Heterogeneous Modeling and Design

- Edward A. Lee (PI) -

Staff
Jennifer Basler
Christopher Hylands
Mary P. Stewart

Postdoctoral Researchers
H. John Reekie

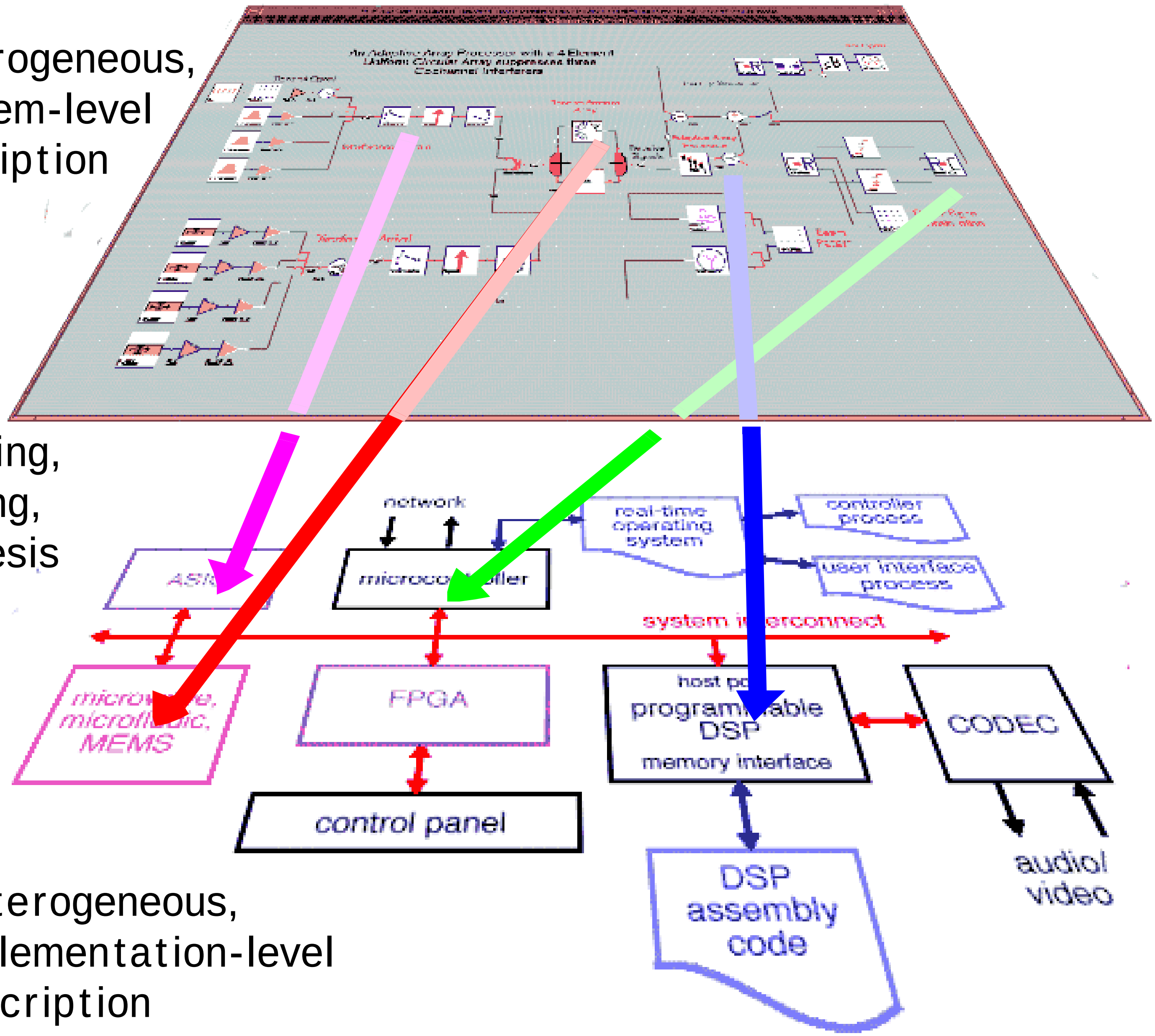
Students
Albert Chen
John Davis, II
Mudit Goel
Bilung Lee
Jie Liu
Xiaojun Liu
Stephen Neuendorffer
Neil Smyth
Michael C. Williamson
William Wu
Yuhong Xiong

Agenda

- 9:00 Introduction and overview of the project
- 9:30 Hierarchical FSMs as a model of discrete control
- 9:45 Ptolemy II architectural overview
- 10:15 break
- 10:30 process networks for concurrent systems modeling
- 10:45 continuous-time modeling in Ptolemy II
- 11:00 Tycho-based user interface toolkit
- 11:45 Wrapup and future plans
- 12:00 Adjourn

Heterogeneous,
problem-level
description

Modeling,
mapping,
synthesis

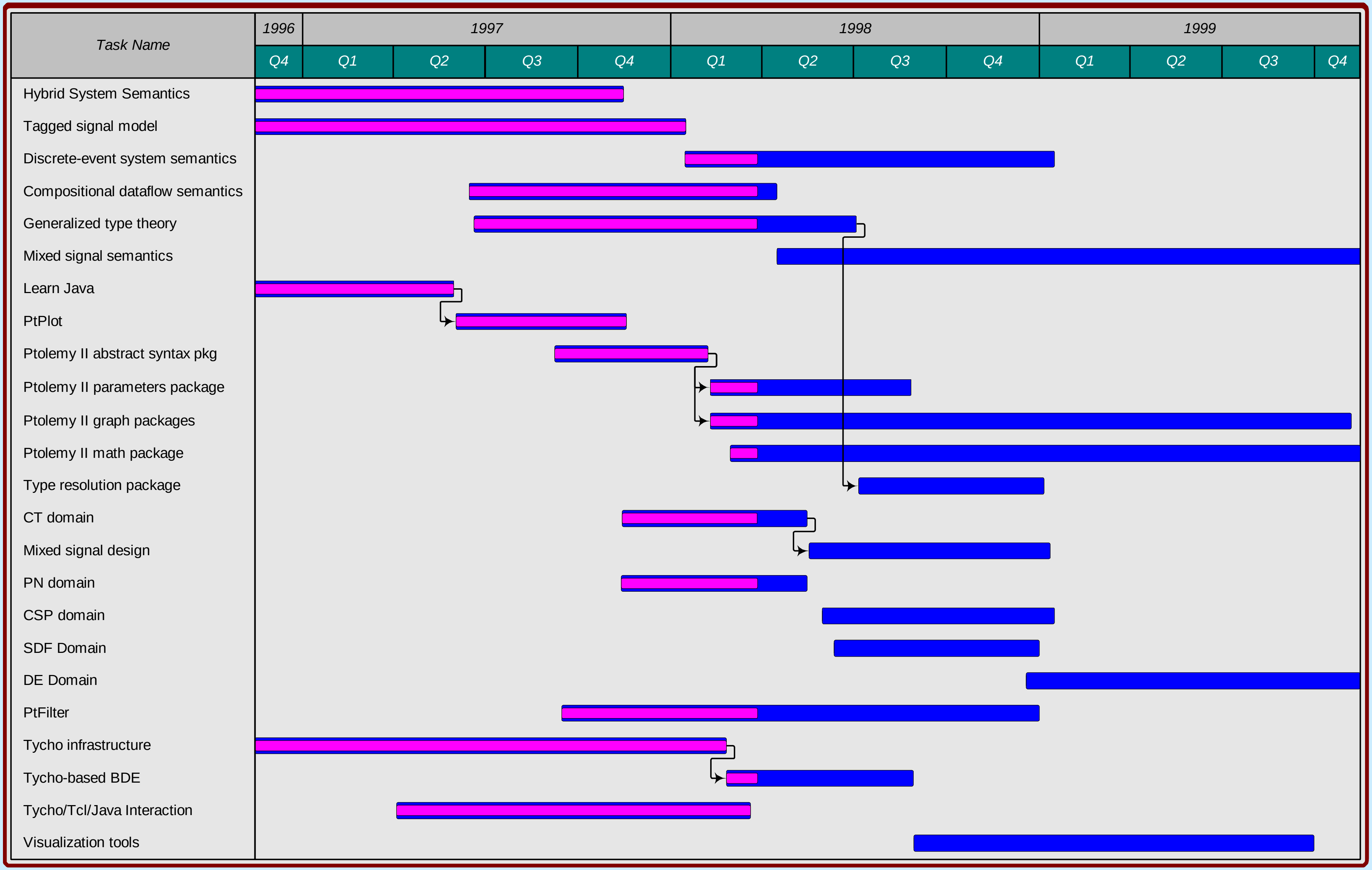


Heterogeneous,
implementation-level
description

Approach

- Theory and techniques for mixing diverse models of computation, e.g. mixed signal, hybrid systems, discrete and continuous events.
- Software architecture for modular, distributed, and heterogeneous design, modeling and visualization tools.
- Theory and software for domain-specific modeling of composite concurrent systems.
- Use of programming language concepts (semantics, type theories, and concurrency theories) for modeling and design of composite systems.
- Emphasis on visual representations.

Schedule

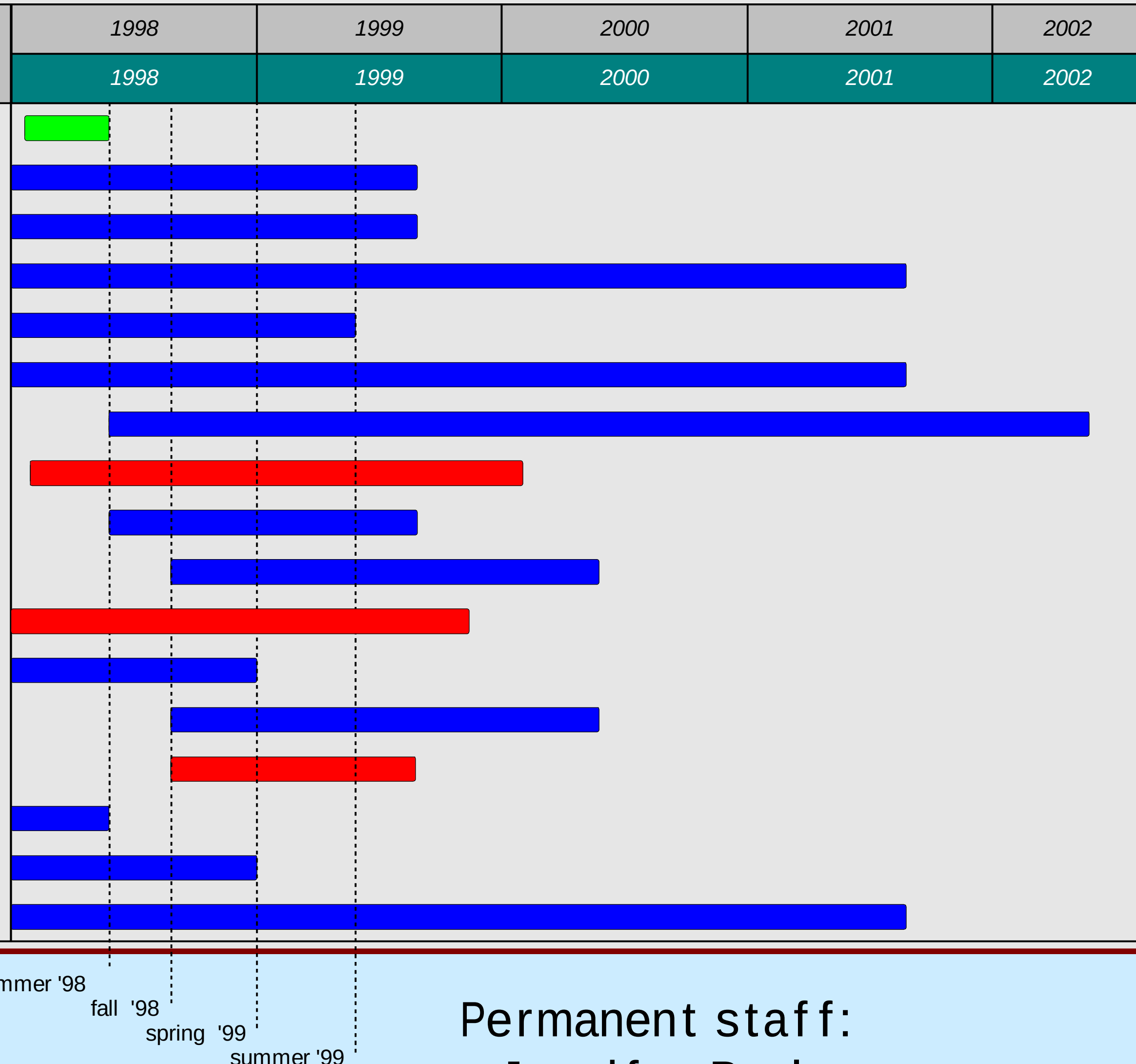


Task Name
1996
1997
1998
1999
Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
Hybrid System Semantics
Tagged signal model
Discrete-event system semantics
Compositional dataflow semantics
Generalized type theory
Mixed signal semantics
Learn Java
PtPlot
Ptolemy II abstract syntax pkg
Ptolemy II parameters package
Ptolemy II graph packages
Ptolemy II math package
Type resolution package
CT domain
Mixed signal design
PN domain
CSP domain
SDF Domain
DE Domain
PtFilter
Tycho infrastructure
Tycho-based BDE
Tycho/Tcl/Java Interaction
Visualization tools

Staff schedule:

Students and Visiting Scholars in the Ptolemy Project

ID	Task Name	Start Date	End Date
1	Albert Chen	1/20/98	5/25/98
2	John Davis	8/15/96	8/27/99
3	Ron Galicia	9/15/94	8/27/99
4	Mudit Goel	1/21/97	8/24/01
5	Bilung Lee	8/15/93	5/27/99
6	Jie Liu	4/20/97	8/24/01
7	Xiaojun Liu	5/26/98	5/24/02
8	James Lundblad	1/28/98	1/31/00
9	Lukito Muliadi	5/26/98	8/27/99
10	Steve Neuendorffer	8/26/98	5/24/00
11	John Reekie	8/15/95	11/12/99
12	Neil Smyth	5/25/97	12/31/98
13	Jeff Tsay	8/26/98	5/24/00
14	Nils Wemhöner	8/26/98	8/24/99
15	Mike Williamson	1/1/92	5/25/98
16	William Wu	8/15/97	12/31/98
17	Yuhong Xiong	1/1/97	8/24/01



→ Having other support

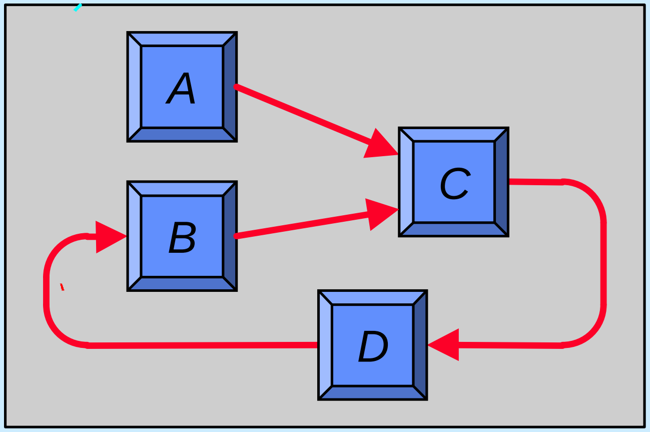
- graduate students
- undergraduate students
- visiting scholars
- possible graduate students

Permanent staff:

- Jennifer Basler
- Christopher Hylands
- Edward A. Lee
- Mary Stewart

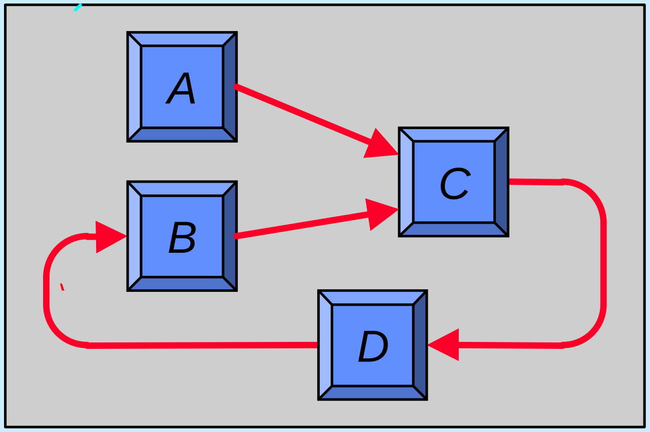
Models of Computation

- Analog computers (differential equations)
- Discrete time (difference equations)
- Discrete-event systems
- Synchronous-reactive systems
- Sequential processes with rendezvous
- Process networks
- Dataflow
- Finite state machines



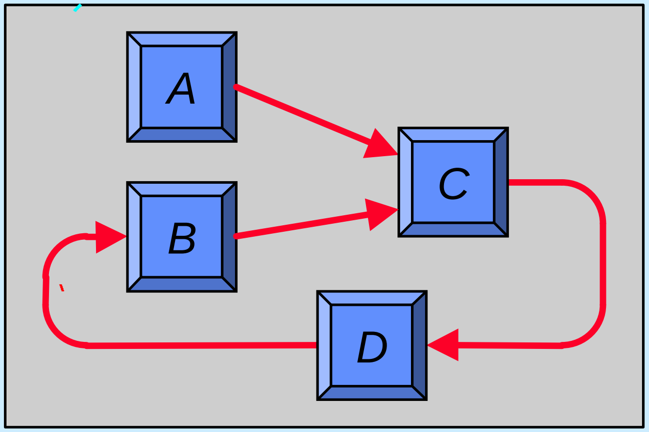
Shared Properties

- Strengths and weaknesses (no silver bullet)
- Domain-specific
- Modular
- Amenable to visual syntaxes
- Hierarchical
- Concurrent (except FSMs)
- Abstract



Issues Being Addressed

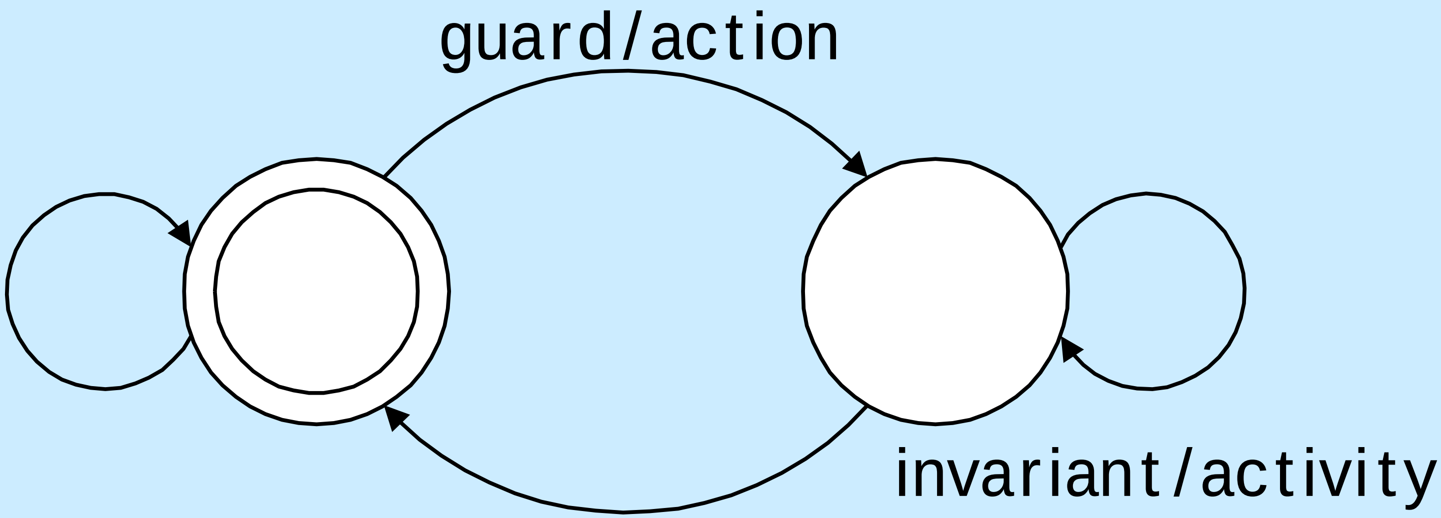
- Semantics (what is a behavior)
- Determinacy (how many behaviors are there)
- Simulation (finding a behavior)
- Analysis (finding properties of behaviors)
- Compositionality (encapsulating subsystems)
- Synthesis (translation to implementation)
- Design (choosing implementations)
- Heterogeneity

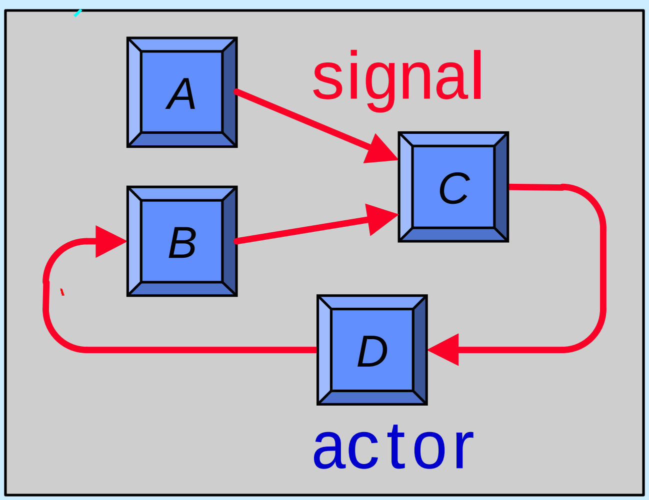


Examples Requiring Heterogeneity

- MEMS device with a discrete controller (differential equations plus discrete-event models)
- Modal models, with regimes of operation (differential equations plus finite-state machines)
- Mixed signal systems (differential equations plus discrete-time and/or discrete-event systems)
- Hardware/software systems (differential equations, discrete-events, discrete-time, finite-state machines, dataflow, rendezvous, process networks, ...)

State Machines & Block Diagrams





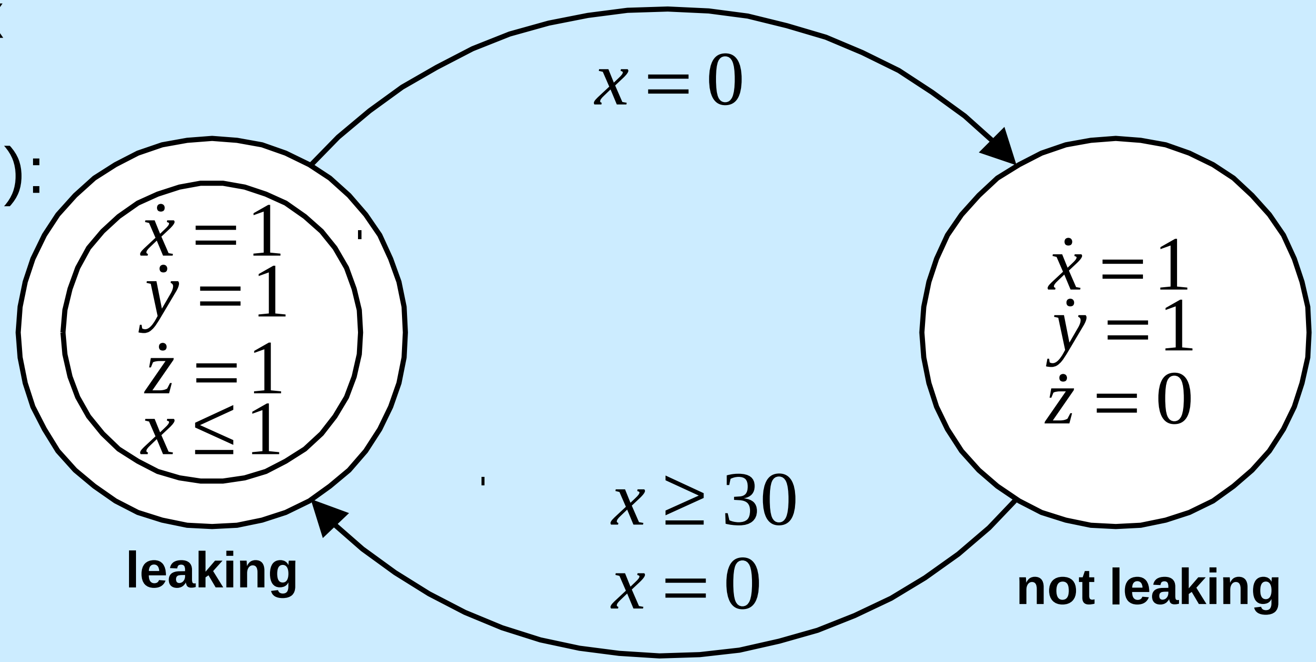
Sequential

Concurrent

Hybrid Systems

A discrete program combined with an analog system.
A combination of automata and analog computers.

Traditional syntax (classic example: leaking gas burner):

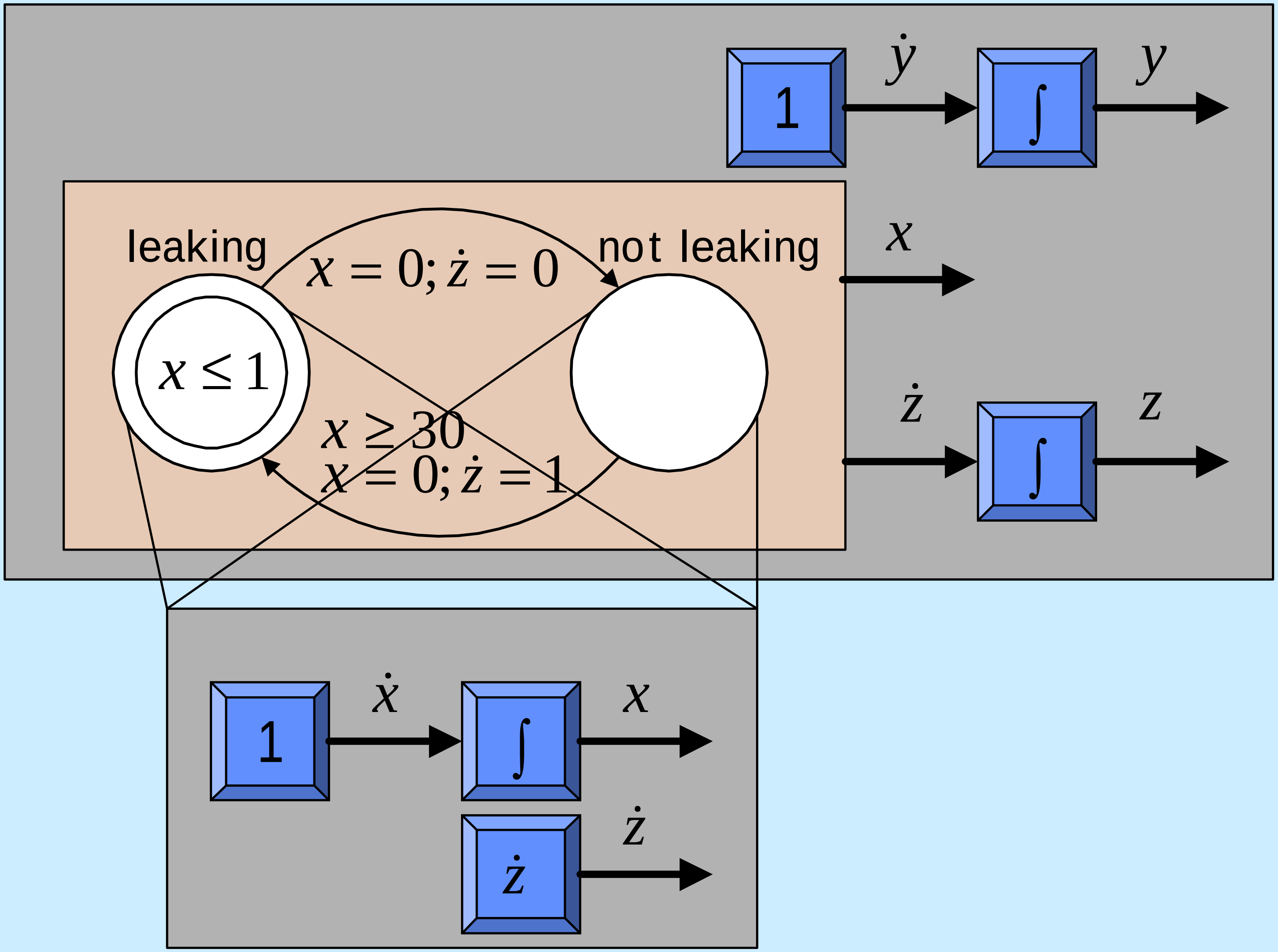


Here, the differential equations hardly look like a concurrency model, but in fact they are.

Alternative View of Hybrid Systems

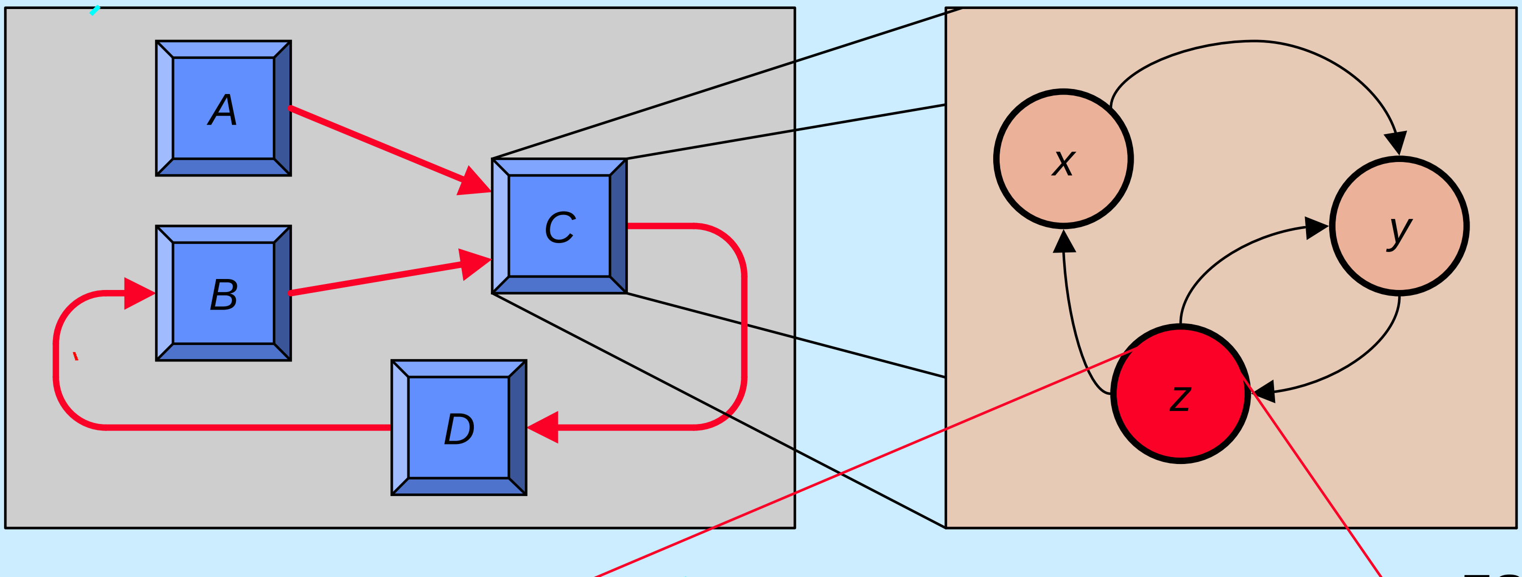
Analog computers hierarchically combined with automata.

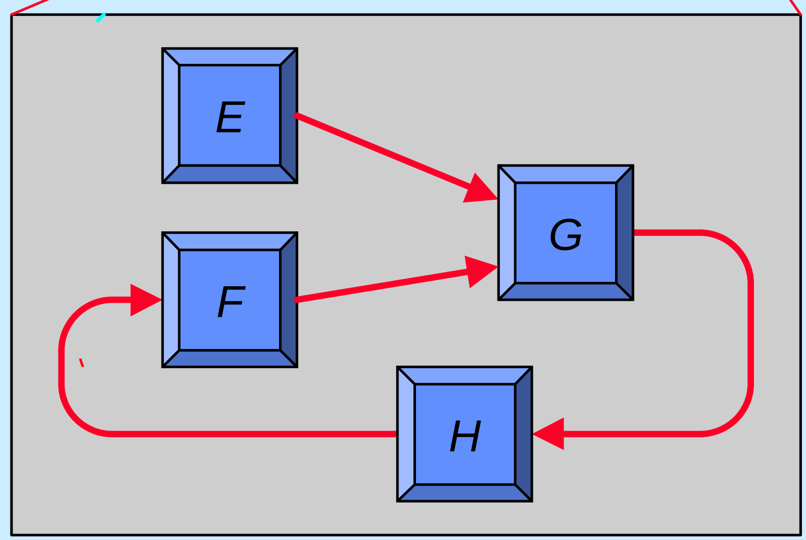
Classic example (leaking gas burner):



Generalized Hybrid Systems

Choice of domain here determines concurrent semantics

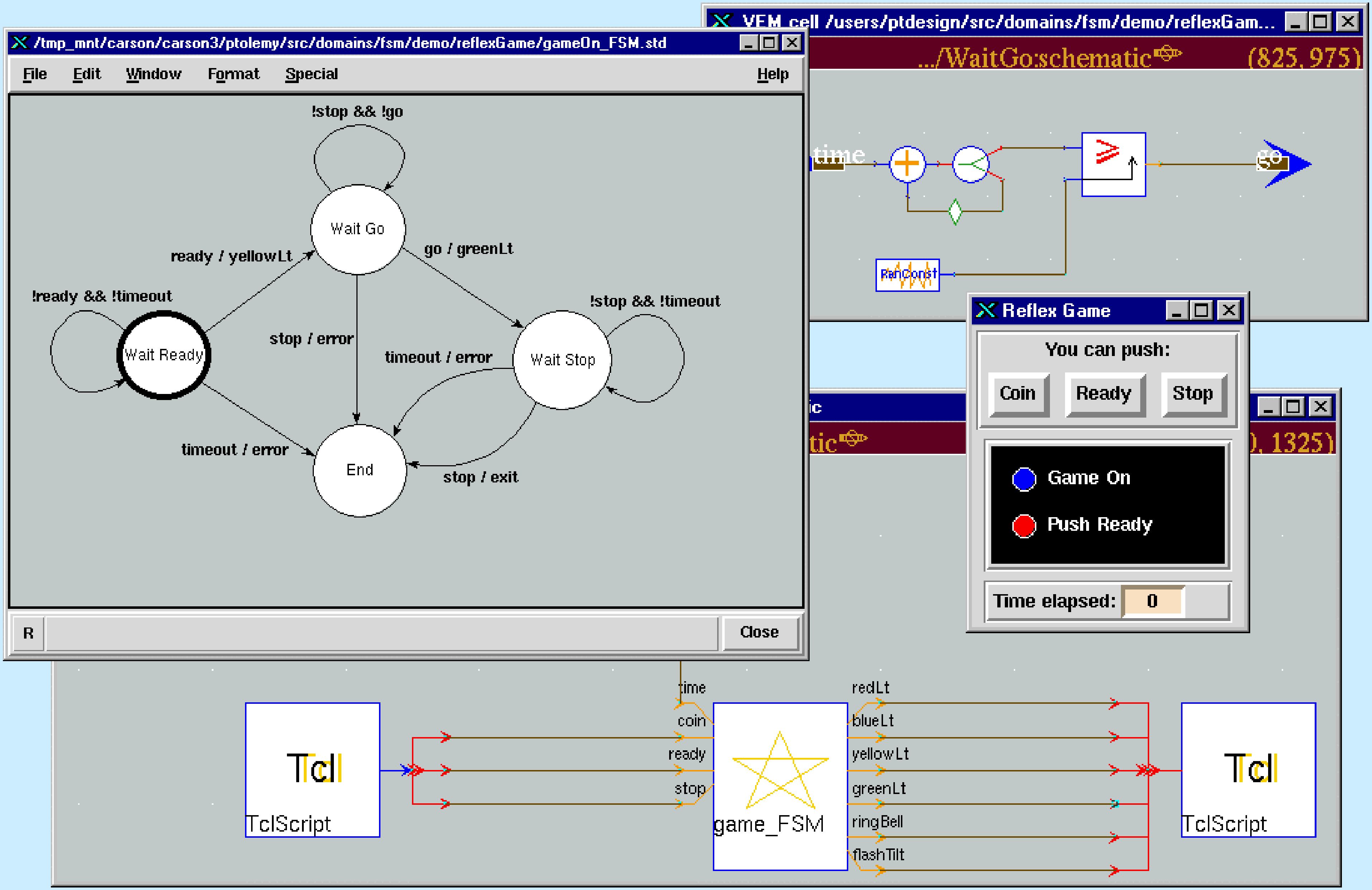




FSM for control

We have formalized the semantics of FSMs combined with discrete-event, dataflow, and synchronous-reactive models.

Ptolemy 0.7 Prototype

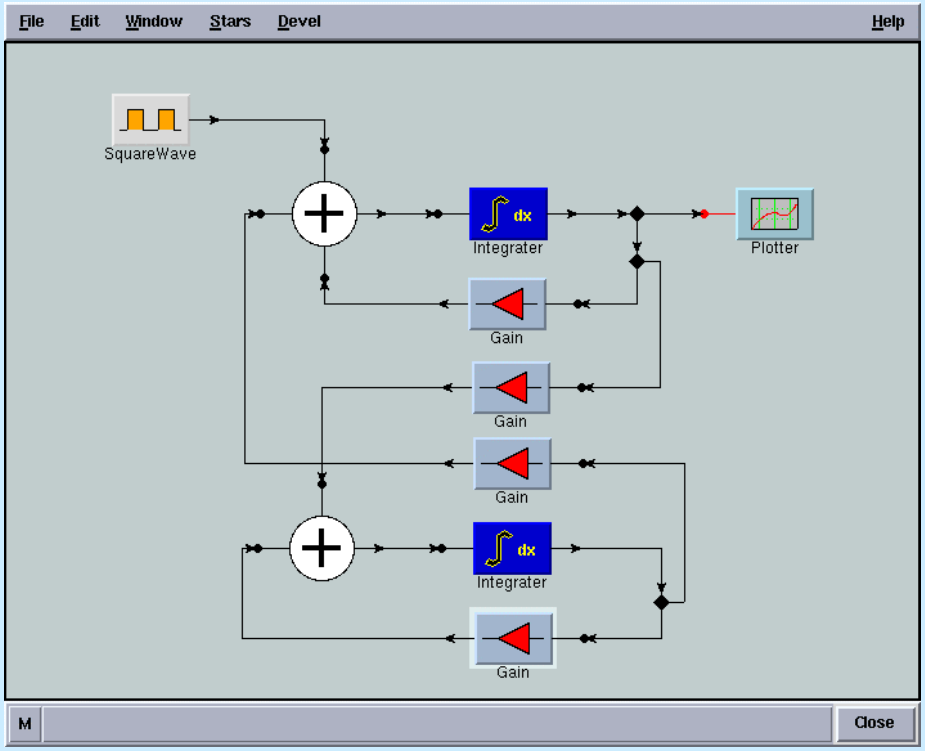


Ptolemy II Hybrid Systems

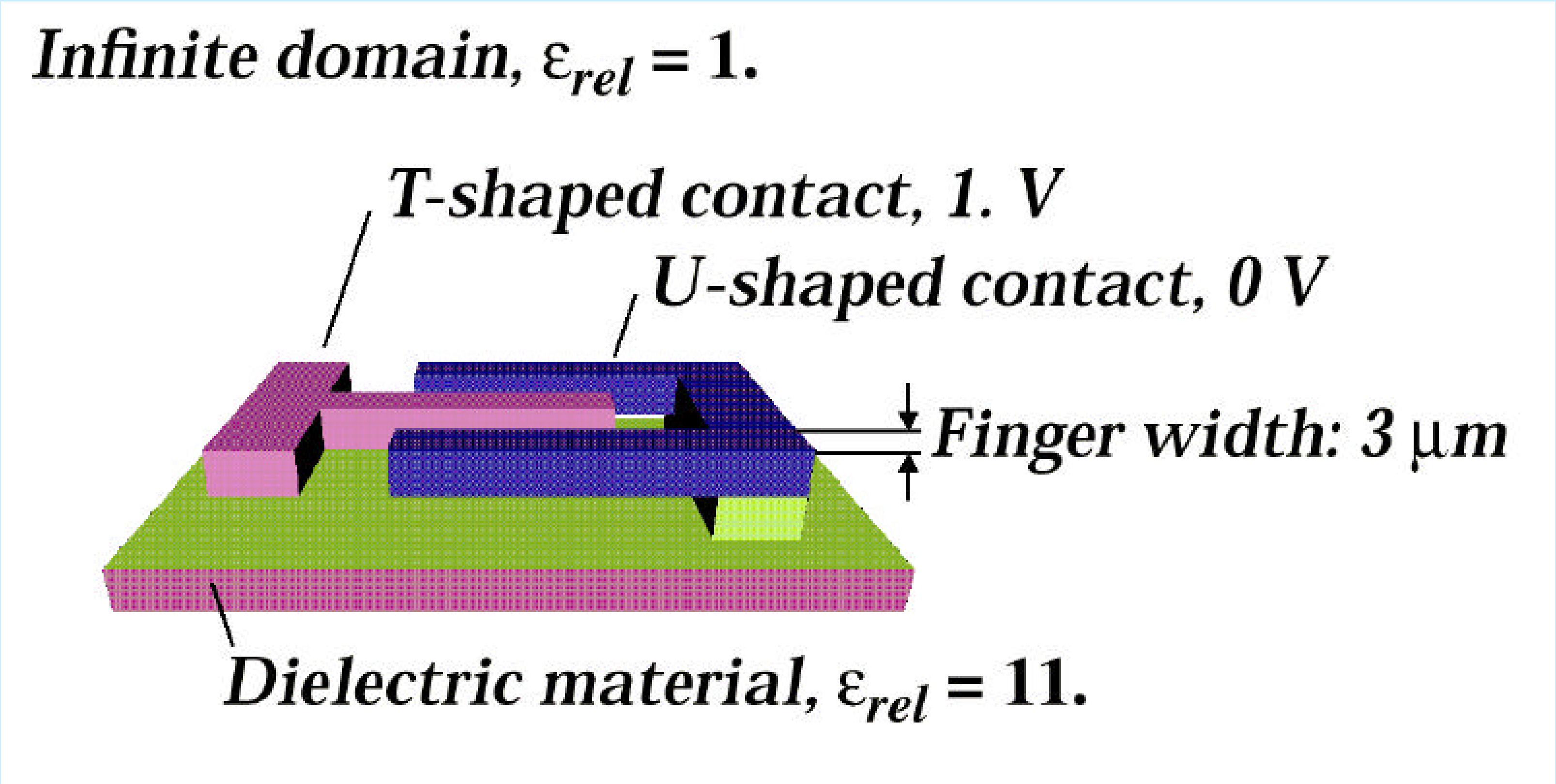
- CT domain: ODE solver in continuous time.
- Generalized wormhole mechanism.
- Emphasis on specification and simulation (not verification).
- Hierarchical visual specifications.
- Interactive, animated simulations.

Continuous-Time Domain in Ptolemy II

- Will support a variety of numerical methods for solving ODEs
  - Time marching, Waveform relaxation, Frequency domain methods, Monte-Carlo methods
- Will mix with:
  - Dataflow
  - DE
  - FSM
- Applications:
  - Mixed signal design
  - MEMS
  - Hybrid systems

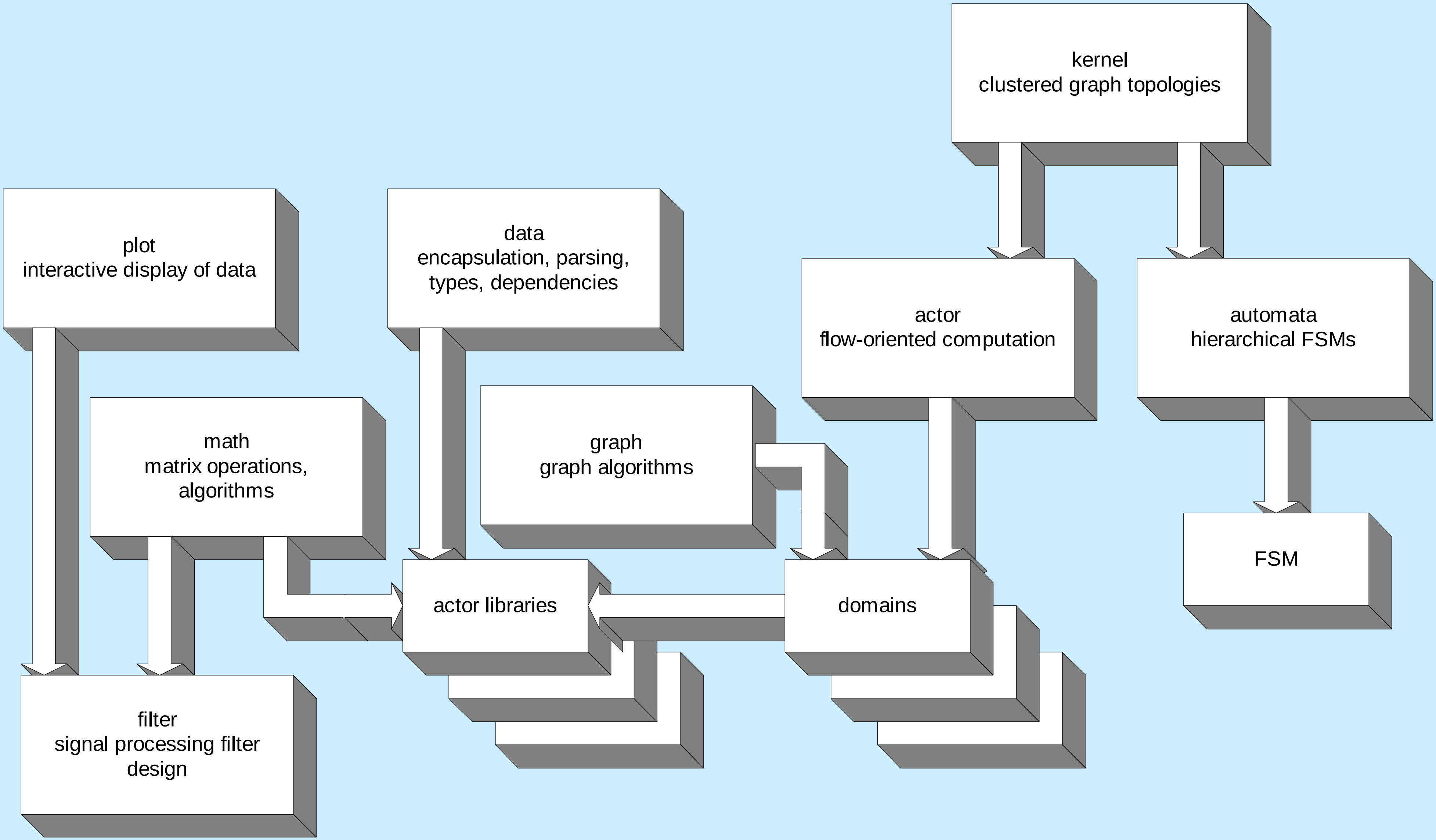


Macro Modeling from Coyote



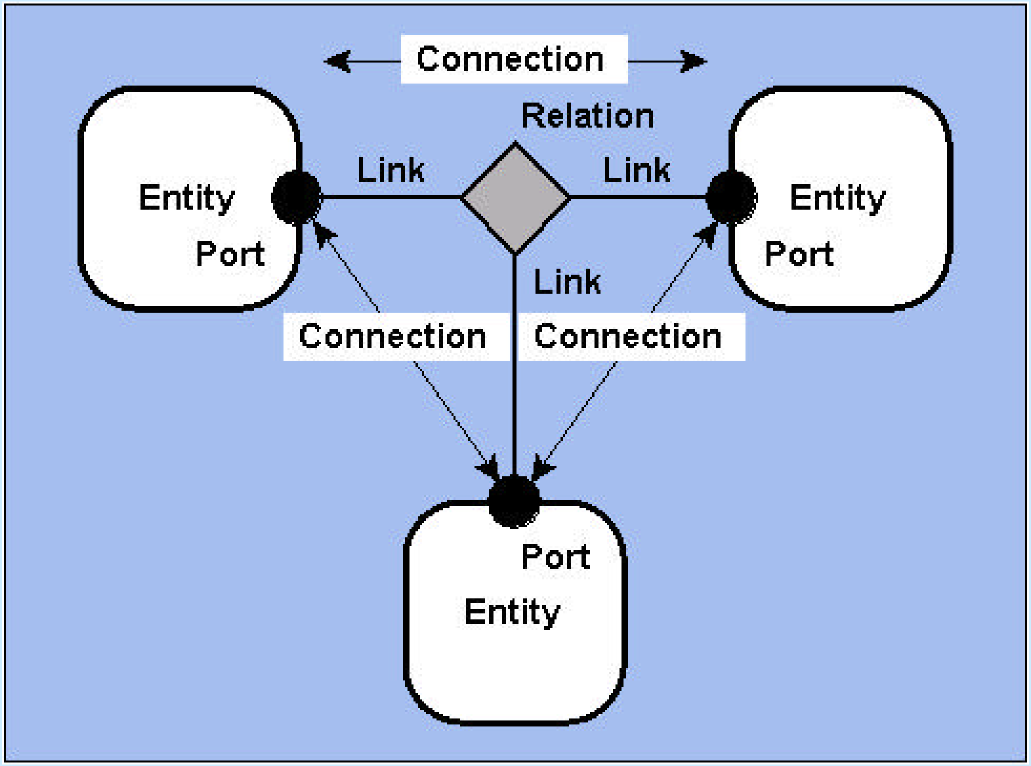
We have incorporated a macro-model of a mechanical comb constructed by Coyote using their electrostatic BEM method.

Package Structure of Ptolemy II



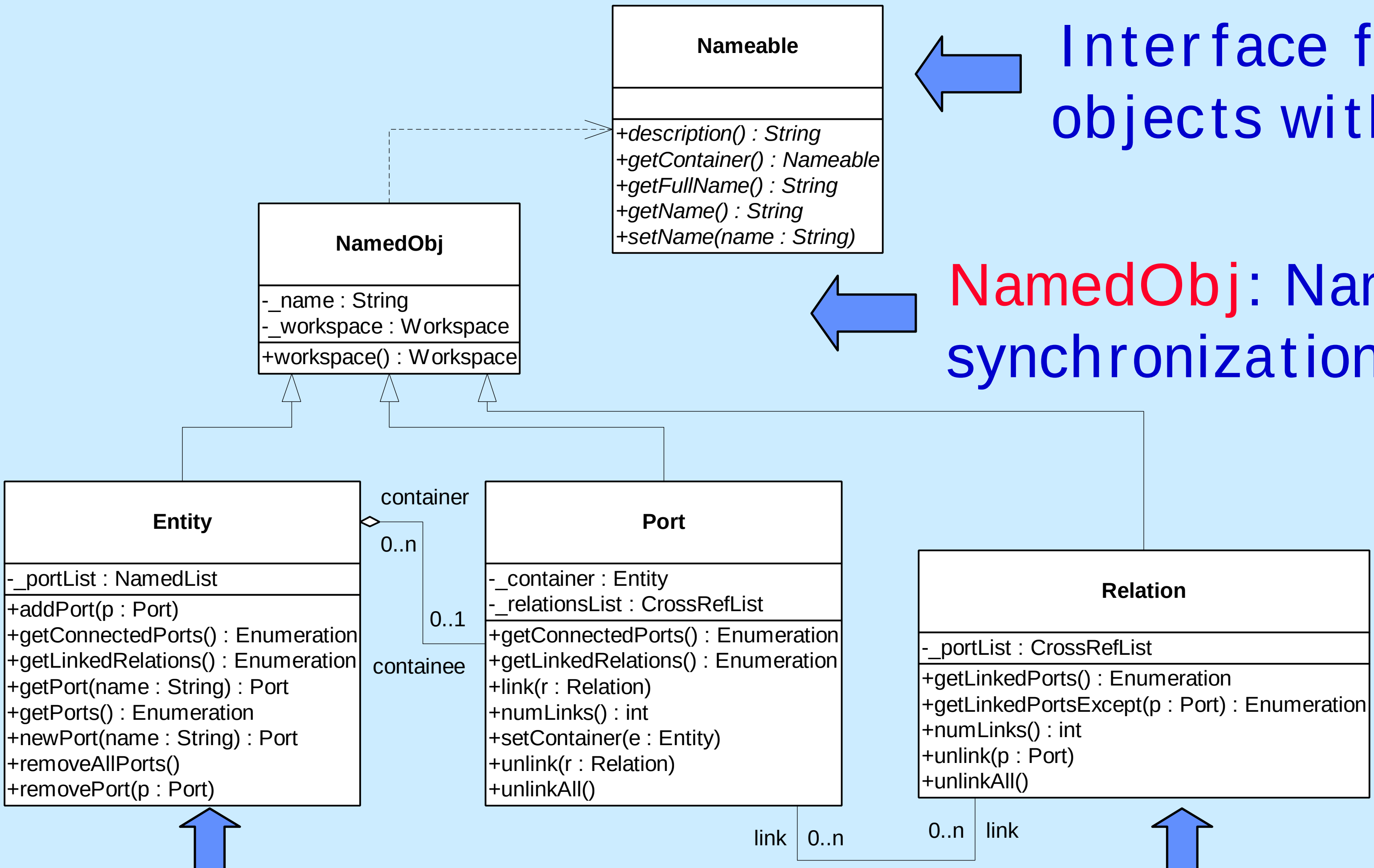
Ptolemy II Abstract Syntax

- Entity/Relation bipartite graphs



- Ports are named aggregations of links
- A topology is a linked collection of entities and relations

Flat Abstract Syntax Classes



Interface for objects with names

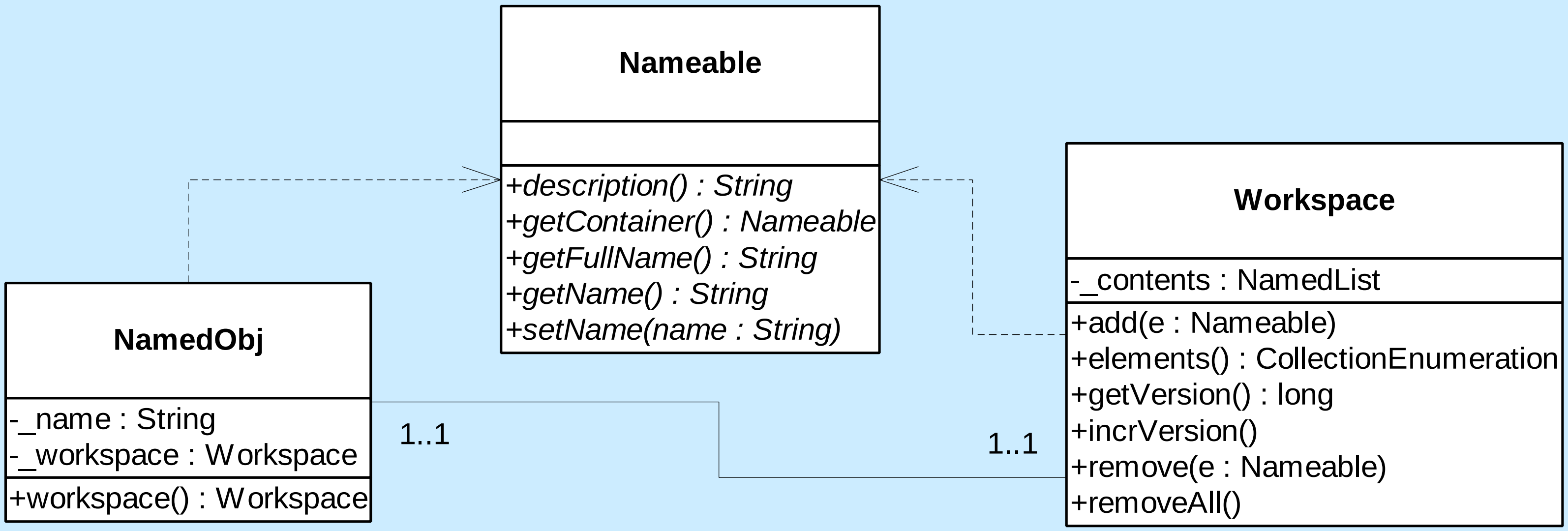
NamedObj: Naming and synchronization services

Entity: aggregation of Ports

Port: aggregation of links to Relations

Relation: aggregation of links to Ports

Synchronization Services

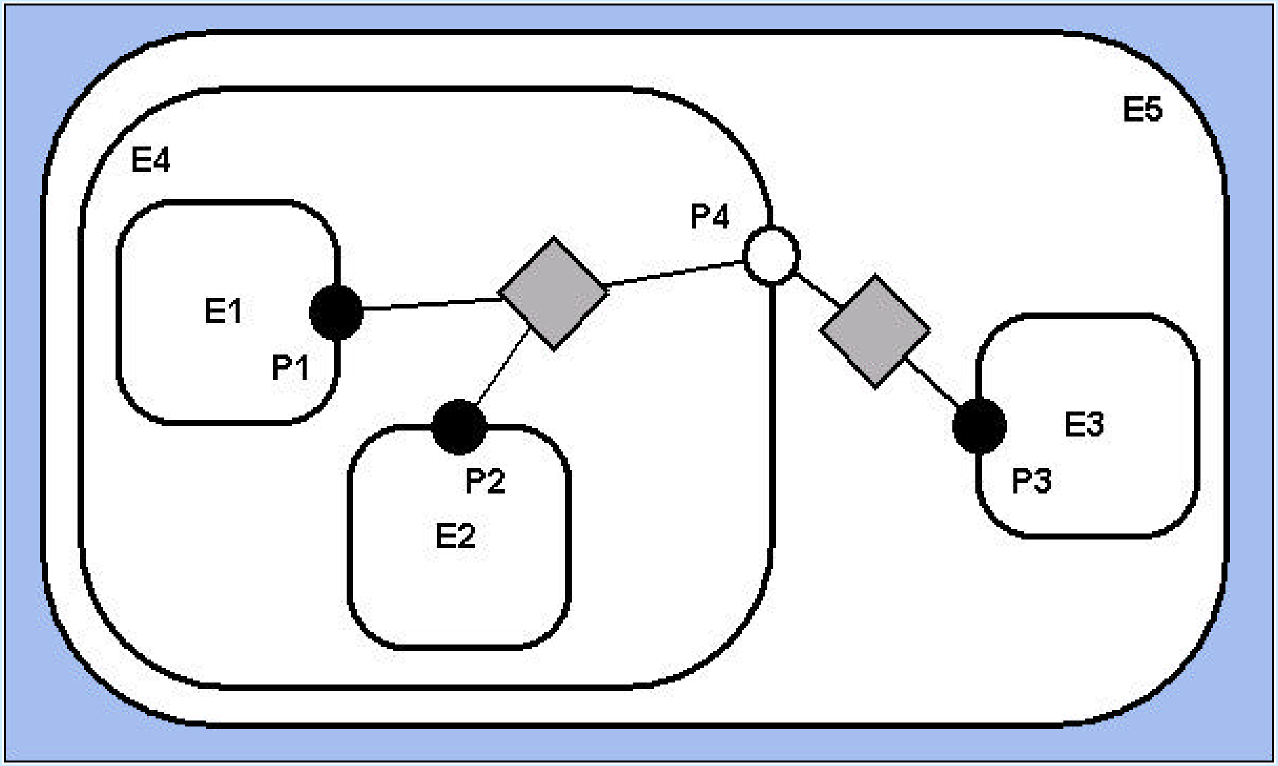


Every NamedObj has an immutable association with an instance of Workspace. A monitor on Workspace is used for thread synchronization, and the workspace tracks versions of a topology.

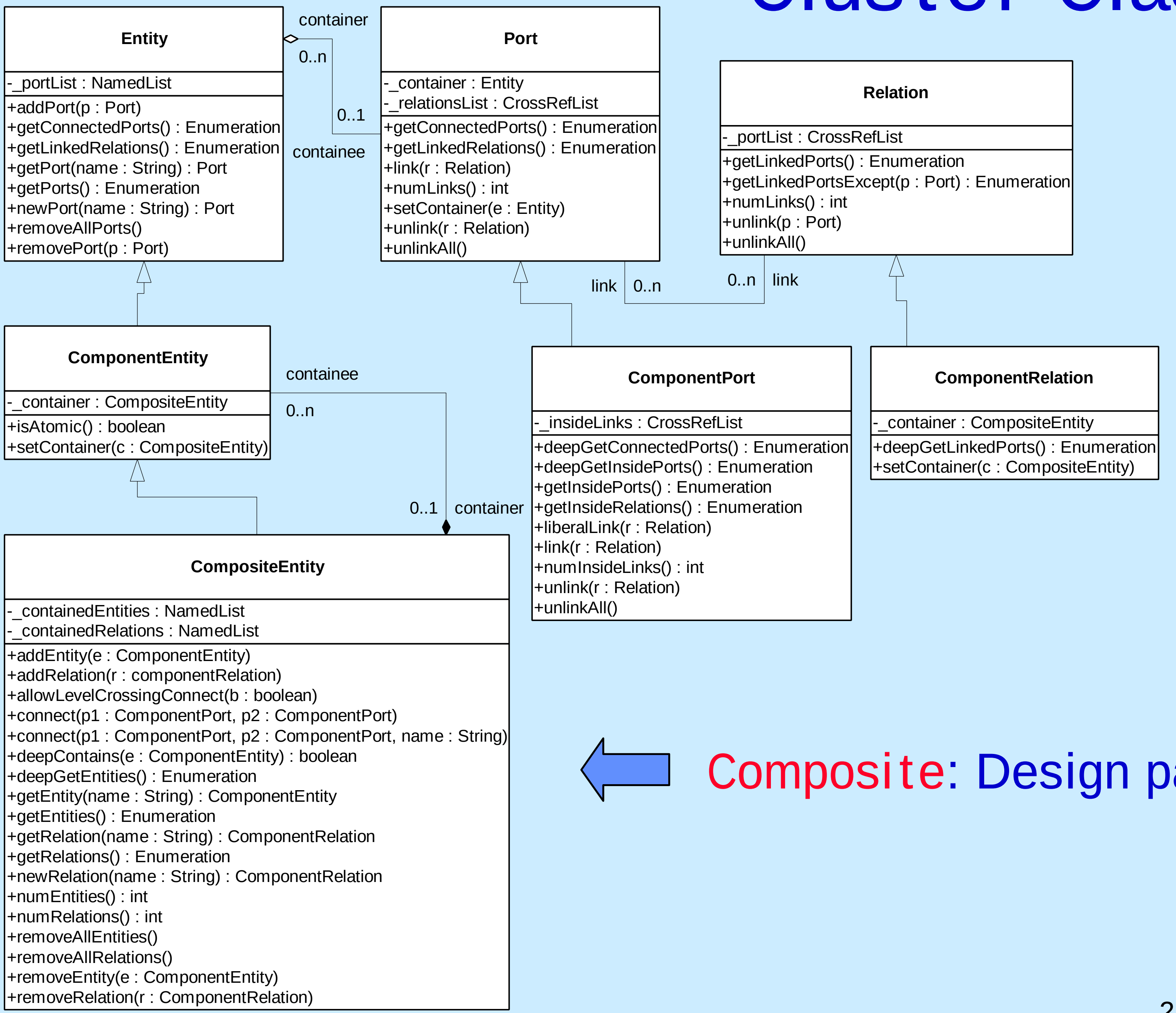
Clustered Graphs

- Transparent ports
- Transparent entities
- Managed containers

Every object has zero or one containers. If zero, then it is known to its workspace

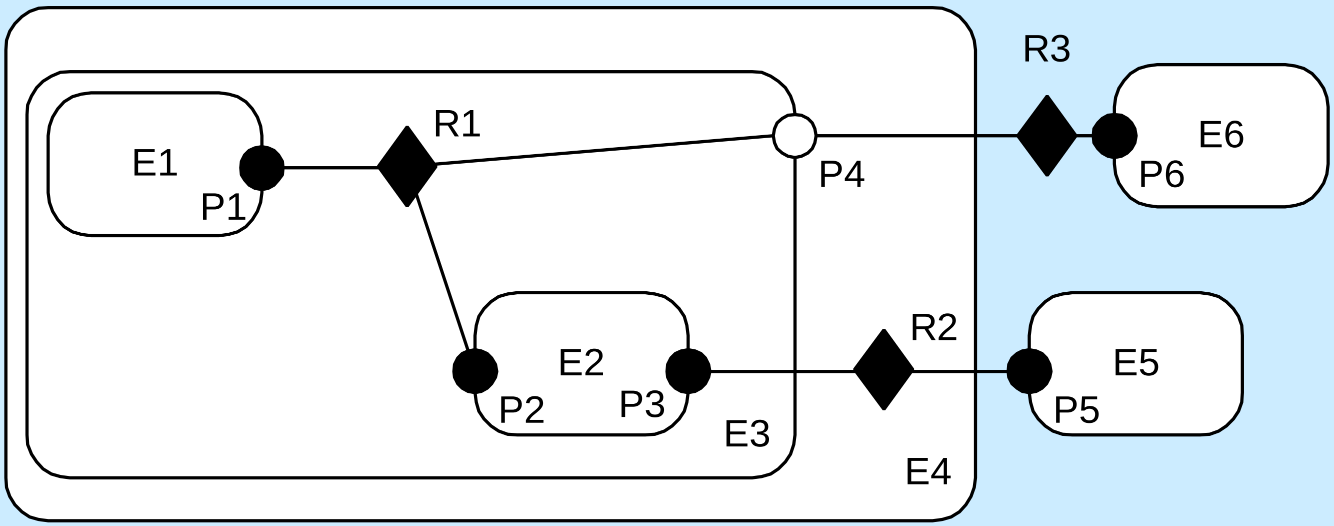


Cluster Classes



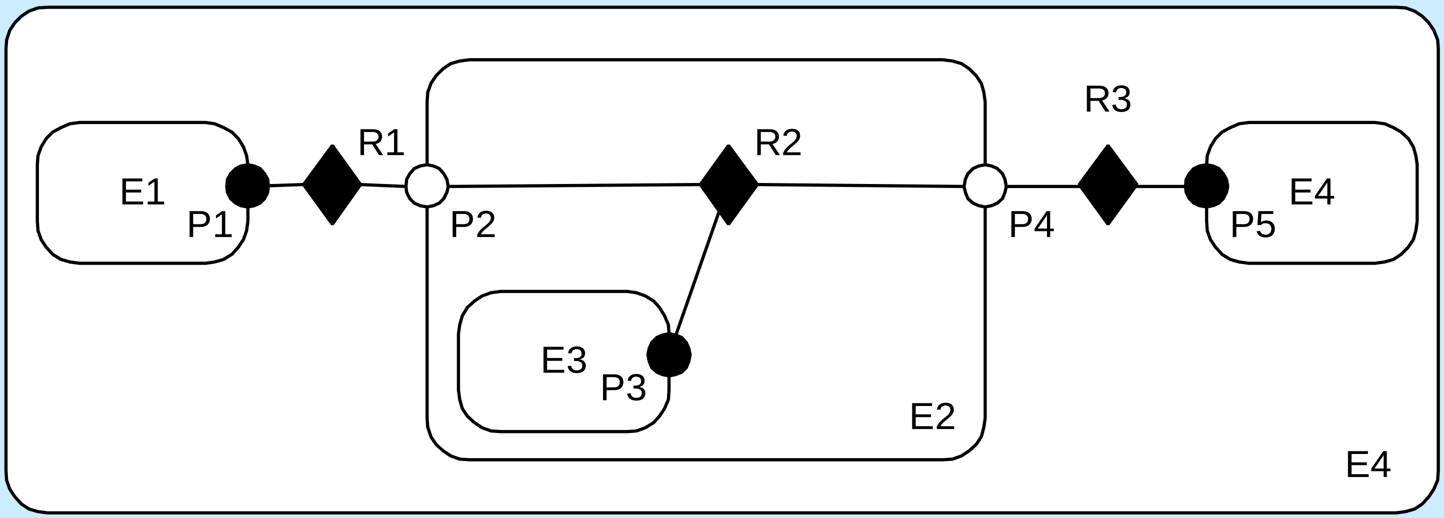
Composite: Design pattern

Level-Crossing Links



These are supported, but discouraged.

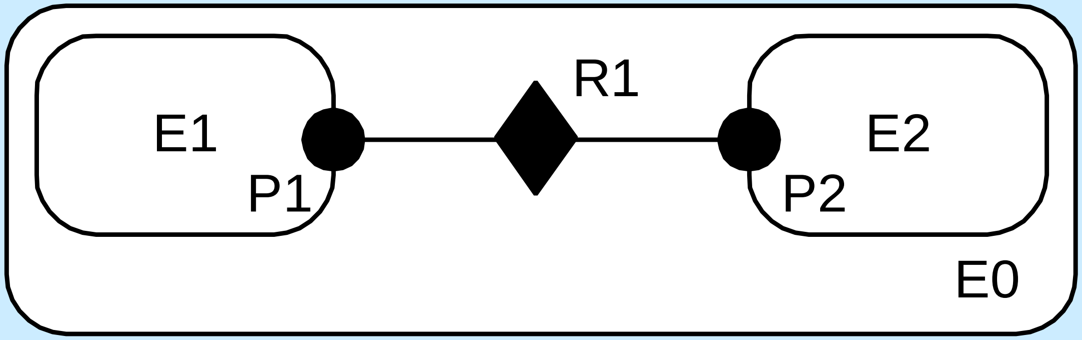
Transparent Entities



Modularity and compositionality require that relations be able to transparently span the hierarchy.

TclBlend Interface

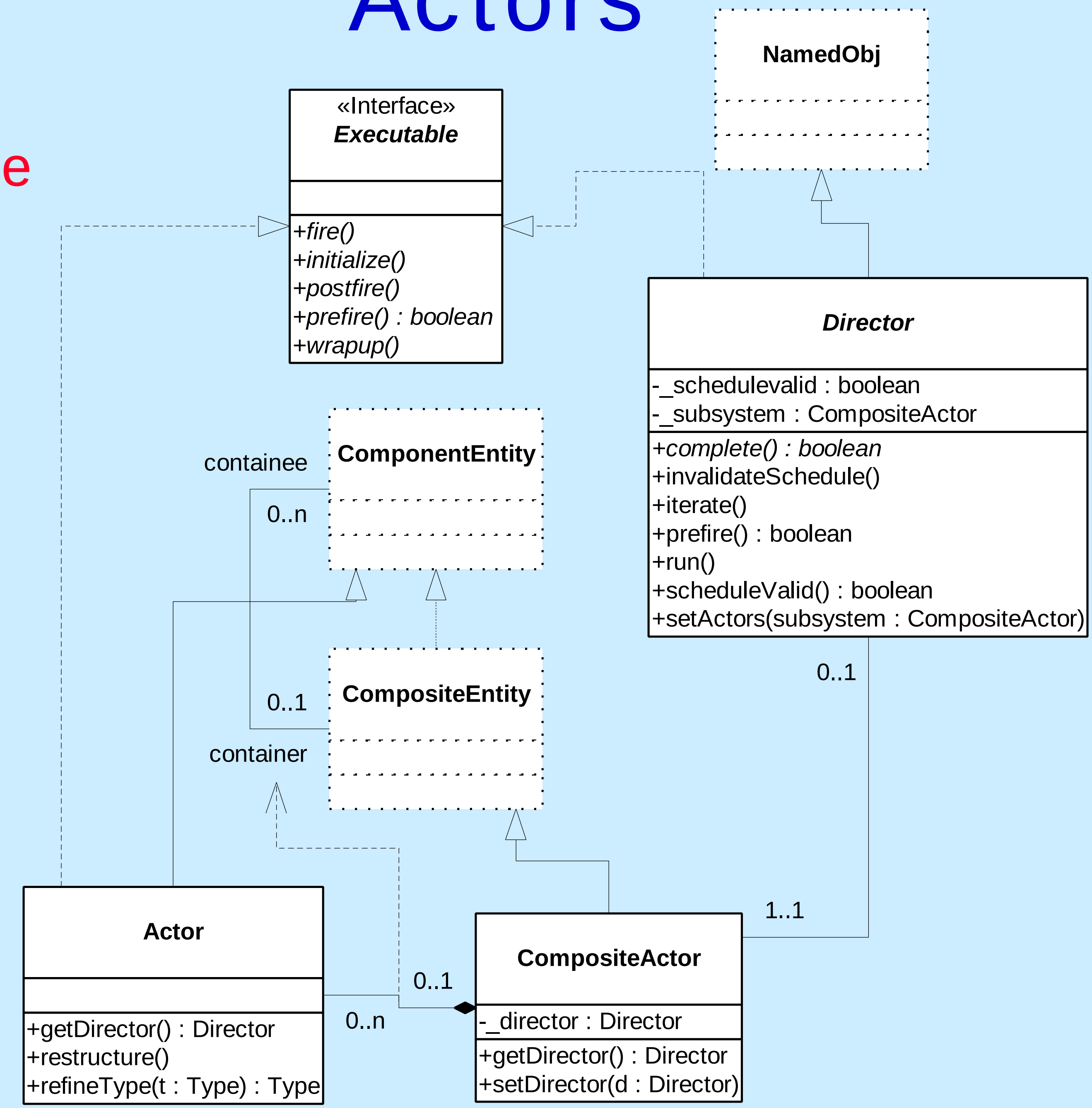
```
set e0 [java::new pt.kernel.CompositeEntity E0]
set e1 [java::new pt.kernel.ComponentEntity E1]
set e2 [java::new pt.kernel.ComponentEntity E2]
set p1 [$e1 newPort P1]
set p2 [$e1 newPort P2]
set r1 [$e0 connect $p1 $p2 R1]
```

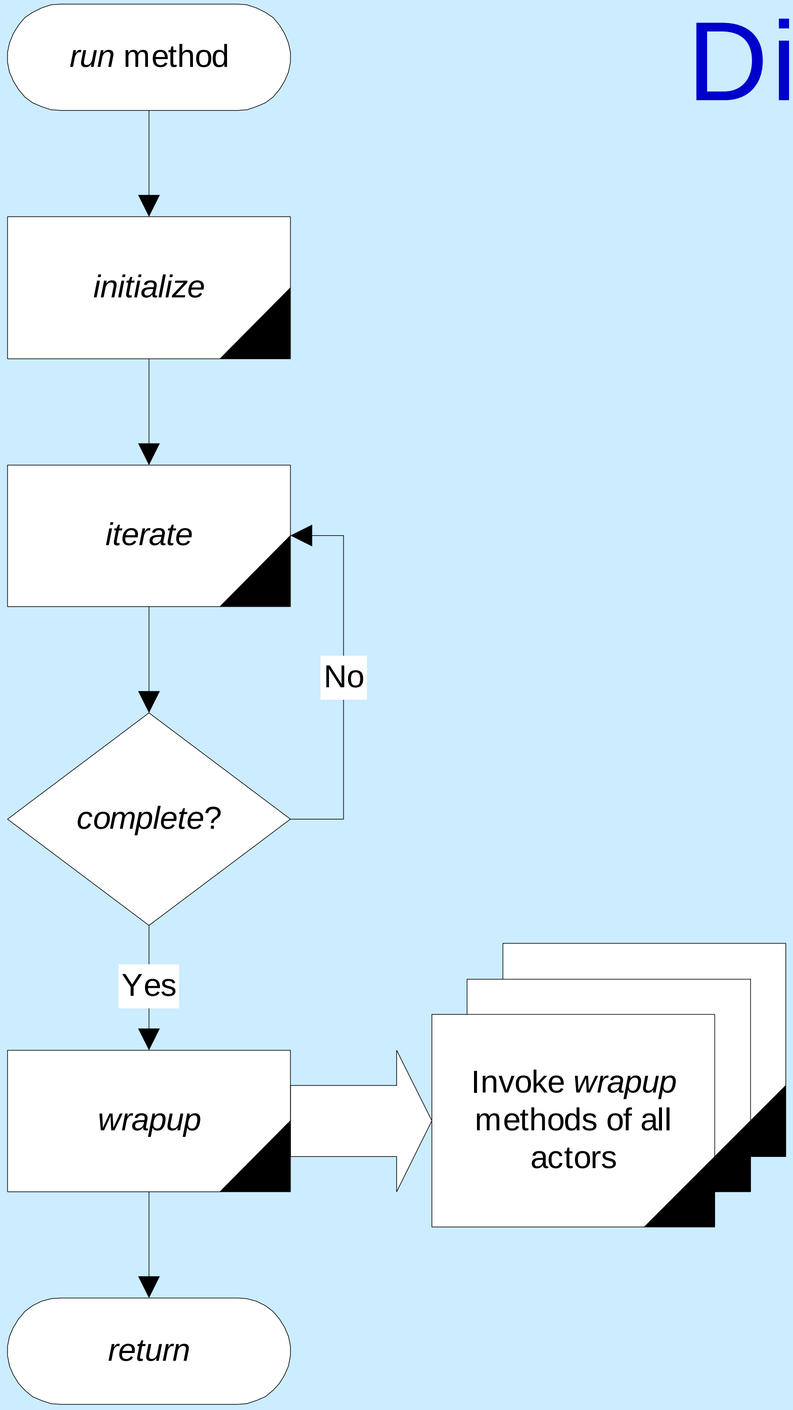


We have an extensive, automated test suite that uses this.

Actors

Make entities executable

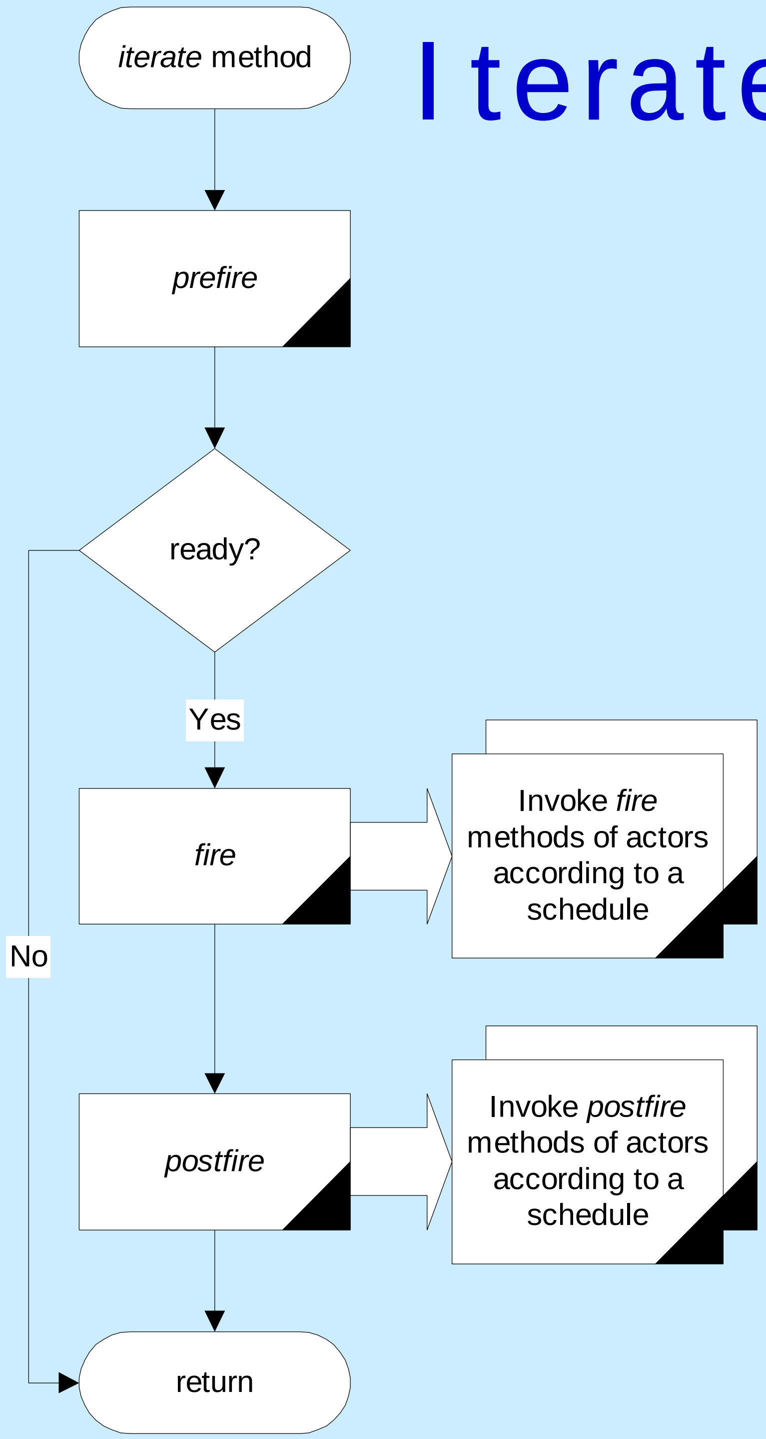




Director

- Provides a standard template for execution of a model.
- Most of the work is done in the iterate method.

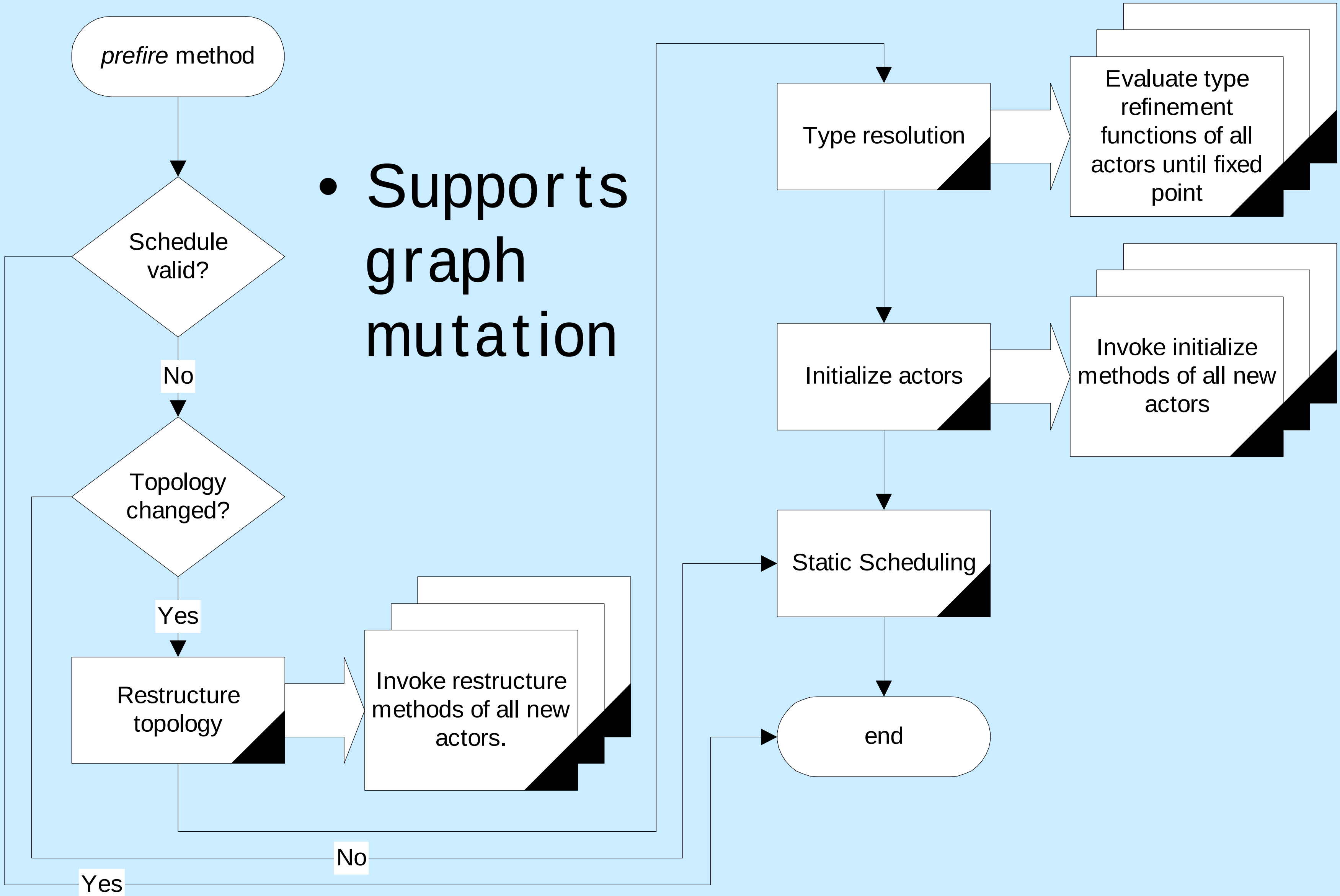
Iterate Method



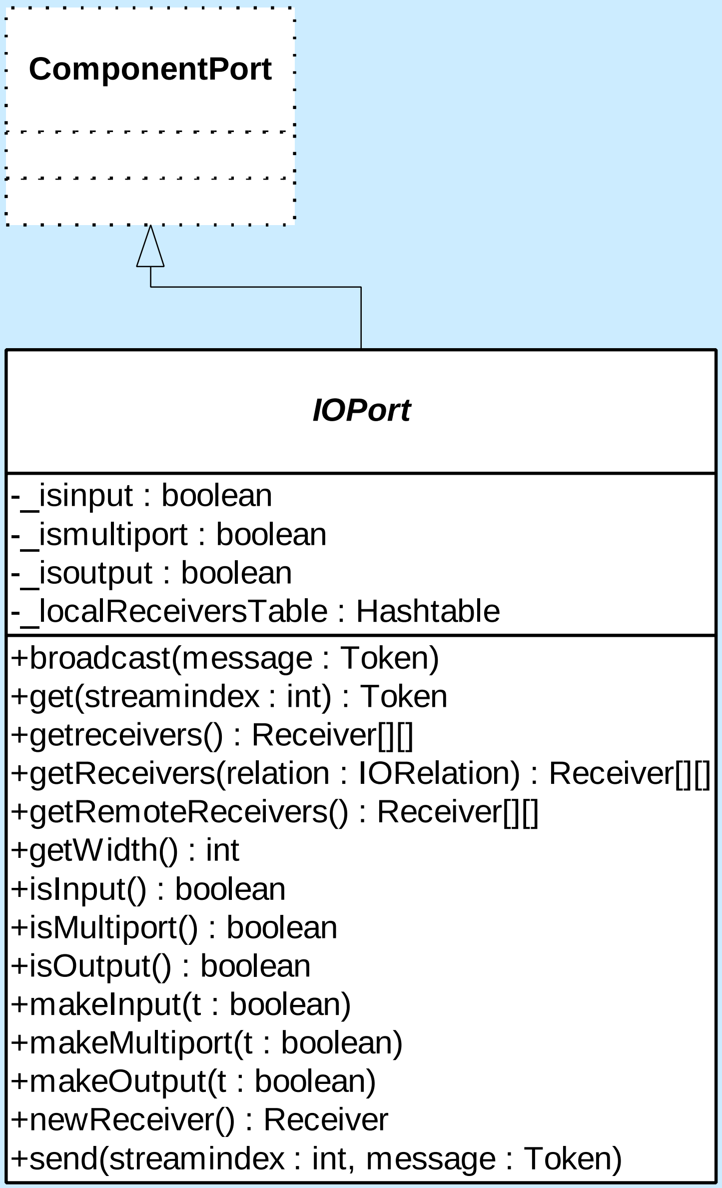
- Fire methods may be invoked multiple times in one iteration.
- Prefire and postfire methods are invoked exactly once.

Prefire Method

- Supports graph mutation



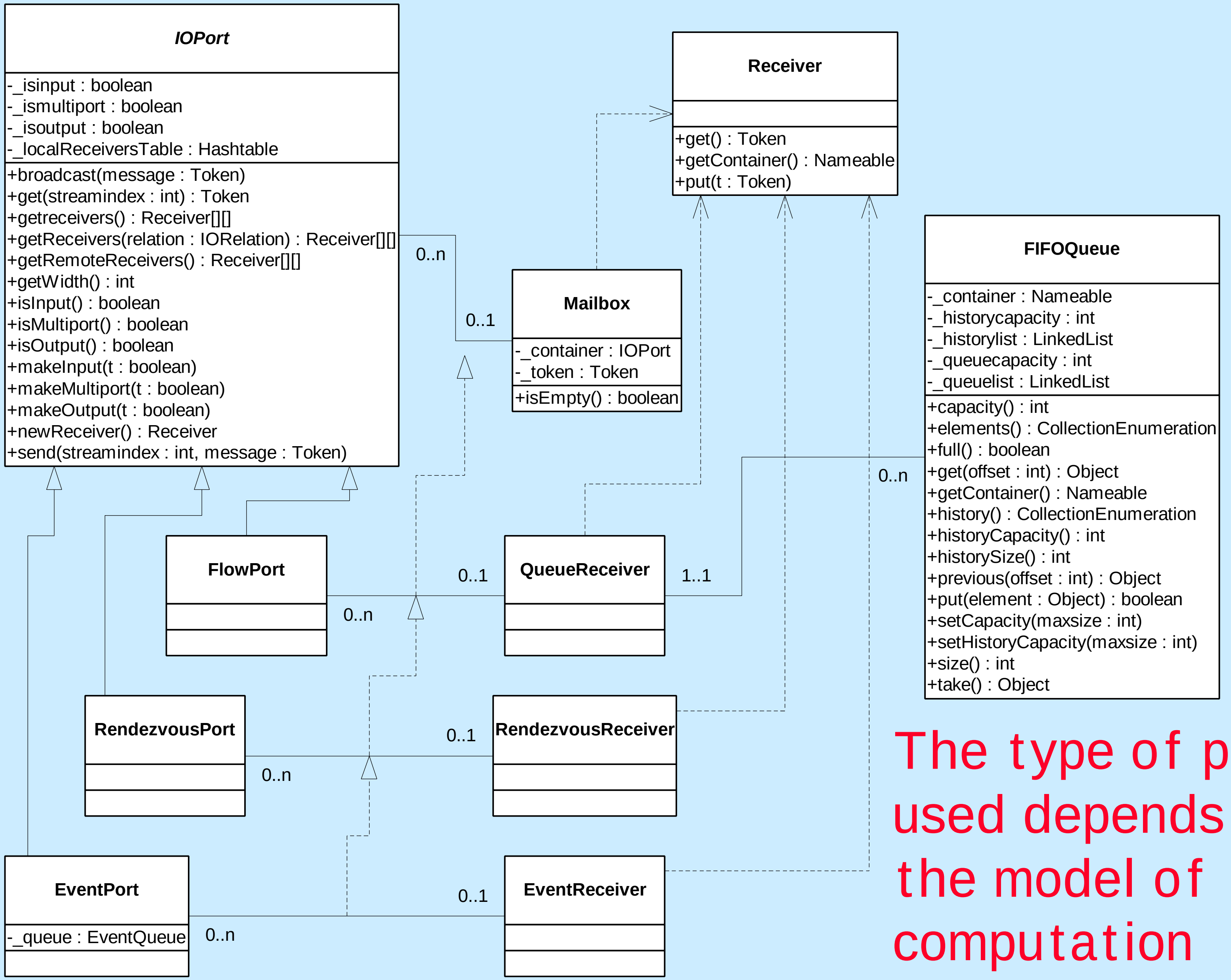
Message Passing



Support for:

- Inputs and Outputs
- Broadcast and Multicast
- Polymorphic in communication protocols

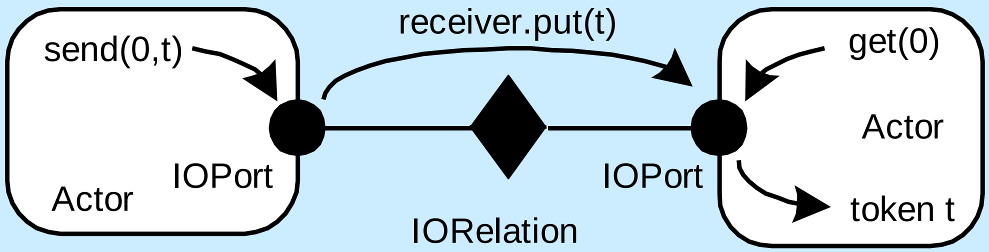
Polymorphic Message Passing



The type of port used depends on the model of computation

Point-to-Point Transport

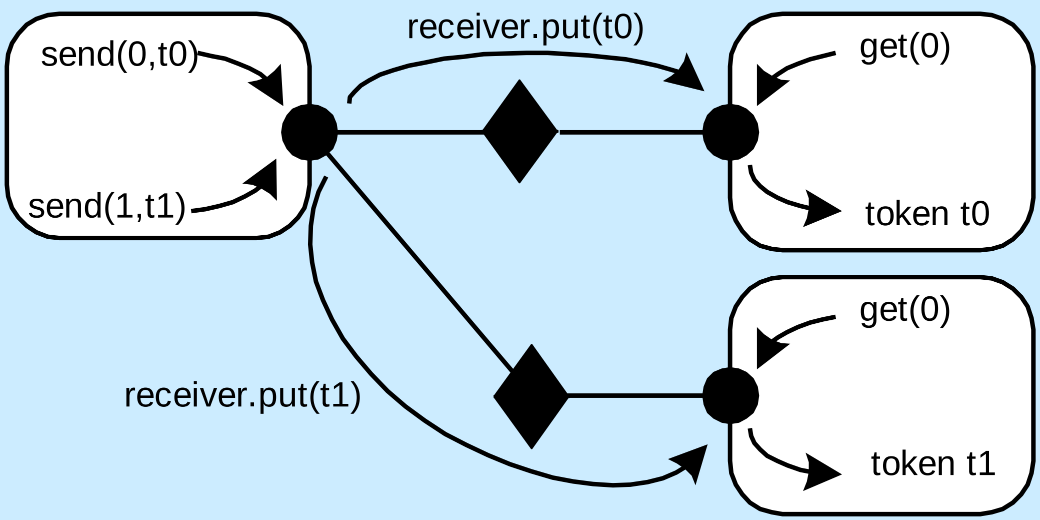
Simple transport:



- Sender calls send(channel,token)
- This calls receiver.put(token) for each receiver
- Receiver calls get(channel)

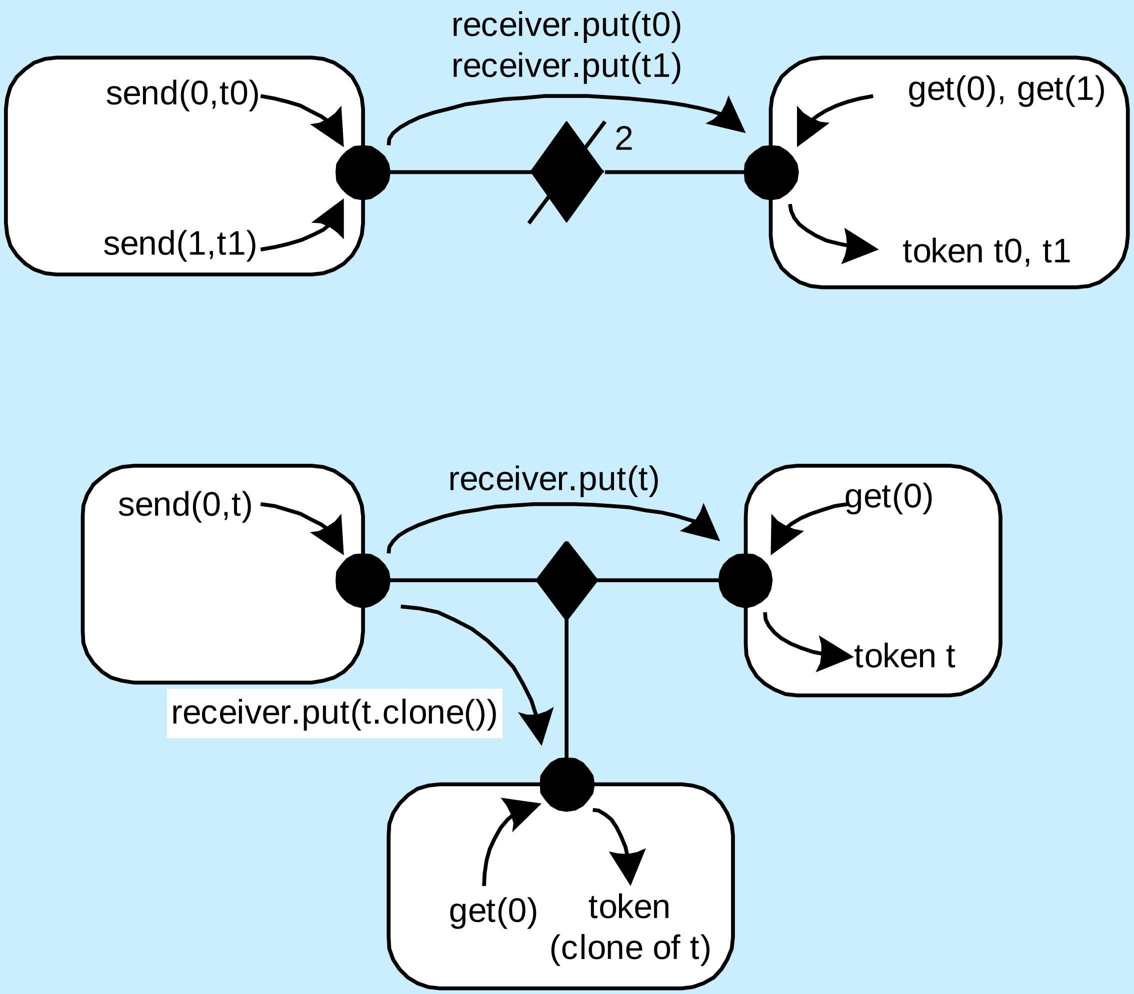
Multiports

Multiple (scalable) distinct channels:



- Sender calls send(channel,token)
- This calls receiver.put(token) for each receiver
- Receiver calls get(channel)

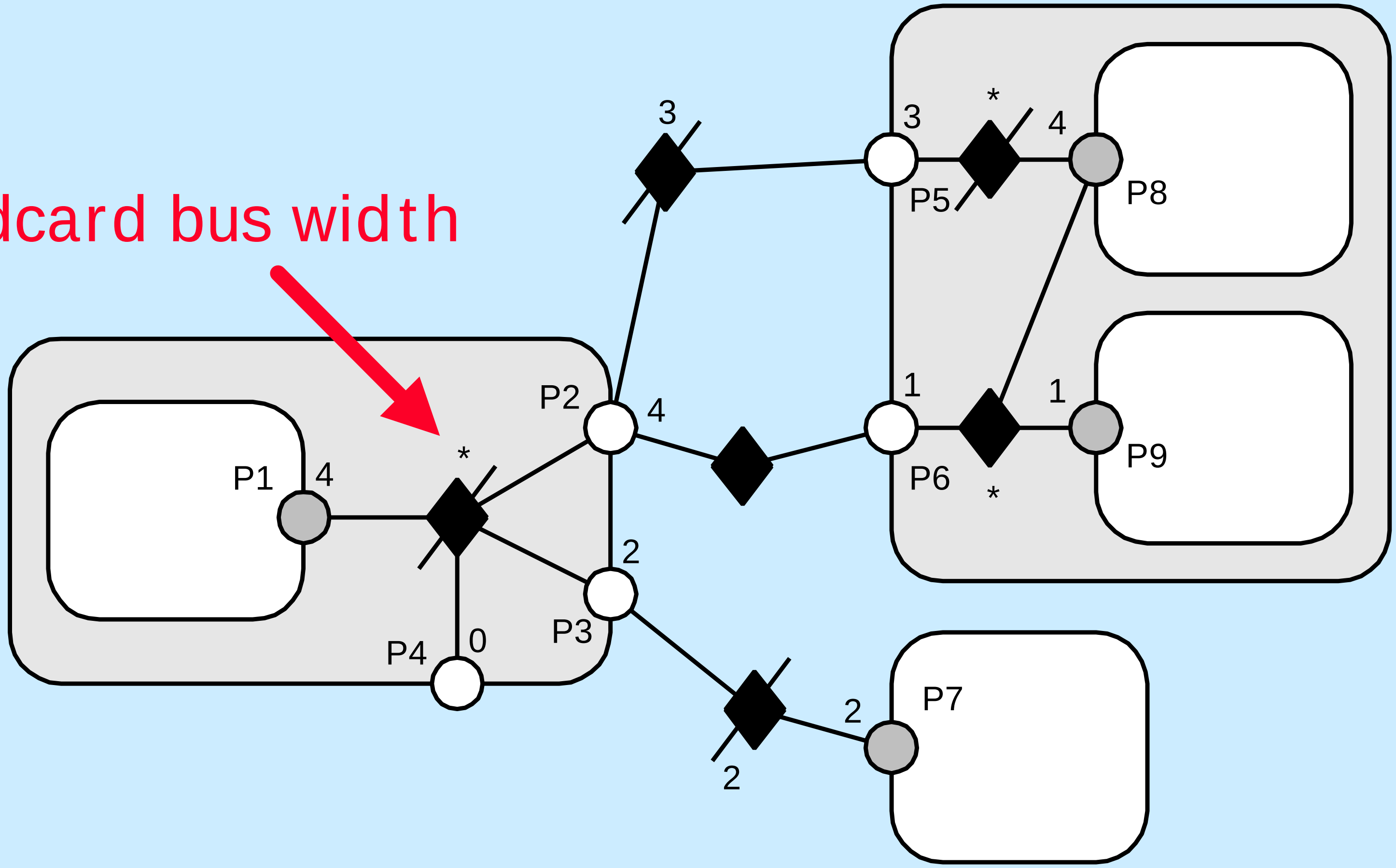
Multicast Transport



- Multiple desitnations for the same data
- Tokens are cloned automatically

Transparent Ports

Wildcard bus width



Sets of destination receivers are cached at the sender for efficiency.

The Data Package

- Tokens
  - encapsulate data for transport
- Parameters
  - attach names and dependencies to tokens
- Expressions
  - operate on tokens
- Type system
  - maximize polymorphism

PtParser

- Created using JavaCC & JJTree

  foo.jjt => JJTREE => foo.jj => JAVACC => foo.java

- Recognizes full range of arithmetic, relational and logical operators
- overloading in some cases to “do the right thing”

  eg 3+4 ==> 7

  3+” hello” ==> “3 hello”

PtParser

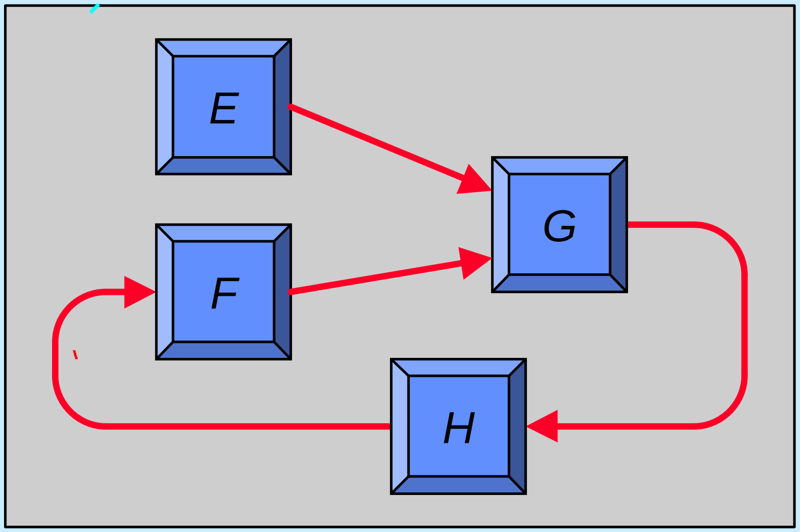
- allows reference to Ptolemy II Params, passed as an argument to the parser
  - eg if `clkFreq` and `duration` are Params in the current Entity
    - then `delay` Param could be given the value `duration/clkFreq` or `(duration*2)/clkFreq` etc.
- using reflection, all the functionality of java.lang.Math is available
- extendable: easy to add new functionality

  eg tcl(…) to invoke tcl to evaluate a string

  readFile("foo.bar") to read the given file as input

Type System

- Two-levels of types:
  - data type of atomic exchanges
  - signal type governing the exchange protocol
- Type hierarchy:
  - a lattice
- Type specification:
  - a monotonic function on this lattice that refines the estimated types.
- Type resolution:
  - iterate to a fixed point.



Use of Infrastructure in CT

- CTDirector
  - uses graph package for topological sort
  - overrides Director methods to implement a 4-th order Runge-Kutta solver.
- IOPort
  - use the default Mailbox receiver
- Data
  - use data package for parameters, signals
- Plot
  - use ptplot for interactive, animated plots

Use of Infrastructure in PN

- PNDirector
  - Kahn process networks model of computation
  - uses Java threads, one for each actor
  - manages deadlocks
- PNPort
  - Uses unbounded FIFO queues
  - Blocking reads
  - Bounded memory, when possible
- Very different from CT, hence stresses the kernel & actors package designs

Applications of PN

- Concurrent digital control processes with dynamic behavior.
- Provides a higher-level concurrency model than threads for programming in Java.
- (Future) With the addition of nondeterminism, model resource management problems.
- (Future) With the addition of time, model mappings of applications to hardware.

Synthesis

- Director manages synthesis process (vs. simulation)
- Separate interface from implementation.
- Support migration from simulation to implementation.

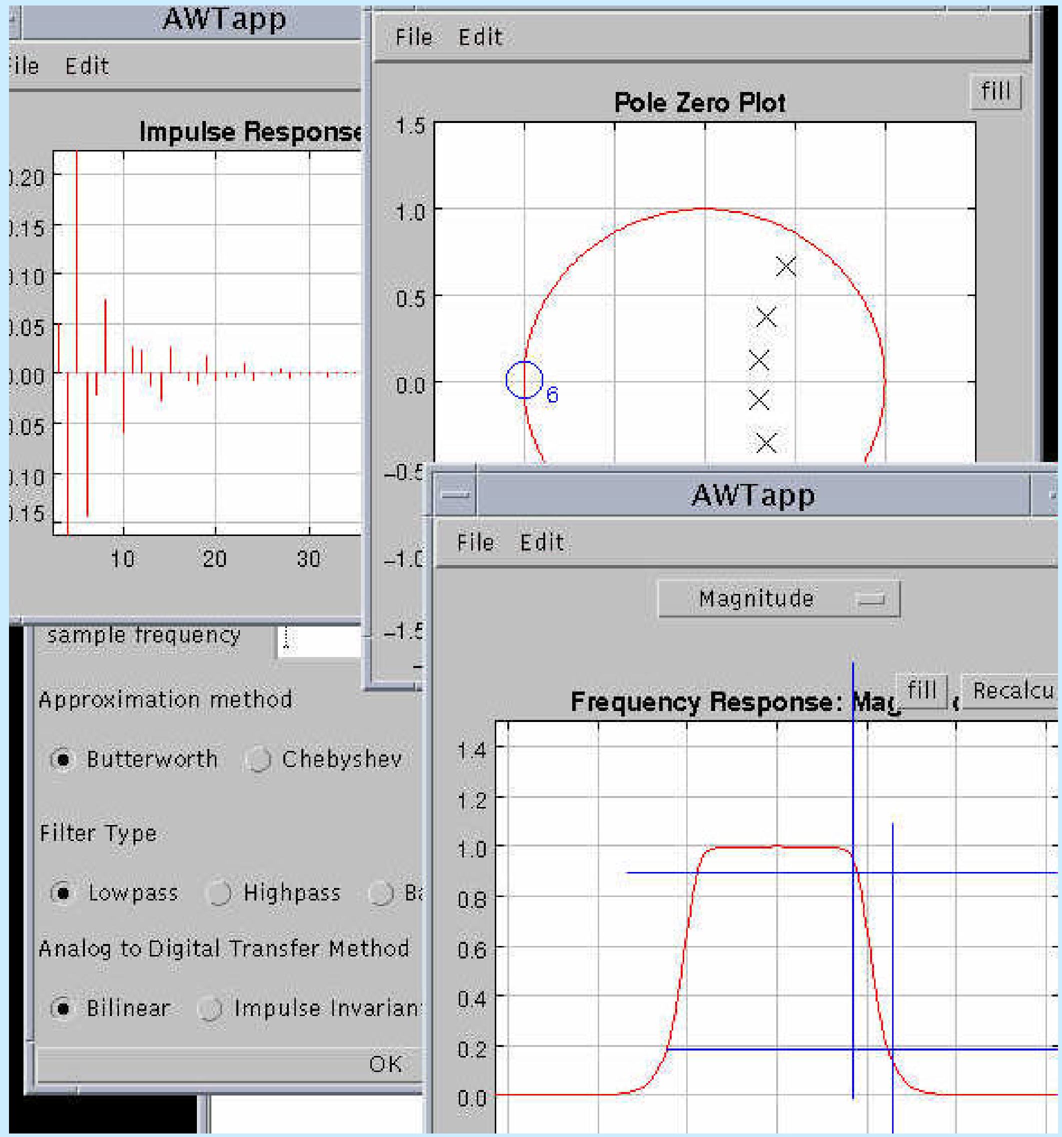
Note: We have a subcontract from Lockheed-Sanders in the adaptive computing program to develop technology for FPGA synthesis.

Additional Packages

- HOF (Higher-order functions)
- Graph (Leda-style graph algorithms)
- Math (Matrix operations, solvers, signal processing)
- Filter (Linear time-invariant systems)
- Plot (interactive, animated signal display)

The Filter Package

- Filter design tool.
- Highly interactive.
- Based on Ptplot.
- Model/view architecture.
- Uses math library.
- Designed to interface to Ptolemy filters.
- Web compatible.



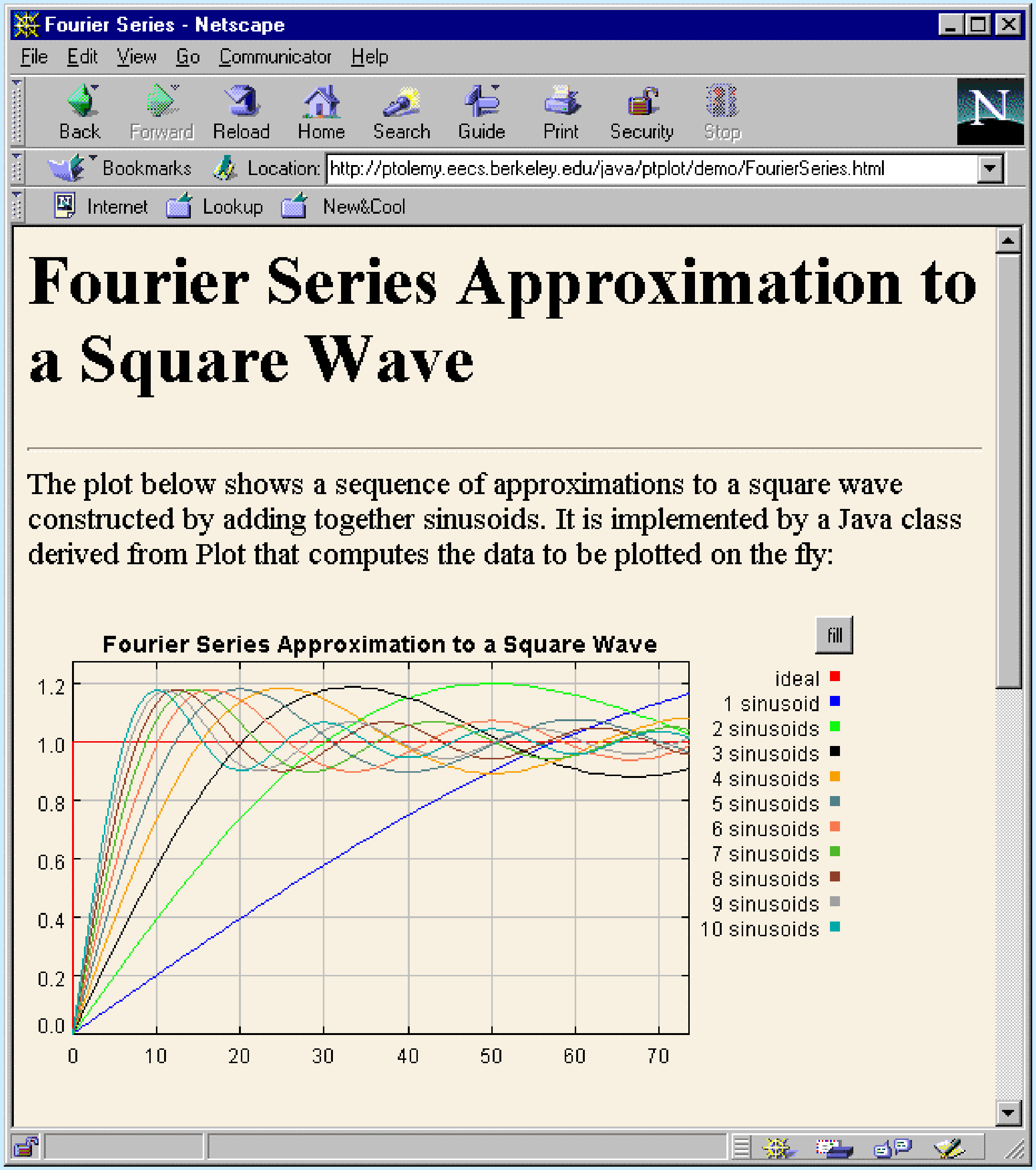
Modular Tools Architecture

- Use Java and Itcl
- Split Ptolemy into Java packages
- Split Tycho into Itcl packages
- Make everything network aware
- Use object modeling
- Use the model-view design pattern
- Use object-request broker technology
- Experiment with reflection, remote method invocation, etc.

First Released Java Module

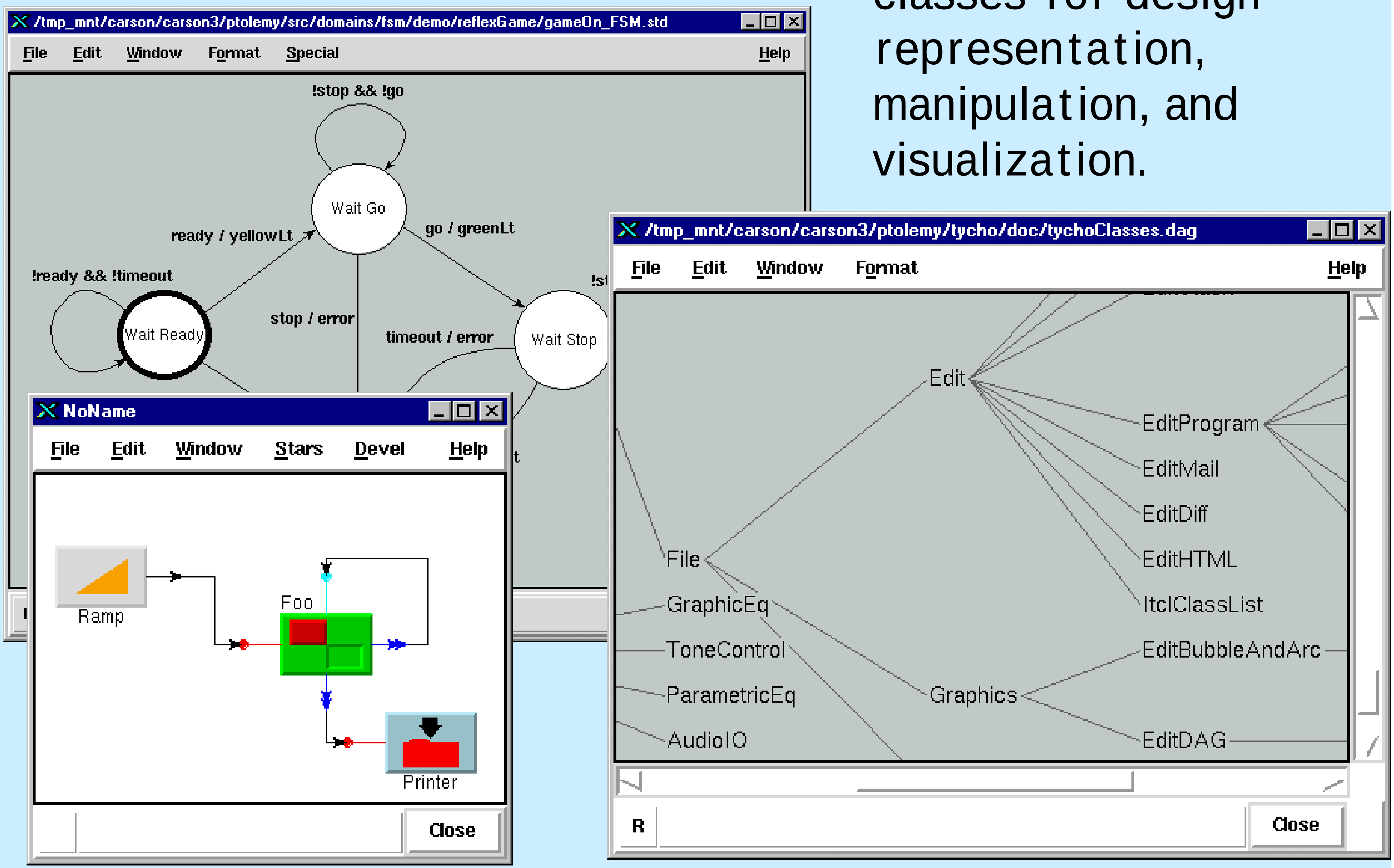
PtPlot is a Java package for interactive, animated signal plotting on the web.

We have used it to learn about Java applets as an interchange and modularization format, and will distribute Ptolemy modules similarly.



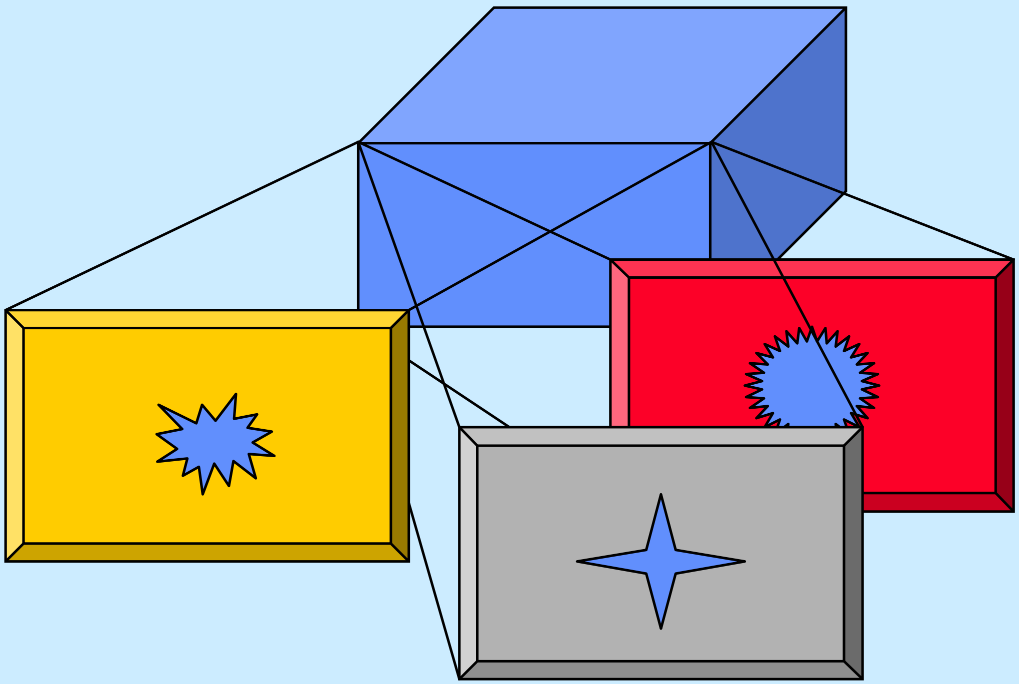
Tycho

Tycho is suite of Itcl classes for design representation, manipulation, and visualization.



Model/View Architecture

- Abstract data types
- Publish & subscribe



Software Engineering

- Code rating system: red, yellow, green, blue.
- Author/reviewer division of responsibilities.
- Automated test suites (scripted, in Tcl).
- Code coverage measurements.
- Integrated documentation.
- Tycho support.

The Ptolemy group has a tradition of emphasizing code and documentation quality.

Technology Transfer

- HP Ptolemy (released in March '98): Supports mixed-signal modeling (DSP + RF).
- BNeD, in cooperation with HP: Design and simulation of optical communication systems based on Ptolemy.
- Cadence: SDF and DE technology in SPW 3.0 and higher.

Interoperability

- CORBA
- Java Beans and the Tcl Bean
- An open architecture
- Small, modular Java packages
- Well-defined semantics

Major Accomplishments so Far

- Ptolemy II kernel, actors, data packages, CT, and PN domains.
- Semantics for hierarchical interaction of finite-state controllers with several models of computation.
- Formal semantics for DE systems.
- Demonstration of a client-server, web-based mechanism supporting Ptolemy simulations.
- Construction of a network-integrated, scripted design management environment (Tycho).
- Design of an "information model" and an associated "model-view" software architecture (Tycho).
- Release on the net of our first Java module, a multipurpose signal plotter.
- Java/Tycho integration.
- A well-attended Ptolemy miniconference.

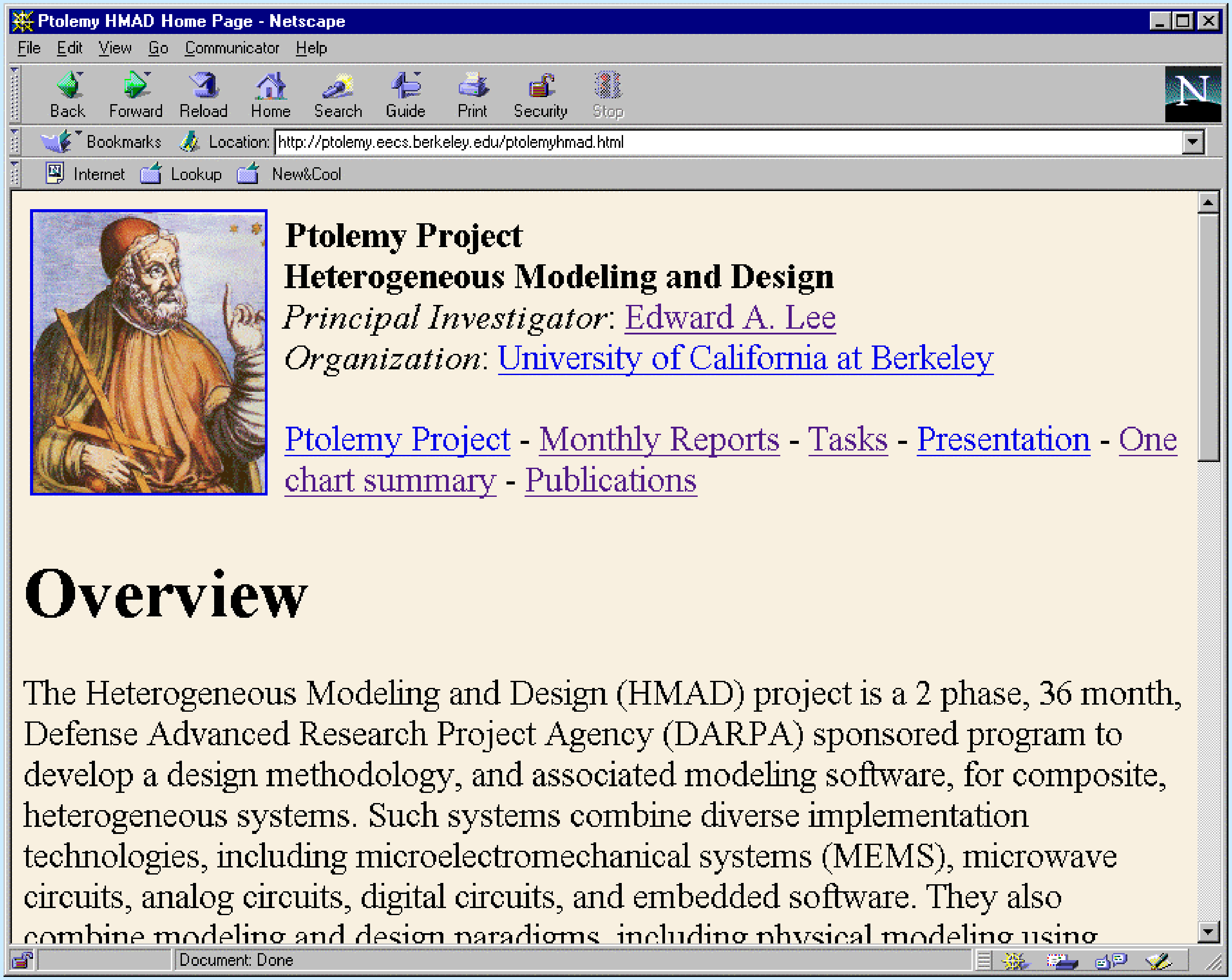
Actual Deliverables

- Reports
  - monthly reports
  - annual reports
- Software
  - Tycho 0.2 released (May, 1997)
  - PtPlot 0.1 released (October, 1997)
  - Ptolemy 0.7.1 alpha (May, 1998)
  - Ptolemy II Modules (September 1998 - December 1999)
  - Annual updates of Tycho (est. October, 1998, 1999)
- Papers
  - Reports, journal, and conference papers.

Future Work

- Finish actors, math, graph, data packages
- Design the wormhole interface
- Implement FSMs in Ptolemy II
- Create hybrid systems modeling
- Create CORBA interface
- Implement dataflow, DE, CSP domains
- Support nondeterminism in PN
- Add time to PN and CSP

More Information



http://ptolemy.eecs.berkeley.edu/ptolemyhmad.html

Publications

- E. A. Lee, “Modeling Concurrent Real-Time Processes Using Discrete Events,” Memorandum UCB/ERL M98/7 March 4th 1998, Electronics Research Laboratory, University of California, Berkeley, CA 94720.
- M. Goodwin, “Adaptive Signal Models: Theory, Algorithms, and Audio Applications,” Ph.D. thesis, University of California, Berkeley, December 1997. Available as UCB/ERL M97/31.
- P. K. Murthy, and E. A. Lee, ``Two Cycle Related Problems for Regular Dataflow Graphs: Complexity and Heuristics,'' Memorandum UCB/ERL M97/76, Electronics Research Laboratory, University of California, Berkeley, CA 94720, October 1997.
- S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee, ``Synthesis of Embedded Software from Synchronous Dataflow Specifications,'' Invited paper, to appear in J. of VLSI Signal Processing, 1998.
- A. Girault, B. Lee, and E. A. Lee, ``A Preliminary Study of Hierarchical Finite State Machines with Multiple Concurrency Models,'' Memorandum UCB/ERL M97/57, Electronics Research Laboratory, University of California, Berkeley, CA 94720, August 1997.

Publications (continued)

- S. Edwards, L. Lavagno, E. A. Lee, and A. Sangiovanni-Vincentelli, ``Design of Embedded Systems: Formal Models, Validation, and Synthesis,'' *Proceedings of the IEEE*, Vol. 85, No. 3, March 1997.
- S. A. Edwards, ``The Specification and Execution of Heterogeneous Synchronous Reactive Systems,'' Ph.D. thesis, University of California, Berkeley, May 1997. Available as UCB/ERL M97/31 .
- C. Hylands, E. A. Lee, and H. J. Reekie, ``The Tycho User Interface System,'' to be presented at the 5th Annual Tcl/Tk Workshop '97, Boston, Massachusetts, July, 1997.
- S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee, ``Software Synthesis for Synchronous Dataflow,'' International Conference on Application Specific Systems, Architectures, and Processors, July, 1997, invited paper.
- W.-T. Chang, S.-H. Ha, and E. A. Lee, ``Heterogeneous Simulation -- Mixing Discrete-Event Models with Dataflow,'' invited paper, Journal on VLSI Signal Processing, Vol. 13, No. 1, January 1997.

Publications (continued)

- E. A. Lee and A. Sangiovanni-Vincentelli, ``A Denotational Framework for Comparing Models of Computation,'' ERL Memorandum UCB/ERL M97/11, University of California, Berkeley, CA 94720, January 30, 1997.
- P. K. Murthy, E. A. Lee, "Some cycle-related problems for regular dataflow graphs: complexity and heuristics," UCB/ERL Tech. Report, UCB/ERL M97/76, July 1997.
- S. Kim and E. A. Lee, ``An Infrastructure for Numeric Precision Control in the Ptolemy Environment'', Proceedings of the 40th Midwest Symposium on Circuits and Systems, August 3-6, 1997.
- Richard S, Stevens (Naval Research Laboratory), Marlene Wan, Peggy Laramie (UCB), Thomas M. Parks (MIT Lincoln Labs), and Edward A. Lee (UCB), "Implementation of Process Networks in Java," UCB/ERL Tech. Report, number pending, November 1997.

Publications (continued)

Under subcontract to UT Austin (Brian Evans):

- D. Arifler, C. Duong, B. L. Evans, S. K. Marwat, C. M. Moy, and A. Yuan, ``A Configurable, Portable, Extensible Framework for Web-Enabled Interactive Simulation of Software for Embedded Programmable Processors,'' submitted.
- A. K. Kulkarni, A. Dube, and B. L. Evans, ``Benchmarking Code Generation Methodologies for Programmable Digital Signal Processors,'' submitted.
- B. Lu, B. L. Evans, and D. V. Tosic, ``Simulation and Synthesis of Artificial Neural Networks Using Dataflow Models in Ptolemy,'' Invited Paper, Proc. IEEE Conf. on Neural Network Applications in Engineering, Sep. 8-9, 1997.