6.836 Project Proposal
Embodied Elevators

Kenneth C. Barr
kbarr@mit.edu

March 26, 2001

Abstract

Today's intelligent elevators rely on a central point of control. I am curious how well elevators would perform if each was a "creature" that made its own decisions based on its environment. By implementing "Embodied Elevators" in a C++ simulation, I hope to improve my grasp of material presented in the first few 6.836 lectures.

Milestones

I am unsure how much effort each phase of the project will take, but I have listed major milestones in hopes that I will reach at least the first three.

- Creature Design

  Decide what sensors are essential for elevator operation. I am also considering several "luxury" sensors such as weight and vision sensors that would likely improve performance (eg, there's no reason for an elevator to respond to a call if it is knows it cannot carry another person). Another important feature would be inter-elevator communication so the elevators could work together reducing redundant trips. Special-purpose elevators could be accounted for as well: elevators that service only certain floors of a skyscraper; service elevators; and/or LCS-style elevators of which only two reach the basement.

- Subsumption framework in C++

  I hope to design a framework for subsumption architecture in C++ so that I can begin with simple behaviors and add more complex behavior layers via suppression, defaulting, and inhibition. After a brief talk with Professor Brooks, I worry that this may be more complex than it sounds, but I will spend part of my time budget on this problem if only to discuss the difficulties.

- Benchmark Suite

  While I have chosen this project for personal exploration rather than optimization of elevator service, I am interested in seeing how well Embodied Elevators perform compared to traditional elevators. It is necessary to model a traditional design and a benchmark suite that can measure performance quantitatively and qualitatively. The benchmarks would also help to refine and minimize the necessary sensors on each elevator.

- Learning

  While it departs from the material presented in the first few lectures, one can hardly resist trying to design “Hitchhiker’s Guide Elevators” that can predict the future. Several modern systems possess this ability, keeping a timestamped database of call requests. It would be interesting to implement this behavior with neural-network-inspired learning to emulate real learning rather than with brute-force memory lookups that likely form the basis of current systems.