Software for the Real World

Edward A. Lee
UC Berkeley

computers without actuators and
sensors are destined to look like this.

Software Design and Productivity Workshop
NSF, April 17-18, 2001

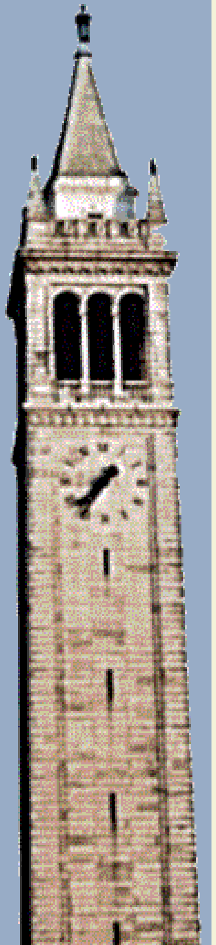
What is an Embedded System?

prime example today

- One or more computers
  - but not first-and-foremost a computer
- Interaction with physical processes
  - sensors, actuators
- Reactive
  - operating at the speed of the environment
- Heterogeneous
  - hardware/software, mixed architectures
- Networked
  - adaptive software, shared data, resource discovery

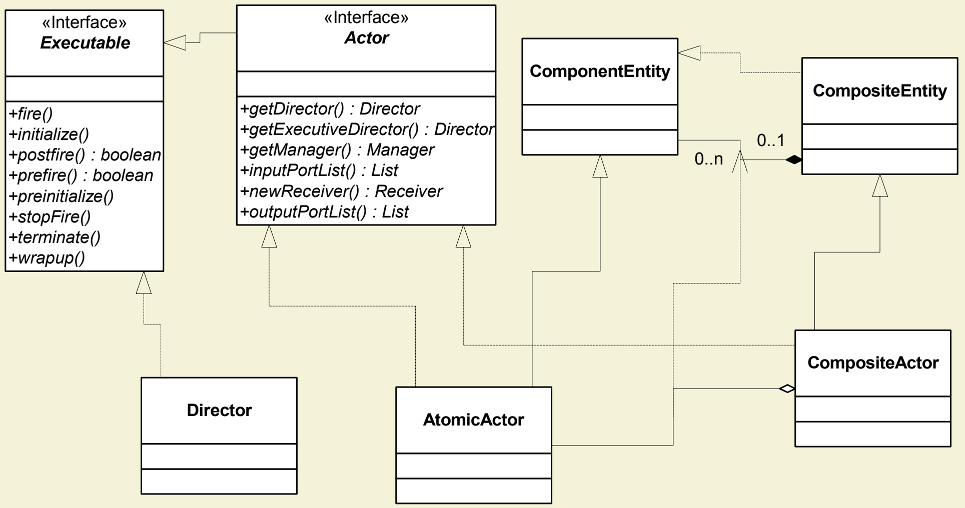
Why is Embedded SW an Issue?

- Embedded systems becoming networked
  - more complex, more vulnerable
  - can no longer use static point designs
- Focus on non-functional properties is new for SW
  - real-time, fault recovery, power, security, robustness
- Neglected area
  - computer science has largely ignored it
  - best-of-class methods don't help much



E.g. Object-Oriented Design

- Call/return imperative semantics
- Concurrency is via ad-hoc calling conventions
  - band-aids: futures, proxies, monitors
- Poorly models the environment
  - which does not have call/return semantics
- Nothing at all to say about time



Object modeling emphasizes inheritance and procedural interfaces.

Actor modeling emphasizes concurrency and communication abstractions.

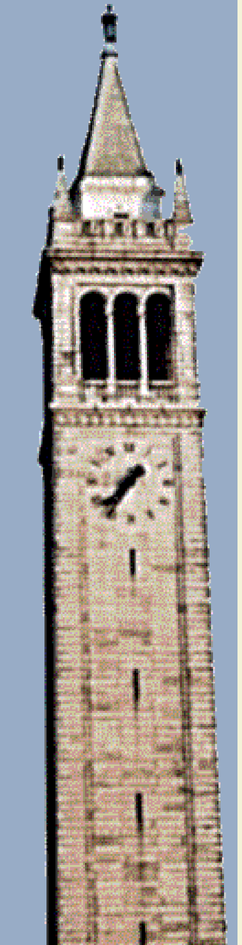
E.g. Real-Time Corba

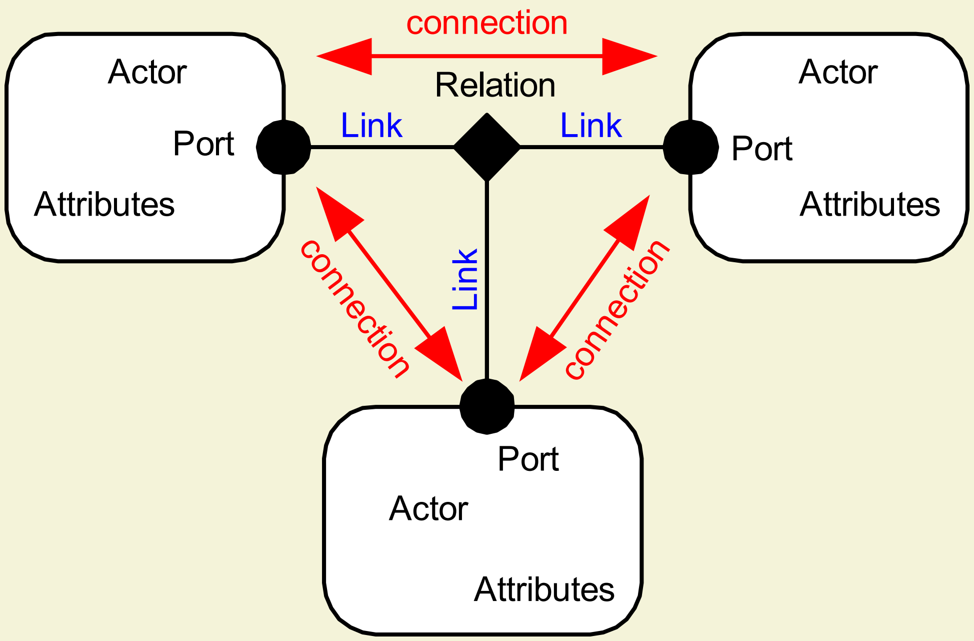
Component specification includes

- worst case execution time
- typical execution time
- cached execution time
- priority
- frequency
- importance

This is an elaborate prayer…

Alternative View of SW Architecture: *Actors* with *Ports* and *Attributes*





Model of Computation:

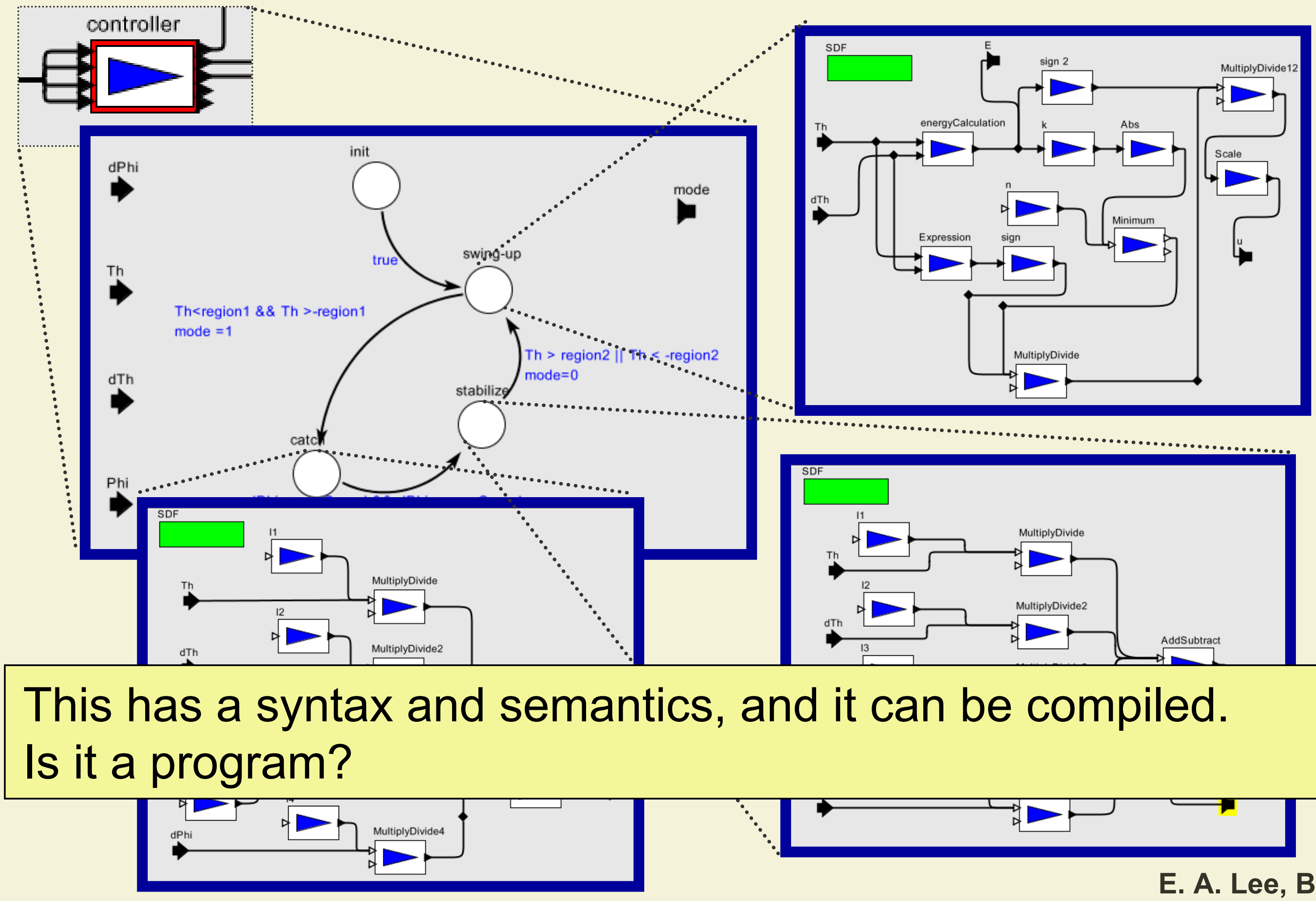
- Messaging schema
- Flow of control
- Concurrency

Examples:

- Time triggered
- Process networks
- Discrete-event systems
- Dataflow systems
- Publish & subscribe

Key idea: The model of computation is part of the framework within which components are embedded rather than part of the components themselves.

What a Program Might Look Like



This has a syntax and semantics, and it can be compiled.
Is it a program?

Opportunities

- Modernize concurrency
  - It’s time to move beyond Dijkstra’s 60’s methods
- Reintroduce time into models of computation
  - let’s get rid of “prioritize and pray”
- Generate Generators
  - translation between models
- Get domain specific
  - specialized modeling means practical formal methods
- Model modeling (meta models)
  - create the language that talks about modeling methods
- Understand heterogeneous modeling

Results exist that show there is promise, such as time-triggered architectures and synchronous/reactive languages