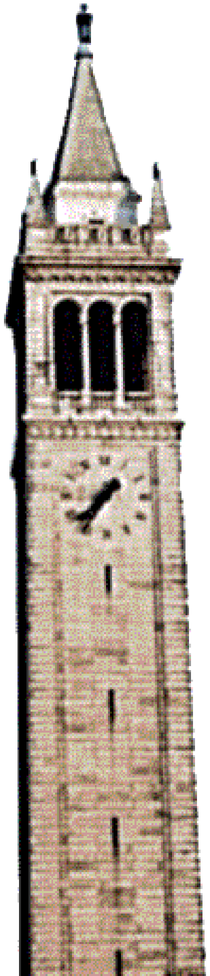


Synthesis of Distributed Real-Time Embedded Software

Edward A. Lee

Robert S. Pepper Distinguished Professor
UC Berkeley

Keynote talk

Electronic System Level Synthesis Conference

June 5-6, 2011

San Diego, California, USA

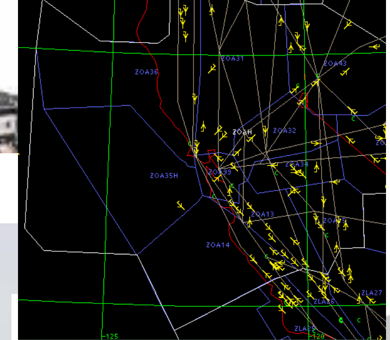
Key Collaborators:

- *Steven Edwards*
- *Sungjun Kim*
- *Isaac Liu*
- *Slobodan Matic*
- *Jan Reinke*
- *Sanjit Seshia*
- *Mike Zimmer*
- *Jia Zou*

Cyber-Physical Systems (CPS):

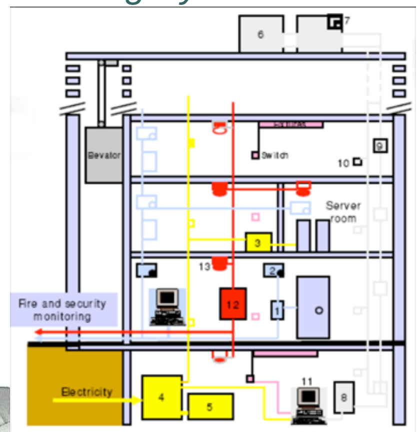
Orchestrating networked computational resources with physical systems

Avionics

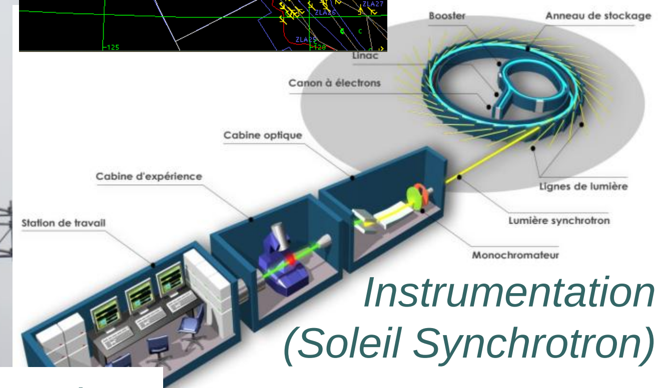


Transportation (Air traffic control at SFO)

Building Systems

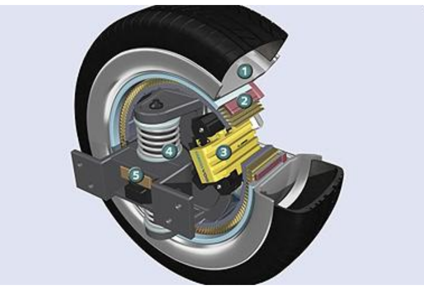


Telecommunications

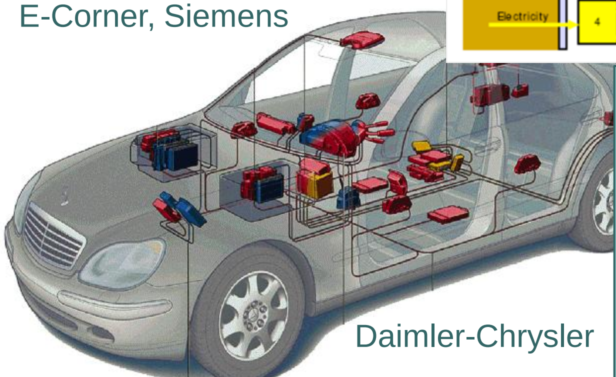


Instrumentation (Soleil Synchrotron)

Automotive



E-Corner, Siemens



Daimler-Chrysler

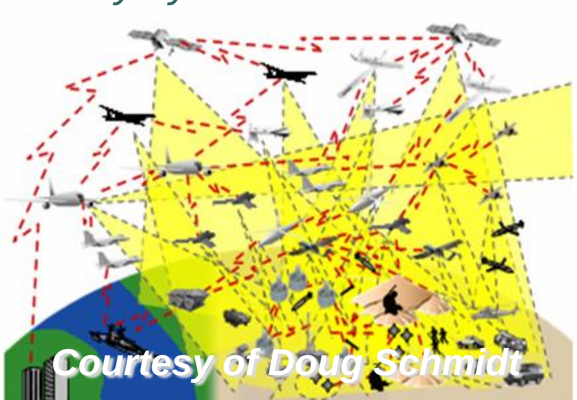
Power generation and distribution

Courtesy of General Electric

Factory automation

Courtesy of Kuka Robotics Corp.

Military systems:



Courtesy of Doug Schmidt

Printing Press Example

Bosch-Rexroth

- Application aspects
  - local (control)
  - distributed (coordination)
  - global (modes)
- Open standards (Ethernet)
  - Synchronous, Time-Triggered
  - IEEE 1588 time-sync protocol
- High-speed, high precision
  - Speed: 1 inch/ms
  - Precision: 0.01 inch

    -> Time accuracy: 10us

Orchestrated networked resources built with
sound design principles on **suitable abstractions**

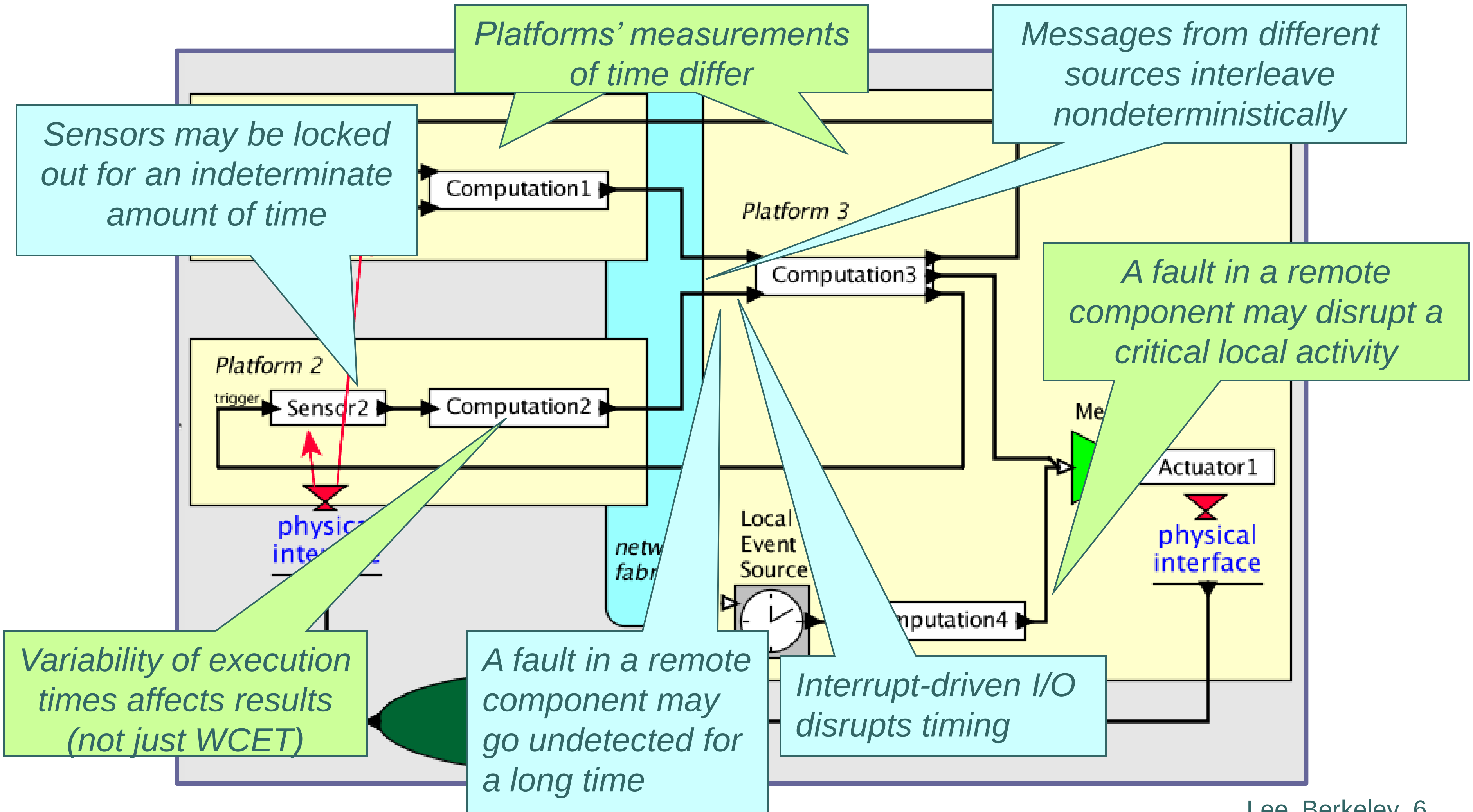
DETERMINISM **TIMED SEMANTICS**

Claim

For CPS, *programs* do not today adequately specify *behavior*.

Structure of a Cyber-Physical System

Problems that complicate analysis of *system* behavior:



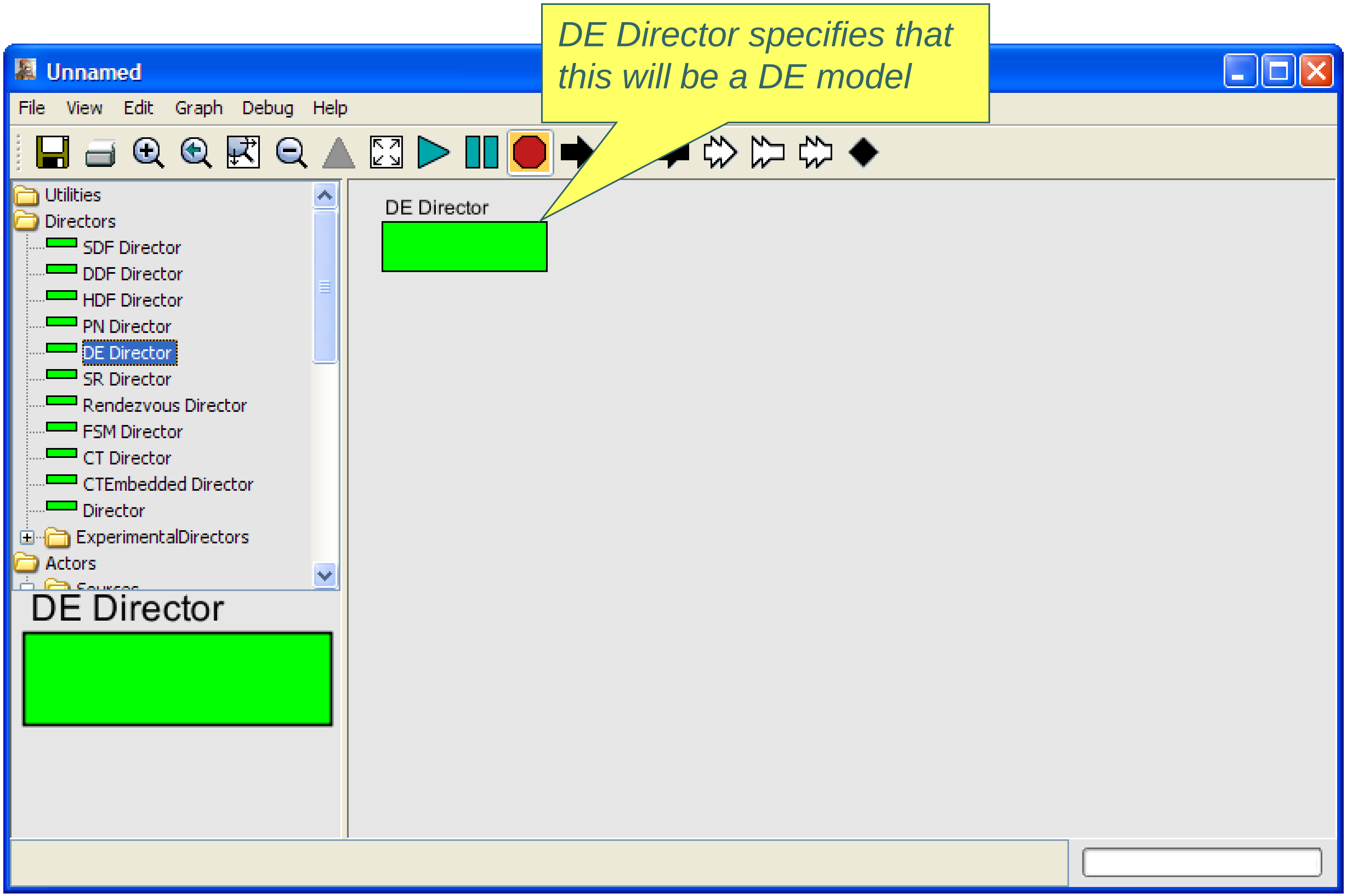
Our Approach is based on Discrete Events (DE)

- Concurrent actors
- Exchange time-stamped messages (“events”)

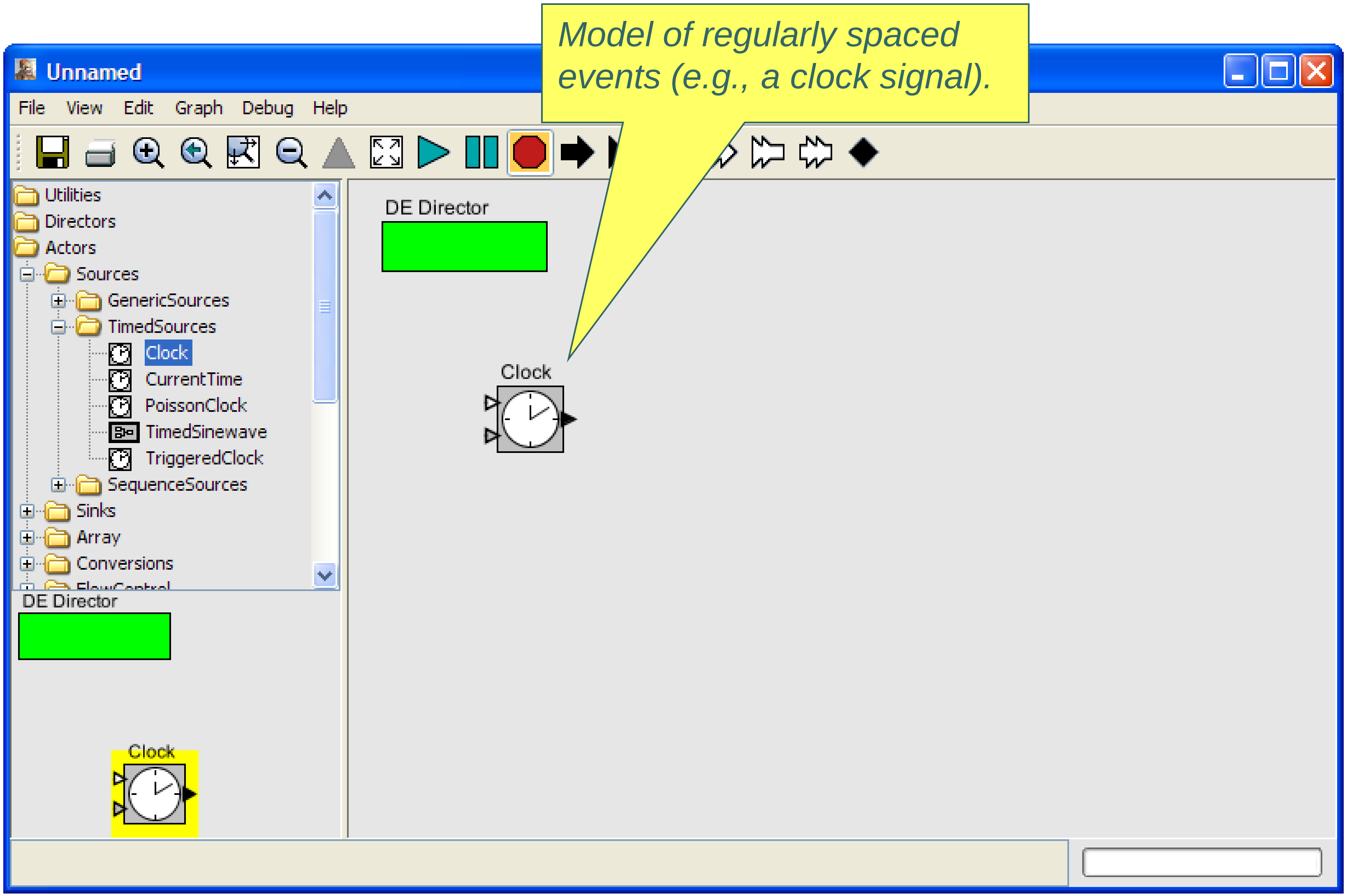
A correct execution is one where every actor reacts to input events in time-stamp order.

Time stamps are in “**model time**,” which typically bears no relationship to “**real time**” (wall-clock time). We use *superdense time* for the time stamps.

Building a DE Model (in Ptolemy II)

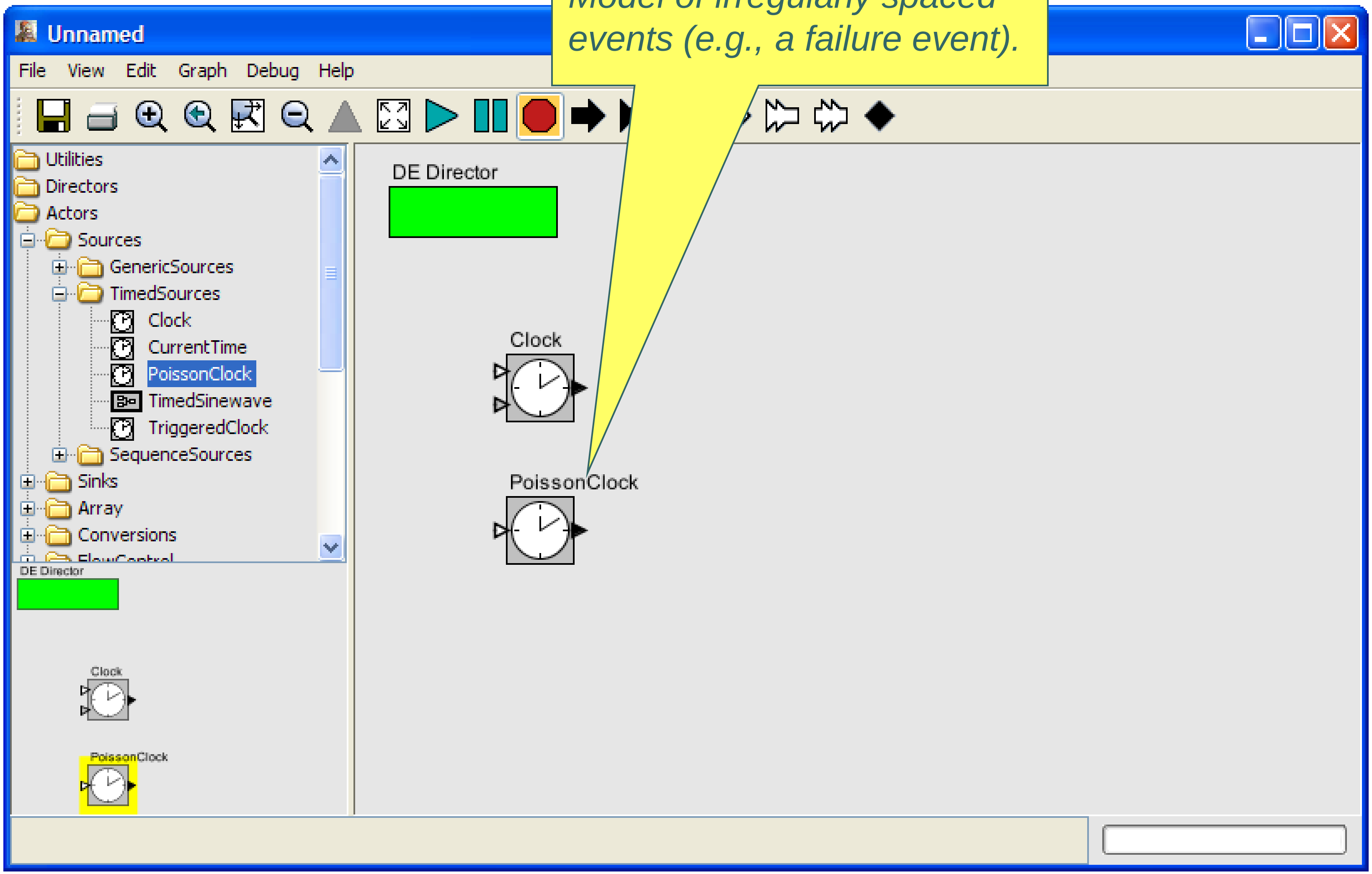


Building a DE Model (in Ptolemy II)



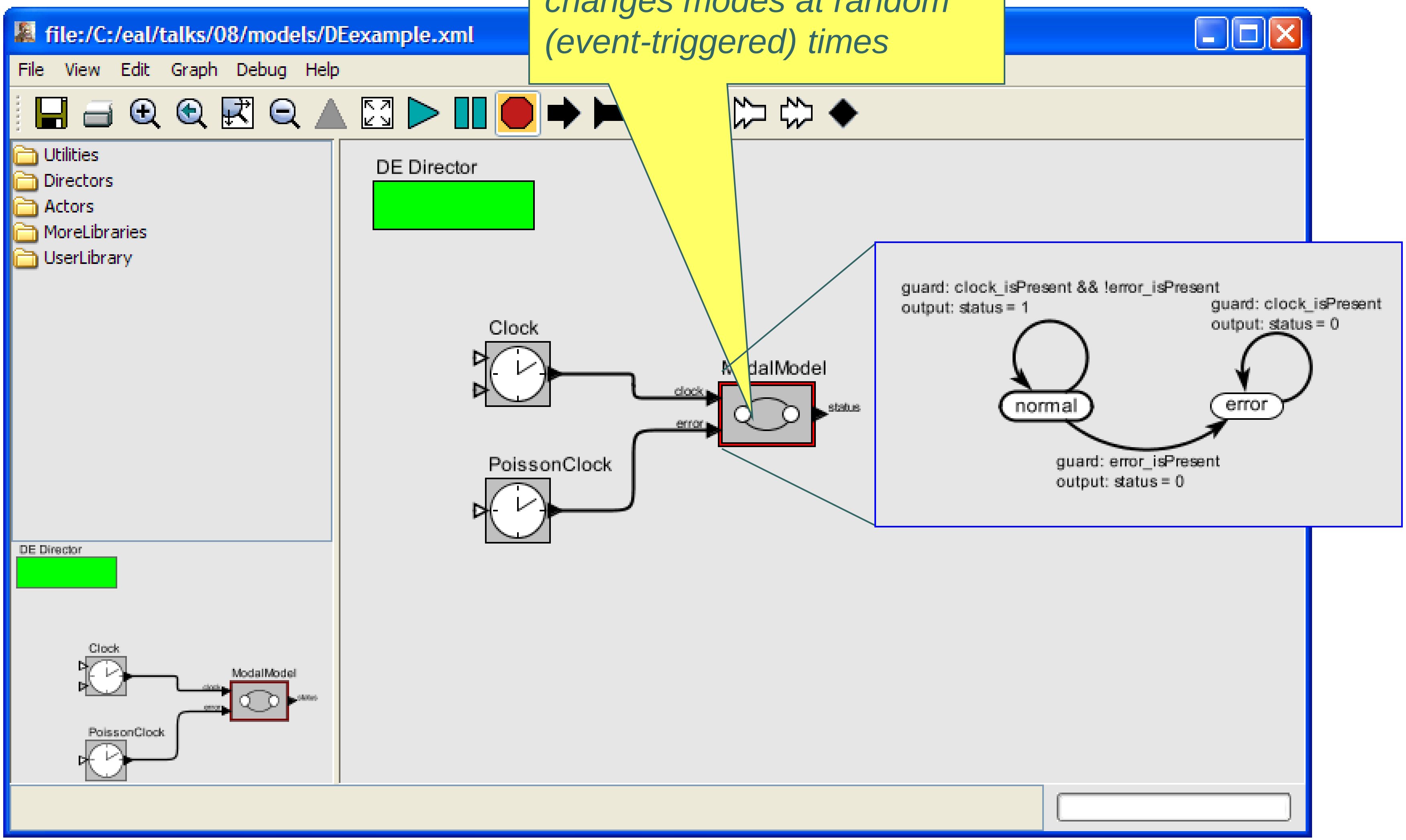
Building a DE Model (in Ptolemy II)

Model of irregularly spaced events (e.g., a failure event).

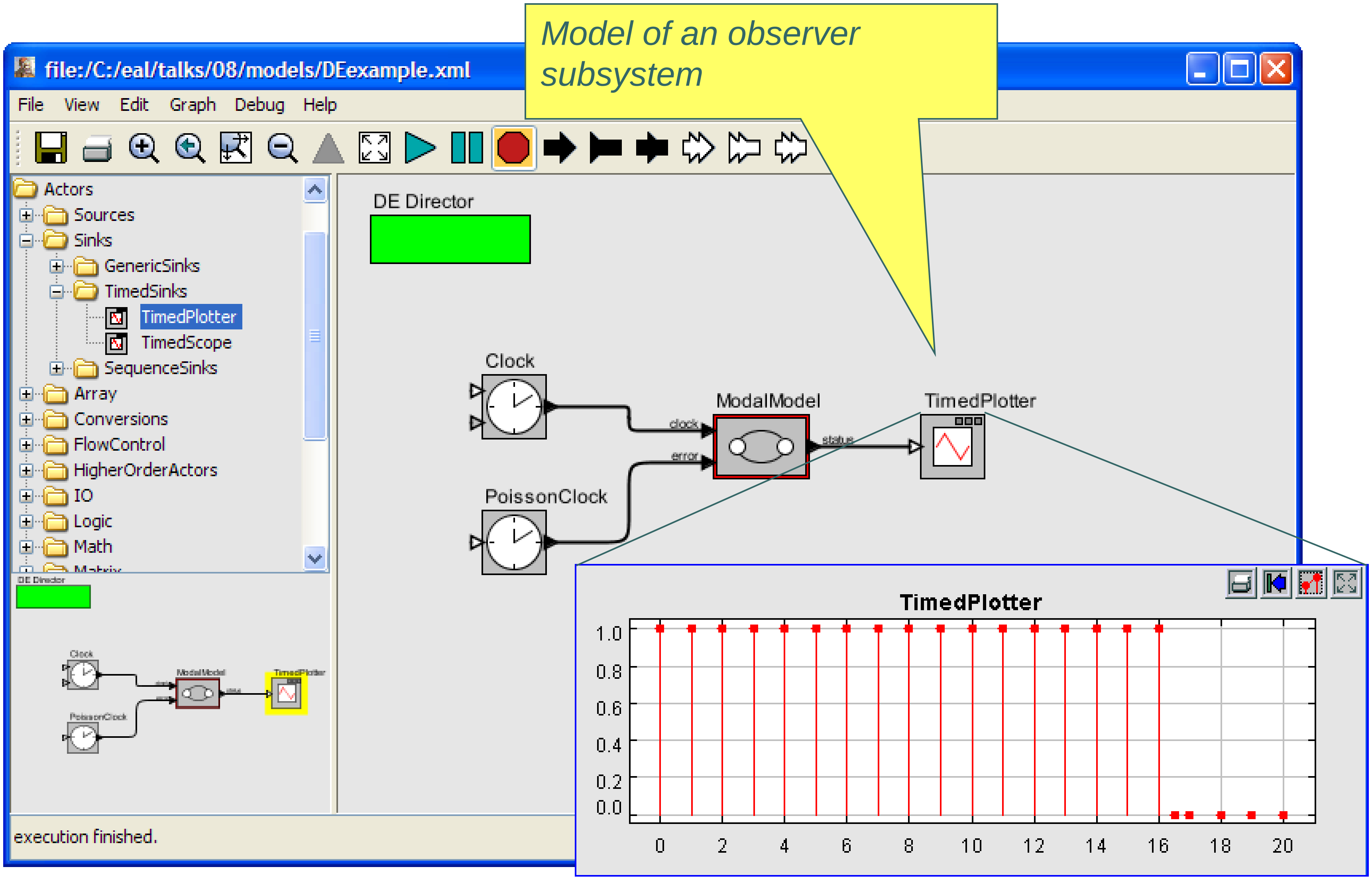


Building a DE Model (in Ptolemy II)

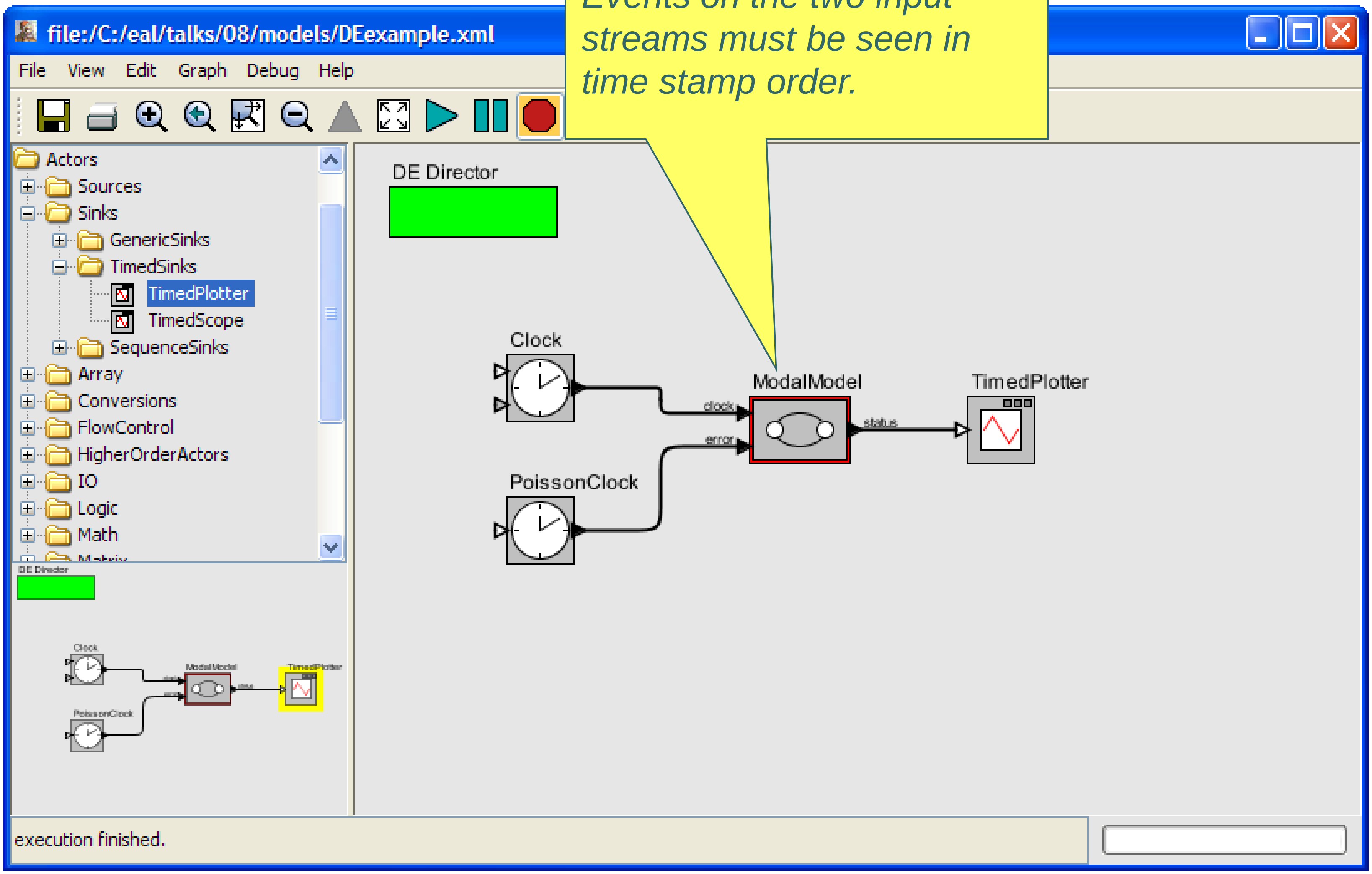
Model of a subsystem that changes modes at random (event-triggered) times



Building a DE Model (in Ptolemy II)



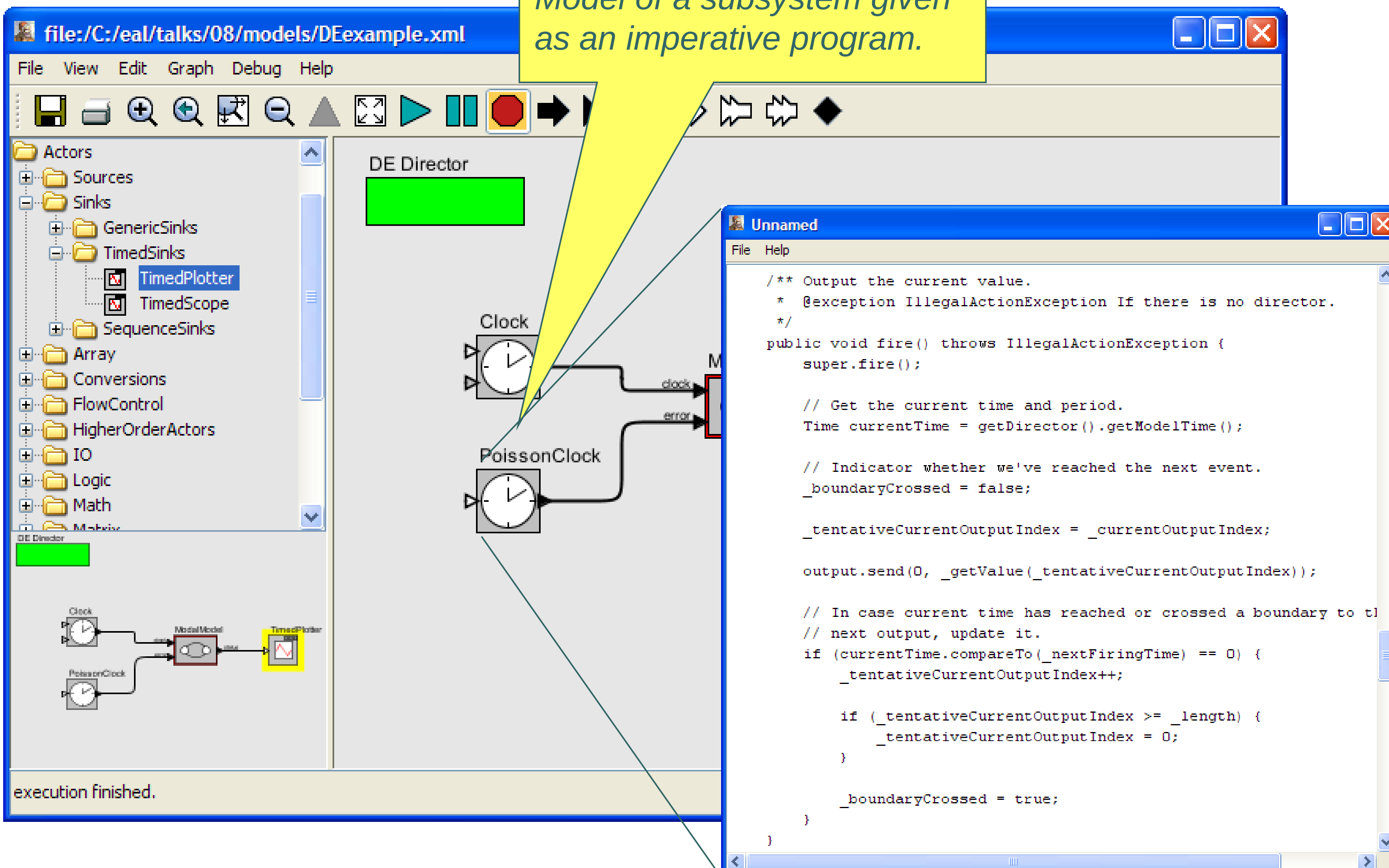
Building a DE Model (in Ptolemy II)



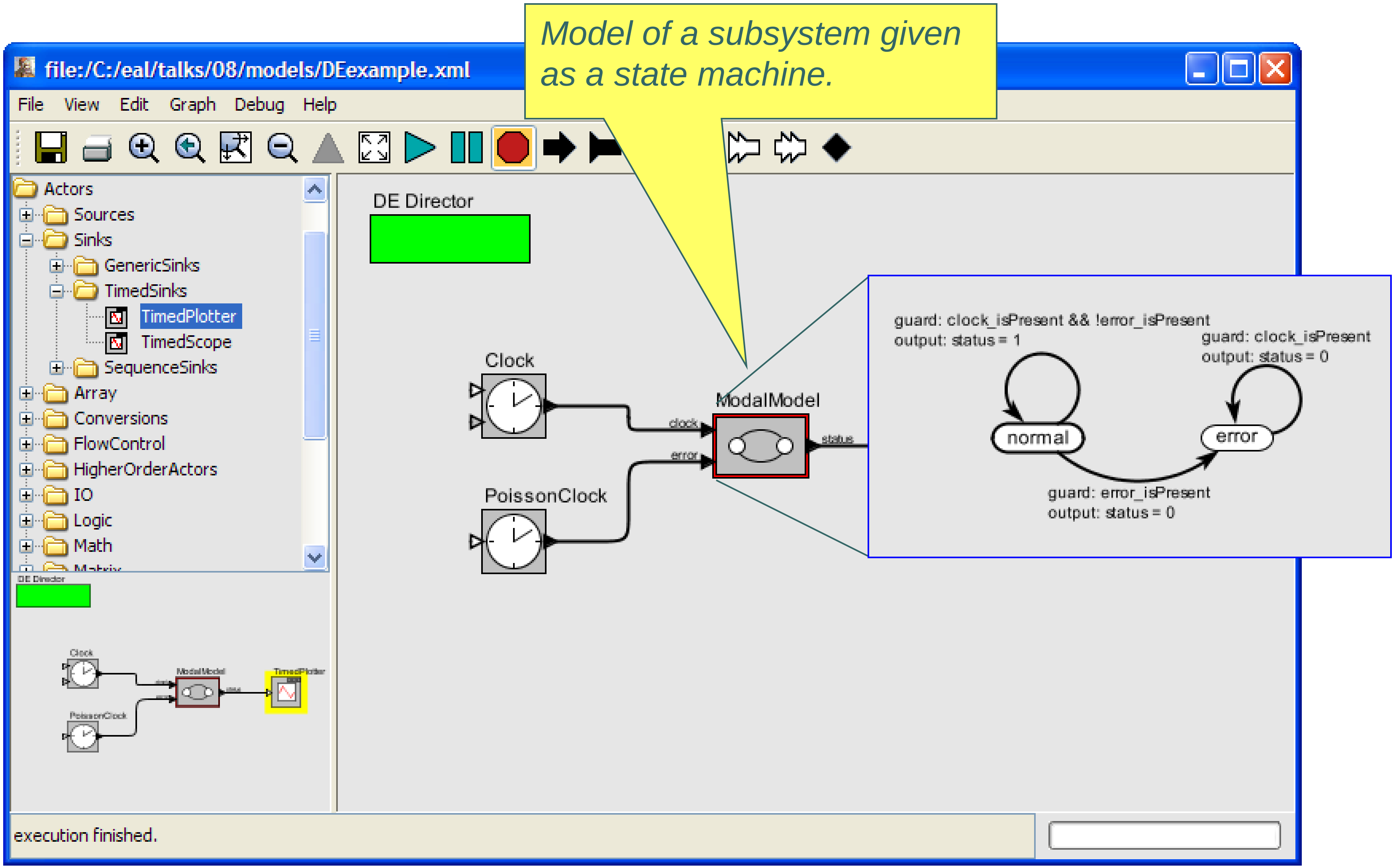
Events on the two input streams must be seen in time stamp order.

This is a Component Technology

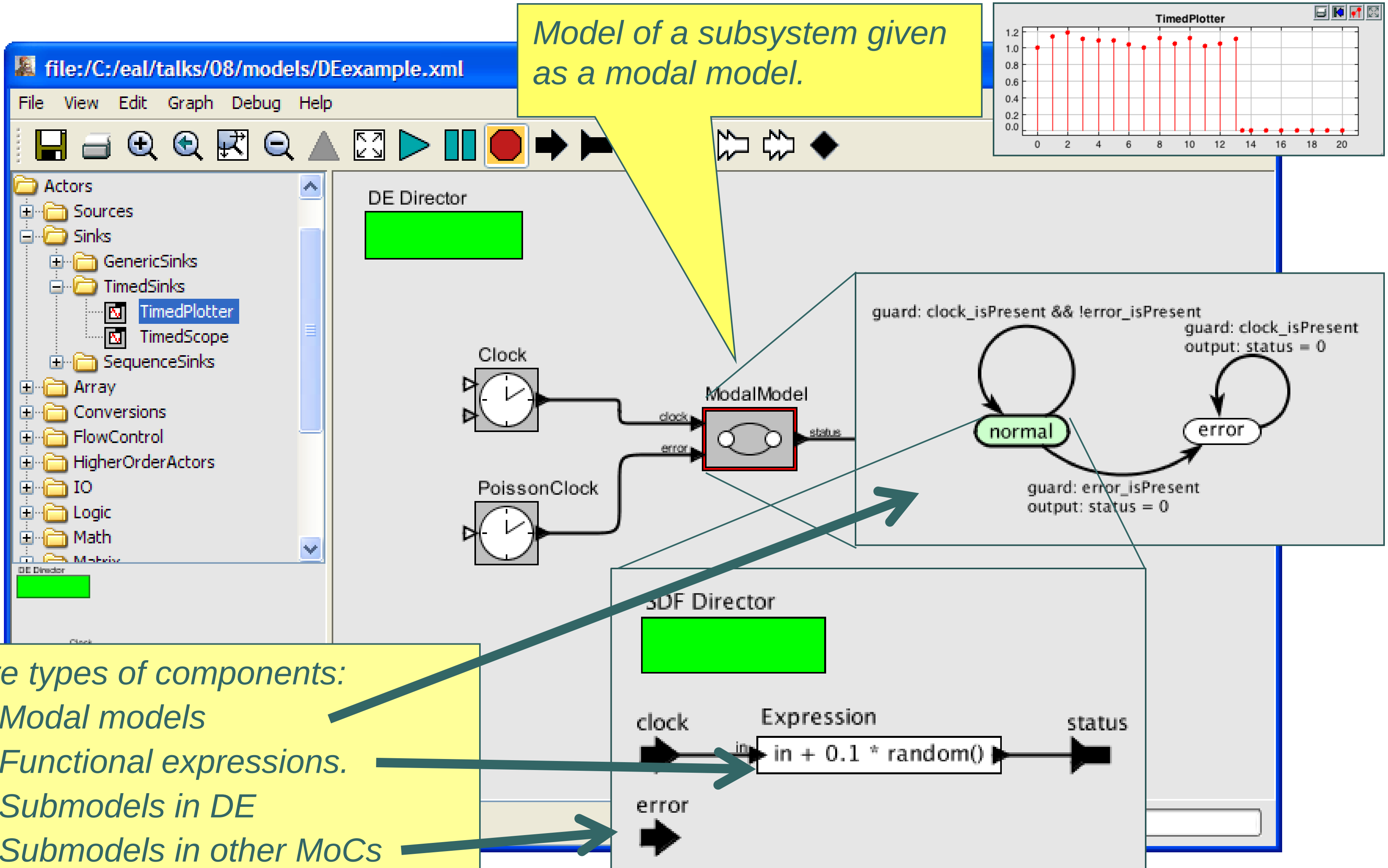
Model of a subsystem given as an imperative program.



This is a Component Technology



This is a Component Technology



More types of components:

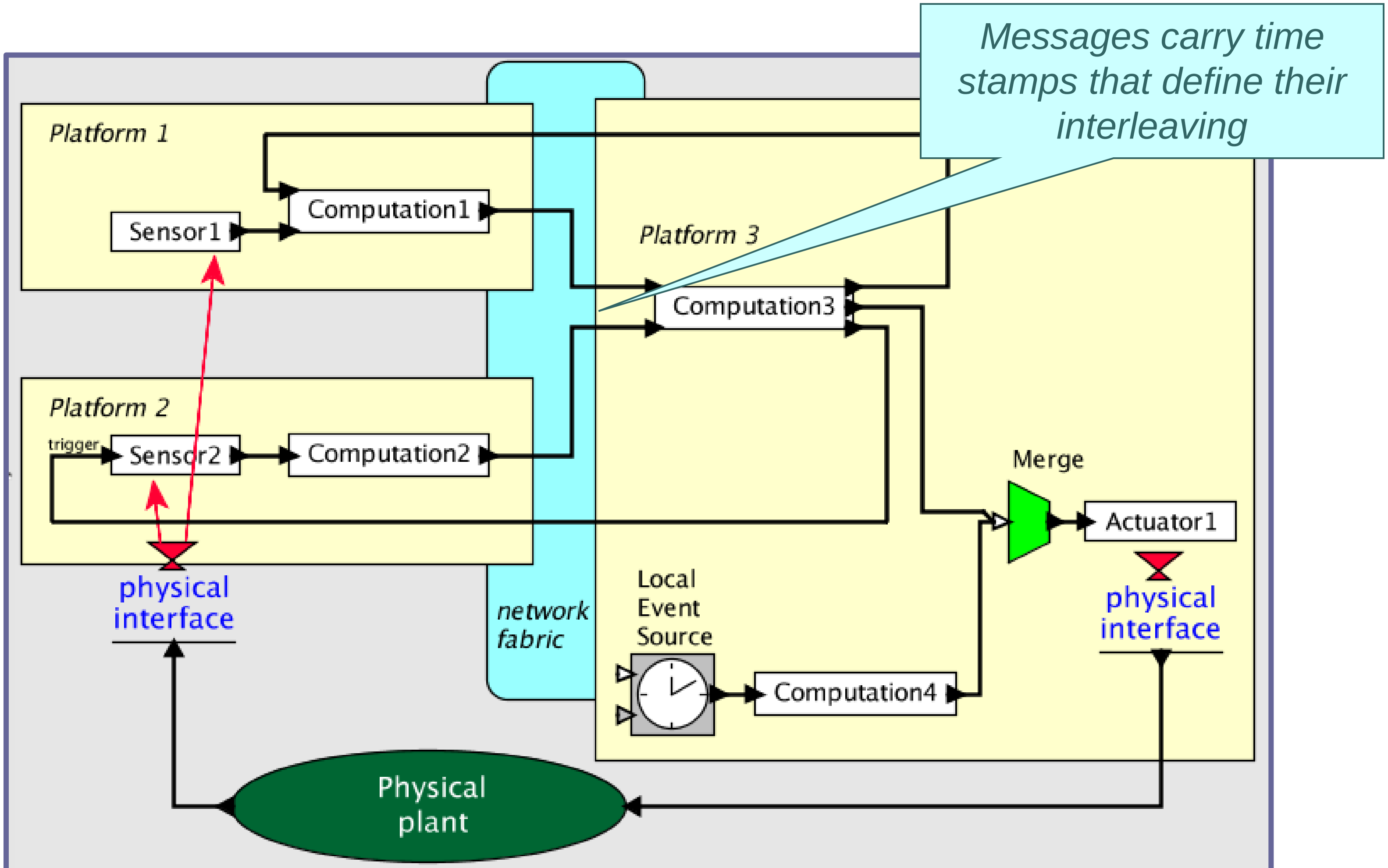
- *Modal models*
- *Functional expressions.*
- *Submodels in DE*
- *Submodels in other MoCs*

Using DE Semantics in Distributed Real-Time Systems

- DE is usually a simulation technology.
- Distributing DE is traditionally done for acceleration.
- Hardware design languages (e.g. VHDL) use DE where time stamps are literally interpreted as real time, or abstractly as ticks of a physical clock.

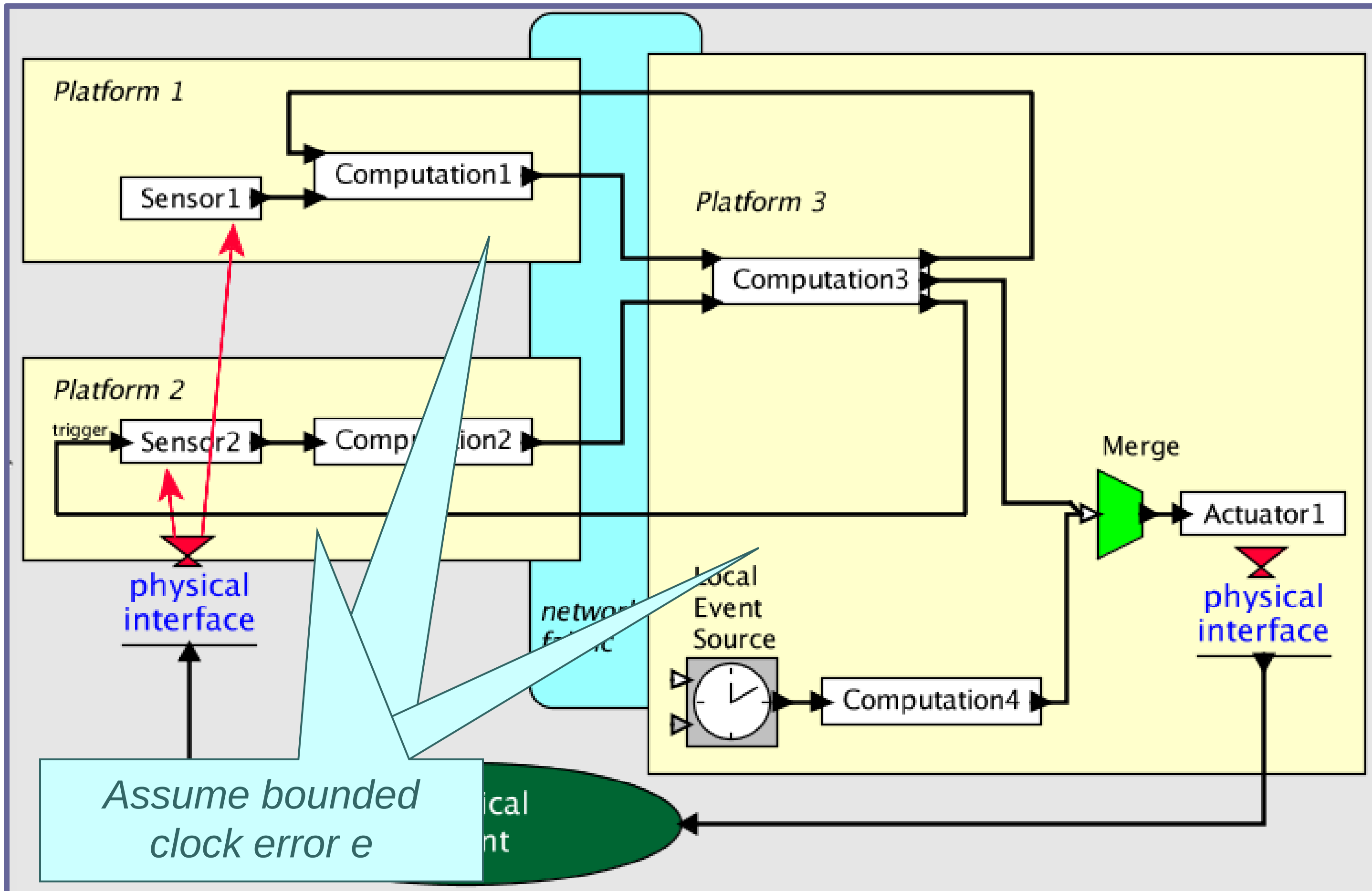
- We are using DE for distributed real-time software, binding time stamps to real time only where necessary.
- *PTIDES*: Programming Temporally Integrated Distributed Embedded Systems

Ptides: First step: Time-stamped messages.



Ptides: Second step: Network time synchronization

GPS, NTP, IEEE 1588, time-triggered busses, etc., all provide some form of common time base. These are becoming fairly common.



Time-Aware Networking Technology Facilitates Network Time Synchronization

- Frequency locking
  - E.g., **synchronous ethernet**: ITU-T G.8261, May 2006
  - Enables integrating circuit-switched services on packet-switched networks
  - Can deliver performance independent of network loading.

Press Release

Zarlink Semiconductor Corp.

Release date: January 31, 2007

Zarlink and Marvell® First to Demonstrate Synchronous Ethernet Solution Supporting Network-Quality Performance

Companies demonstrate synchronization over Ethernet physical layer using Zarlink PLL (phase locked-loop) and Marvell Ethernet PHY technologies

OTTAWA, Jan. 31 /- Zarlink Semiconductor (NYSE/TSX:ZL) and Marvell® (NASDAQ:MRVL) today announced the successful demonstration of a synchronous Ethernet solution using already available products from both companies that will allow carriers to support real-time services over packet-based networks.

- Time synchronization
  - E.g., **IEEE 1588** standard set in 2002.
  - Synchronized time-of-day across a network.

Precision Time Protocol (PTP) Standardized for Ethernet

Press Release October 1, 2007

National Semiconductor
The Sight & Sound of Information

NEWS RELEASE

For More Information Contact

Media Contact
Naomi Mitchell
National Semiconductor
(408) 721-2142
naomi.mitchell@nsc.com

Reader Information
Design Support Group
(800) 272-9959
www.national.com

Industry's First Ethernet Transceiver with IEEE 1588 PTP Hardware Support from National Semiconductor Delivers Outstanding Clock Accuracy

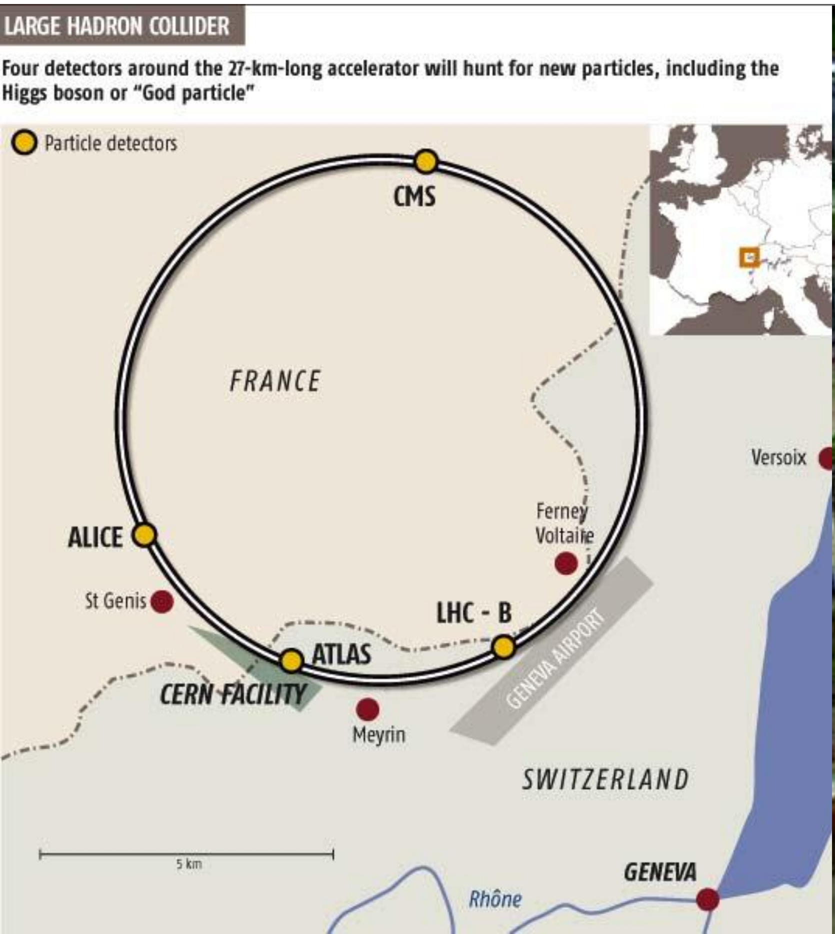
Using DP83640, Designers May Choose Any Microcontroller, FPGA or ASIC to Achieve 8- Nanosecond Precision with Maximum System Flexibility

This may become routine!

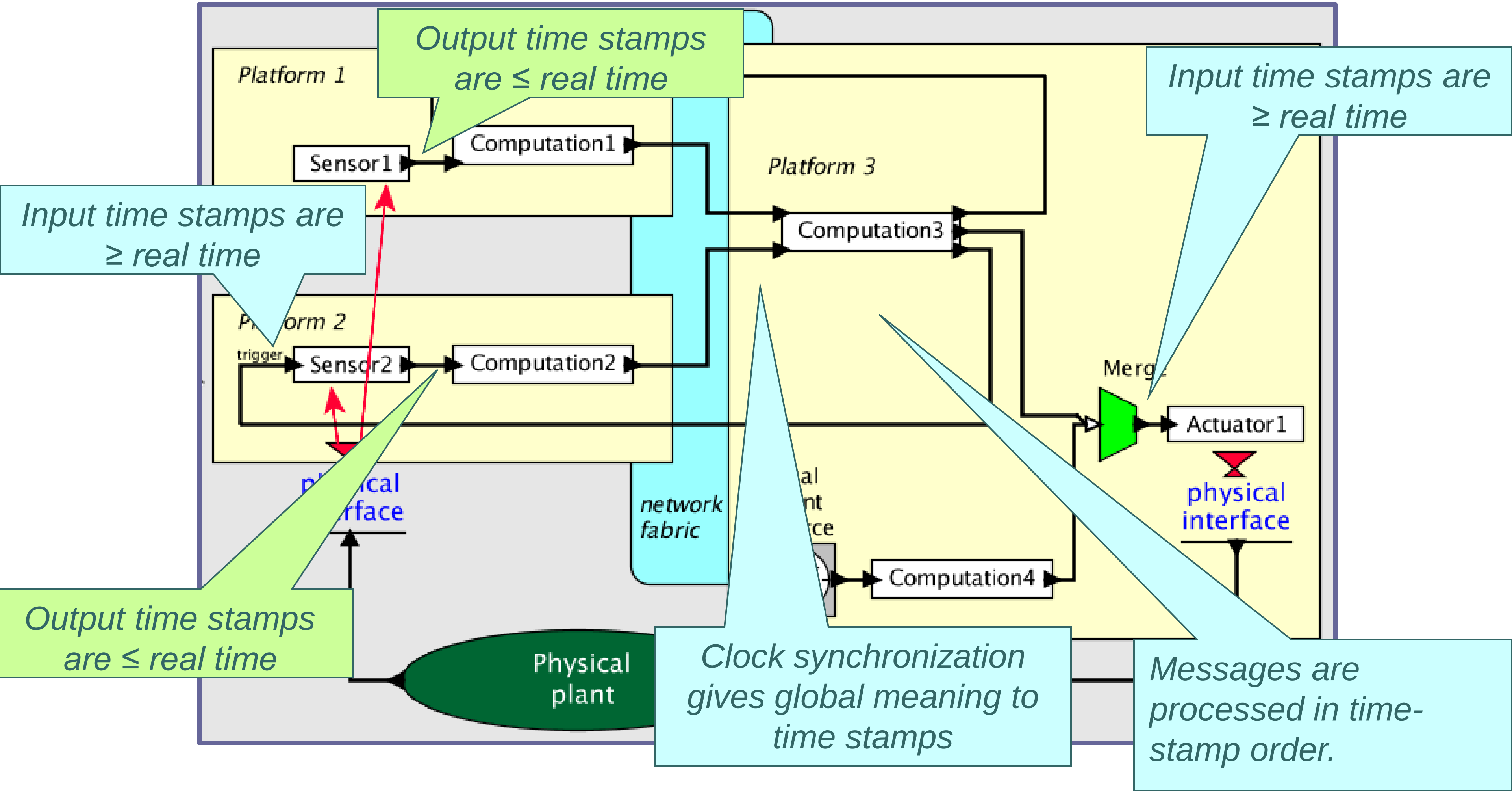
With this PHY, clocks on a LAN agree on the current time of day to within 8ns, far more precise than older techniques like NTP.

An Extreme Example: The Large Hadron Collider

The WhiteRabbit project at CERN is synchronizing the clocks of computers 10 km apart to within about 80 psec using a combination of IEEE 1588 PTP and synchronous ethernet.

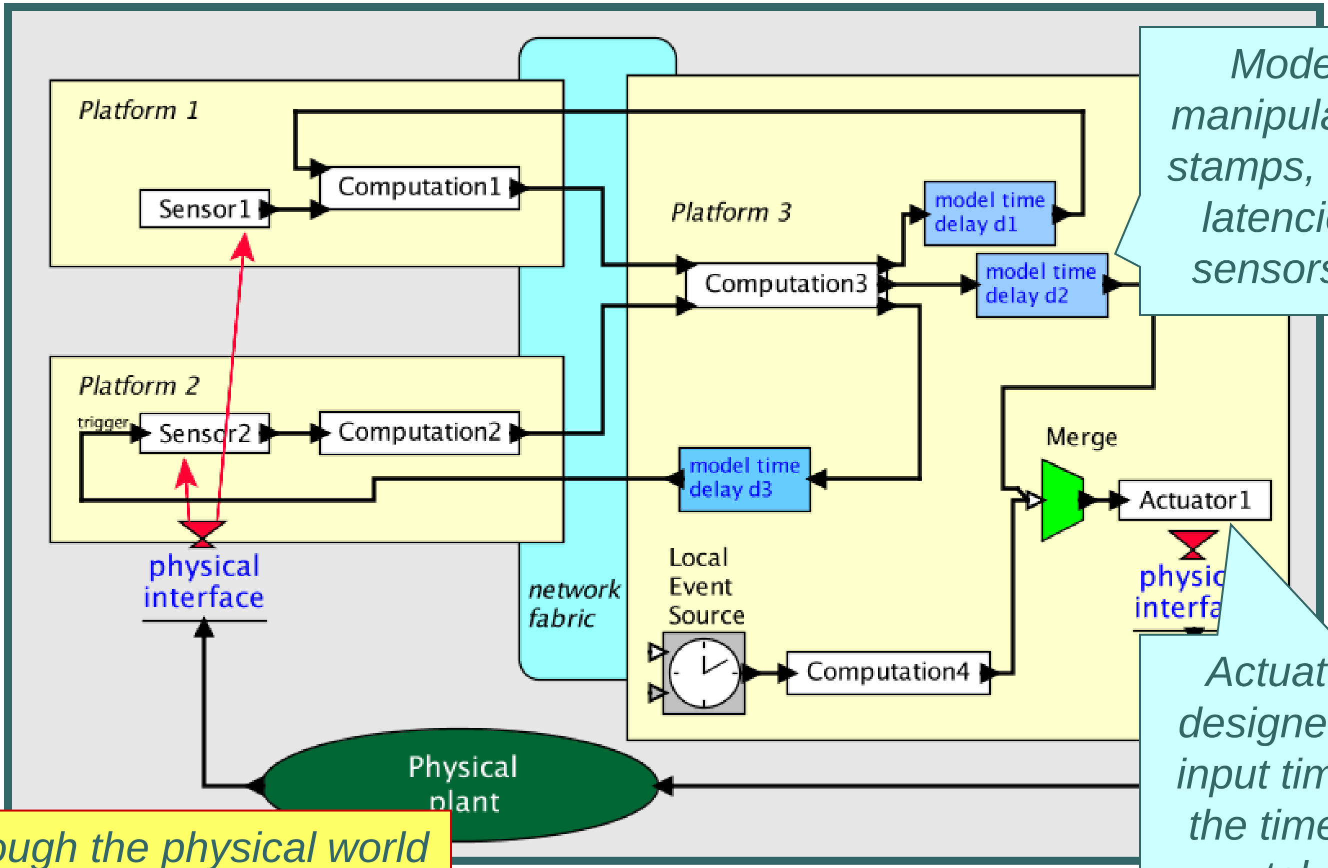


Ptides: Third step: Bind time stamps to real time at sensors and actuators



Ptides: Fourth step:
Specify latencies in the model

Global latencies between sensors and actuators become controllable, which enables analysis of system dynamics.



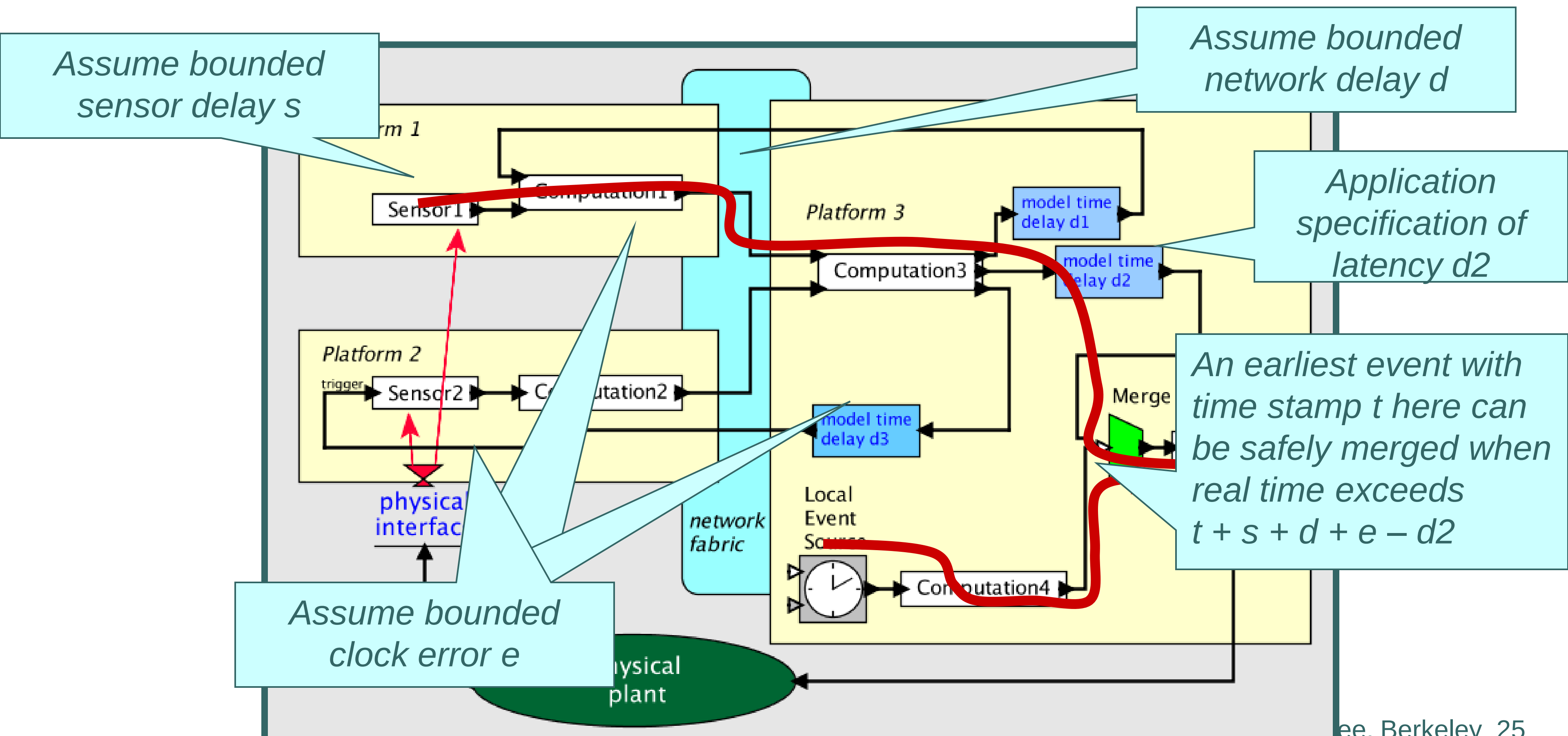
Model includes manipulations of time stamps, which control latencies between sensors and actors

Actuators may be designed to interpret input time stamps as the time at which to take action.

Feedback through the physical world

Ptides: Fifth step
Safe-to-process analysis (ensures determinacy)

Safe-to-process analysis guarantees that the generated code obeys time-stamp semantics (events are processed in time-stamp order), given some assumptions.



Delivering Bounded Network Delay

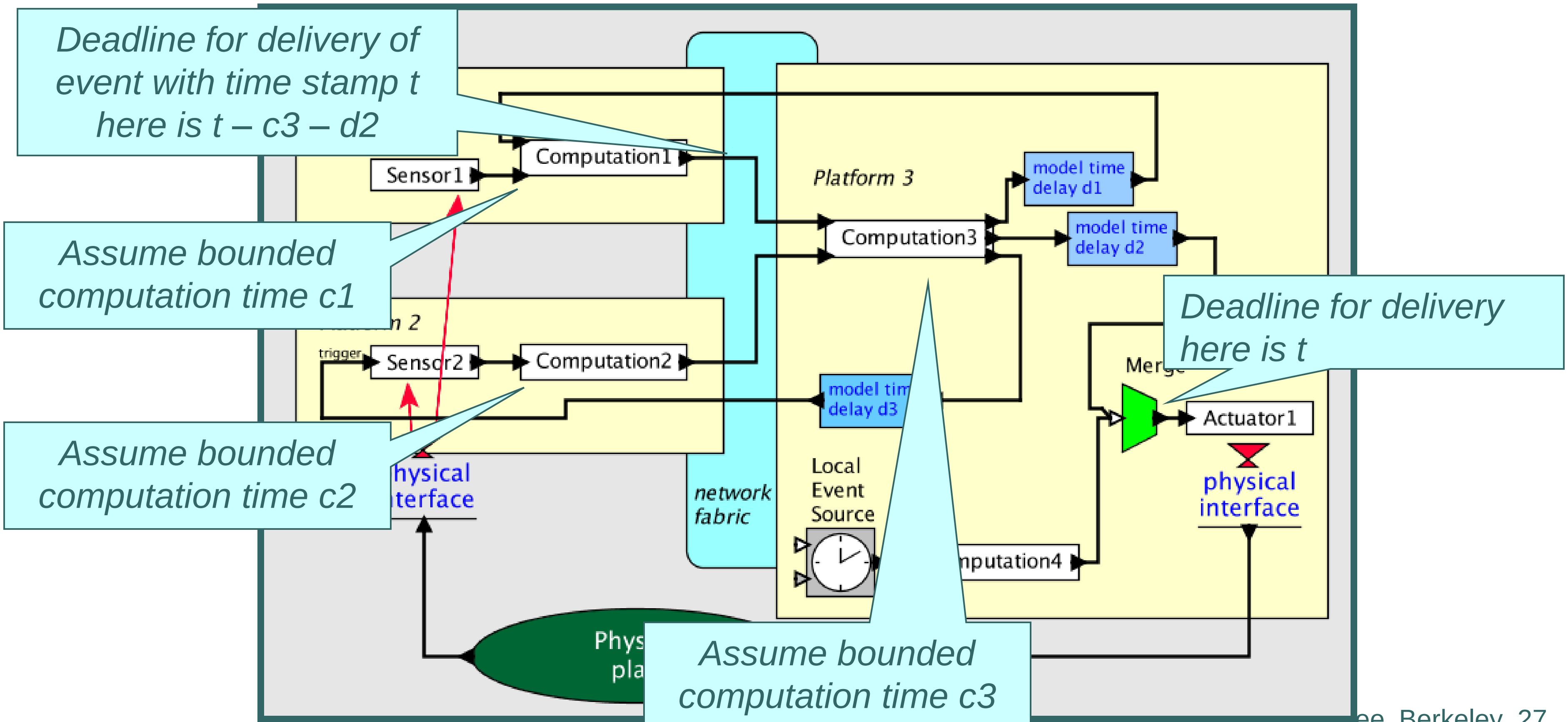
Domain-Specific Networks capable of Bounded Delay

- **WorldFIP** (Factory Instrumentation Protocol)
  - Created in France, 1980s, used in train systems
- **CAN**: Controller Area Network
  - Created by Bosch, 1980s/90s, ISO standard
- Various **ethernet** variants
  - PROFInet, EtherCAT, Powerlink, …
- **TTP/C**: Time-Triggered Protocol
  - Created around 1990, TU Vienna, supported by TTTech
- **MOST**: Media Oriented Systems Transport
  - Created by a consortium of automotive & electronics companies
  - Under active development today
- **FlexRay**: Time triggered bus for automotive applications
  - Created by a consortium of automotive & electronics companies
  - Under active development today

Ptides Schedulability Analysis

Determine whether *deadlines* can be met

Schedulability analysis incorporates computation times to determine whether we can guarantee that deadlines are met.



PtidyOS: A lightweight microkernel supporting Ptides semantics

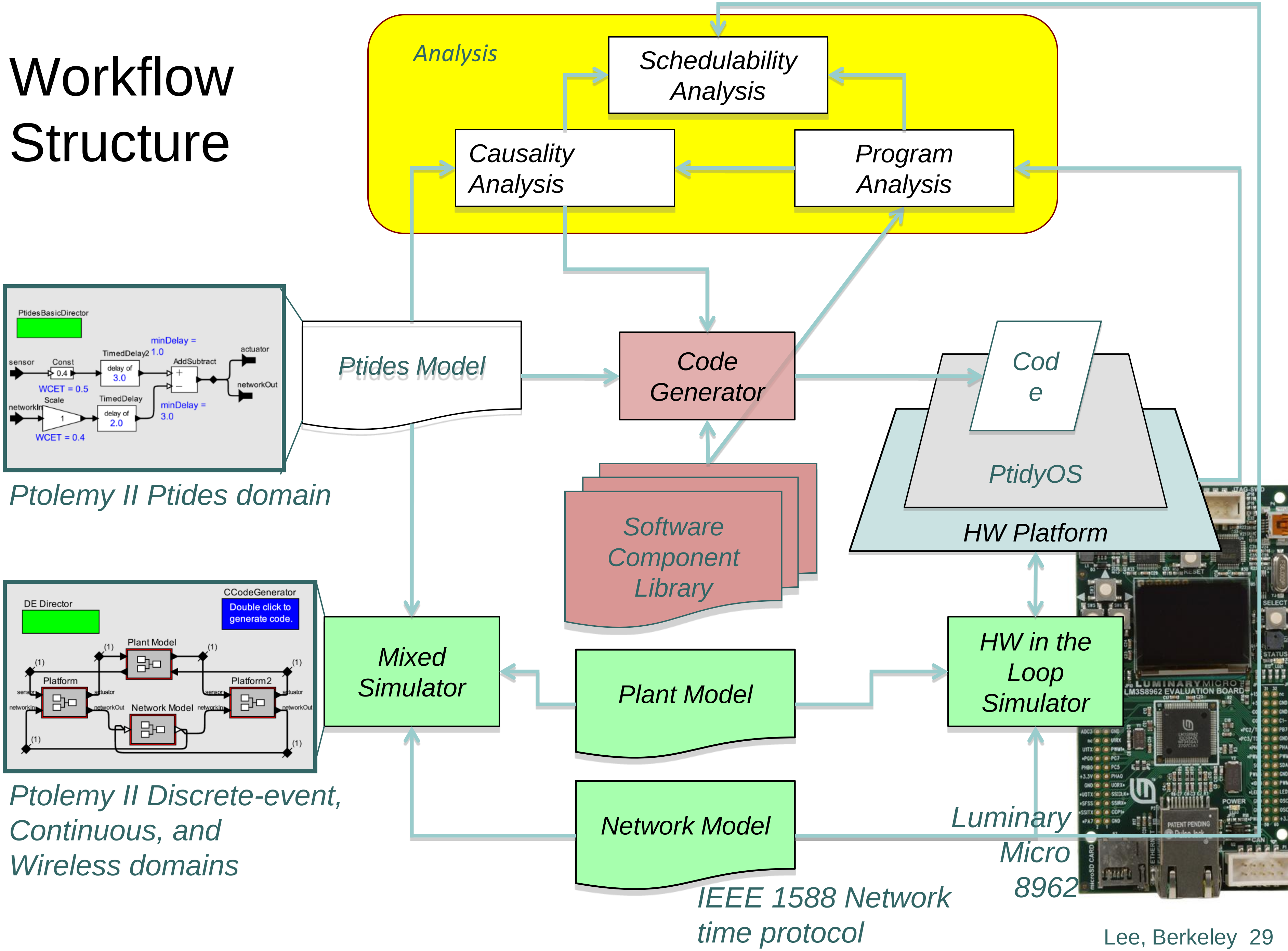
Current prototype runs on a COTS Arm platform (Luminary Micro) with rudimentary support for IEEE 1588 network time synchronization. Occupies about 16 kbytes of memory.

Currently porting to Renesas and PRET platforms.

An interesting property of PtidyOS is that despite being highly concurrent, preemptive, and EDF-based, it does not require threads.
A single stack is sufficient!

Luminary Micro 8962

Workflow Structure

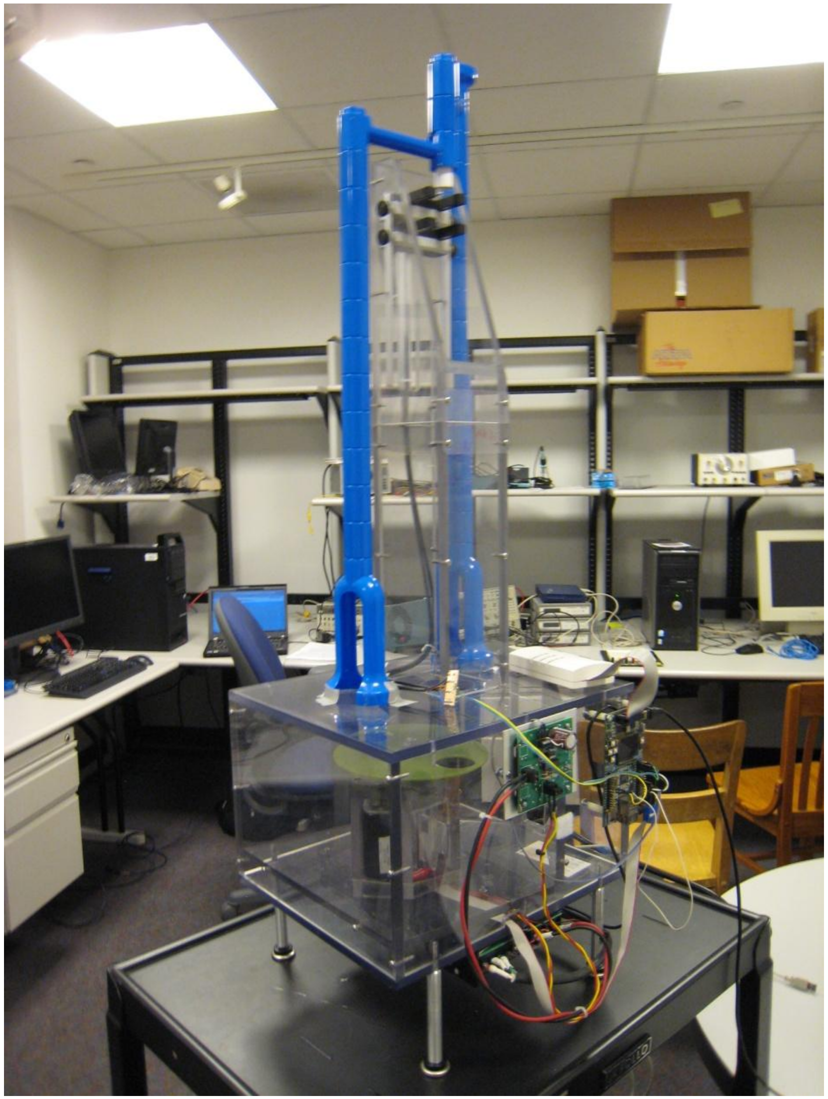


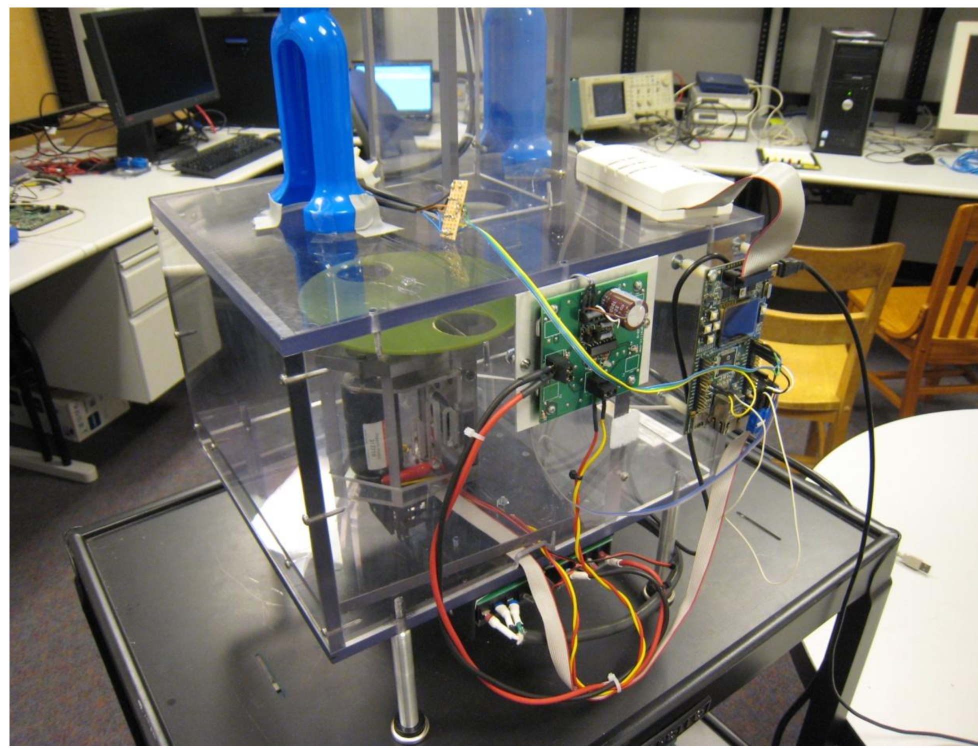
A Test Case for PtidyOS

This device, designed by Jeff Jensen, mixes periodic, quasi-periodic, and sporadic real-time events.

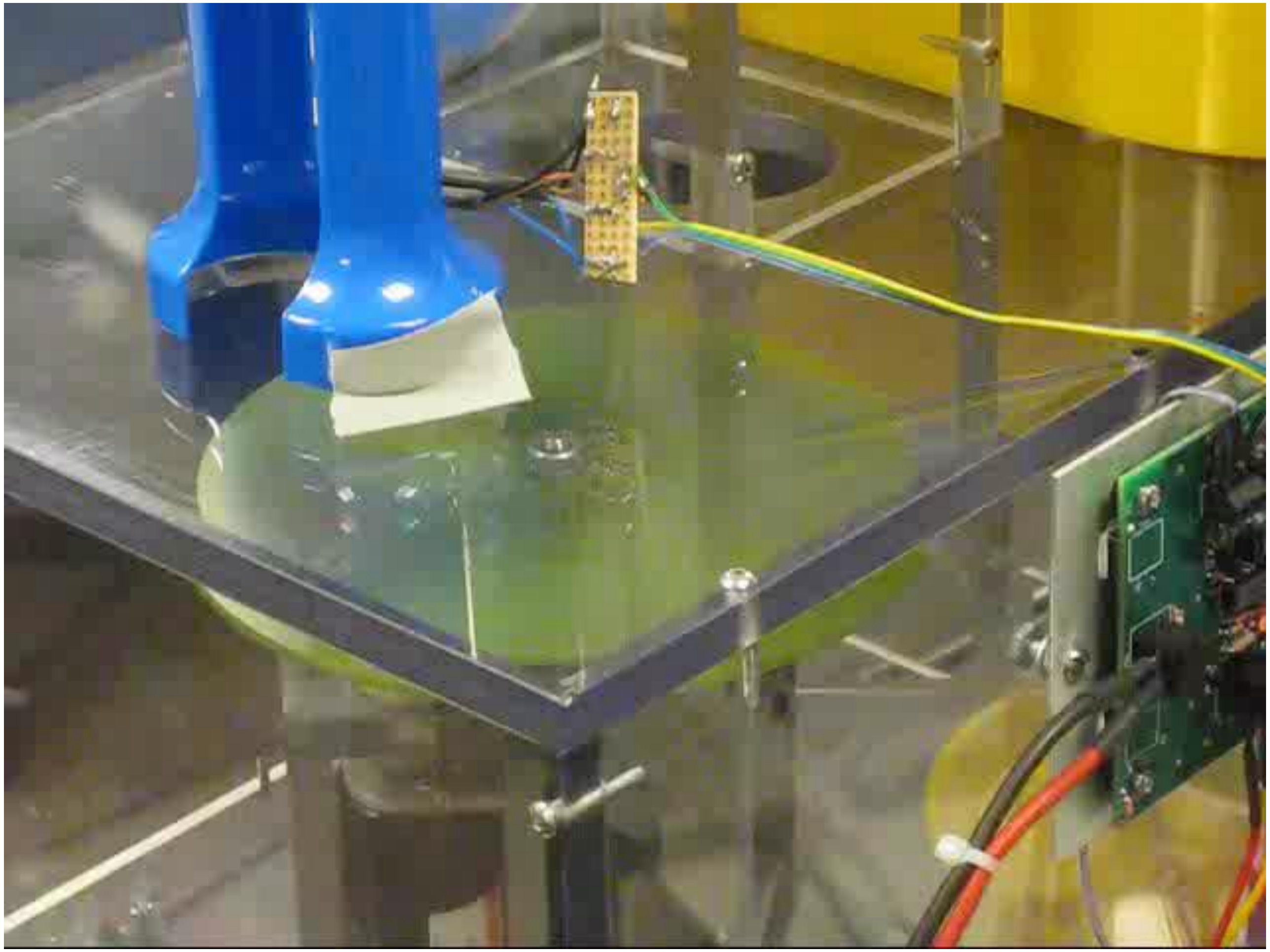
Tunneling Ball Device

- *sense ball*
- *track disk*
- *adjust trajectory*

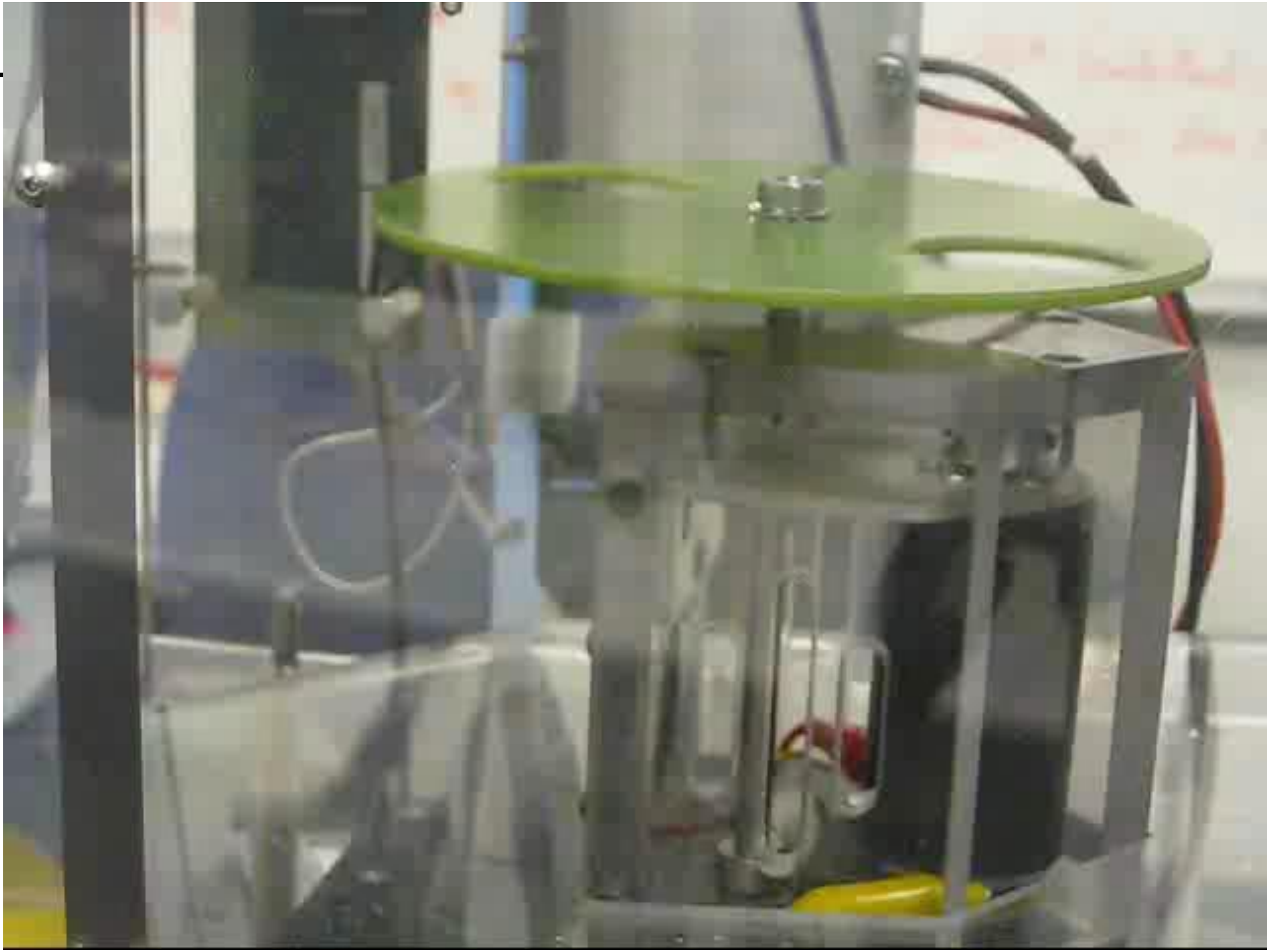




Tunneling Ball Device in Action

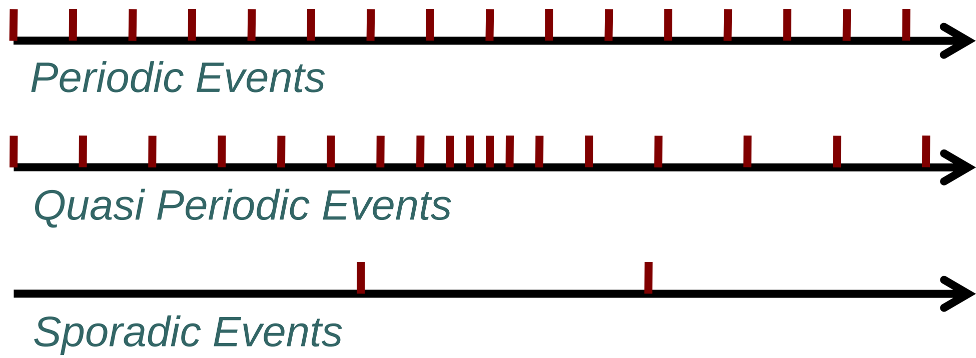


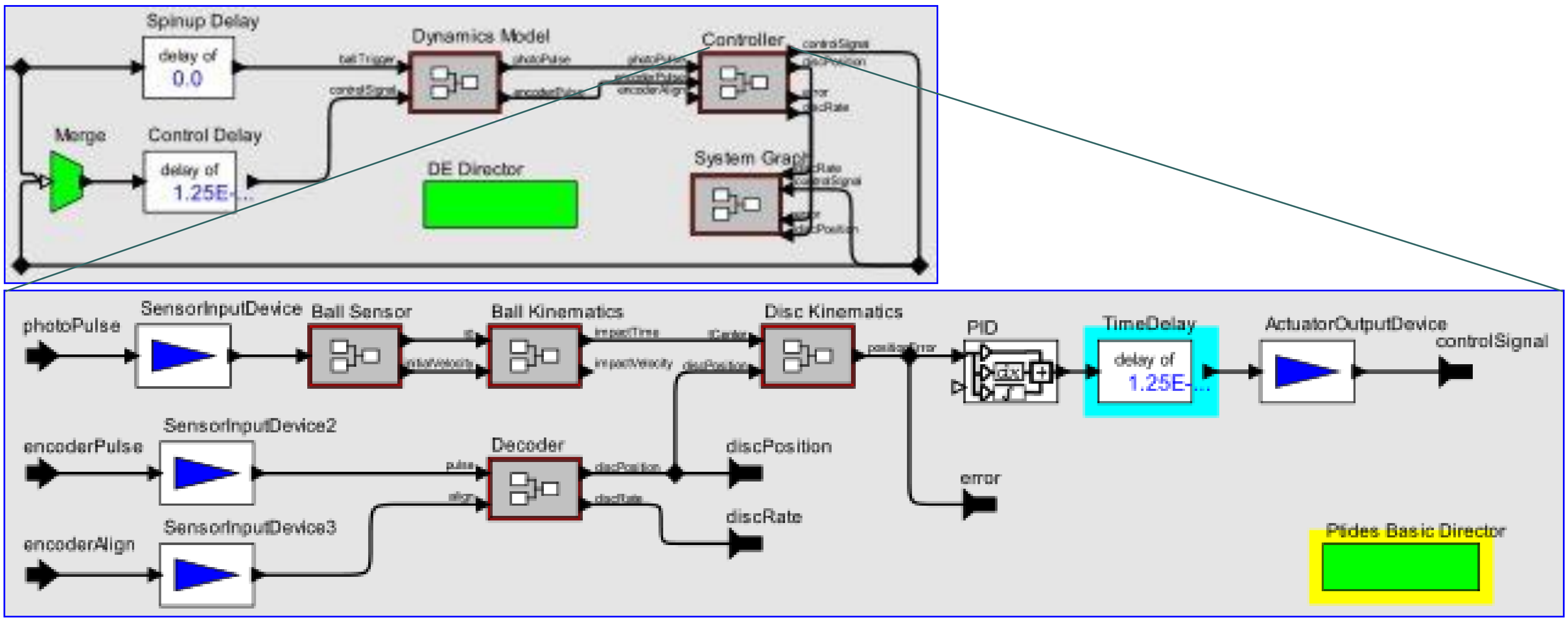
T



Tunneling Ball Device

Mixed event sequences





Ptides Project Status

- Seed funding from ARL got the project going.
- Ongoing NSF effort (CPS Medium)
  - Sanjit Seshia focused on WCET & schedulability analysis
  - Ptolemy II-based simulator supports multiform clocks
  - PtidyOS being prepped for open-source release

Ptides Publications

- Y. Zhao, J. Liu, E. A. Lee, "**A Programming Model for Time-Synchronized Distributed Real-Time Systems**," RTAS 2007.
- T. H. Feng and E. A. Lee, "**Real-Time Distributed Discrete-Event Execution with Fault Tolerance**," RTAS 2008.
- P. Derler, E. A. Lee, and S. Matic, "**Simulation and implementation of the ptides programming model**," DS-RT 2008.
- J. Zou, S. Matic, E. A. Lee, T. H. Feng, and P. Derler, "**Execution strategies for Ptides, a programming model for distributed embedded systems**," RTAS 2009.
- J. Zou, J. Auerbach, D. F. Bacon, E. A. Lee, "**PTIDES on Flexible Task Graph: Real-Time Embedded System Building from Theory to Practice**," LCTES 2009.
- J. C. Eidson, E. A. Lee, S. Matic, S. A. Seshia and J. Zou, "**Time-centric Models For Designing Embedded Cyber-physical Systems**," ACES-MB 2010.

Overview Reference:
E. A. Lee. Computing needs time. CACM, 52(5):70–79, 2009

Conclusions

Today, timing behavior is a property only of *realizations* of software systems.

Tomorrow, timing behavior will be a semantic property of *programs* and *models*.

Raffaello Sanzio da Urbino – The Athens School