DCOSS 2006

PTIDES: Programming Temporally Integrated Distributed Embedded Systems

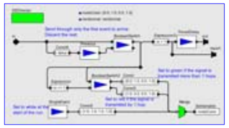
Yang Zhao*, Edward Lee*, Jie Liu† (*EECS, UC Berkeley, †Microsoft Research)

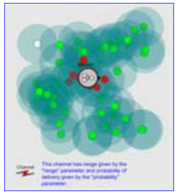
This work extends discrete-event models with the capability of mapping certain events to physical time and proposes them as a programming model, called PTIDES. We seek analysis tools and execution strategies that can preserve the deterministic behaviors specified in DE models without paying the penalty of totally ordered executions.

Discrete-Event (DE) Systems

- Typically used for modeling physical systems where atomic events occur on a time line. Examples:
  - VHDL
  - OPNET Modeler
  - NS-2
  - VisualSense

model of a sensor node





- Time is only a modeling property, DE systems are primarily used in performance modeling and simulation.

Time Synchronization

- Provides a convenient coordination mechanism for coordinated actions over distances.
  - NTP (standard networks, ~ms)
  - IEEE1588 (Ethernet, ~ns)
  - RBS (wireless network)
- A key question that arises in the face of such technologies is how they can change the way software is developed.

PTIDES

- Uses model time to define execution semantics, and constraints that bind certain model time events to physical time.
  - PTIDES programs are constructed as networks of actors.
  - The interface of actors contains ports.
  - Designate a subset of the input ports to be real-time ports. Time-stamped events must be delivered to these ports before physical-time exceeds the time stamp.
- The global notion of time that is intrinsic in DE models is used as a binding coordination agent.
- The focus here is not about speed of execution but rather about timing determinism.

Relevant Dependency Analysis

Relevant dependency analysis gives a formal framework for analyzing causality relationships to determine the minimal ordering constraints on processing events. The key idea is that events only need to be processed in time-stamp order when they are causally related.

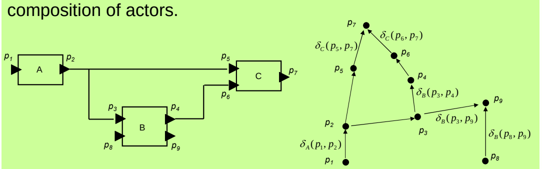
Causality Interface

- Declares the dependency that output events have on input events.

$$\delta_a : P_i \times P_o \rightarrow D$$

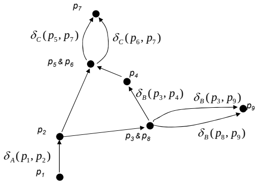
- D is an ordered set associated with the min ($\oplus$) and plus ($\otimes$) operators.
- The dependencies between any two ports in a composition can be determined by using ($\otimes$) for serial composition and ($\oplus$) for parallel composition.

The dependency graph for computing the causality interface of a composition of actors.



Relevant Dependency

- Relevant dependency on any pair of input ports p_1 and p_2 specifies whether an event at p_1 will affect an output signal that may also depend on an event at p_2.



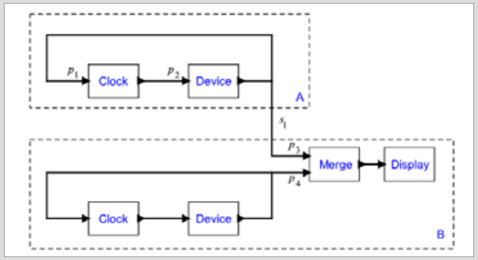
- $d(p_1, p_2) = r$ means any event with time stamp t_2 at p_2 can be processed when all events at p_1 are known up to time stamp $t_2 - r$.
- $d(p_1, p_2) = \infty$ means that events at p2 can be processed without knowing anything about events at p1.

Relevant Order

- Relevant dependencies induce a partial order, called the relevant order, on events.
- $e_1 <_r e_2$ means that e_1 must be processed before e_2.
- If neither $e_1 <_r e_2$, nor $e_2 <_r e_1$, i.e. $e_1 \|_r e_2$, then e_1, e_2 can be processed in any order.
- This technique can be adapted to distributed execution.

Motivating Example

- At two distinct sensor nodes A and B we need to generate precisely timed samples under the control of software. Moreover, the devices that generate these samples provide some sensor data to the software after generating the event.
- A distributed DE model to be executed on a two-sensor, time-synchronized platform A and B is shown in the right figure.



Towards Deployability Analysis

- A key requirement for preserving runtime determinism of PTIDES programs is that each event with model time t at a real-time port must be received before the physical time exceeds $t - \tau$.
- When the execution time is negligible comparing to the network delays and setup time, deployability checking becomes straightforward by using the relevant order.
- A full analysis, when the execution time is not negligible is ongoing.