Ptolemy Project Status and Overview

Edward A. Lee
Professor
Ptolemy Project Director

Ptolemy Miniconference
May 9, 2003
Berkeley, CA

Project Participants

Director:

- Edward A. Lee

Staff:

- Christopher Hylands
- Susan Gardner (Chess)
- Nuala Mansard
- Mary P. Stewart
- Neil E. Turner (Chess)
- Lea Turpin (Chess)

Postdocs, Etc.:

- Joern Janneck, Postdoc
- Rowland R. Johnson, Visiting Scholar
- Kees Vissers, Visiting Industrial Fellow
- Daniel Lázaro Cuadrado, Visiting Scholar

Graduate Students:

- J. Adam Cataldo
- Chris Chang
- Elaine Cheong
- Sanjeev Kohli
- Xiaojun Liu
- Eleftherios D. Matsikoudis
- Stephen Neuendorffer
- James Yeh
- Yang Zhao
- Haiyang Zheng
- Rachel Zhou

Chess: Center for Hybrid and Embedded Software Systems

Seeded by a Major NSF/ITR project
(National Science Foundation/Information Technology Research)
Project: *Foundations of Hybrid and Embedded Software Systems*
A collaboration with Vanderbilt's ISIS Institute and U. of Memphis

Chess Board of Directors

- Tom Henzinger, tah@eecs.berkeley.edu
- Edward A. Lee, eal@eecs.berkeley.edu
- Alberto Sangiovanni-Vincentelli, alberto@eecs.berkeley.edu
- Shankar Sastry, sastry@eecs.berkeley.edu

Other key faculty

- Alex Aiken, aiken@eecs.berkeley.edu
- Dave Auslander, dma@me.berkeley.edu
- Ruzena Bajcsy, bajcsy@eecs.berkeley.edu
- Karl Hedrick, khedrick@me.berkeley.edu
- Kurt Keutzer, keutzer@eecs.berkeley.edu
- George Necula, necula@eecs.berkeley.edu
- Masayoshi Tomizuka, tomizuka@me.berkeley.edu
- Pravin Varaiya, varaiya@eecs.berkeley.edu

Software Legacy of the Project

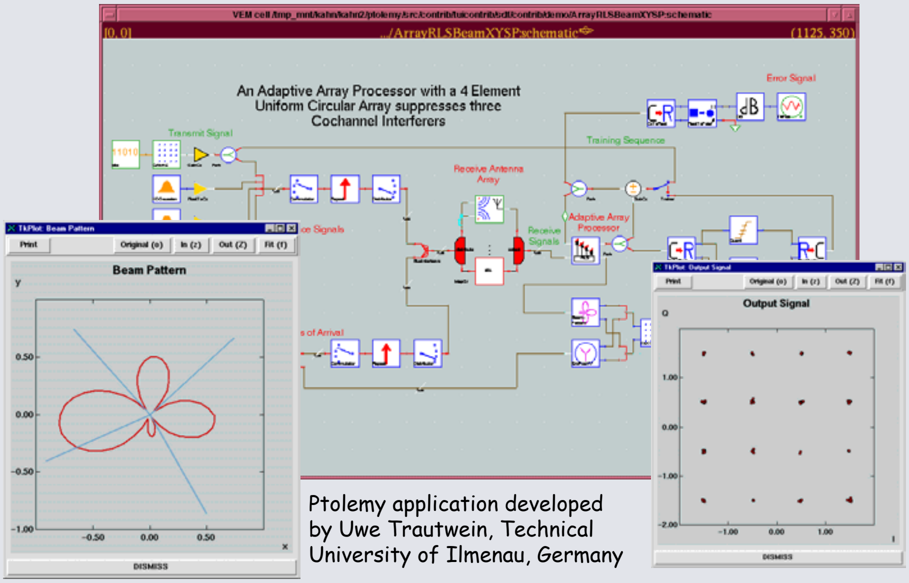
- Gabriel (1986-1991)
  - Written in Lisp
  - Aimed at signal processing
  - Synchronous dataflow (SDF) block diagrams
  - Parallel schedulers
  - Code generators for DSPs
  - Hardware/software co-simulators
- Ptolemy Classic (1990-1997)
  - Written in C++
  - Multiple models of computation
  - Hierarchical heterogeneity
  - Dataflow variants: BDF, DDF, PN
  - C/VHDL/DSP code generators
  - Optimizing SDF schedulers
  - Higher-order components
- Ptolemy II (1996-2022)
  - Written in Java
  - Domain polymorphism
  - Multithreaded
  - Network integrated
  - Modal models
  - Sophisticated type system
  - CT, HDF, CI, GR, etc.

Each of these served us, first-and-foremost, as a laboratory for investigating design.

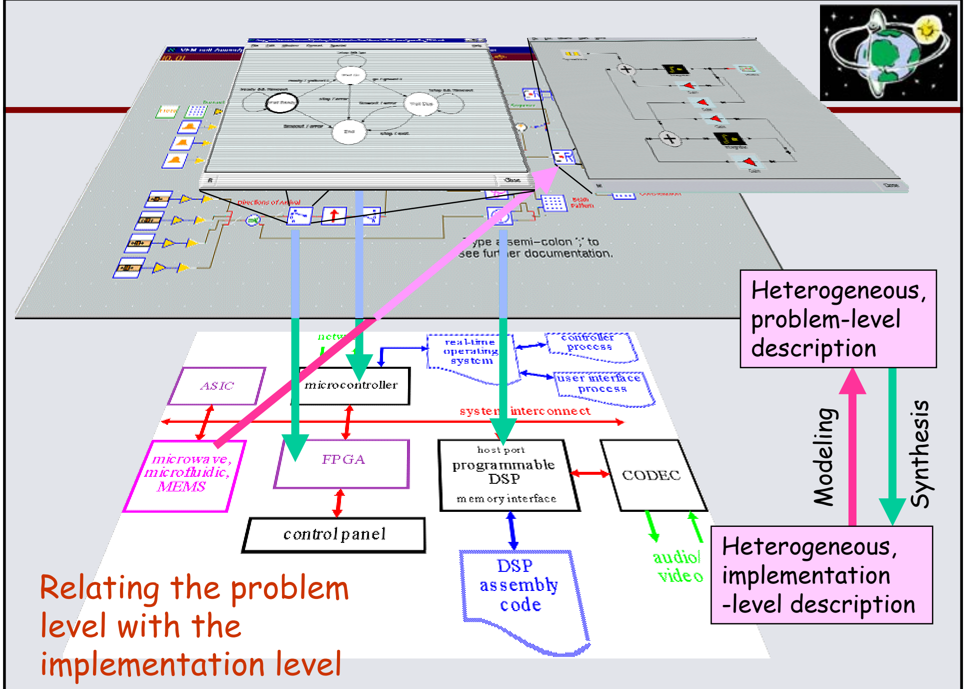
- PtPlot (1997-??)
  - Java plotting package
- Tycho (1996-1998)
  - Itcl/Tk GUI framework
- Diva (1998-2000)
  - Java GUI framework

All open source.
All *truly* free software (cf. FSF).

Ptolemy Classic Example



Ptolemy application developed by Uwe Trautwein, Technical University of Ilmenau, Germany

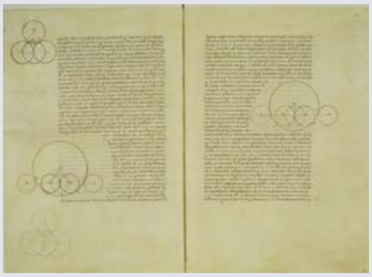


Heterogeneous, problem-level description

Heterogeneous, implementation -level description

Relating the problem level with the implementation level

Foundations

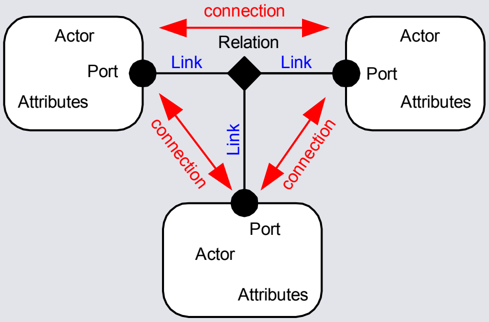


Our contributions:

- Behavioral Types
- Domain Polymorphism
- Responsible Frameworks
- Hybrid Systems Semantics
- Dataflow Semantics
- Tagged Signal Model
- Starcharts and Modal Model Semantics
- Discrete-Event Semantics
- Continuous-Time Semantics

Giving structure to the notion of "models of computation"

Actor-Oriented Design
Actors with Ports and Attributes



Model of Computation:

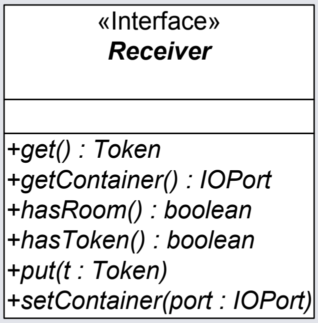
- Messaging schema
- Flow of control
- Concurrency

Examples:

- Push/Pull
- 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.

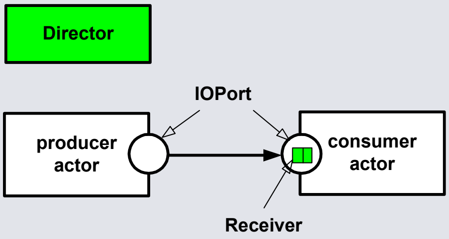
Receiver Interface



«Interface»
Receiver

+get() : Token
+getContainer() : IOPort
+hasRoom() : boolean
+hasToken() : boolean
+put(t : Token)
+setContainer(port : IOPort)

These polymorphic methods implement the communication semantics of a domain in Ptolemy II. The receiver instance used in communication is supplied by the director, not by the component.



Director

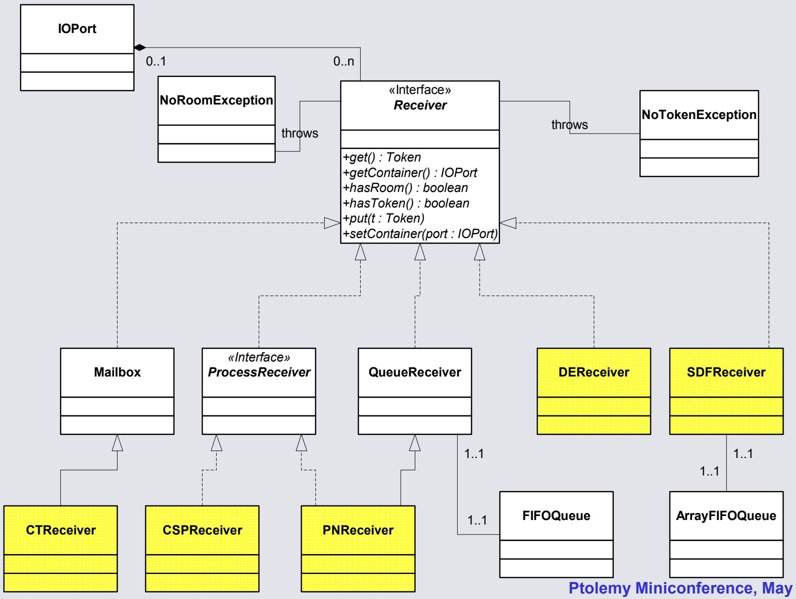
IOPort

producer actor

consumer actor

Receiver

Key to Domain Polymorphism: Receiver Object Model



IOPort

0..1

0..n

NoRoomException

throws

«Interface»
Receiver

+get() : Token
+getContainer() : IOPort
+hasRoom() : boolean
+hasToken() : boolean
+put(t : Token)
+setContainer(port : IOPort)

throws

NoTokenException

Mailbox

«Interface»
ProcessReceiver

QueueReceiver

DEReceiver

SDFReceiver

1..1

1..1

1..1

1..1

CTReceiver

CSPReceiver

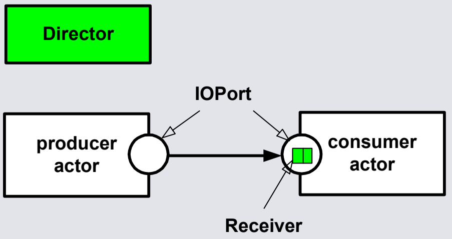
PNReceiver

FIFOQueue

ArrayFIFOQueue

Behavioral Types – Codification of Domain Semantics

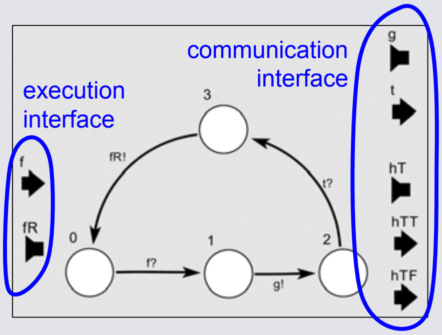
- Capture the dynamic interaction of components in *types*
- Obtain benefits analogous to data typing.
- Call the result *behavioral types.*



- Communication has
  - data types
  - behavioral types
- Components have
  - data type signatures
  - domain type signatures
- Components are
  - data polymorphic
  - domain polymorphic

Behavioral Type System

- We capture patterns of component interaction in a type system framework.
- We describe interaction types and component behavior using *interface automata.*
- We do type checking through *automata composition* (detect component incompatibilities)
- Subtyping order is given by the alternating simulation relation, supporting *domain polymorphism.*



A type signature for a consumer actor.

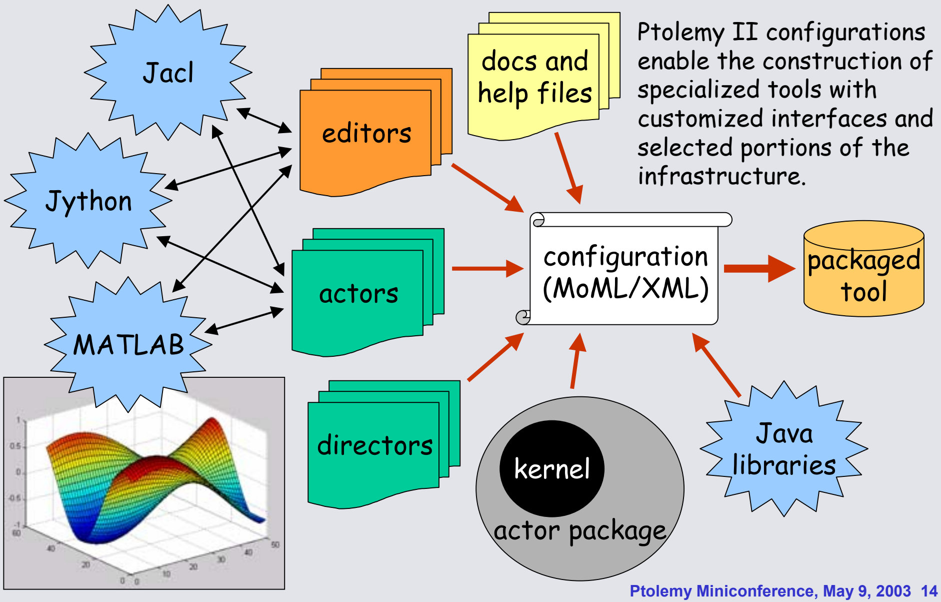
Software

Our contributions:

- Visual notations
  - Modal models
  - Higher-order components
- Domains as sandboxes
- Actor-orientation
- Polymorphism in actor-oriented design
- Tool integration as a semantics problem
- Scheduling and code generation methods
- Influenced many commercial products
- Effective software engineering in a research context
- Configurable tool architectures

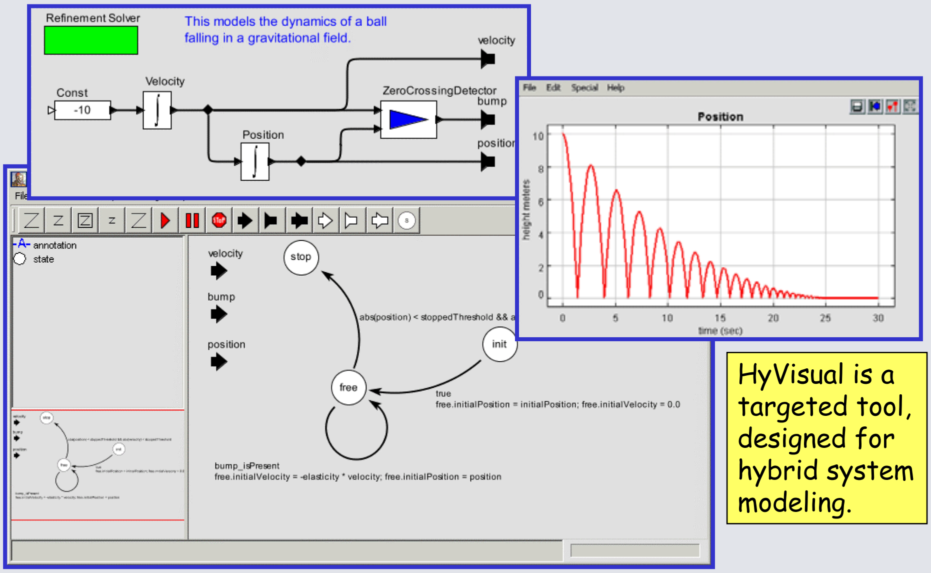
Providing architecture for “models of computation”

Configurable Tool Architecture



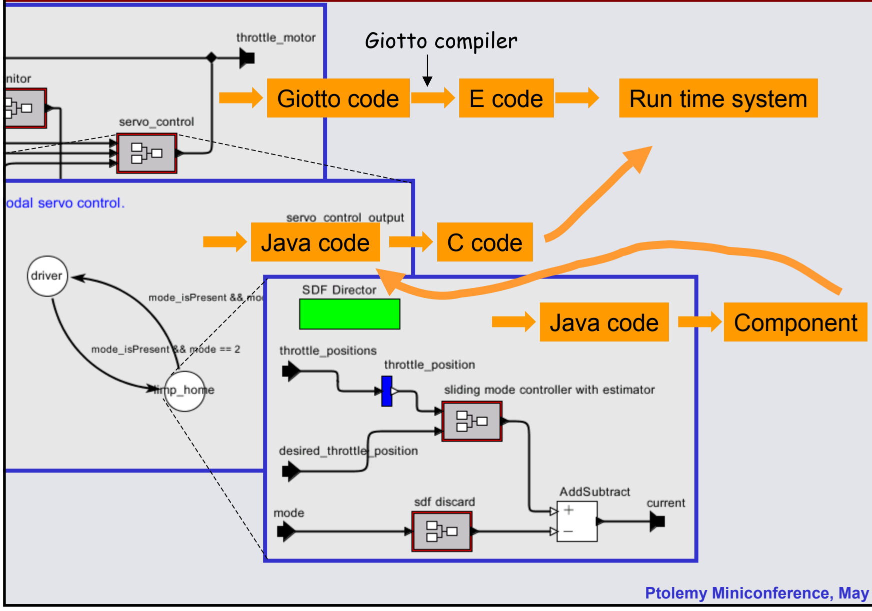
Ptolemy II configurations enable the construction of specialized tools with customized interfaces and selected portions of the infrastructure.

HyVisual – Hybrid System Modeling Tool Based on Ptolemy II, Released Jan. 2003


Refinement Solver
This models the dynamics of a ball falling in a gravitational field.
velocity
Const
-10
Velocity
ZeroCrossingDetector
bump
Position
position
File Edit Special Help
Position
height meters
10
8
6
4
2
0
0
5
10
15
20
25
30
time (sec)
annotation
state
velocity
bump
position
stop
abs(position) < stoppedThreshold && a
init
free
true
free.initialPosition = initialPosition; free.initialVelocity = 0.0
bump_isPresent
free.initialVelocity = -elasticity * velocity; free.initialPosition = position

HyVisual is a targeted tool, designed for hybrid system modeling.

Code Generation and Component Specialization


Giotto compiler

Giotto code

E code

Run time system

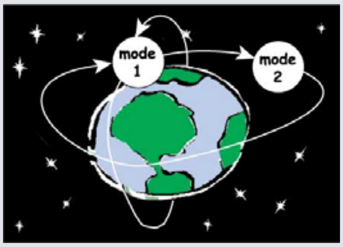
Java code

C code

Java code

Component

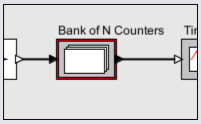
Much Else…



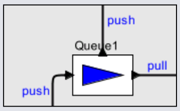
- Hybrid systems semantics
- CAL actor definition language
- Real-time semantics
- CI domain – push/pull
- Giotto and TM domains
- Interface definition/checking
- Units system
- Expression language semantics
- Modal models
- Meta models (IA domain)
- Image and video processing library
- Communications library
- …

Community Involvement
Recent Third Party Software Contributions

- Many enhancements contributed by RIM:
  - Transition refinements
  - Higher-order components
  - Performance improvements
  - Expression language improvements
  - Matlab integration
  - Emacs integration



- Enhancements contributed by Agile Design
  - Undo/Redo
  - Port positioning
  - Icon customization



- Hardware synthesis capability from BYU
- Distributed optimization package from Spain
- Graduate class on MoCs at Virginia Tech
- Ice-cube project: paper on neutrino detection