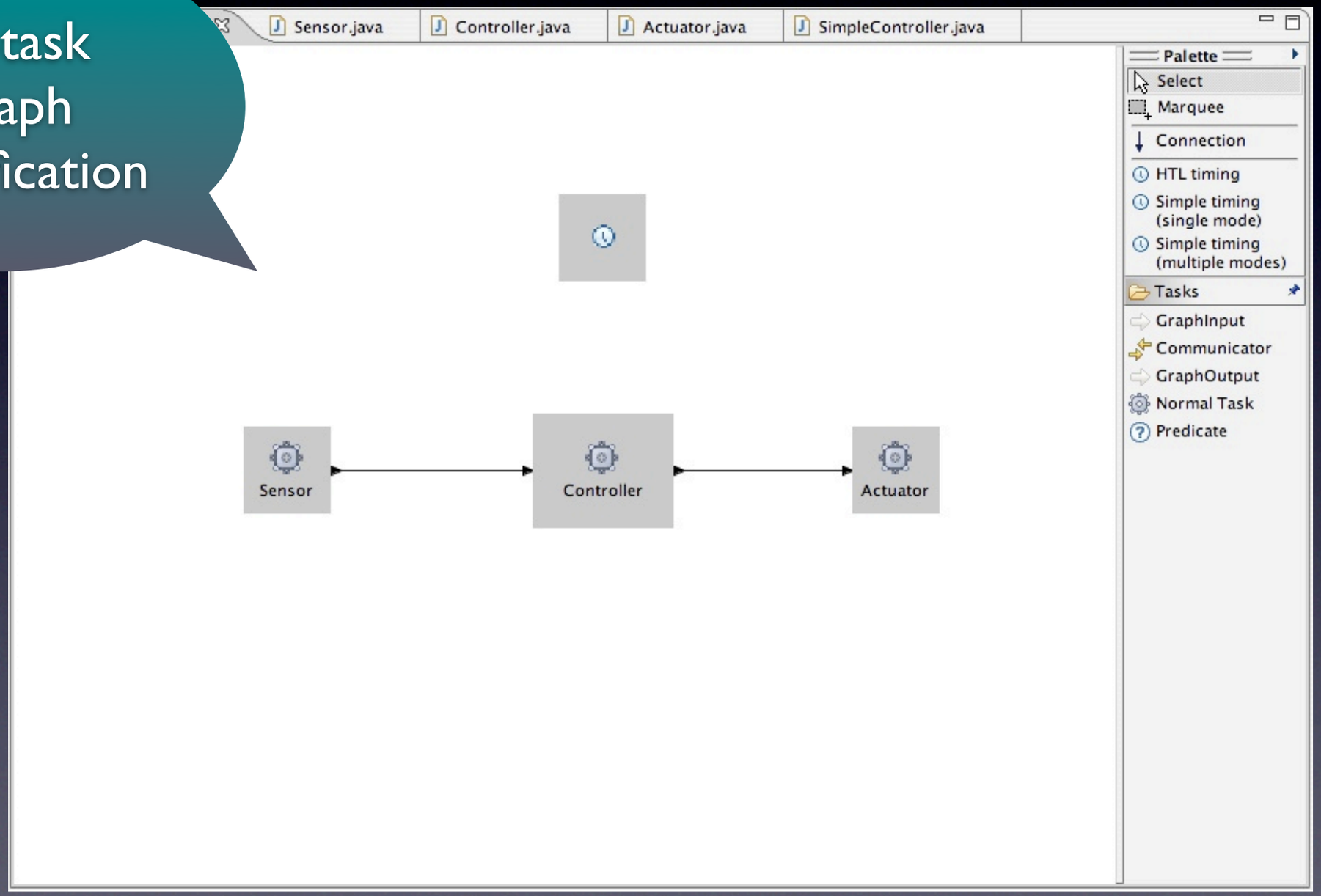
Time-Portable Real-Time Programming with Exotasks

Christoph Kirsch
Universität Salzburg

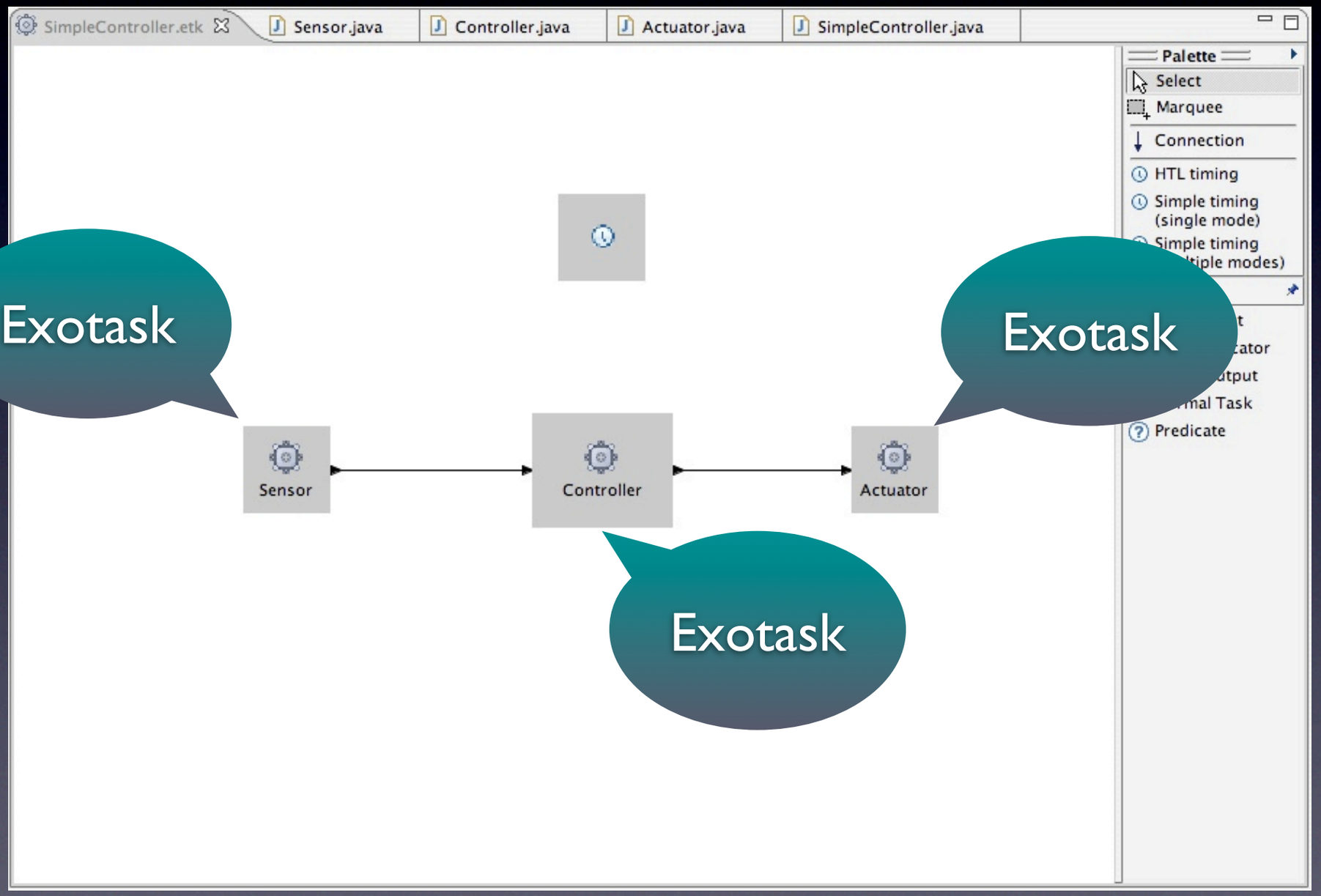
Joint work with J. Auerbach, D. Bacon, D. Iercan, V.T. Rajan, H. Röck, and R. Trummer
CHESS Seminar, UC Berkeley, February 2007

Exotask Eclipse Plugin

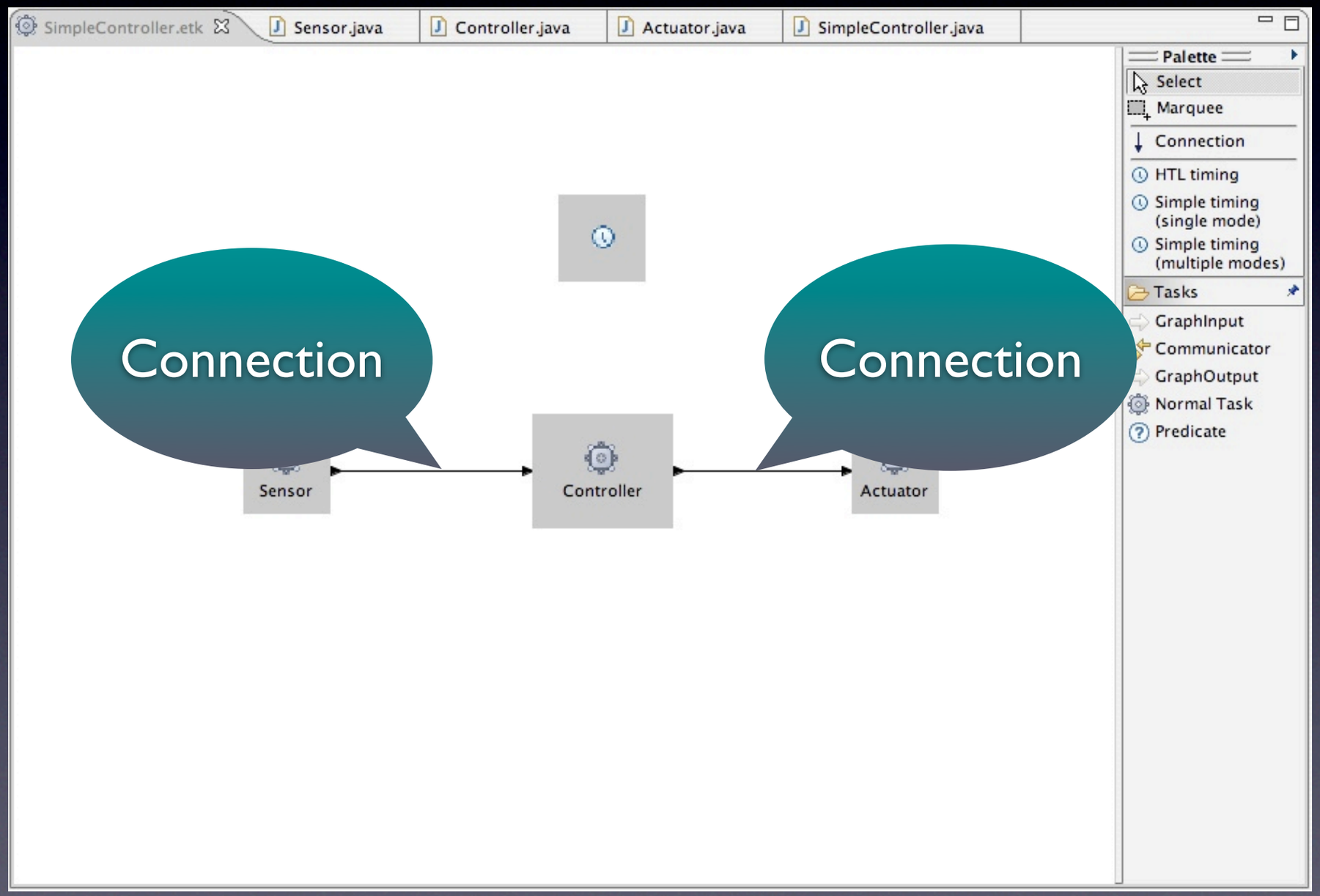
Exotask Graph Specification



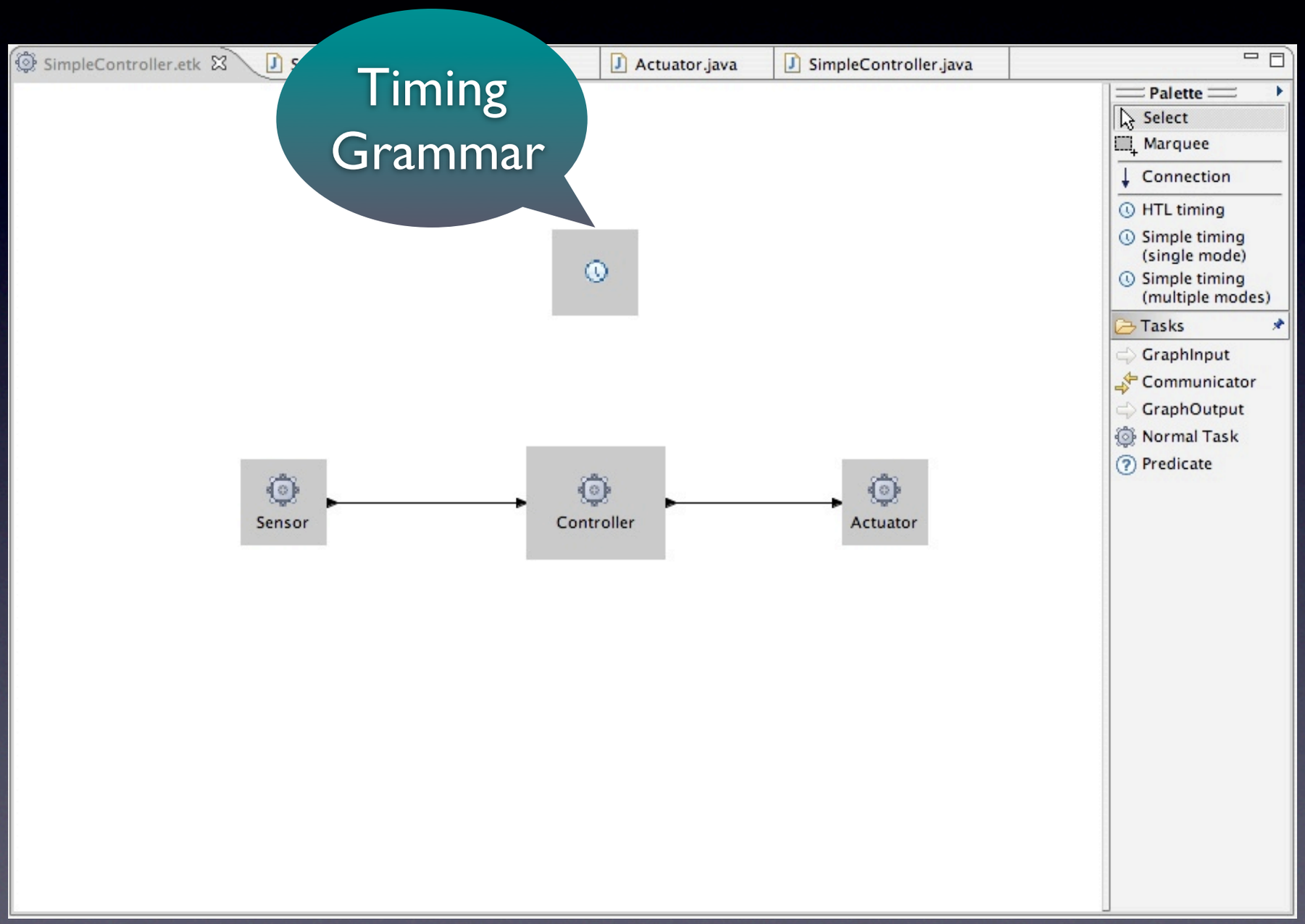
Exotask Eclipse Plugin



Exotask Eclipse Plugin



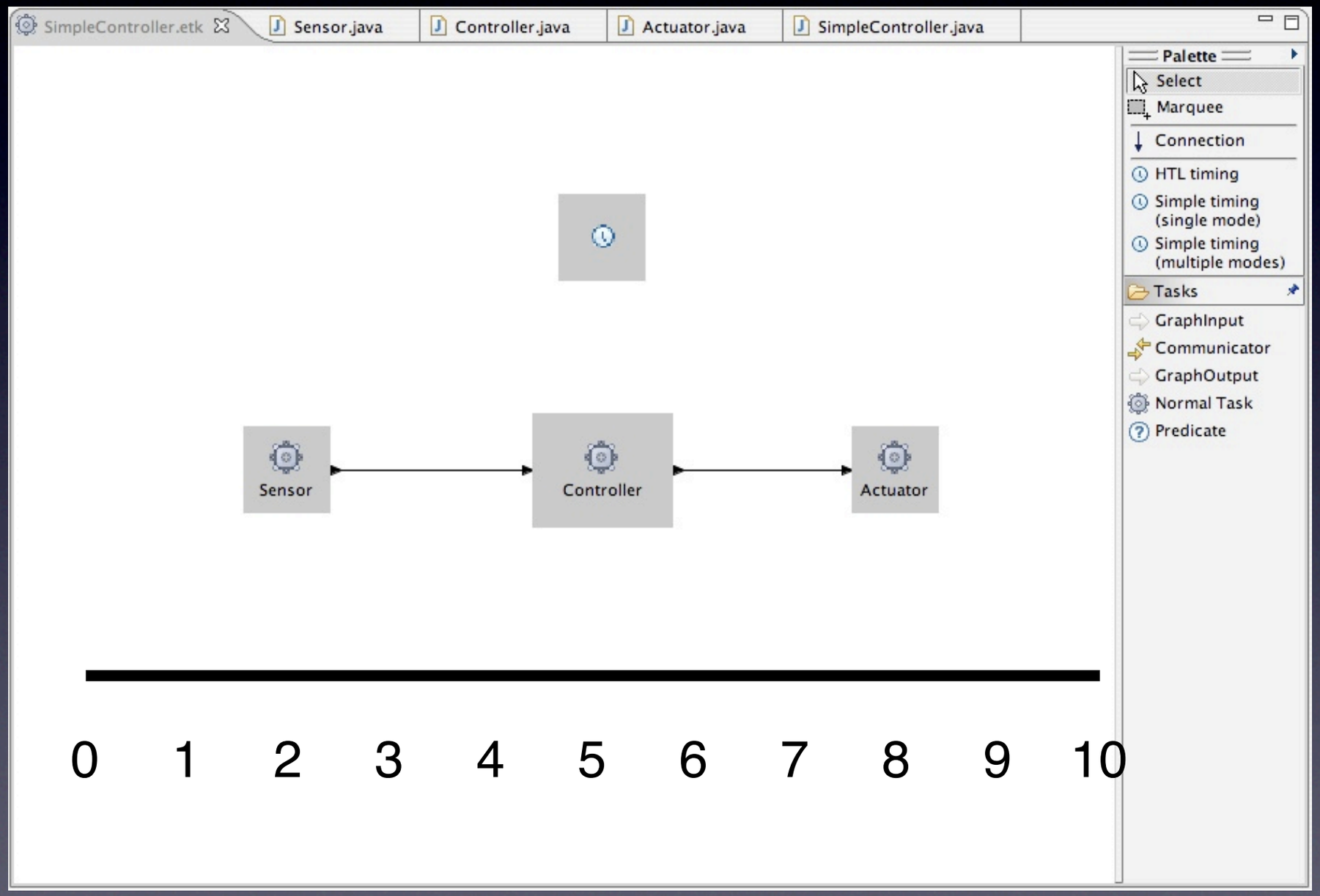
Exotask Eclipse Plugin



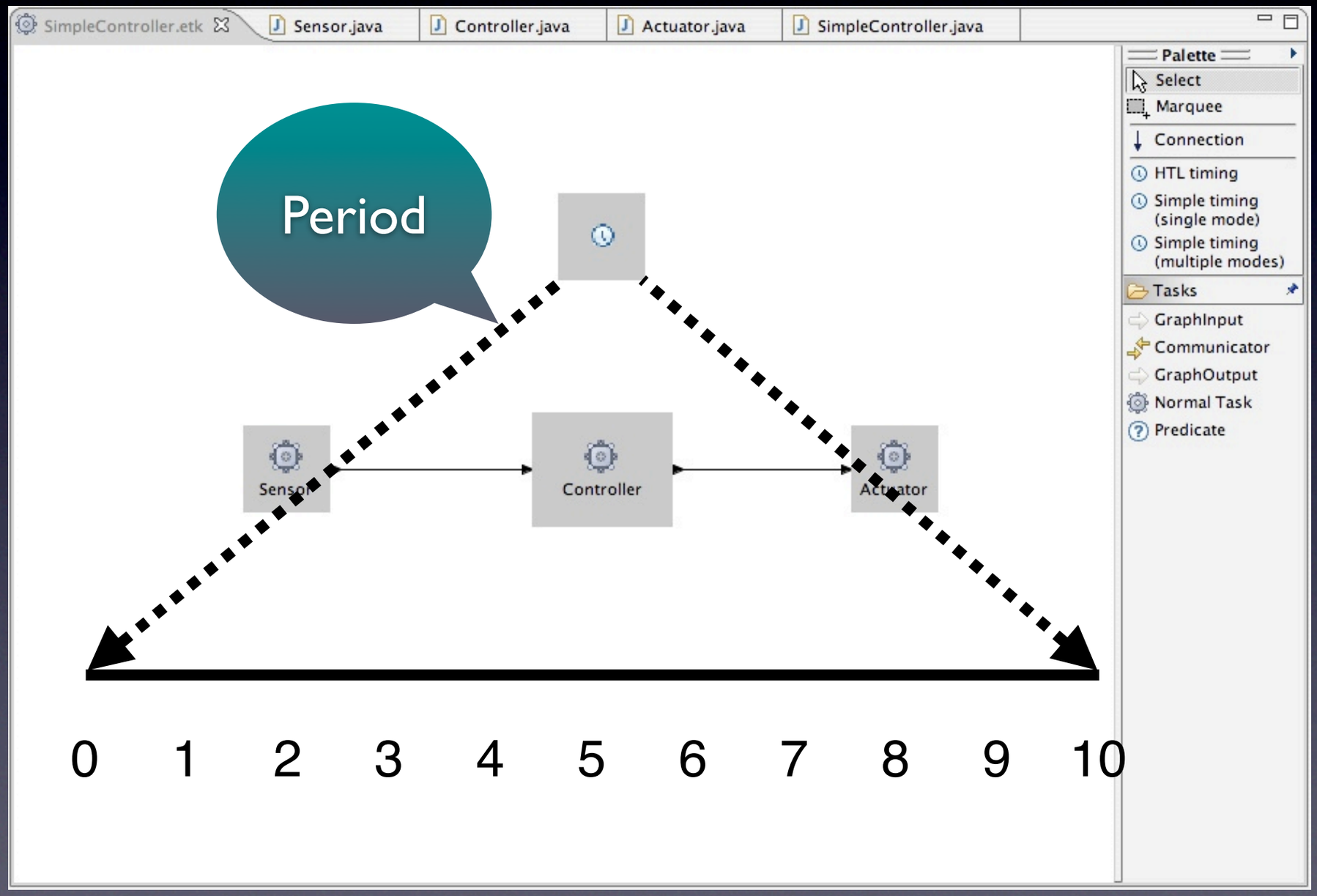
Exotasks in Java

```
public class Controller implements Runnable {
  // Code generated from the specification:
  private ExotaskInputPort in0;
  private ExotaskOutputPort out0;
  public Controller(ExotaskInputPort in0,
                    ExotaskOutputPort out0) {
    this.in0 = in0;
    this.out0 = out0;
  }
  // Code written by the user:
  public void run() {
    double[] sensorData =
      (double[]) in0.getValue();
    out0.setValue(
      new Double(control(sensorData)));
  }
  private double control(double[] sensors)
  { ... the actual control algorithm ... }
}
```

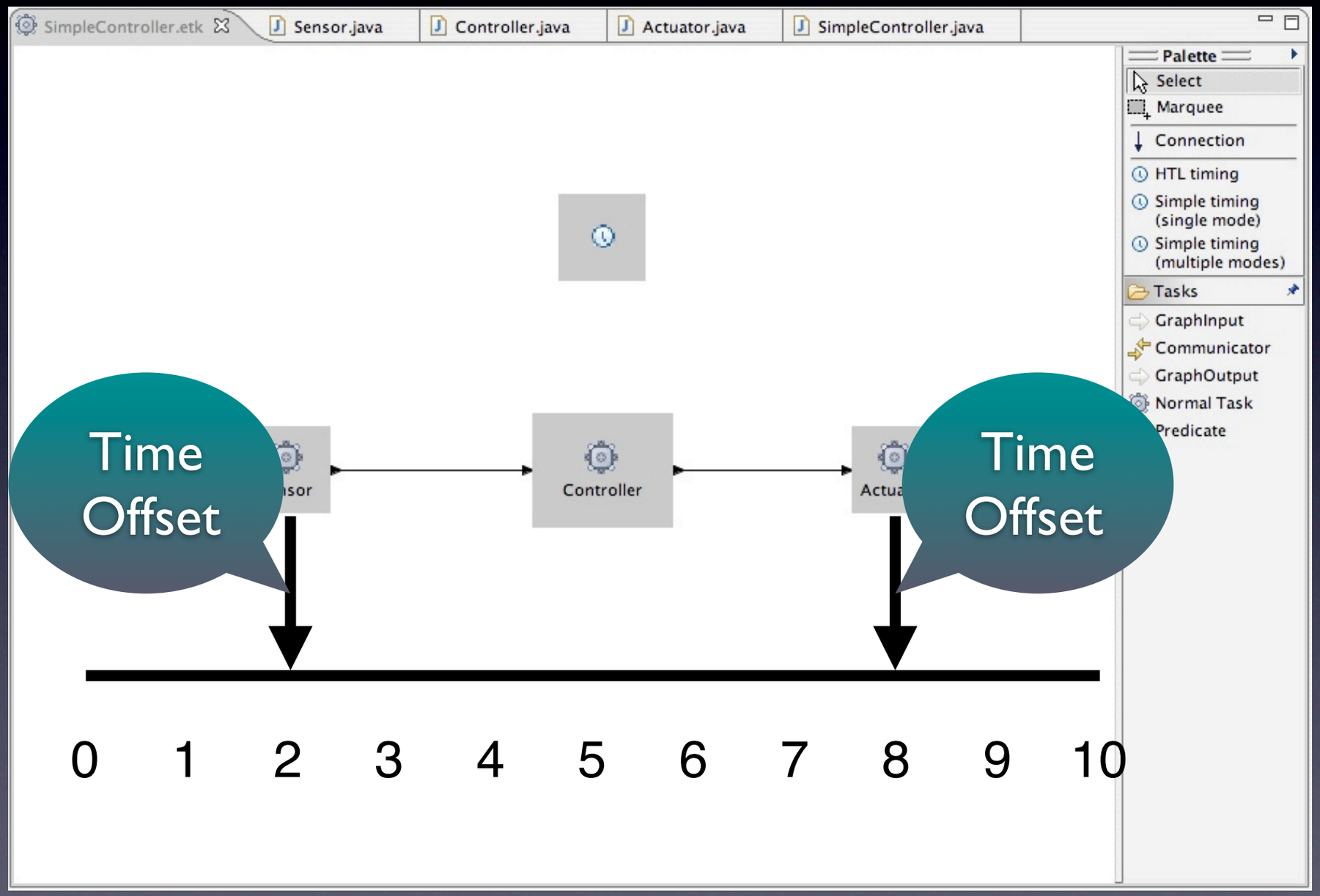
TT Grammar



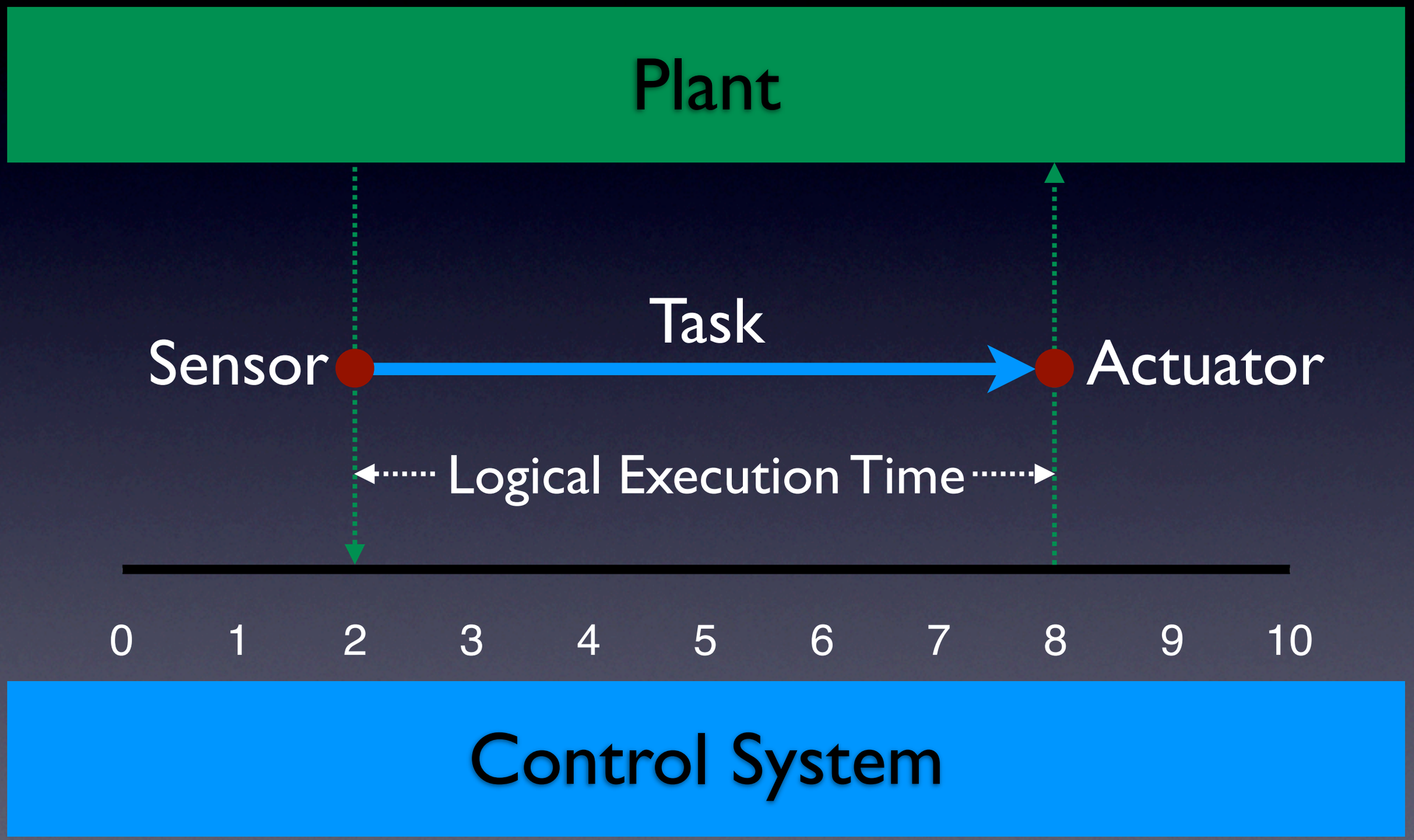
TT Grammar



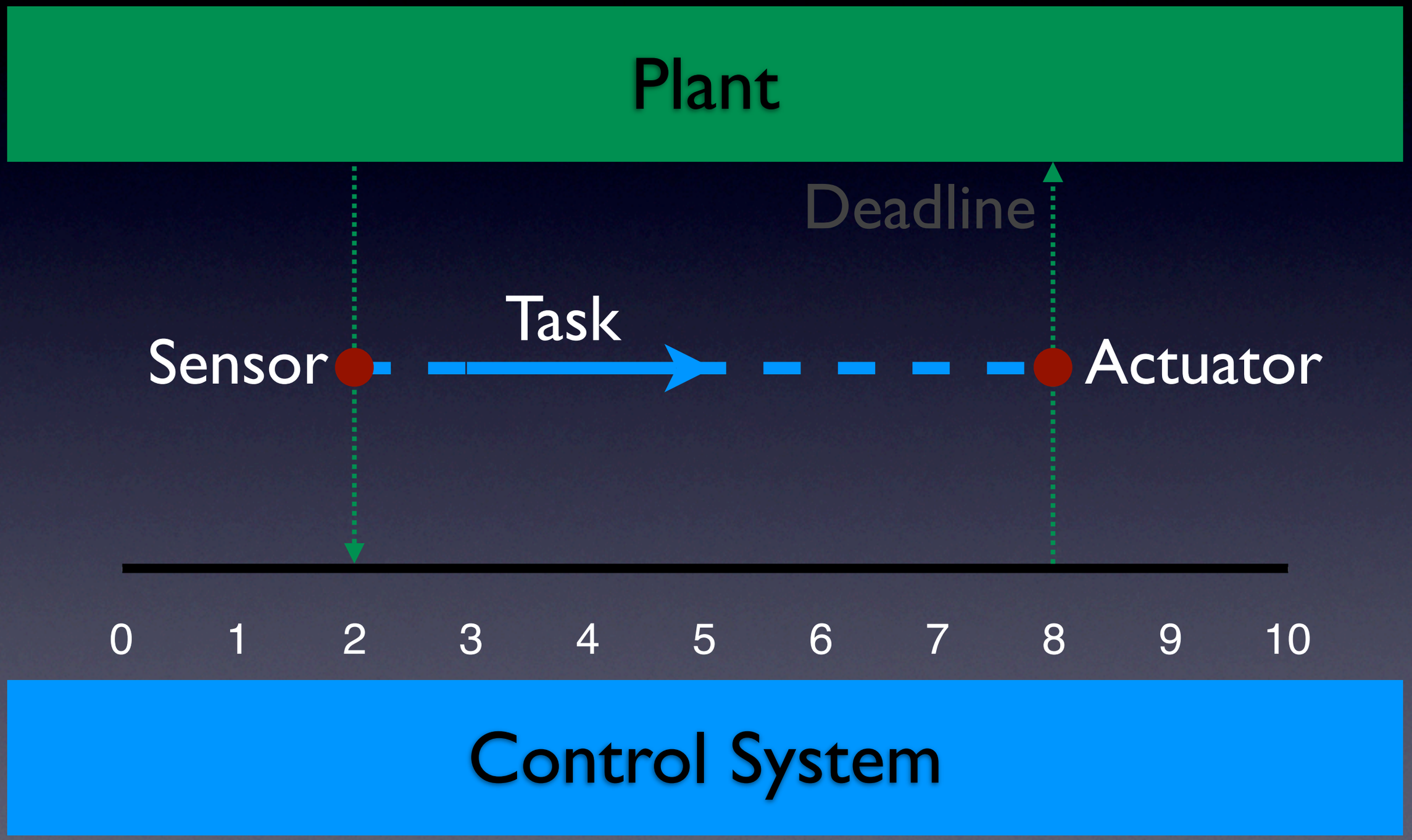
TT Grammar



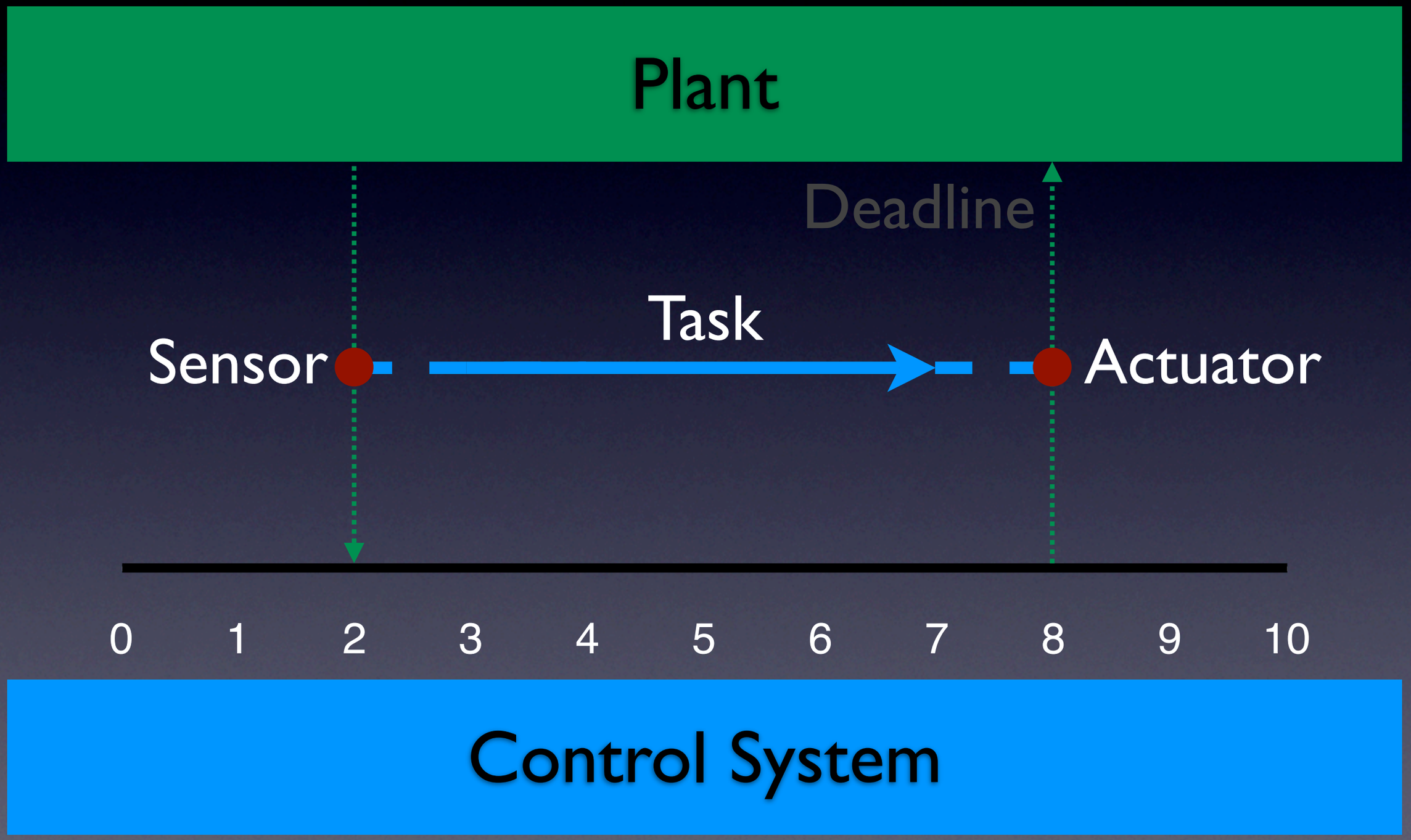
Logical Execution Time



Actual Execution Time



Actual Execution Time



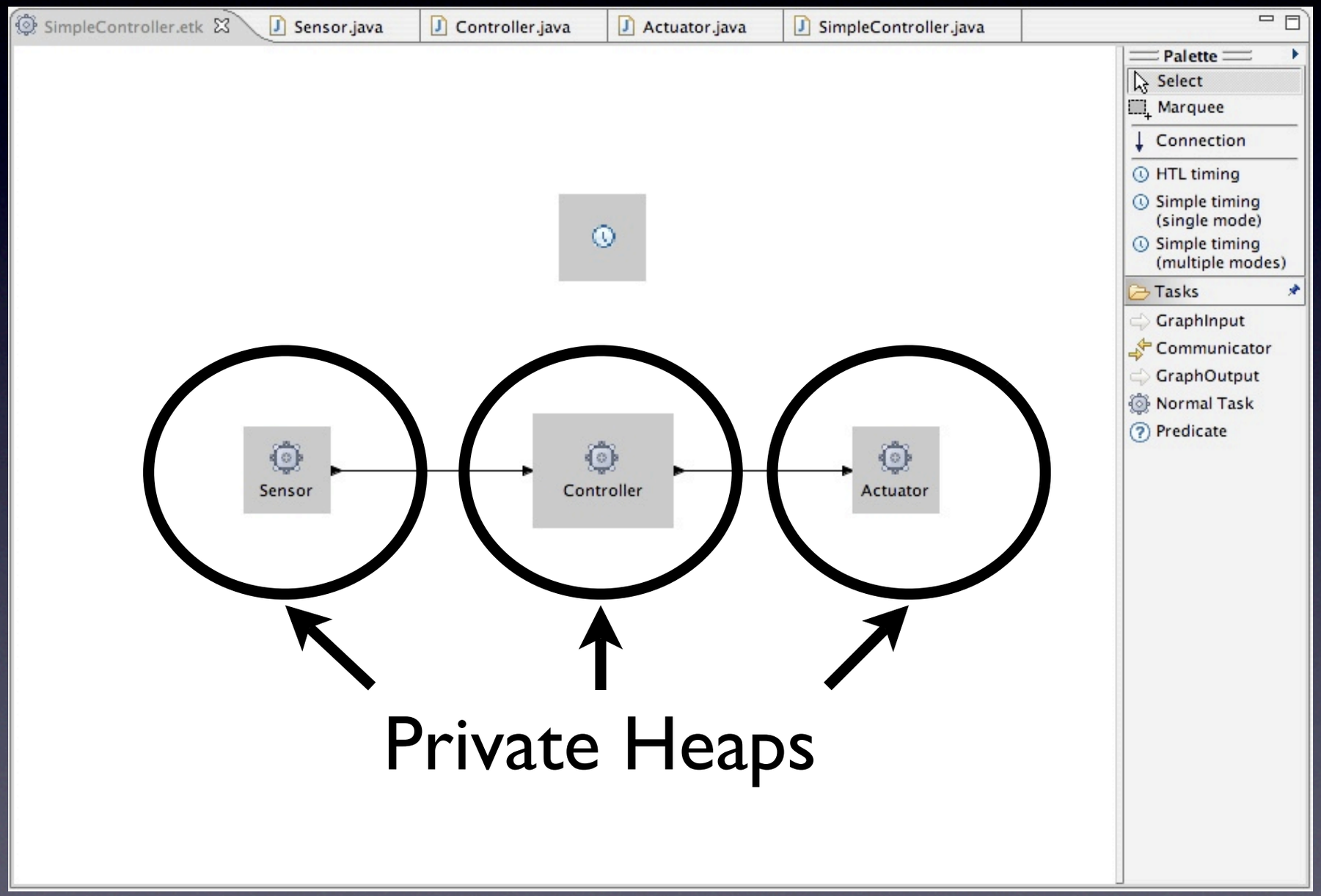
Time Portability

Plant

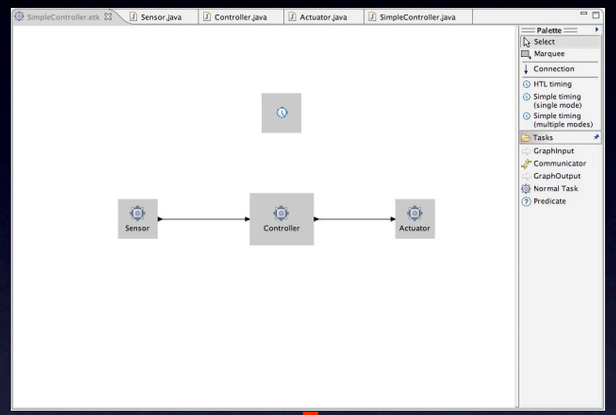
A system's I/O behavior is *input-determined* if,
for all sequences *I* of input values and times,
the system always produces
unique sequences *f(I)* of output values and times.

Control System

Exotask Isolation



1. Validate, 2. Instantiate, 3. Schedule & Execute



Exotask Graph Specification

instantiate

validate

Exotask Graph

invokes

Exotask Code

```
public class Controller implements Runnable {
  // Code generated from the specification:
  private ExotaskInputPort in0;
  private ExotaskOutputPort out0;
  public Controller(ExotaskInputPort in0,
                    ExotaskOutputPort out0) {
    this.in0 = in0;
    this.out0 = out0;
  }
  // Code written by the user:
  public void run() {
    double[] sensorData =
      (double[]) in0.getValue();
    out0.setValue(
      new Double(control(sensorData)));
  }
  private double control(double[] sensors)
  { ... the actual control algorithm ... }
}
```

schedule

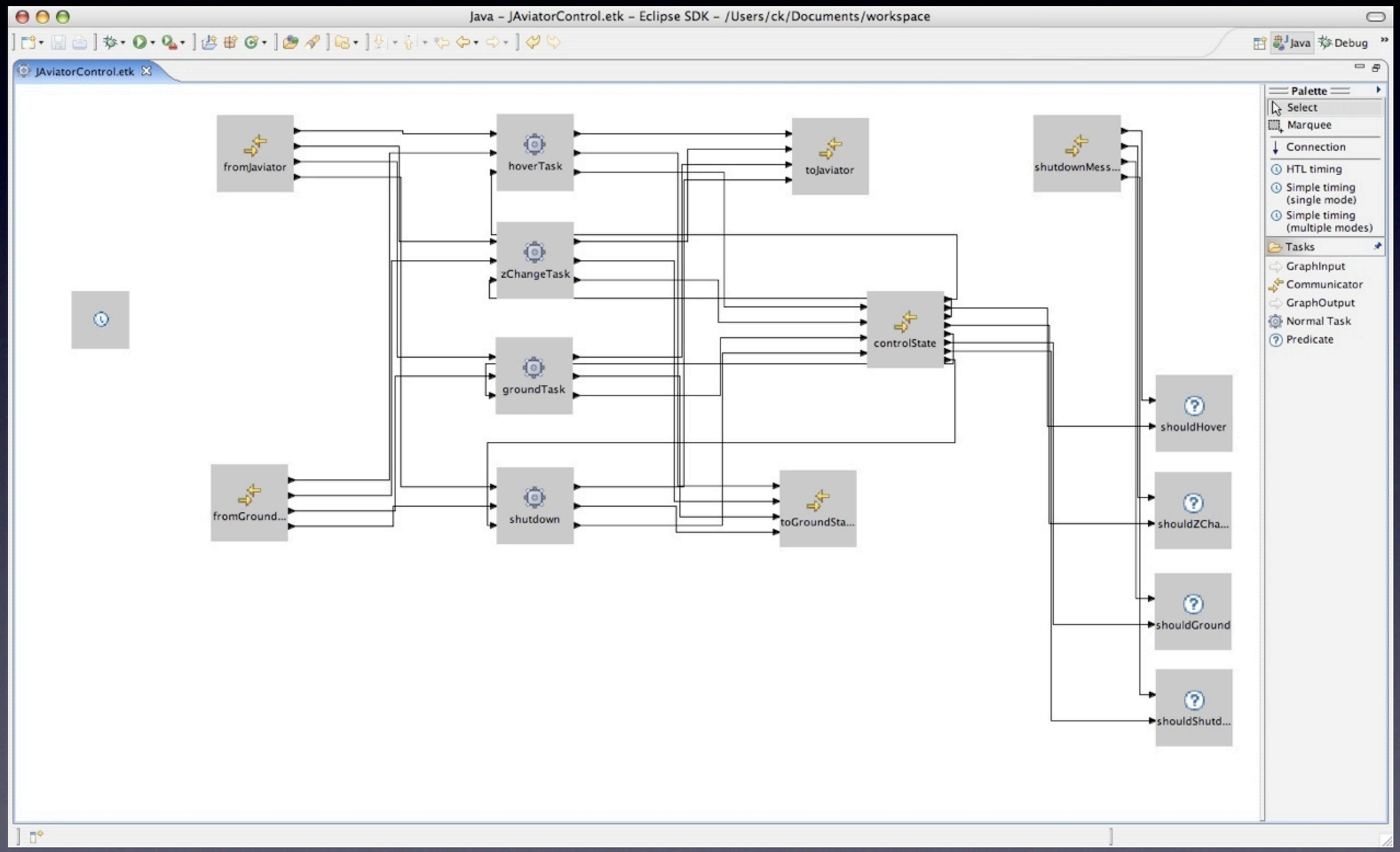
execute

Exotask Runtime System

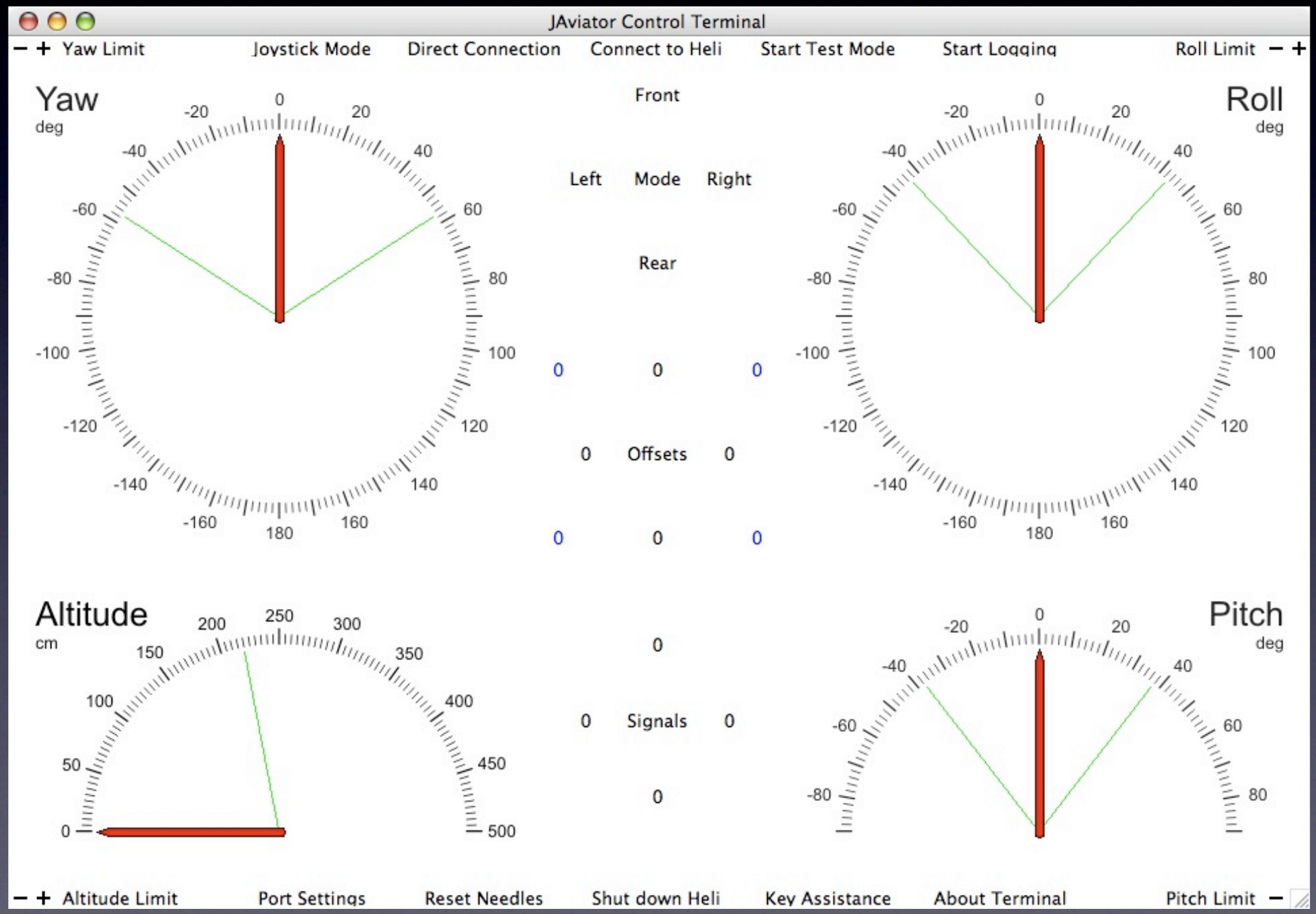
The JAviator

50Hz Hard Real Time

JAviator Controller

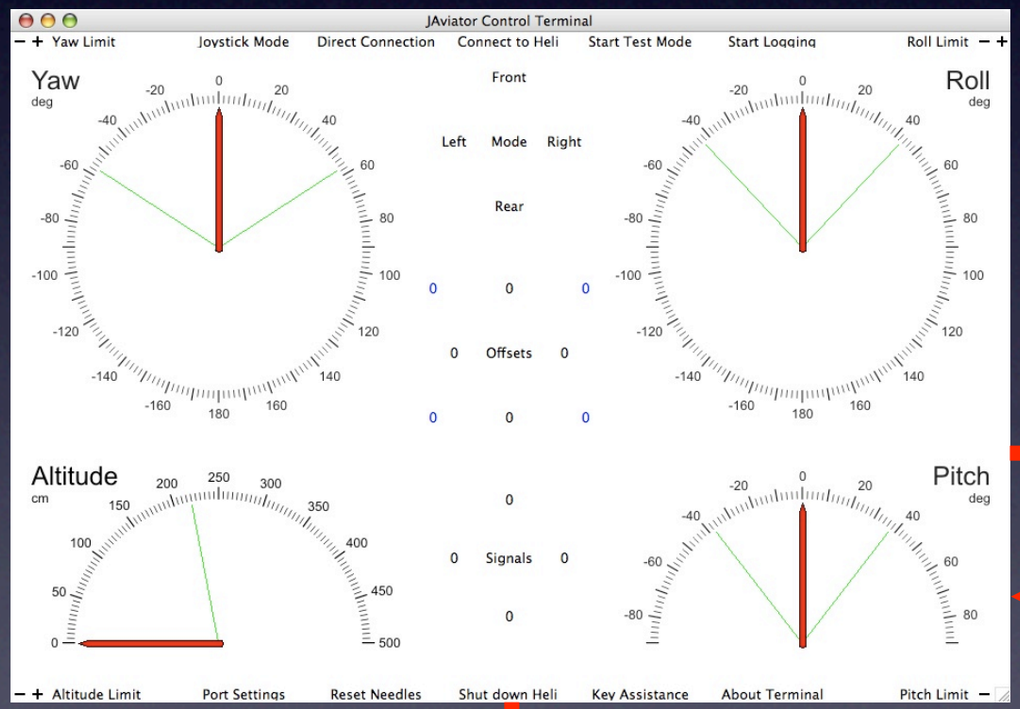


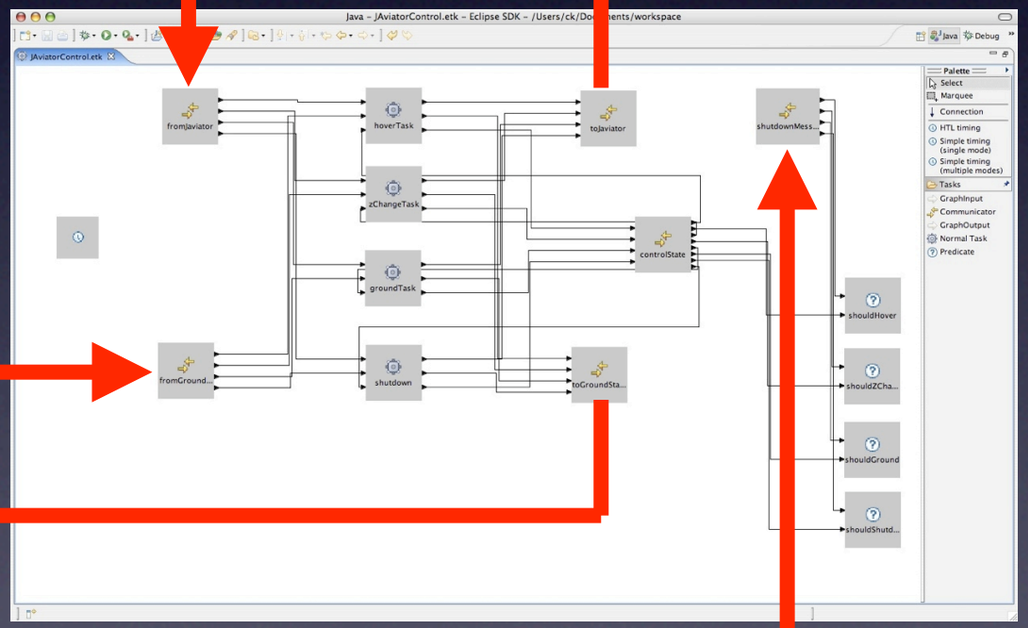
Control Terminal



Mock JAviator

Demo

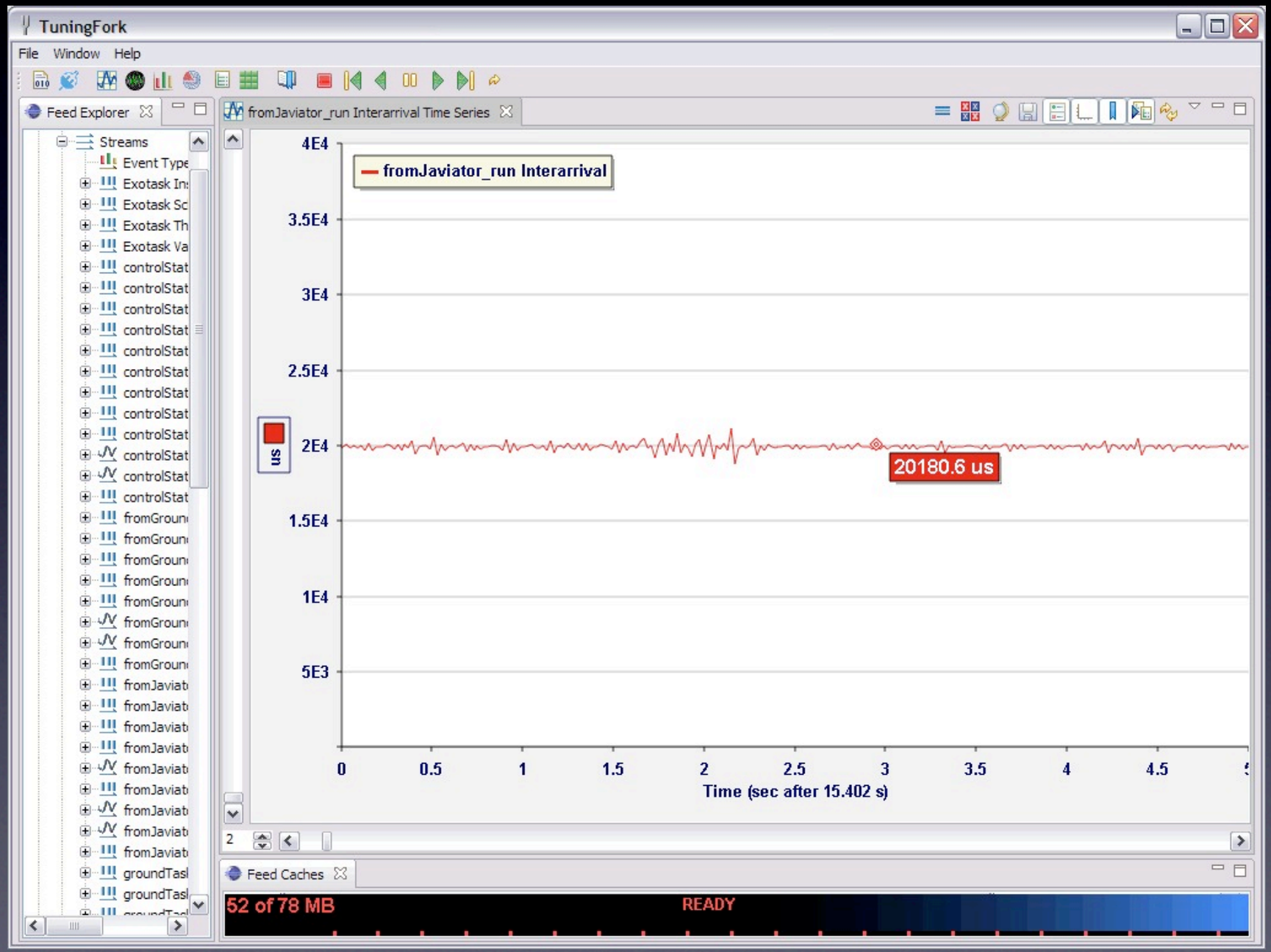




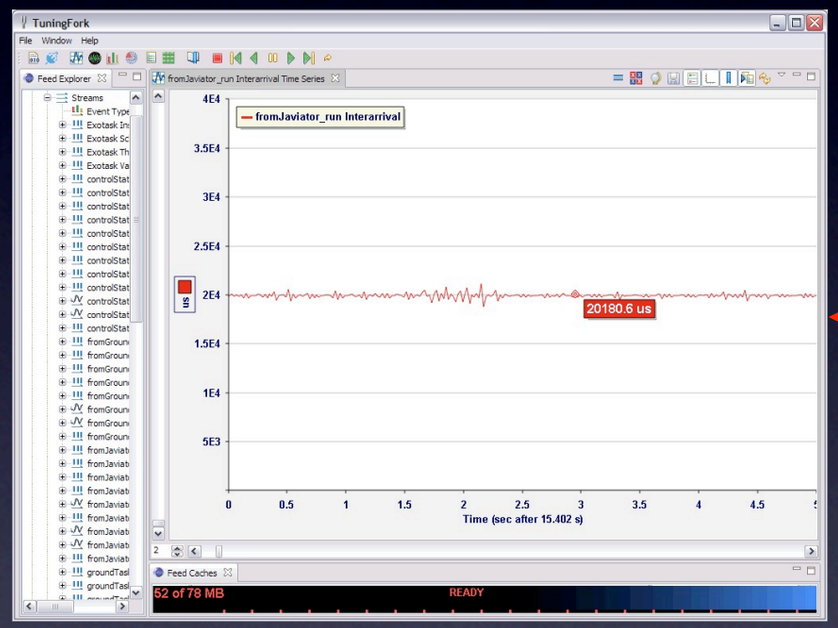
Control Terminal

Exotask Controller

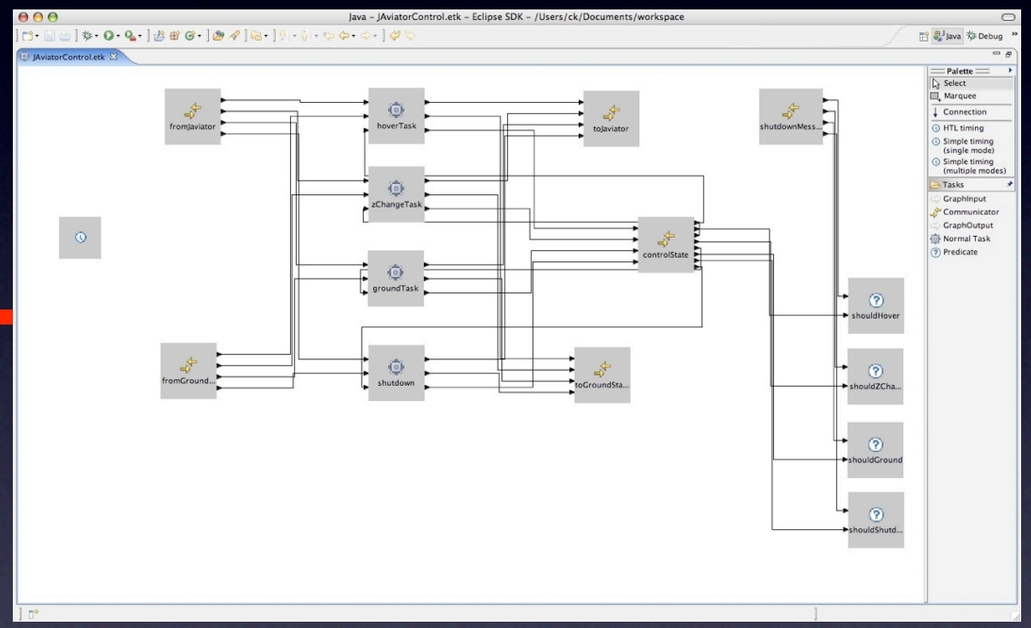
IBM's TuningFork



Experiments

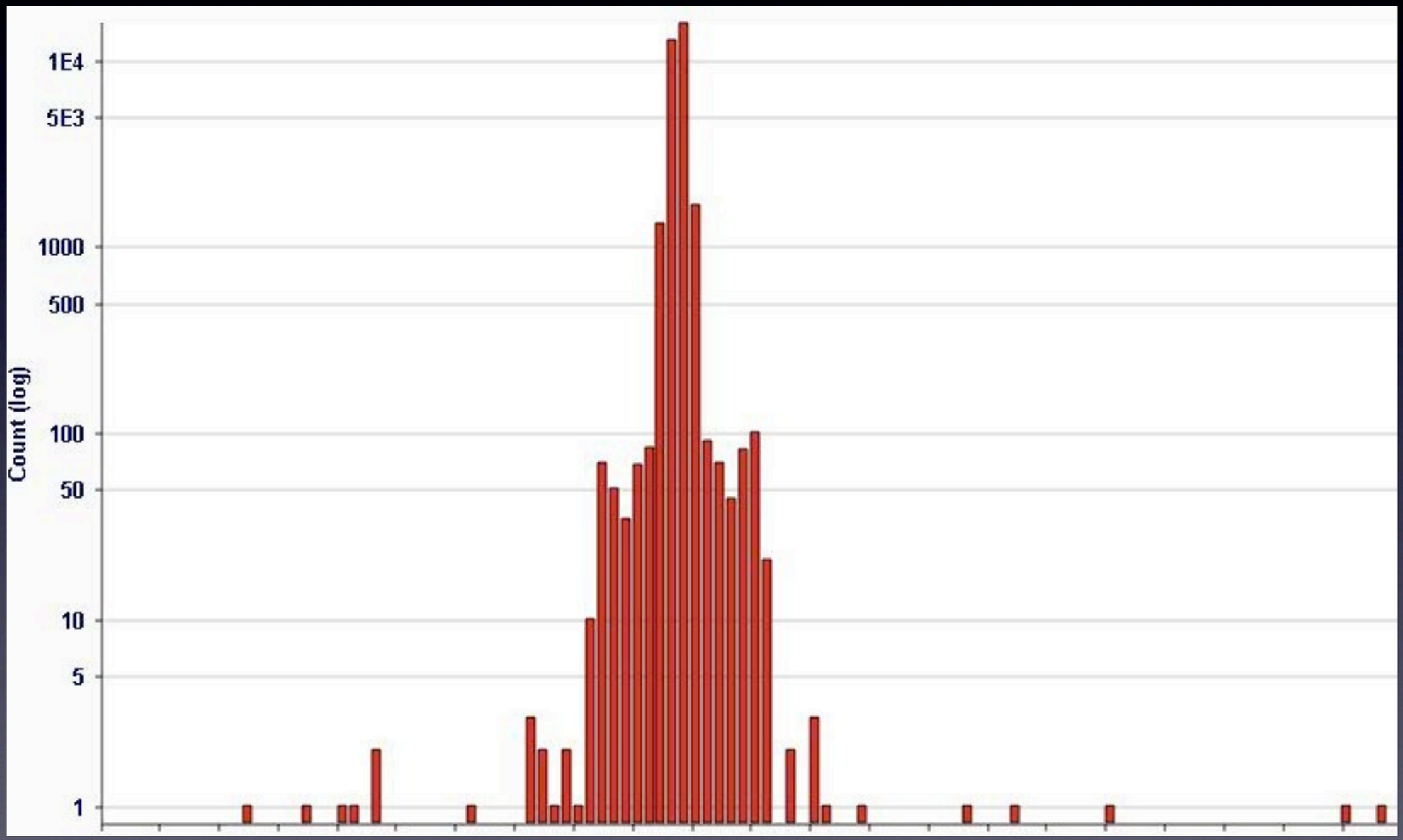


TuningFork



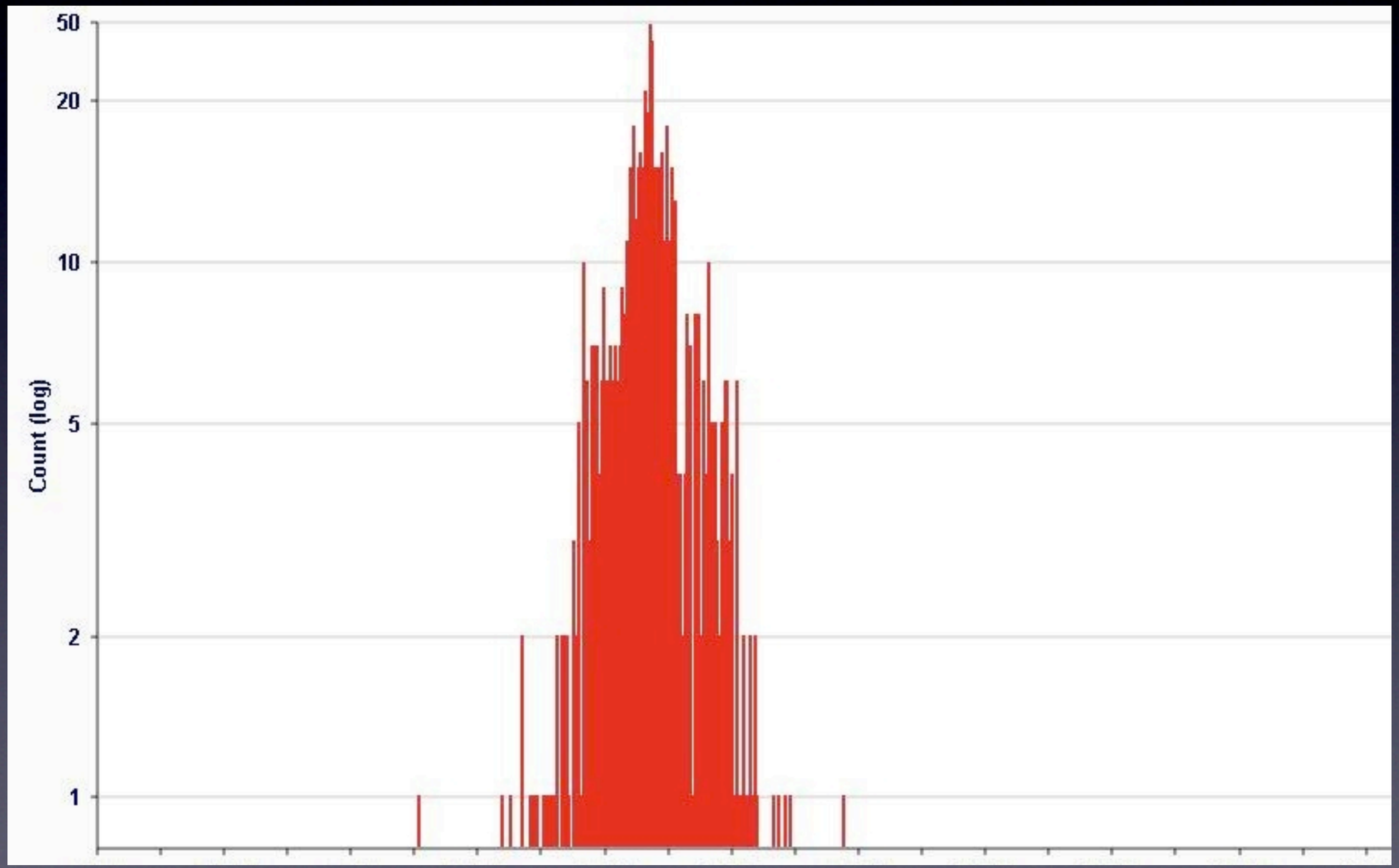
Instrumented Exotask System

No GC in Progress

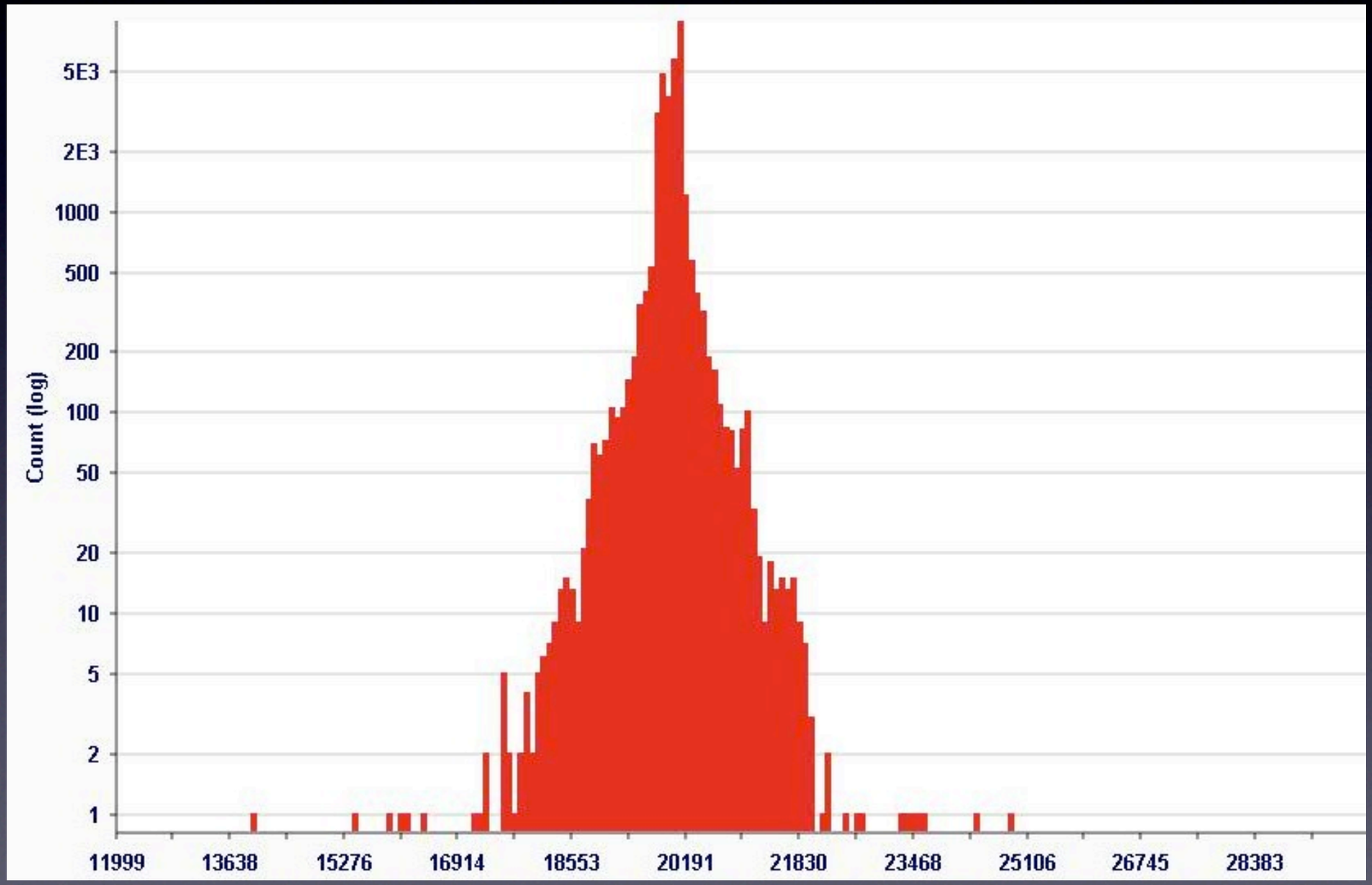


~3ms

During GC

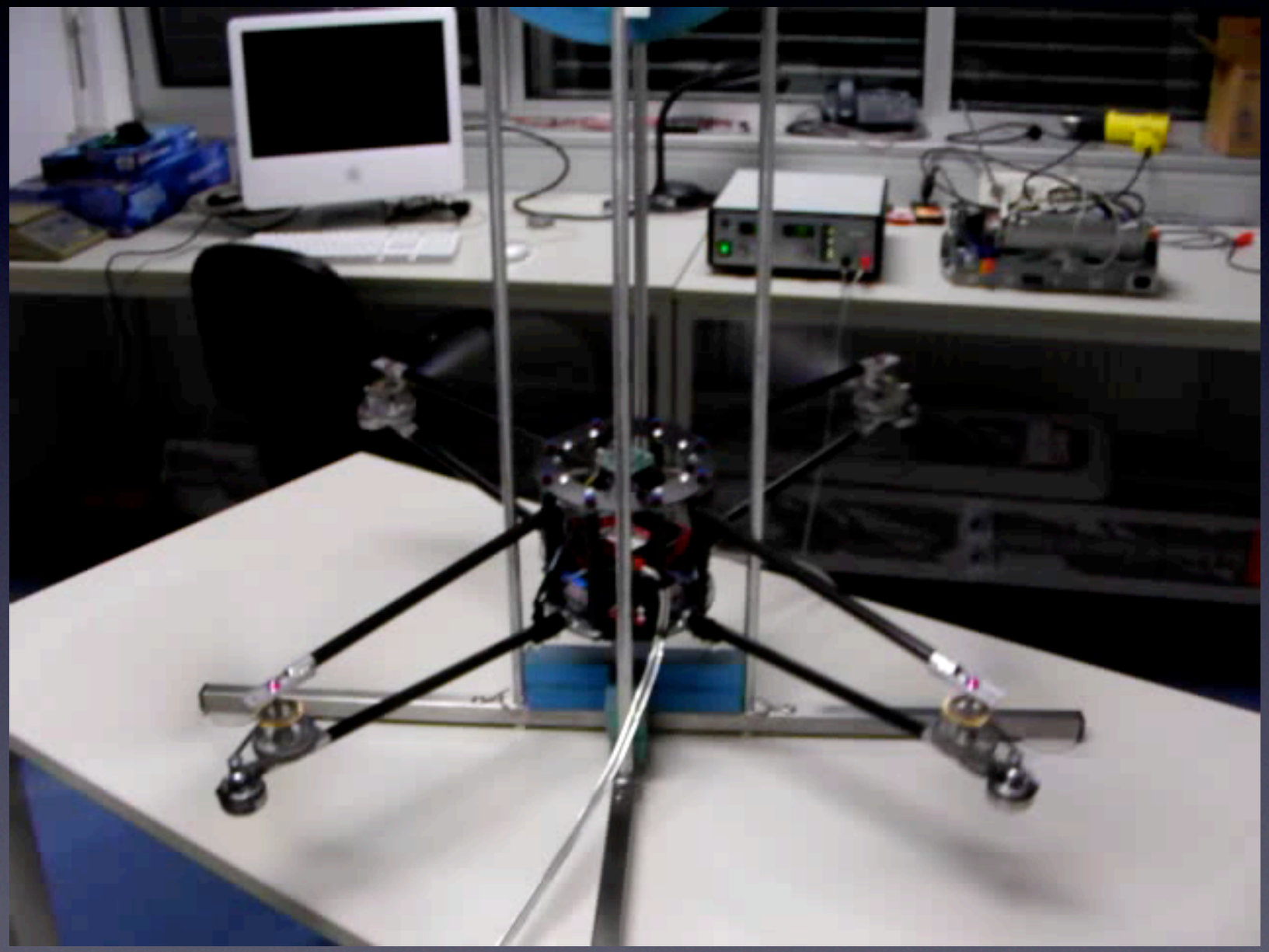


1.4GHz Laptop



~4ms

Clamp-free Hover



Thank you