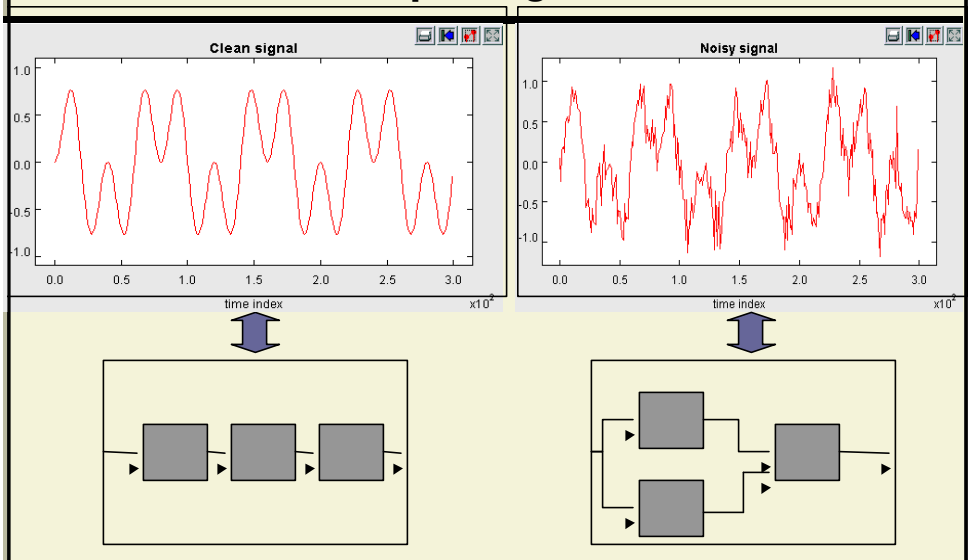
Heterochronous Dataflow in Ptolemy II

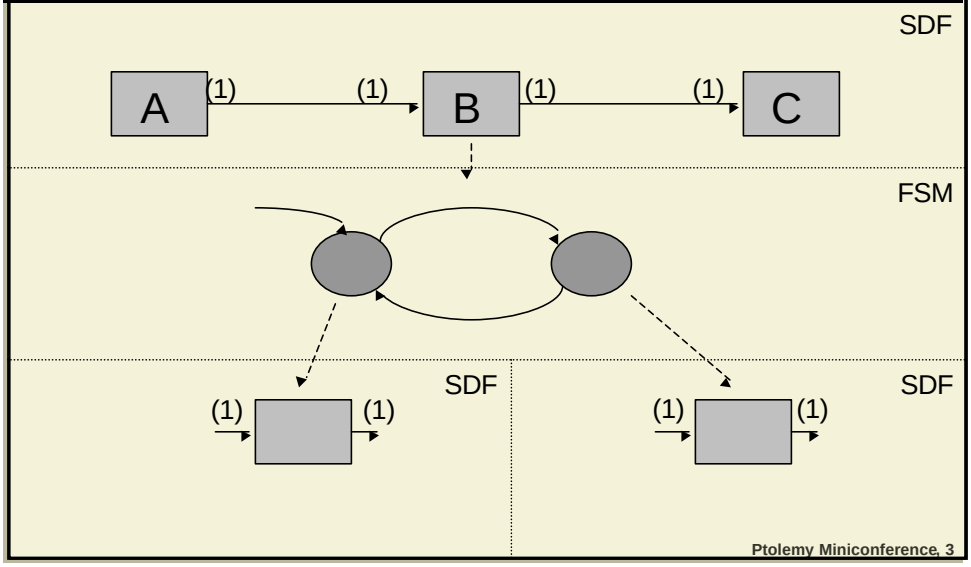
Brian K. Vogel
UC Berkeley

4-th Biennial Ptolemy Miniconference
March 22-23, 2001, Berkeley, CA

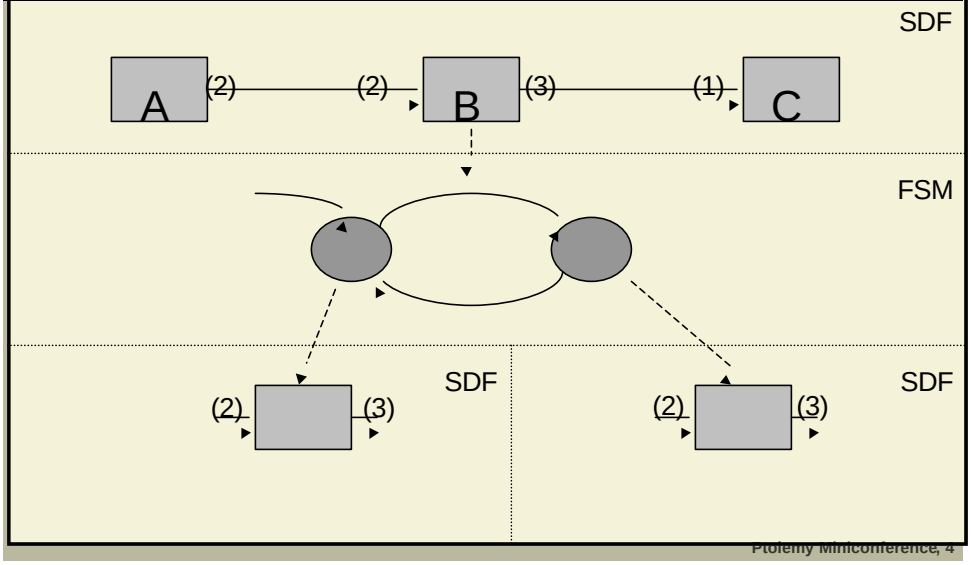
Motivation for composing SDF with FSMs



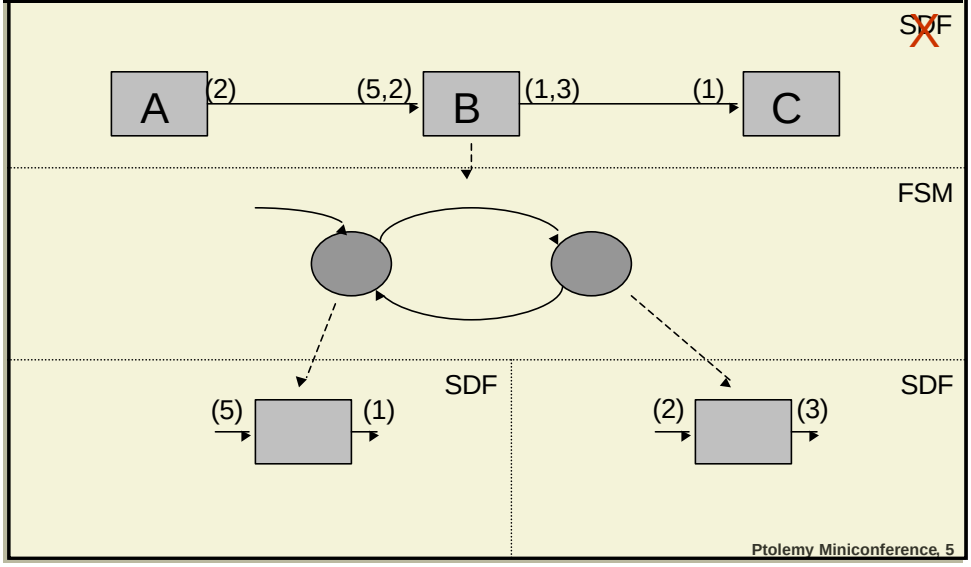
Example 1: homogeneous SDF + FSM



Example 2: non-homogeneous SDF + FSM



Example 3: no longer SDF



Applications of heterochronous dataflow (HDF)

- HDF can represent a class of dataflow models that are more general than cyclo-static dataflow (CSDF), but less general than dynamic dataflow (DDF)
- HDF fits elegantly into the *charts formalism (A. Girault, B. Lee, and E. A. Lee '99)
  - *charts specifies a semantics for hierarchically composing FSMs with various concurrent models of computation
  - HDF with FSMs in *charts is a natural generalization of SDF with FSMs in *charts
  - Useful for representing applications with both control flow and data flow
- HDF can provide an elegant representation for a class of models that contain actors that do not need to be invoked at certain points during the model execution

Properties of HDF

- HDF is a generalization of synchronous dataflow (SDF), and of CSDF
- HDF has a finite state space and retains the desirable decidability properties of SDF
  - Deadlock is decidable
  - Bounded memory channels
  - Static scheduling is possible
- The number of states can be exponential in the number of actors, so static scheduling may not always be practical

HDF semantics

- An HDF actor has a finite number of combinations of port rates
  - A particular combination of ports rates is called a type signature
- An HDF actor has an initial type signature when execution starts
- Can solve the balance equations => find an iteration
- The type signatures are only allowed to change after the last firing of an actor in an iteration

HDF semantics

- Since there are a finite number of type signatures, the set of possible schedules is finite and can be computed at compile-time
  - Deadlock is decidable
  - Bounded channels are decidable
- Unlike CSDF, the order in which the schedules is used is not cyclic

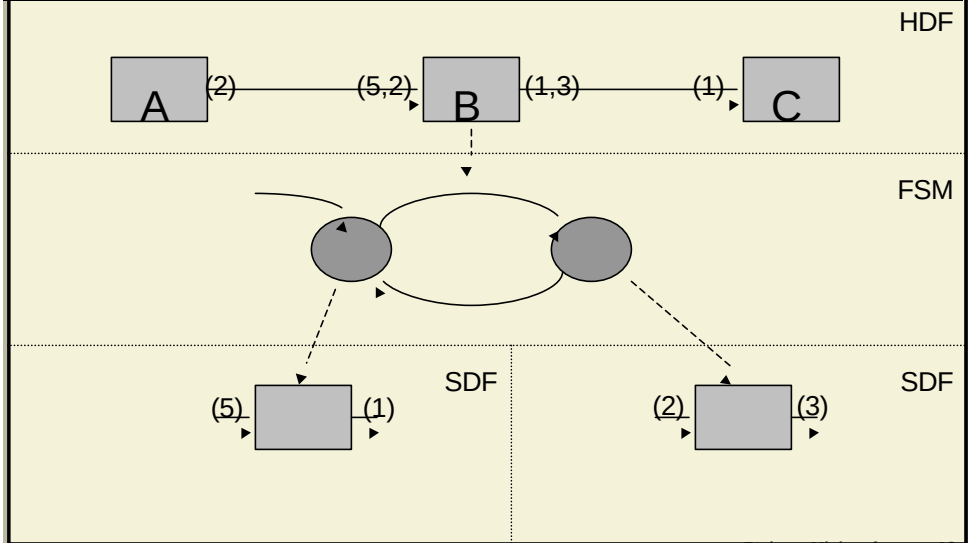
HDF semantics

- It is often useful to use an FSM to control when a type signature change may occur
- An interaction semantics for HDF with hierarchical FSMs is specified in *charts (A. Girault, B. Lee, and E. A. Lee '99)
- *charts is a family of MOCs that specifies an operational semantics for hierarchical FSMs composed with multiple concurrency models

*charts semantics of HDF with FSM

- In *charts, an HDF actor can refine to an FSM
  - The current state of the FSM determines the type signature
  - The type signature may change when a state transition occurs
  - A state is allowed to refine to an SDF graph, an HDF graph, or another FSM
- Arbitrary nesting is allowed in the hierarchy, but the top level must be an HDF graph

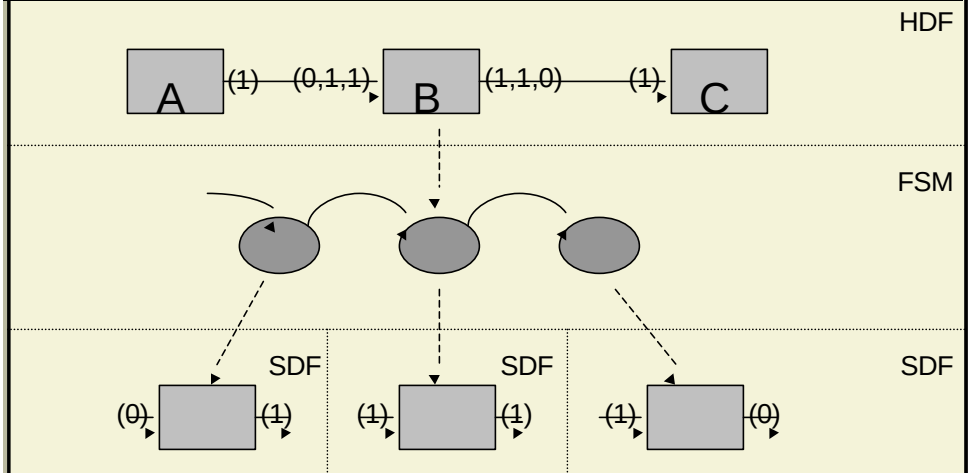
HDF example



Zero-rate ports in HDF

- Some models may contain actors that do not need to be invoked at certain points in the execution
- HDF can elegantly represent a class of models with this property by making use of zero-rate ports

Zero-rate ports example



Ptolemy II Implementation

- An experimental HDF domain has been implemented in Ptolemy II
- Makes use of the existing SDF and FSM domains
- Supports hierarchical heterogeneity by supporting models that hierarchically compose HDF/SDF and FSM subsystems according to the *charts semantics
- HDF models can be constructed visually using the Vergil graph editor

HDF domain features

- Schedules are computed dynamically and cached
- Actors with zero-rate ports are supported
- The *charts SDF with FSM semantics are supported, since this is a special case of HDF with FSM
- Arbitrary levels of nesting in the hierarchy are supported with the constraint that the top level be HDF or SDF

Future work

- Port the HDF implementation to a revised version of the FSM domain, which is under development
- Consider adding parameterized synchronous dataflow (PSDF) (S. Bhattacharyya ‘2000) features to the HDF domain
  - PSDF actors externally have HDF dataflow semantics
  - It may be interesting to to extend the HDF domain to support PSDF parameter flow
  - It may be interesting to consider combining *charts HDF/FSMs with PSDF