The Tycho User Interface System

Christopher Hylands

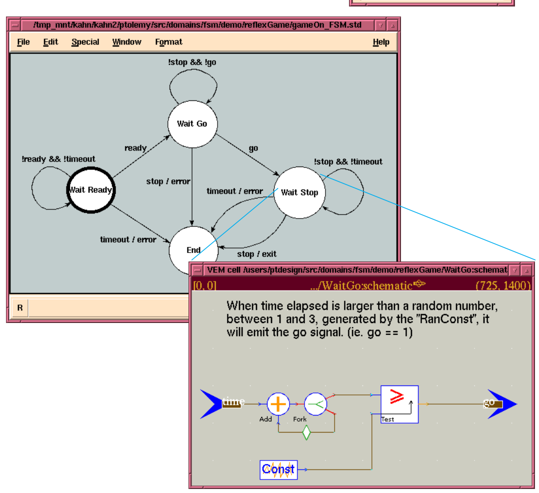
Edward A. Lee

John Reekie

UC Berkeley
Dept. of EECS

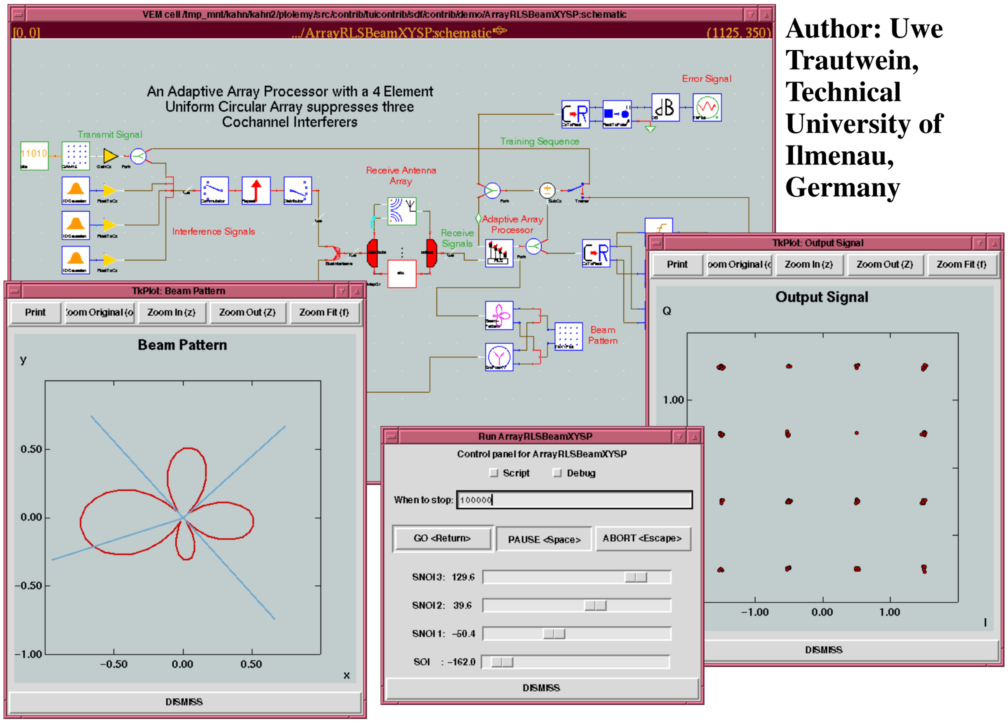
Ptolemy: heterogeneous modeling

- Design methodology for reactive systems—signal processing, telecommunications, hardware-software codesign
- Heterogeneity—multiple, domain-specific, semantic models:
  - synchronous/reactive models
  - multi-rate dataflow
  - discrete-event simulation
  - modular hierarchical finite state machines



UNIVERSITY OF CALIFORNIA AT BERKELEY

Ptolemy: interactive, high-level simulation



Author: Uwe Trautwein, Technical University of Ilmenau, Germany

Tycho

Goal: to extend the non-dogmatic nature of the Ptolemy kernel—multiple semantic models—to the user interface. Thus: multiple syntactic models.

Authors:

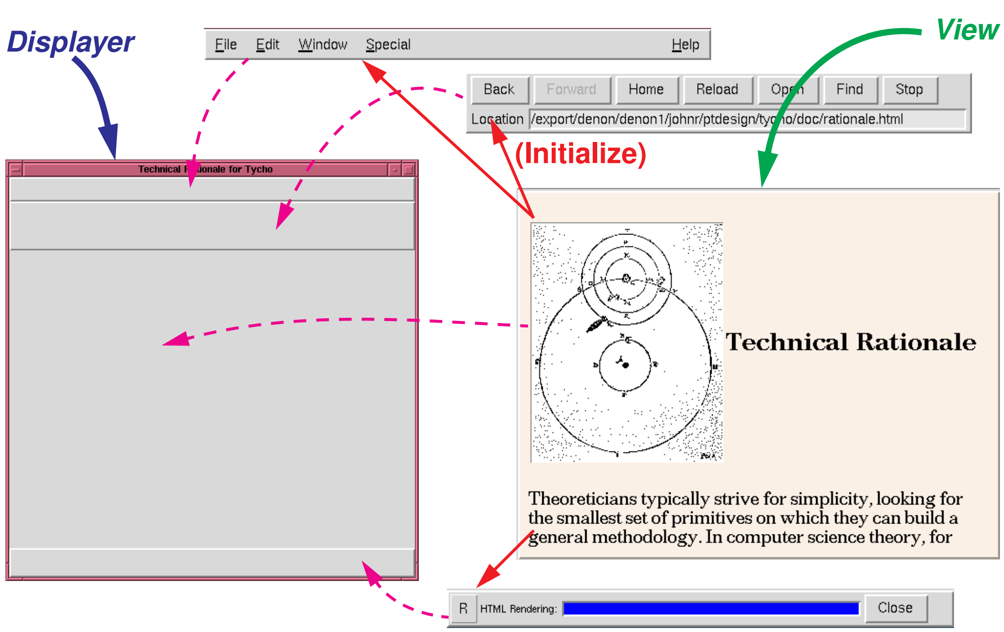
Edward A. Lee
Christopher Hylands
John Reekie
Kevin Chang
Cliff Cordeiro
Wan-teh Chang
Wei-Jen Huang
Joel King
Farhana Sheikh

Motivation

- **Cross-platform portability**
  - ✔ [incr Tcl]/[incr Tk] — object-oriented scripting
  - ✔ Java — portable, clean, system-level language
  - ✘ C/C++ — ?
- **New Ptolemy GUI**
  - ✔ Domain-specific graphical editors
  - ✔ Design visualization and exploration
  - ✘ Binary internal representations, complex databases
- **Extensible, re-usable user-interface framework**
  - ✔ Sub-classable user-interface widgets
  - ✔ Application framework
  - ✘ Ad-hoc Tcl scripts

Frameworks: UI layout

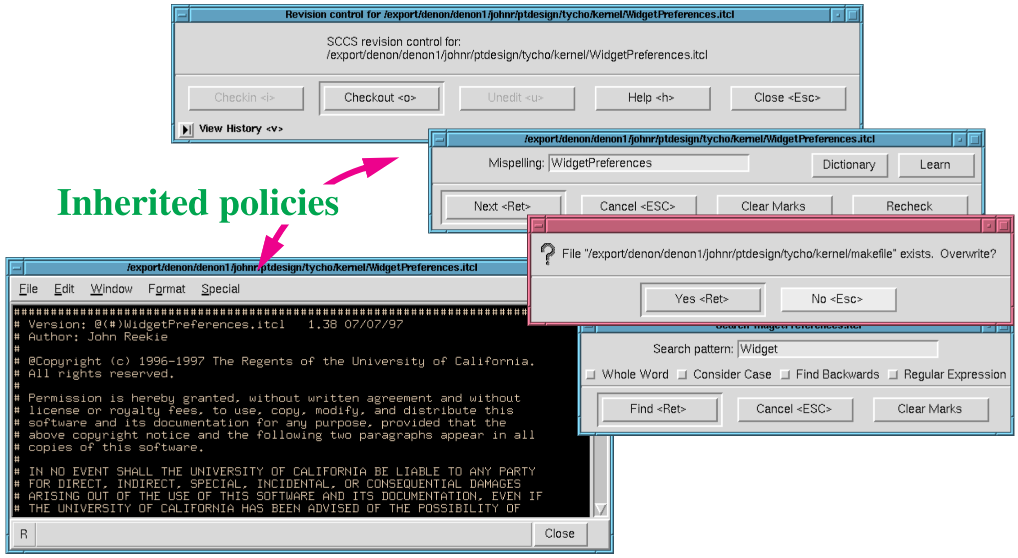
- **A *displayer* contains *views***
- **Menu bar, tool bar, and status bar come “for free”**



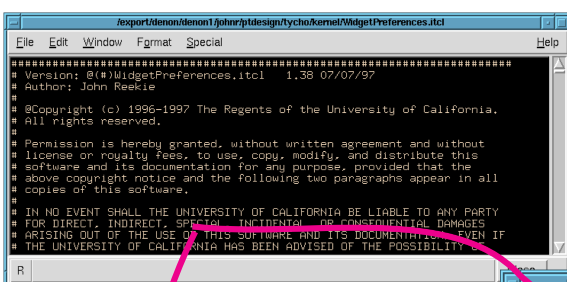
UNIVERSITY OF CALIFORNIA AT BERKELEY

Frameworks: file management

- File load/save policies
- Access management
- Look-and-feel

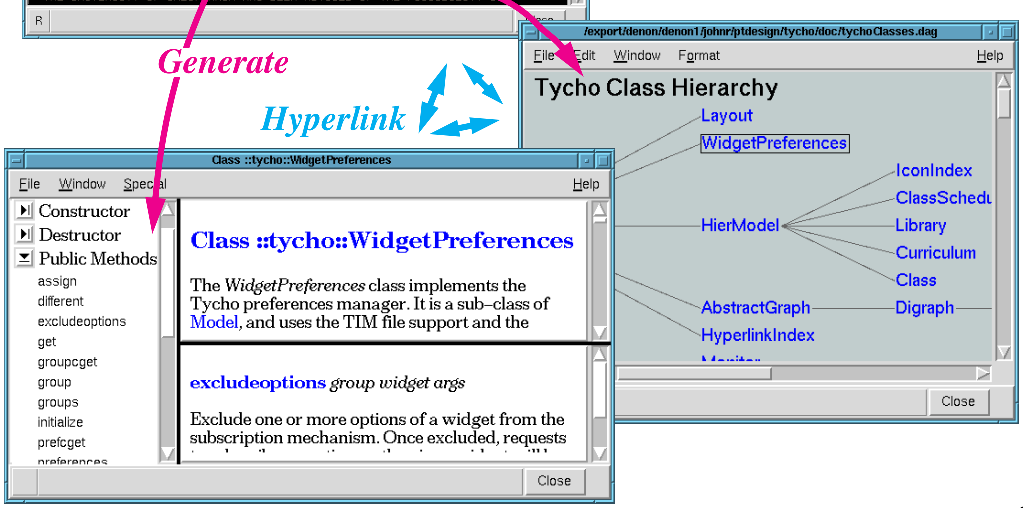


Frameworks: documentation



Multiple languages:

- ✔ [incr Tcl]
- ✔ Java

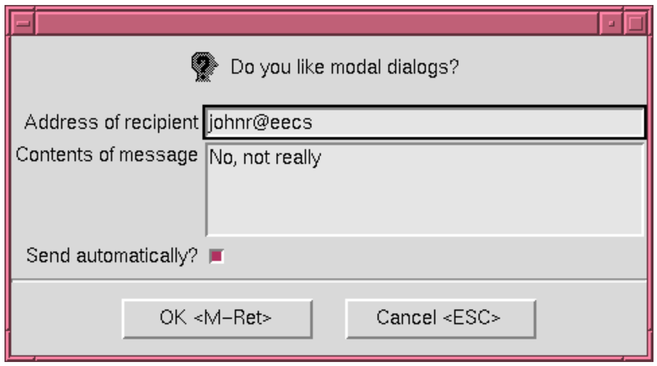


User interface components

Components are customizable by:

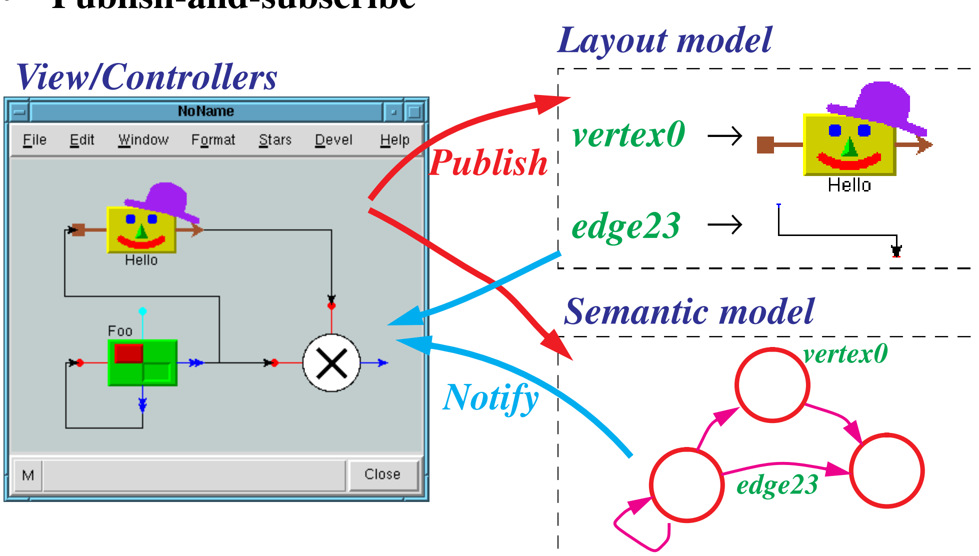
- parameters
- inheritance

```
::tycho::query "Do you like modal dialogs?" {
  {line address "Address of recipient" {johnr@eecs}}
  {lines contents "Contents of message" {No, not really} 4}
  {check auto "Send automatically?" 1}
}
```

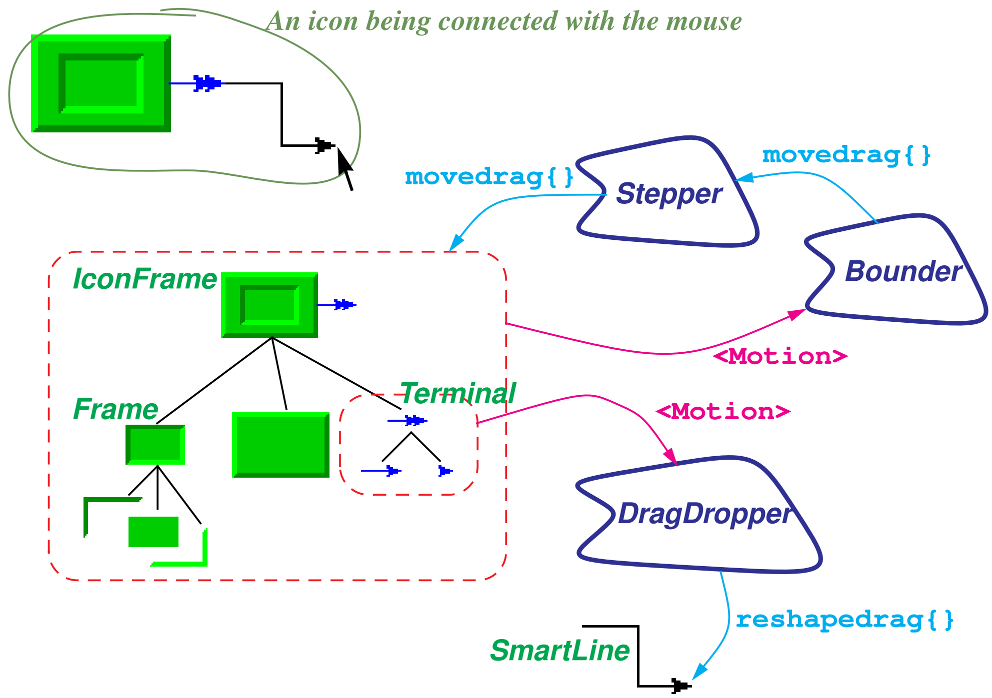


Model-view architecture

- **Flexible data representations**
- **Unbounded history mechanism**
- **"Publish-and-subscribe"**

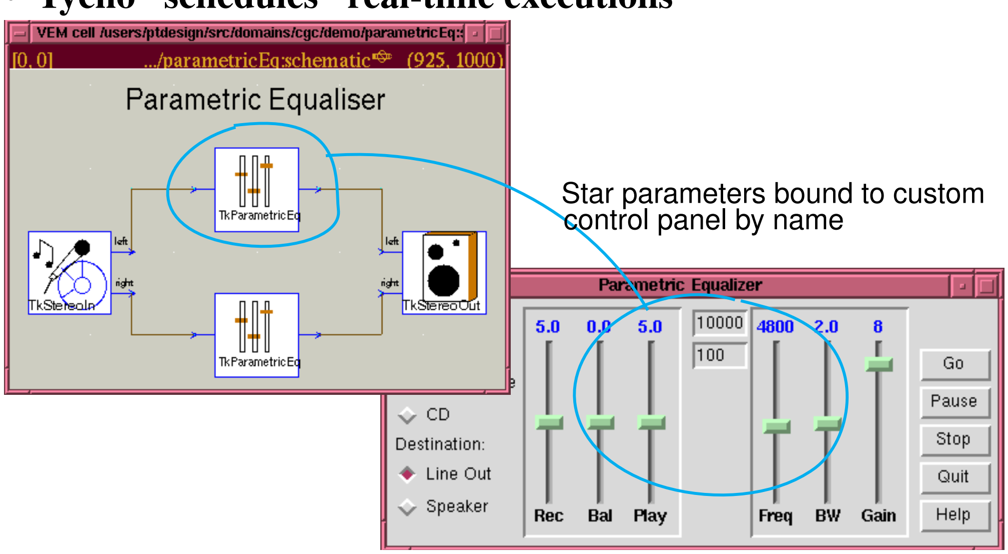


Hierarchical graphics and interactors



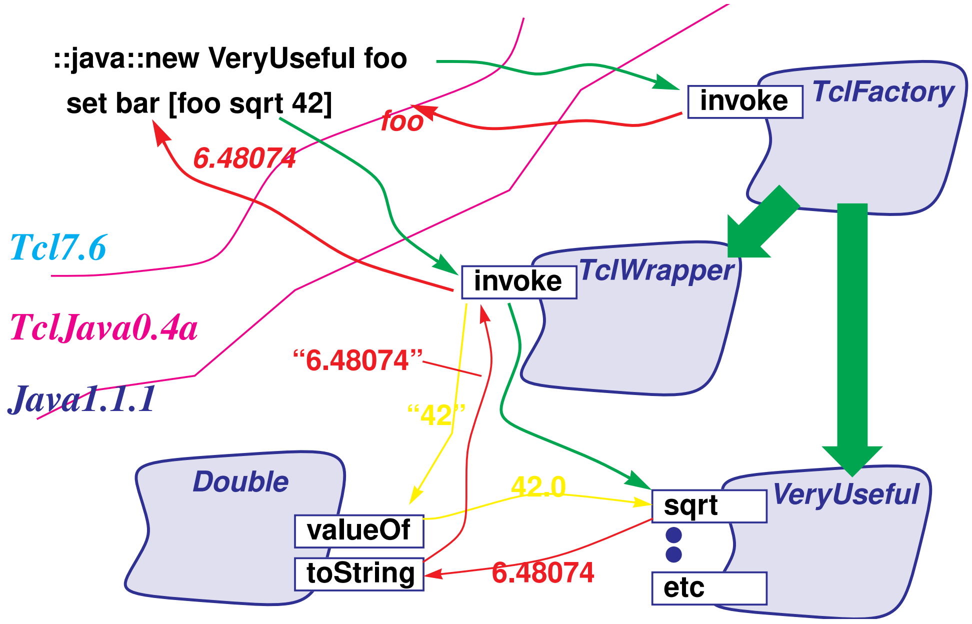
Custom simulation interfaces

- Ptolemy generates a Tcl-Tk package, which Tycho compiles
- User-interface uses Tycho infrastructure
- Tycho “schedules” real-time executions



Tcl ↔ Java

- **WELD i/f prototype: Tcl U/I, Java network access**
- **Updating needed: Tcl8.0, Java 1.1.x, object references**



Status

- **Version 0.2 released June 97**

  http://ptolemy.eecs.berkeley.edu/tycho/

  ✔ **Application framework**

  ✔ **Text editors and user interface components**

  ✔ **Automatic documentation**

  ☞ **Graphical and tree viewers, Tcl profiler**

- **Version 0.2.1 scheduled for Fall 97**

  ☞ **Itcl 3.0 (the byte-compiler version!)**

  ☞ **New Java interface**

  ✔ **"Network-aware": HTTP, FTP, mailto, WELD**

  ☞ **Graphical editing support**

Concluding remarks

- **For user interface development, Tk is flexible and powerful.**
- **[incr Tcl] and [incr Tk] greatly improve program structure and clarity.**
- **[Incr Tcl]/[incr Tk]is portable: Tycho runs on UNIX and Windows.**
- **Development at this scale (85 kloc) is hampered by poor performance and lax parsing.**
- **Our experiments with Tcl + Java are very positive:**
  - **Java is easy to learn and use, and has extensive libraries.**
  - **Tcl/Tk simplifies user interface development and scripting.**
  - **An updated, platform-independent interface is essential.**
- **Hearty thanks to those who developed infrastructure we have used: Michael McLennan, John Ousterhout, Stephen Uhler, Mark L. Ulferts.**