# Using Action Abstraction to Evolve Effective Controllers

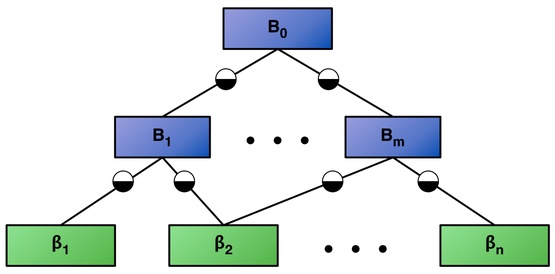
Brent E. Eskridge[1,2] & Dean F. Hougen[1]

[1]University of Oklahoma, Norman, OK, USA
[2]Southern Nazarene University, Bethany, OK, USA

## Introduction

- Development of controller for a simple task is straightforward, but
- Becomes impractical with more tasks added
- Naïve approach is to develop **monolithic** controller
- Doesn't scale well!
- Need an approach to make development of complex controllers more practical

## Fuzzy Behavior Hierarchies

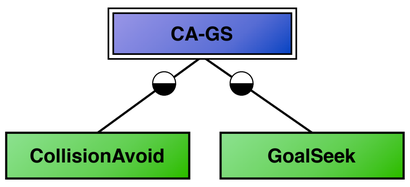



Generic hierarchical decomposition of behavior

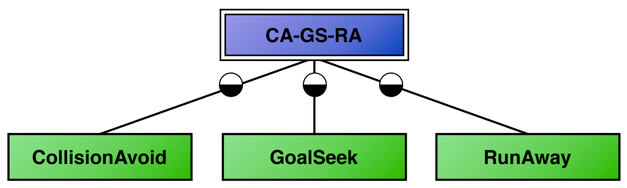
- Developed by Tunstel
- Keeps simple tasks separate
- Uses a hierarchy of controllers
- Single controller for each simple task
- High-level controllers to coordinate
- Implemented used fuzzy rule sets
- Coordination uses action abstraction
- Allows for the use of state abstraction also

## Navigation Problem

- Used two conceptually simple navigation tasks
- Examples of relatively simple composite tasks




CA-GS task: Navigate to goal while avoiding obstacles




CA-GS-RA task: Navigate to goal while avoiding obstacles and steering clear of hazards

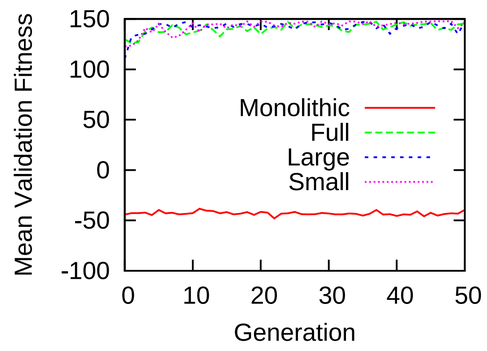
## Evolving Fuzzy Rulesets

- Tunstel originally used genetic programming
- Ensuring valid fuzzy rules is computationally expensive
- Used grammatical evolution instead
- Only valid rules are generated by grammar

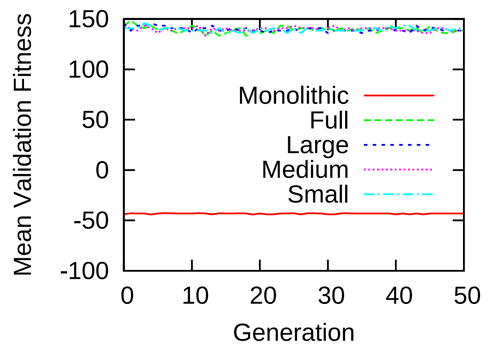
## Experiments

- Environments were unbounded and continuous
- 2-D and 3-D environments were used
- Tested effects of 4 levels of state abstraction on performance
- Used existing controllers for simple tasks
- Evolved coordination controller only

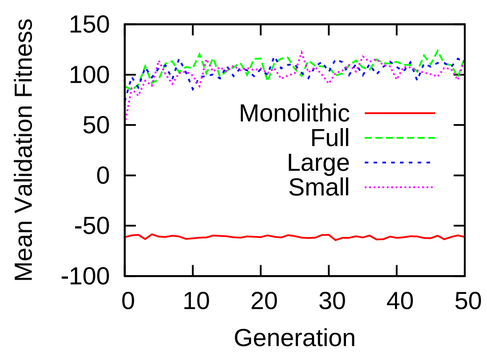
## Results



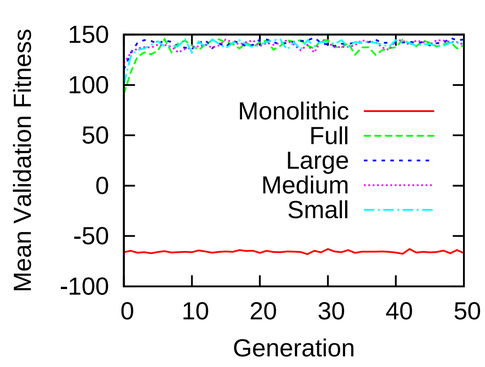

2-D CA-GS




3-D CA-GS




2-D CA-GS-RA




3-D CA-GS-RA

## Discussion

- State-action space for monolithic CA-GS-RA controller is smaller than hierarchical, but
- Effective hierarchical controllers were evolved quickly
- Could not evolve effective monolthic controllers
- Success due to abstract action-space of hierarchical controllers
- No performance difference in state abstraction levels
- Results don't reflect effort to develop controllers for simple tasks
- Can develop those with minimal effort

## Conclusions

- Action abstraction was fundamental to success
- Could not evolve effective monolithic controllers
- Hierarchical controllers were evolved quickly
- Existing controllers for simple tasks were reused
- Development of more complex controllers can be more practical


eskridge@ou.edu