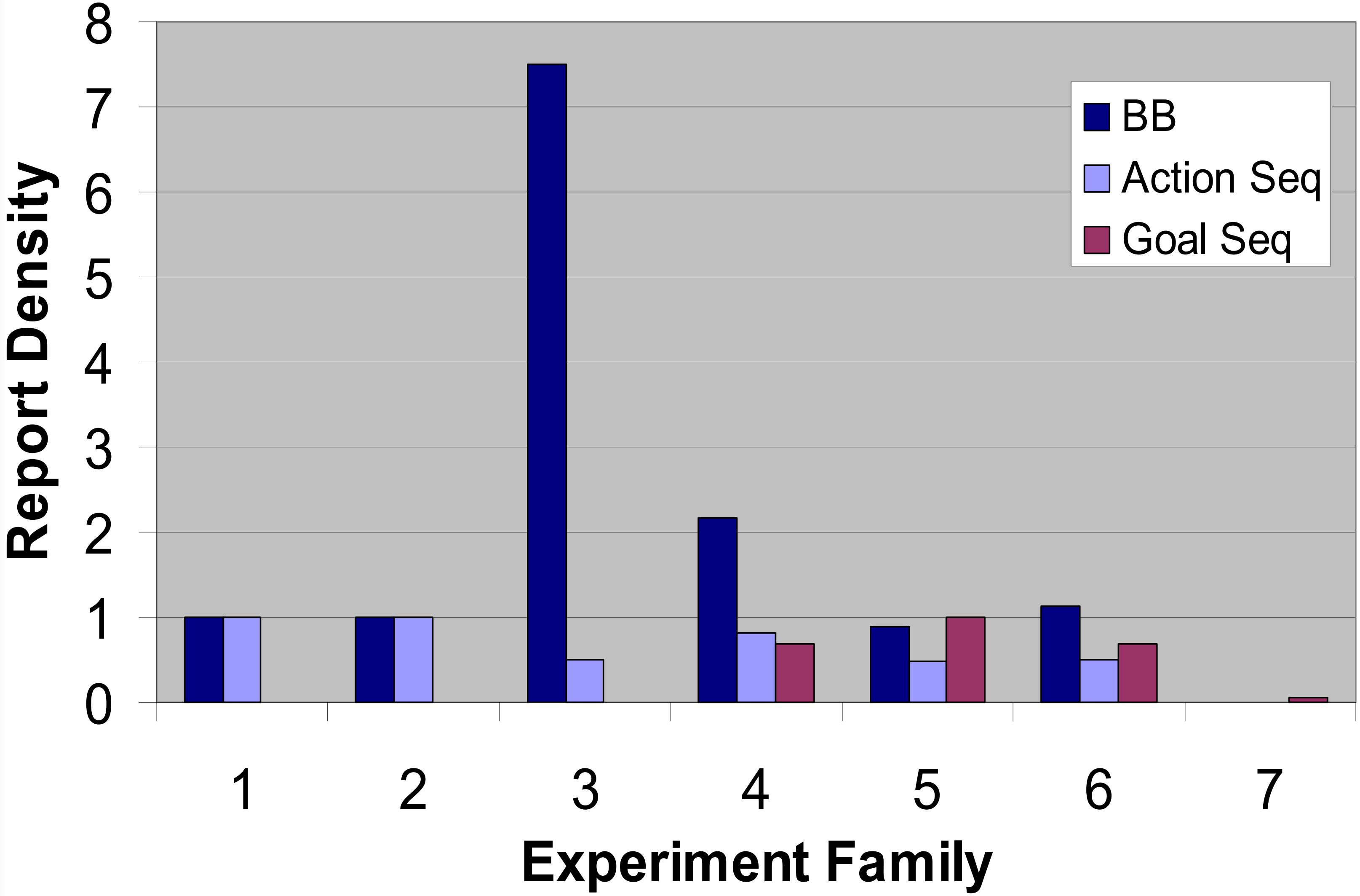
Measuring Efficacy

- Rate method based on the quality of data in the summary

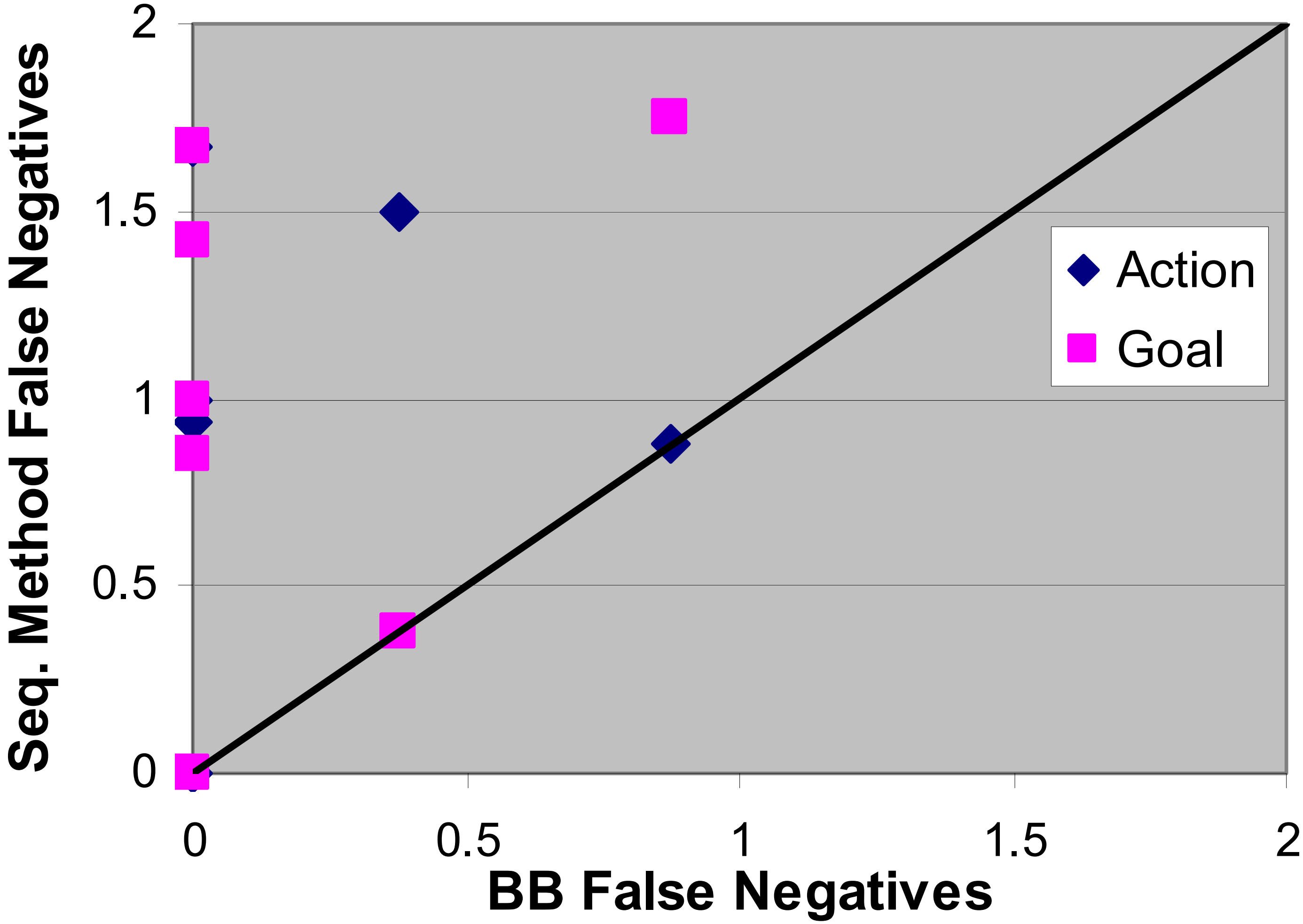
$$Report\ \ Density = \frac{True\ \ Positives}{Reported}$$

- Also want few undetected errors: *False Negatives*

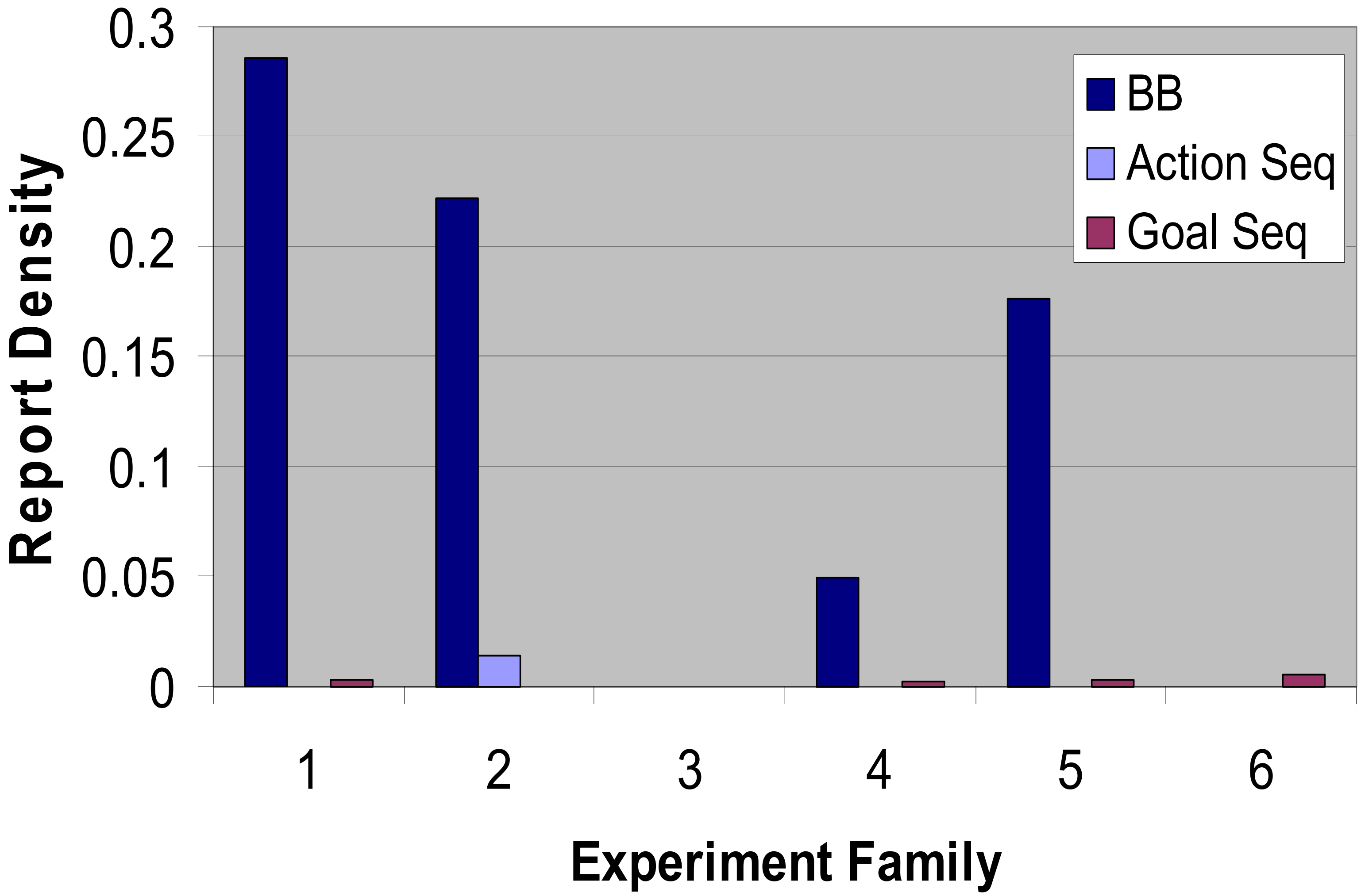
Results: Object Retrieval Domain



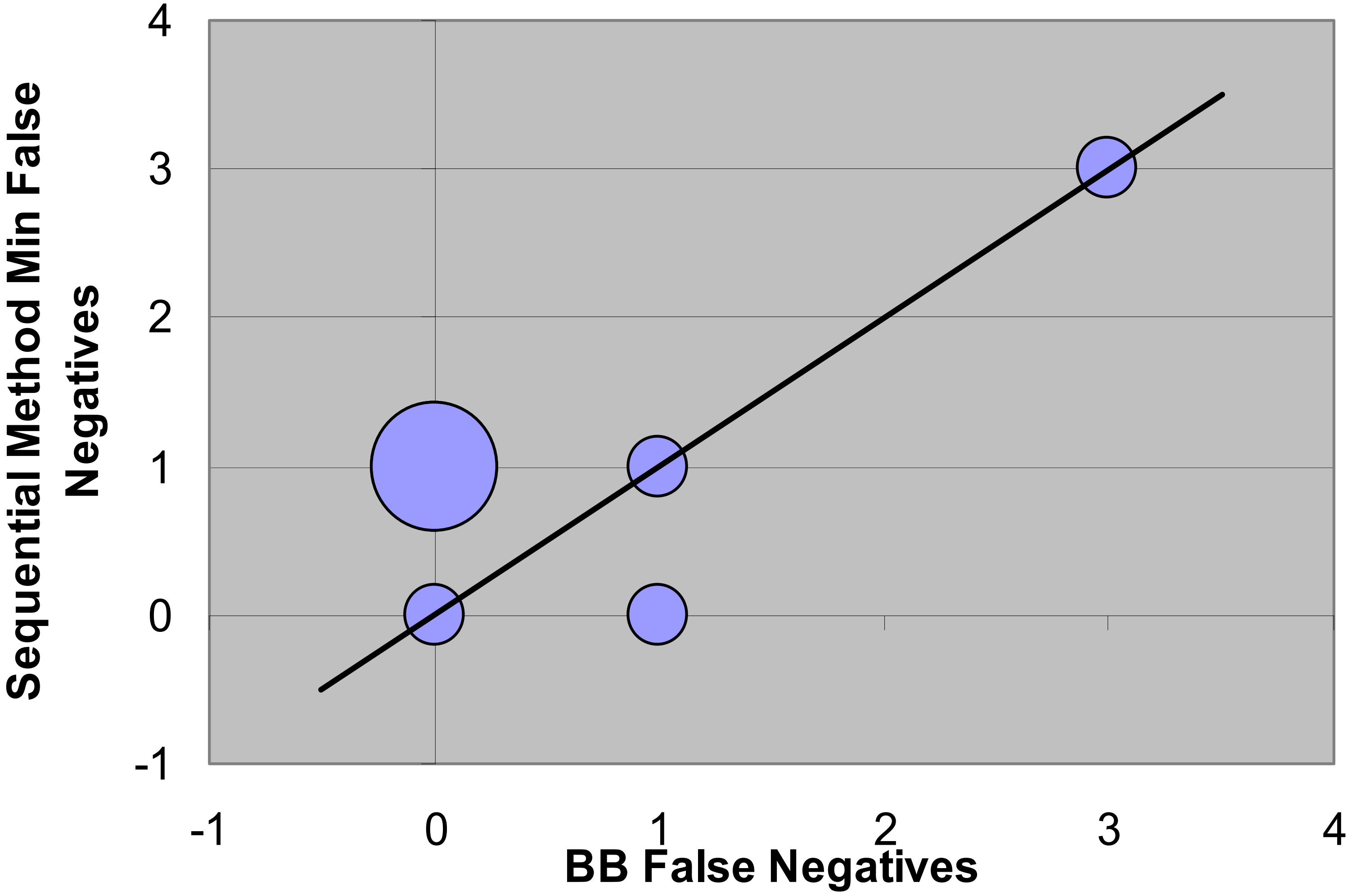
False Negatives in Object Retrieval Domain



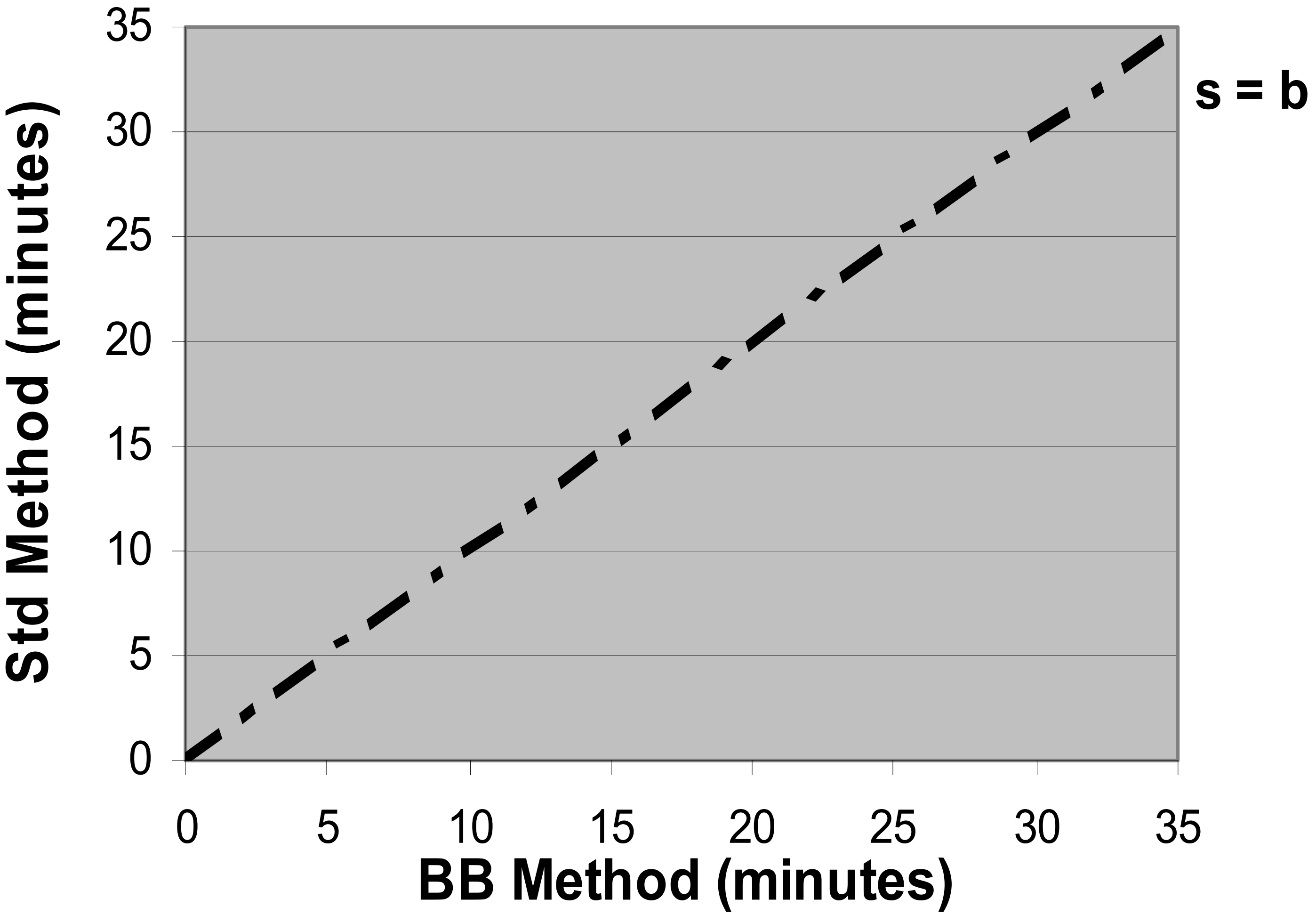
Results: MOUT Domain



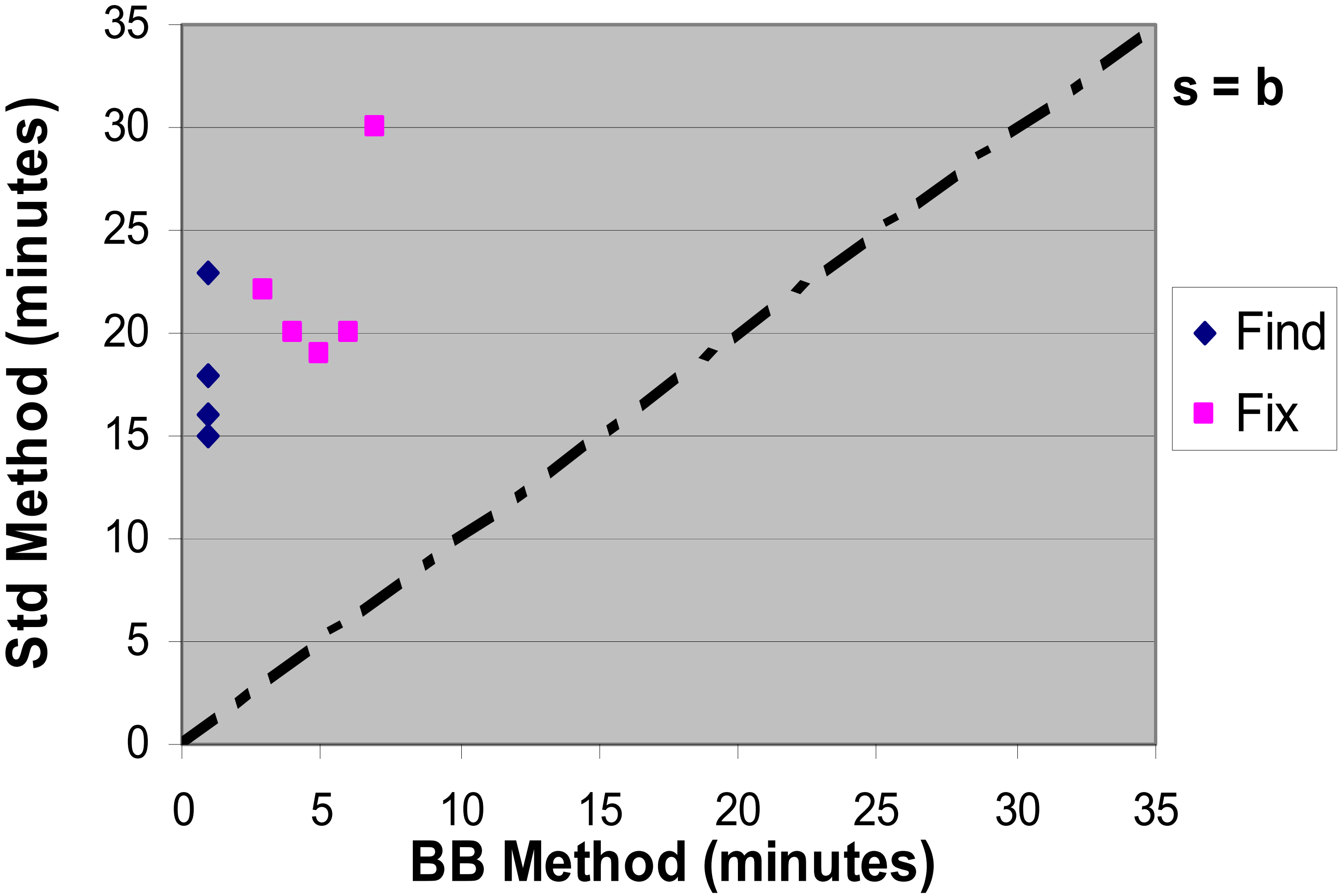
False Negatives in MOUT Domain



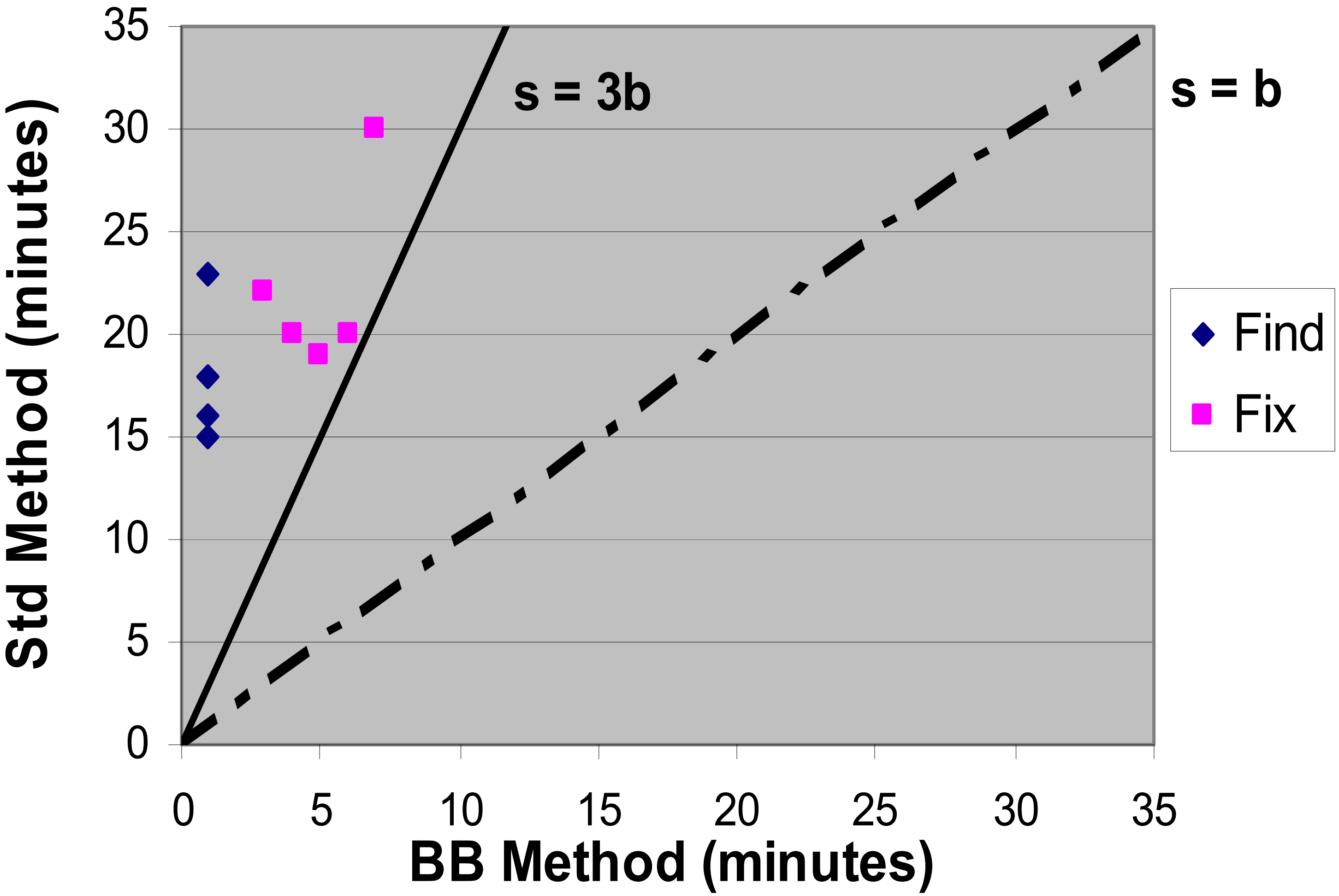
Behavior Bounding as a Validation Tool



Behavior Bounding as a Validation Tool



Behavior Bounding as a Validation Tool



Nuggets & Coal

- Simple, general behavior representation
- Leverages the natural organization of Soar agents
- Low cost to generate
- Performs well compared to sequential approach

- Simplicity leaves it susceptible to overgeneralization