MODELING DISTRIBUTED REAL-TIME SYSTEMS WITH PTOLEMY II

Patricia Derler, Jia Zou, Slobodan Matic, John Eidson, Edward A. Lee

University of California, Berkeley
DISTRIBUTED REAL-TIME SYSTEMS

Multiple computers, comprising of sensors and actuators, connected on a network that act and react on events to meet timing constraints.

Automotive

Telecommunications

Military systems

Factory automation

Building Systems

Transportation (Air traffic control at SFO)

Instrumentation (Soleil Synchrotron)
MODELING DISTRIBUTED REAL-TIME SYSTEMS
OVERVIEW

• Challenges: How to model
  • Time
  • Network
  • Execution time
  • Execution semantics

• Address modeling challenges in PTIDES
Distributed platforms have different notions of time

Platform clocks drift

Platform clocks drift at varying rates
MODELING DISTRIBUTED SYSTEMS

Director mediates between actors
Difficult to maintain different notions of time

One hierarchy level is not enough
MODELING DISTRIBUTED SYSTEMS

Hierarchies:
- Opaque composite actors
- Embedded directors maintain time
MODELING DISTRIBUTED SYSTEMS

Top level: Oracle time

Every platform time is defined with respect to oracle time

Oracle time = \( t_0 \)

Platform1 time = \( f_1(t_0) \)

Platform2 time = \( f_2(t_0) \)

Platform3 time = \( f_3(t_0) \)
CLOCK SYNCHRONIZATION

adjust clock rate:

adjust clock value:
MODELING NETWORKS

Distributed platforms communicate via networks

Networks have latencies

e.g. CAN Bus, TTEthernet
Physical connections vs. Logical connections

Logical connections are lost
MODELING NETWORKS

Aspect-oriented modeling

Quantity managers [Balarin03] and schedulers to simulate network latency

MODELING EXECUTION TIME

Platform 1

Sensor1 → Computation1

execution time

Platform 2

Sensor2 → Computation2

physical interface

physical interface

Merge → Actuator1

Local Event Source

Computation3

Network fabric

execution time

execution time

Physical plant

execution time

execution time
How is execution time computed?

Which time line to use for specifying execution time?

Aspect-oriented programming
Discrete-Event (DE) for simulation

DE as a application specification language which serves as a semantic basis for obtaining determinism in distributed real-time systems.
DISTRIBUTED DISCRETE-EVENT MODELS

Logical time describe the execution semantics

New time line: logical time

Oracle time

Platform time

Logical time

Platform1 time
- Logical time

Platform3 time
- Logical time

Platform2 time
- Logical time

Physical Plant

Oracle time

f(t)
PTIDES: AN APPLICATION

Programming temporally integrated distributed event systems [Zhao07]

- Discrete event model for execution
- Relates logical time to platform time whenever necessary
- Requires bounded error between platform clocks: Relies on clock synchronization
- Events are processed in time-stamped order

A PTIDES MODEL

- Increase time stamp
- Send time stamp and value and over network
- Time stamp ≥ platform time
- Platform time progresses
- Time stamp ≤ platform time
- Execution time ≤ logical time delay

Diagram:

Platform 1
- Sensor 1
- Model time delay d4
- Computation

Platform 2
- Sensor 2
- Model time delay d5
- Computation 2
- Trigger
- Physical interface

Platform 3
- Model time delay d1
- Computation
- Model time delay d2
- Computation
- Model time delay d3
- Merging
- Physical interface

Local Event Source
- Clock
- Computation 4

Physical Plant
SUMMARY

• Distributed embedded systems

• Each distributed platform has its own notion of time

• Modeling distributed systems with different notions of time and clock drifts

• Clock synchronization

• Modeling networks

• Modeling distributed discrete event systems

• PTIDES