COMP522 Modelling and Simulation

Hybrid and Multi-Formalism Modelling

Rifeng Ding

rifeng.ding@mail.mcgill.ca

Overview

- Hybrid Modeling System

- Purpose of the Project

- Tool: HyVisual in Ptolemy II

- Project: A Train System

- Demo

- Conclusion

Hybrid Modelling System [1]

- Heterogeneous systems including continuous-time subsystems interaction with discrete events

- Effective to model physical systems interacting with software or undergoing discrete mode changes

[1] Edward A.Lee and Haiyang Zheng. "Operational Semantics of Hybrid Systems." in Hybrid Systems: Computation and Control: 8th International Workshop, 2005, pp.25-53

Hybrid Modelling System

- Continuous Subsystem

  - Using differential equations

- Discrete Event

  - Using finite state machines

  - Transitions between states represent either discrete mode changes or actions taken by software subsystems

Purpose of this project

- Explore how to use hybrid modelling formalism to stimulate real-world system:
  - Tools: HyVisual in Ptolemy II
  - Simulated system: A Train System
- Show the advantages of hybrid modelling formalism

5

Tool: HyVisual in Ptolemy II

- Ptolemy Project [2]
    - UC Berkeley EECS Department
    - Study modeling , simulation, and design of concurrent, real-time, embedded systems
    - http://ptolemy.eecs.berkeley.edu/index.htm
- Ptolemy II
    - A open-source software framework supporting simulation experimentation with actor-oriented design

[2] UC Berkeley EECS Dept. "Ptolemy Project." Internet: http://ptolemy.eecs.berkeley.edu/, Oct. 7, 2005 [Apr. 20. 2012]
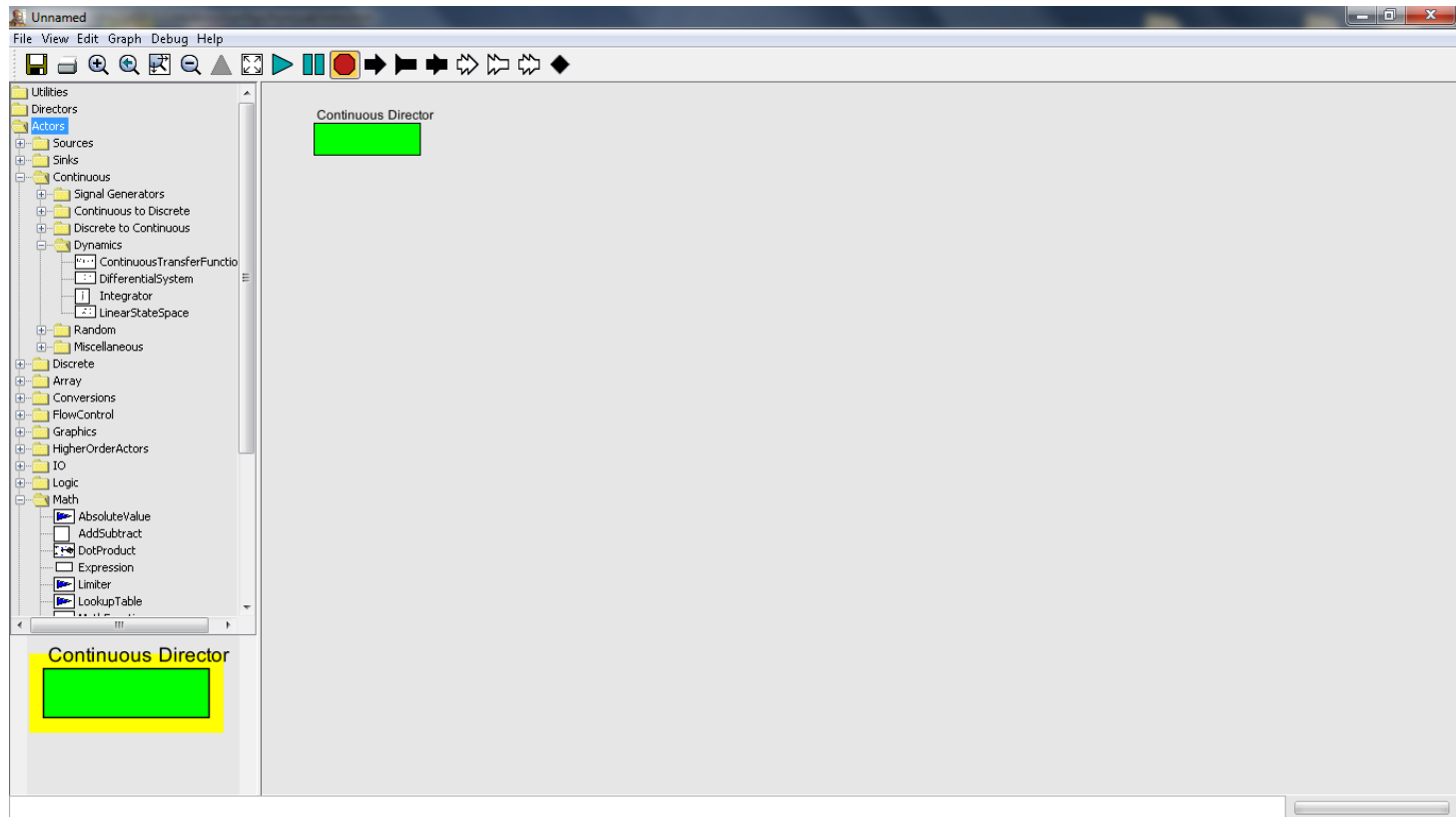
Tool: HyVisual in Ptolemy II

- HyVisual [3]

  - Hybrid System Visual Modeler

  - Block-diagram editor and simulator for continuous-time dynamical systems and hybrid systems

  - Supporting construction of hierarchical hybrid systems

    - Using block-diagram representation ordinary differential equations (ODEs) to define continuous dynamics

    - Using finite state machine to define discrete events

    - Allowing mix of continuous-time signals with events that are discrete in time

[3] UC Berkeley EECS Dept. "HyVisual – Hybrid System Visual Modeler." Internet: ptolemy.eecs.berkeley.edu/Hyvisual/index.htm, Oct. 7, 2005 [Apr. 20. 2012]
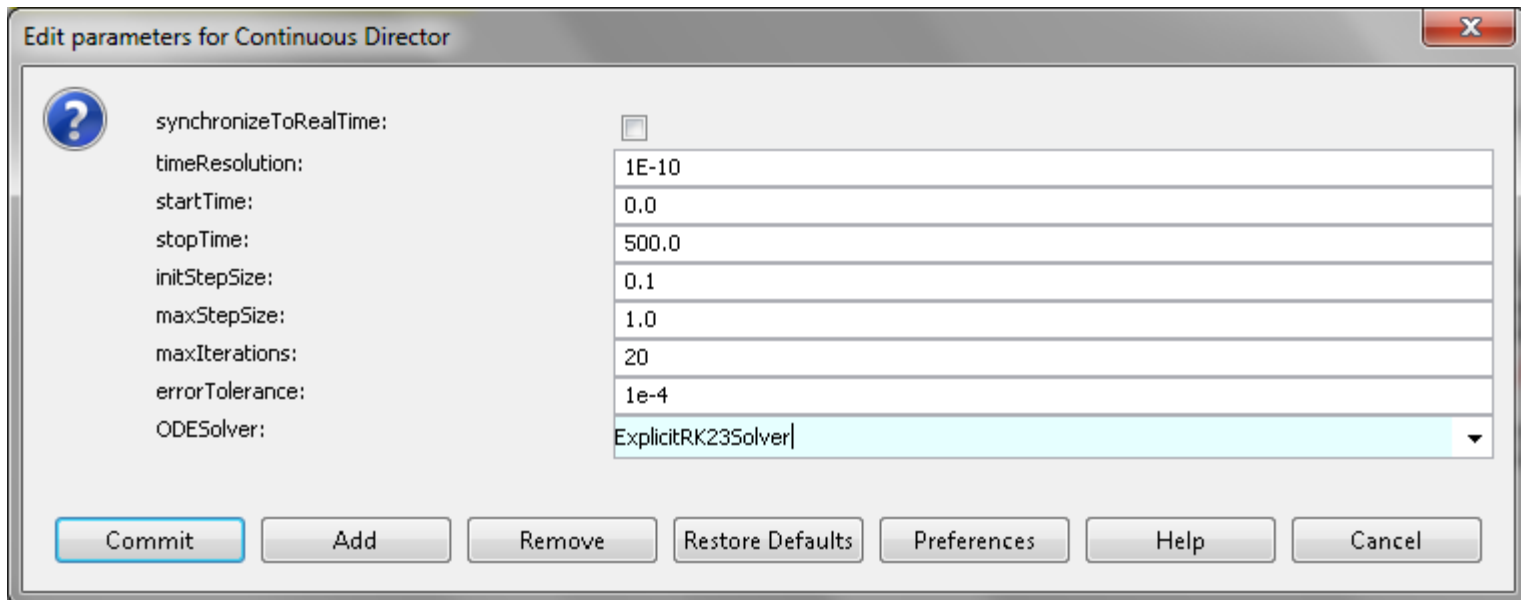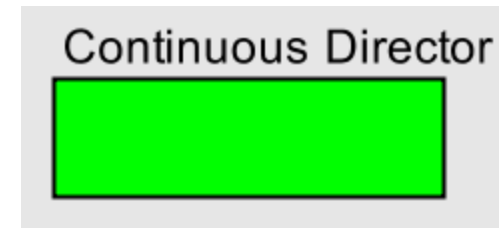
7

Tool: HyVisual in Ptolemy II
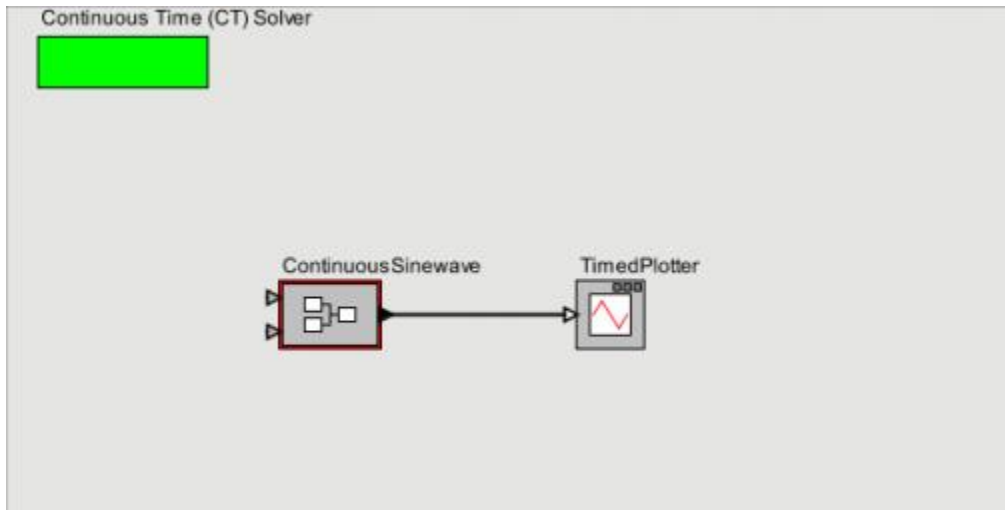
- HyVisual:

HyVisual: Directors

- Continuous Directors (Simulator)
  - Manage the continuous simulation
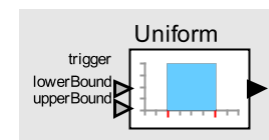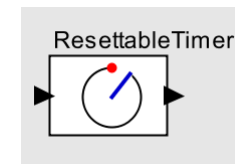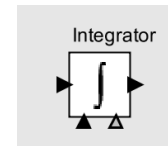 - Contains a sophisticated ODE solver

Continuous Director

Edit parameters for Continuous Director	
synchronizeToRealTime:	☐
timeResolution:	1E-10
startTime:	0.0
stopTime:	500.0
initStepSize:	0.1
maxStepSize:	1.0
maxIterations:	20
errorTolerance:	1e-4
ODESolver:	ExplicitRK23Solver

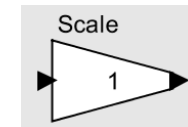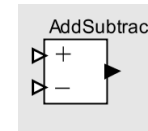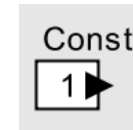Commit | Add | Remove | Restore Defaults | Preferences | Help | Cancel

HyVisual: Actors

- Actors: software components that can execute concurrently and communicate through messages sent via inter connected ports
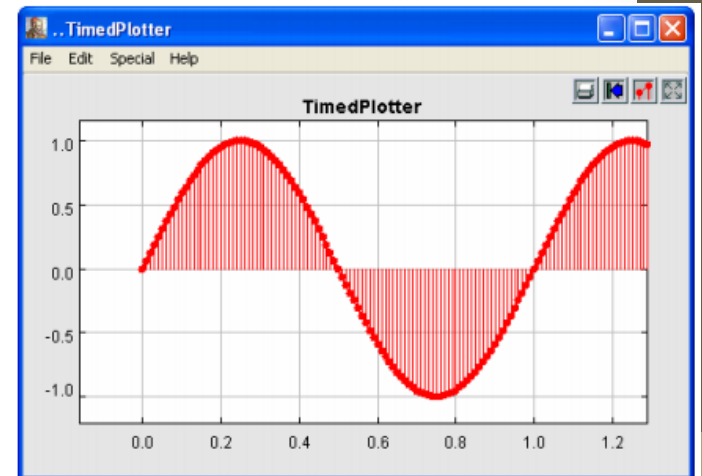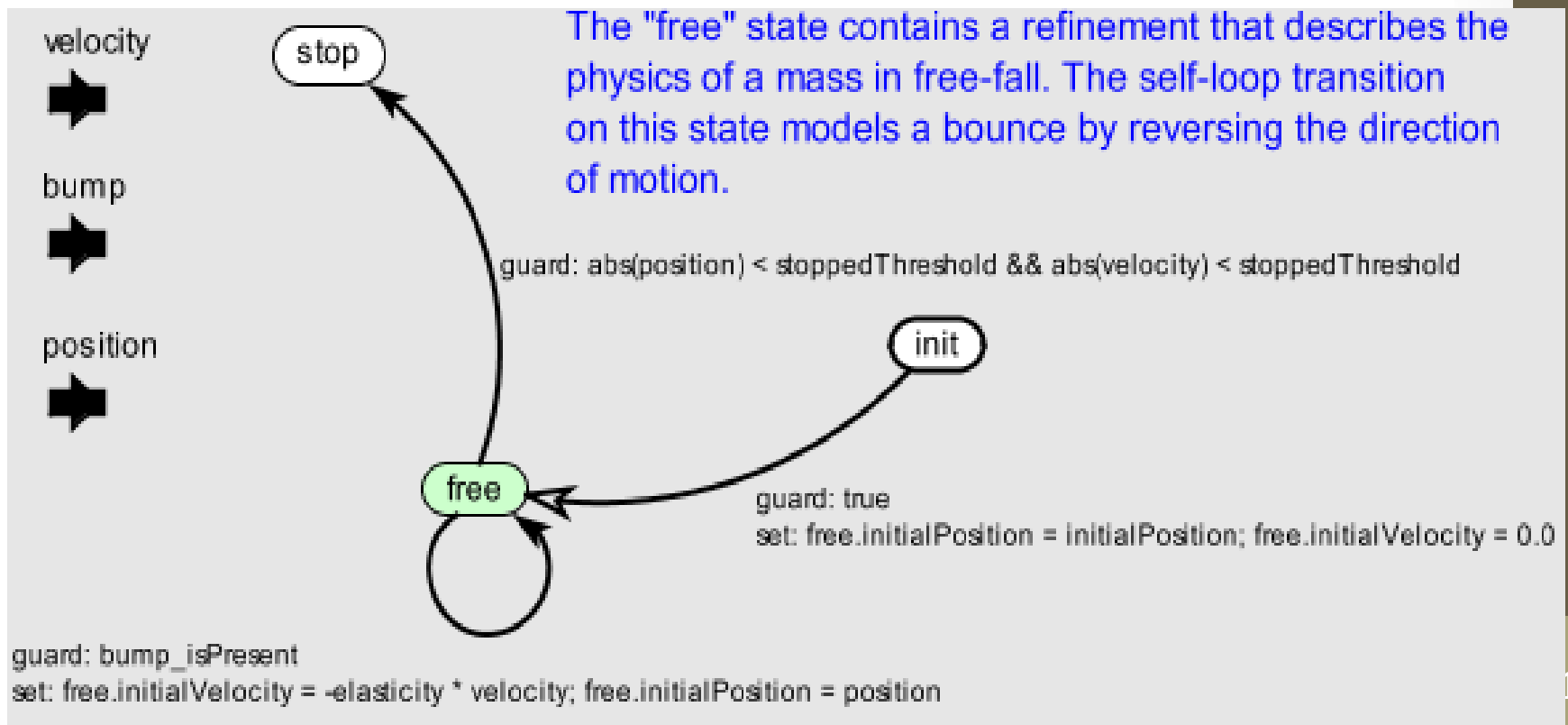
HyVisual: Actors

- Actors used in this project:
  - Modal Model Actor
  - Source Actor:
    - const actor
  - Mathematical Actors:
    - Add/Abstract actor
    - Scale actor
  - Continuous Actors:
    - Integrator actor
    - Resettable timer actor
  - Random Actors:
    - uniform random Actor
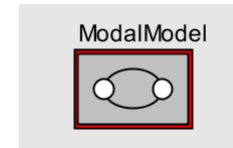
HyVisual: State Machine

- An example: bouncing ball

velocity

bump

position

The "free" state contains a refinement that describes the physics of a mass in free-fall. The self-loop transition on this state models a bounce by reversing the direction of motion.

guard: abs(position) < stoppedThreshold && abs(velocity) < stoppedThreshold

stop

init

free

guard: true
set: free.initialPosition = initialPosition; free.initialVelocity = 0.0

guard: bump_isPresent
set: free.initialVelocity = -elasticity * velocity; free.initialPosition = position

12

HyVisual: Hierarchical Models

Continuous Actors

Input/output ports

Continuous Model

(a Modal Model Actor)

State

State

Transactions

State

Continuous Model

13
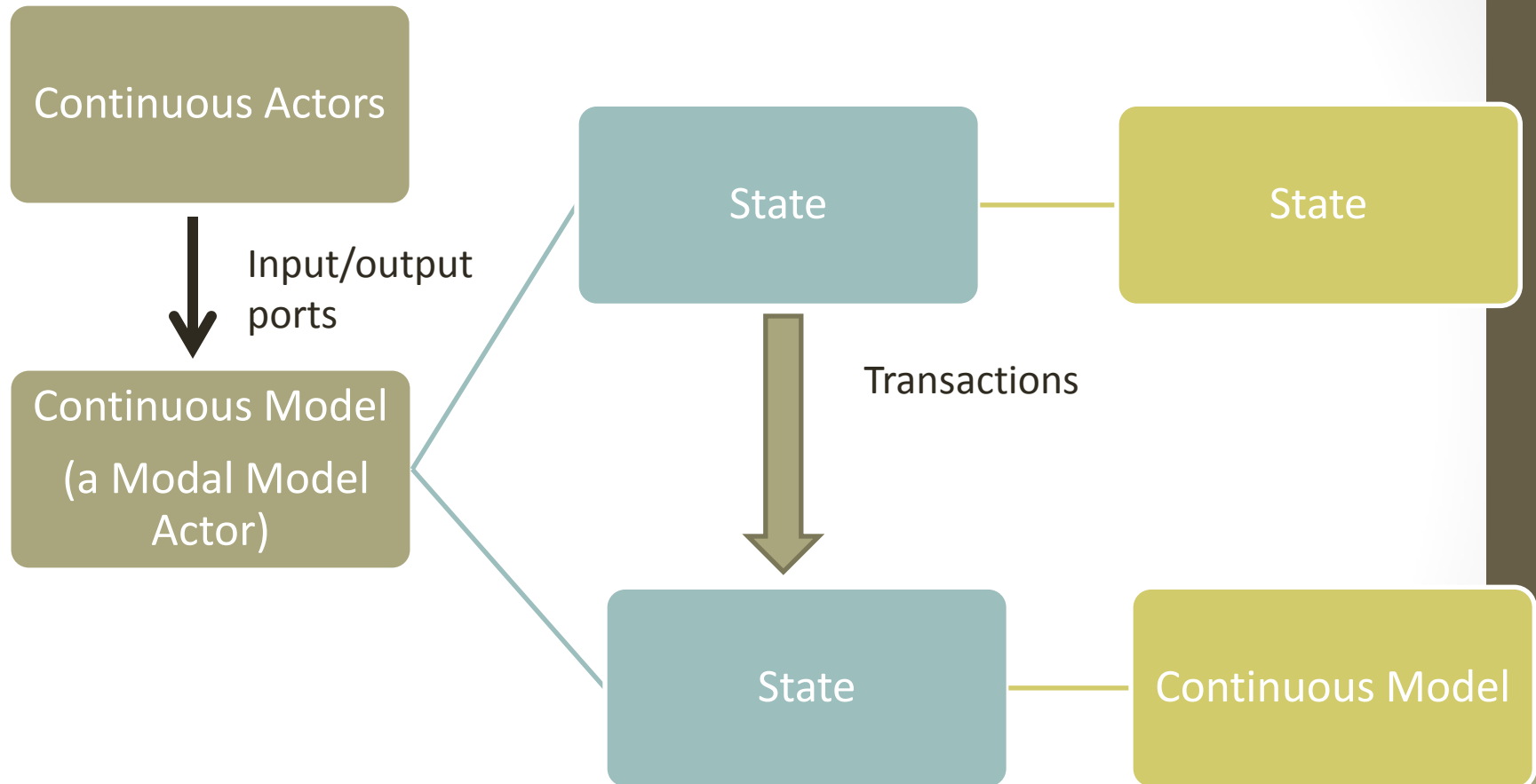
Project: Simulation of A Train System
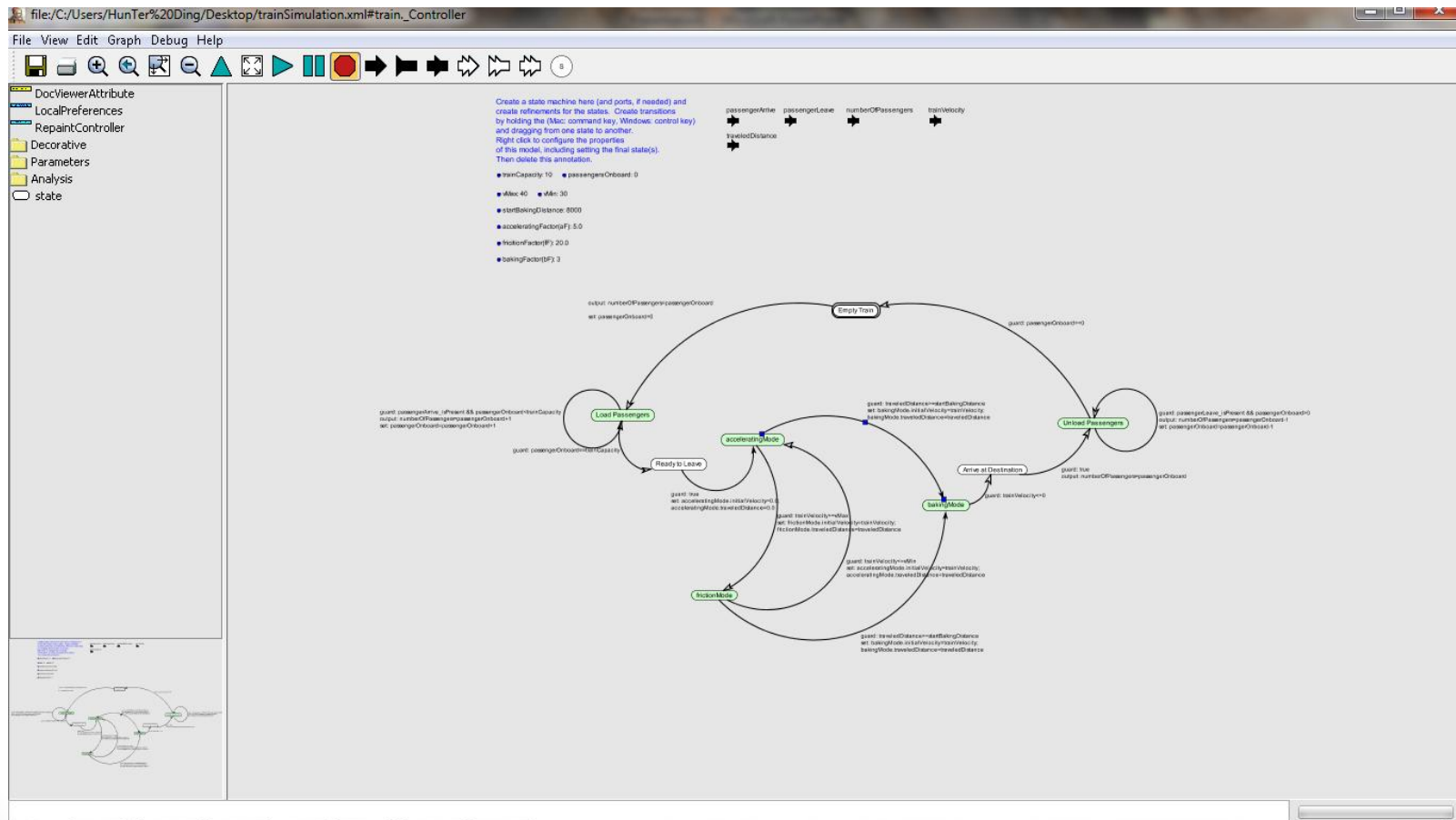

- Discrete Event:
  - loading and unloading passengers
  - Change between modes
- Continuous dynamics:
  - Train accelerating and decelerating

14

Project: Simulation of A Train System

- Empty train (discrete):
  - Load passengers, until the train is full
- Full train (continuous & discrete):
  - Leave the station, start accelerating
  - Velocity control: bang-bang control
  - Get close to the destination, start baking until stop at the destination's station
- Arrive at the destination (discrete):
  - Unload passengers

More Details & Demo

Conclusion

- Build a hybrid system: the Train System using HyVisual
  - Based on block-diagram – straightforward, easily handle complicated hierarchical structures
- Hybrid Modelling :
  - Embrace discontinuities and discrete events along with the usual piecewise continuous signals
  - Provide modelling semantics and results that are not only predictable, but easily understandable

Hybrid and Multi-Formalism Modelling

Question?