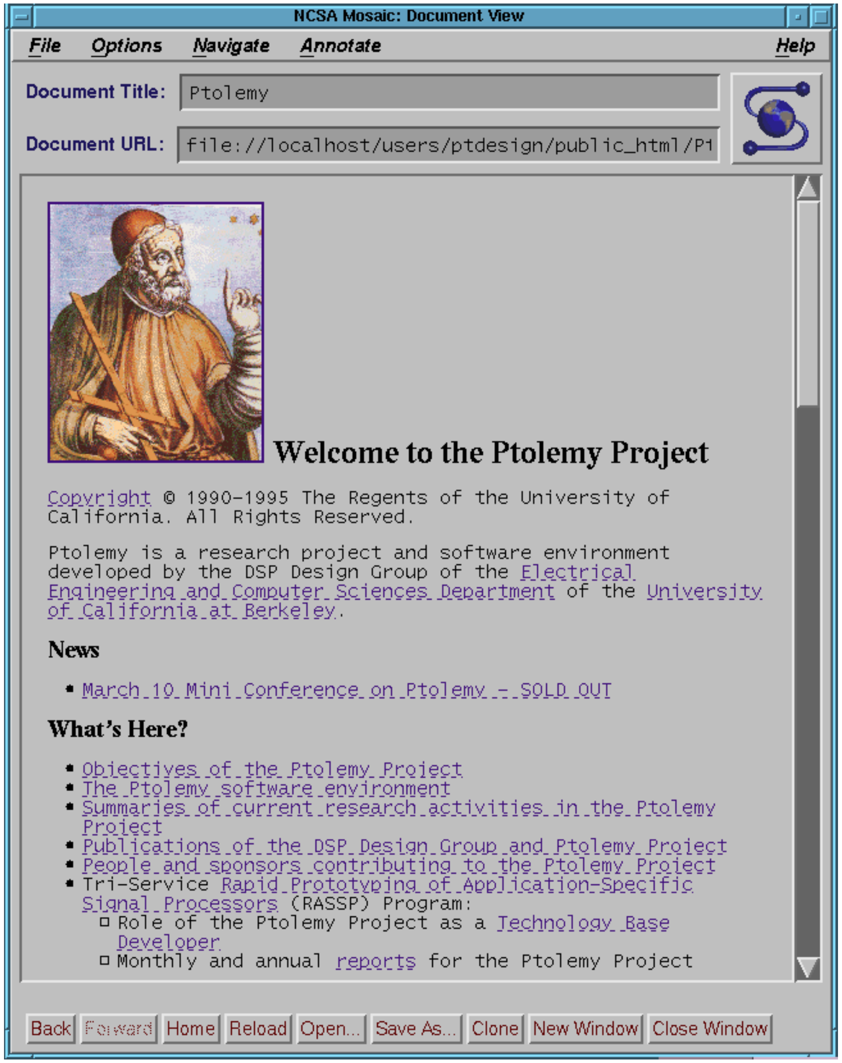
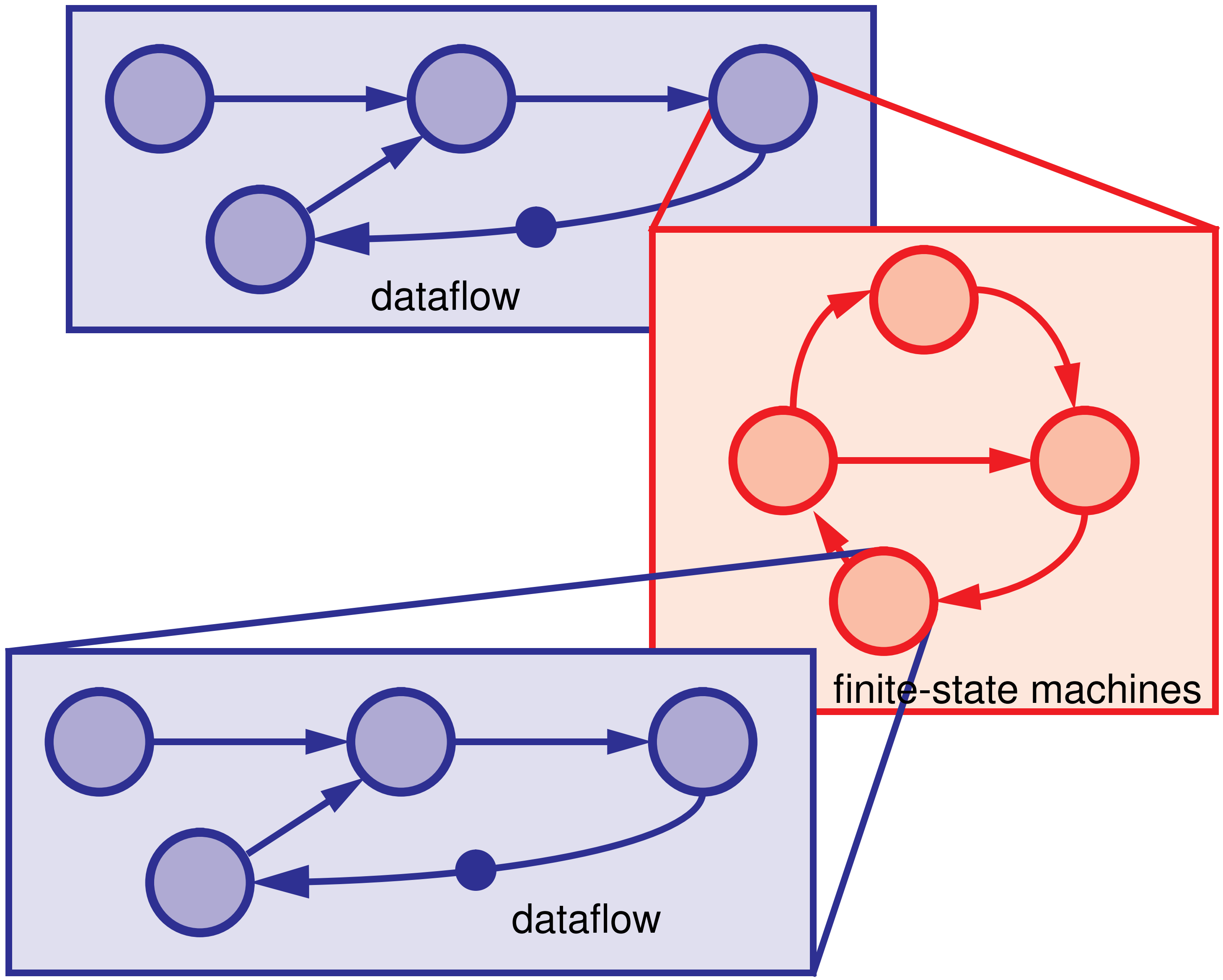
World Wide Web Server



- Complete distribution of version 0.5.2, including all source code.
- Distribution of Ptiny Ptolemy, a small demonstration version.
- An evolving quick tour of Ptolemy with animations of simple demos.
- *The Almagest*, a four-volume manual for Ptolemy, in PostScript.
- User's manual in hypertext form.
- Publications from the Ptolemy group.
- Keyword searching for publications.
- Directory of project participants and sponsors.
- Copy of the FAQ and info about mailing lists and newsgroups.

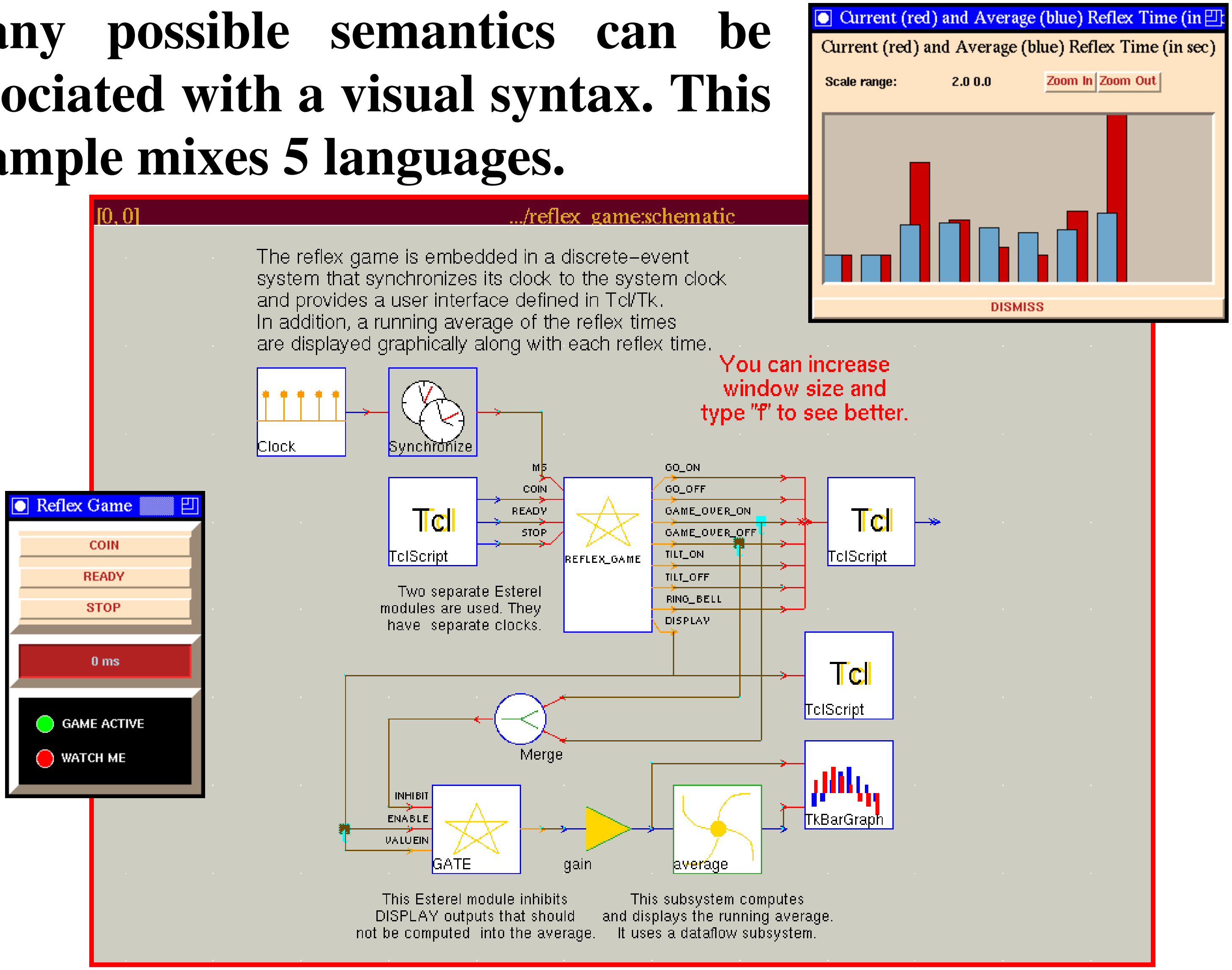
http://ptolemy.eecs.berkeley.edu

Mixing Control-Oriented Models with Others



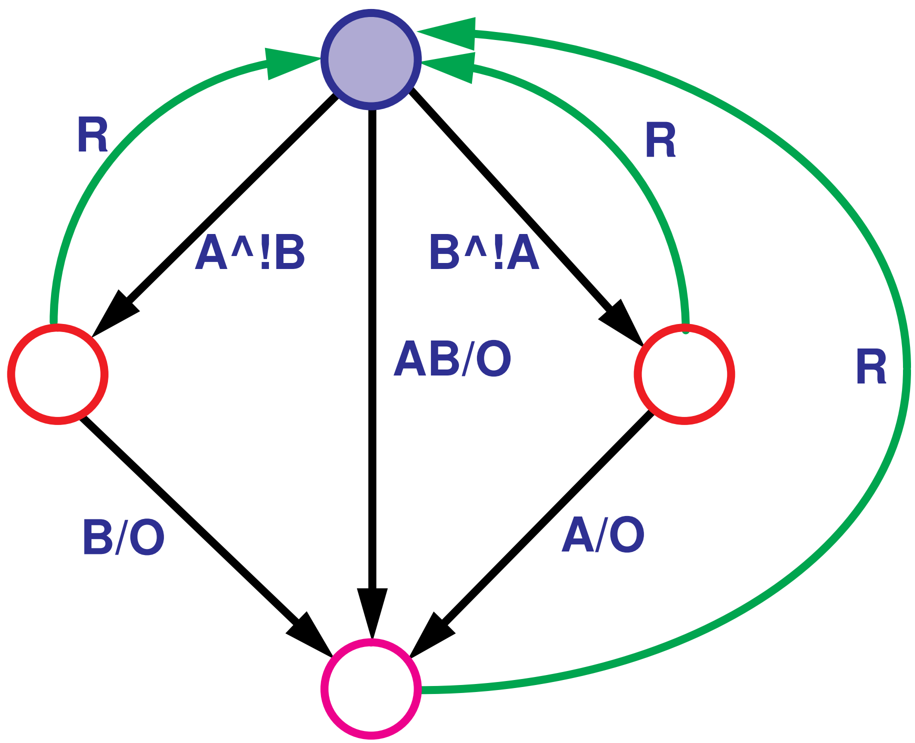
Multilingual Design

Many possible semantics can be associated with a visual syntax. This example mixes 5 languages.



Esterel — An Imperative Language

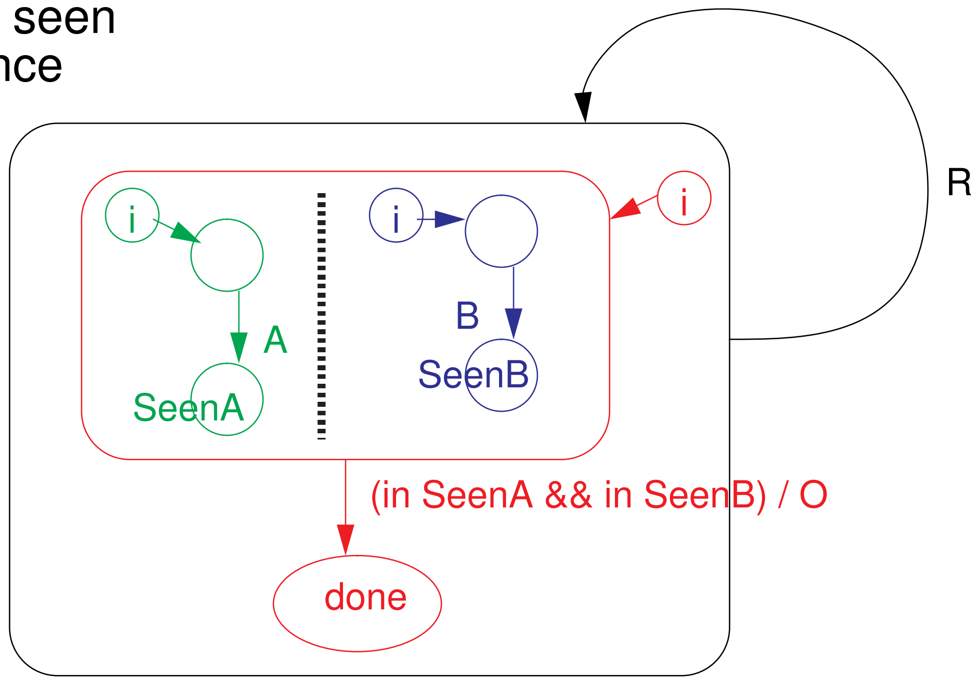
```
module EsTest:
input A, B, R;
output O;
loop
    do
        [await A || await B];
        emit O;
        halt
    watching R
end loop
end module
```



Esterel has the flavor of familiar imperative languages, but with the modeling power of hierarchical FSMs.

Hierarchical Finite-State Machines

This hierarchical FSM reaches "done" when it has seen signals A and B since the last reset.

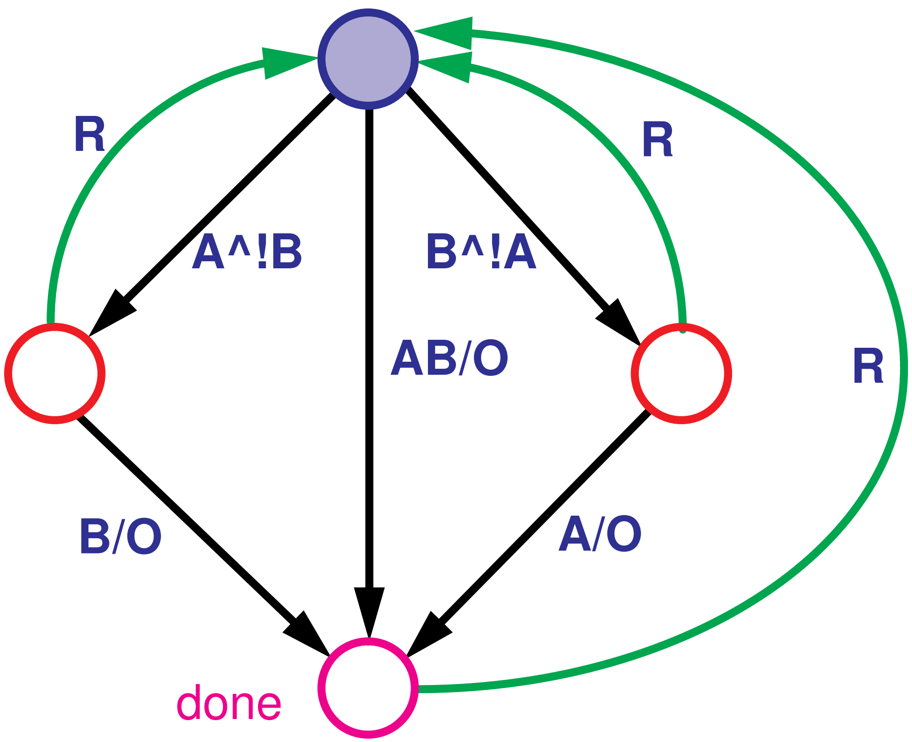


The statecharts model [Harel] permits a state in an FSM to represent being in any state of a sub-FSM. Moreover, it permits concurrent FSMs that communicate through global signals. There are at least 22 variants of this [von der Beeck].

Finite State Machines

Properties of FSMs

- Good for sequential control.
- Formally verifiable reachability.
- State explosion on real systems.
- Not Turing complete.



This FSM reaches "done" when it has seen signals A and B since the last reset.

Key Objectives

- To build a genuinely object-oriented user interface, where the visual syntax can be domain-specific, and interaction with the user can be customized at any level (stars, parameters, galaxies, universes, palettes, etc).
- To extend the non-dogmatic nature of the Ptolemy kernel (which supports multiple semantic models) to the user interface (which will support multiple syntactic models).
- To experiment with design visualization, broadening the perspective beyond a schematic or block-diagram perspective, and exploring new visual and mixed visual/textual syntaxes for design representation/understanding.
- To leverage off work in the Tcl/Tk community to get portable (Windows, Macintoshes) and sophisticated code.
- To integrate sophisticated, interactive documentation into the Ptolemy system.

Tycho

Tycho is an object-oriented front-end for the Ptolemy system. It is based on itcl, an object-oriented extension to Tcl.

The interface to the Ptolemy kernel will be through ptcl, the Tcl extensions that provide an interpreted command language for the kernel.

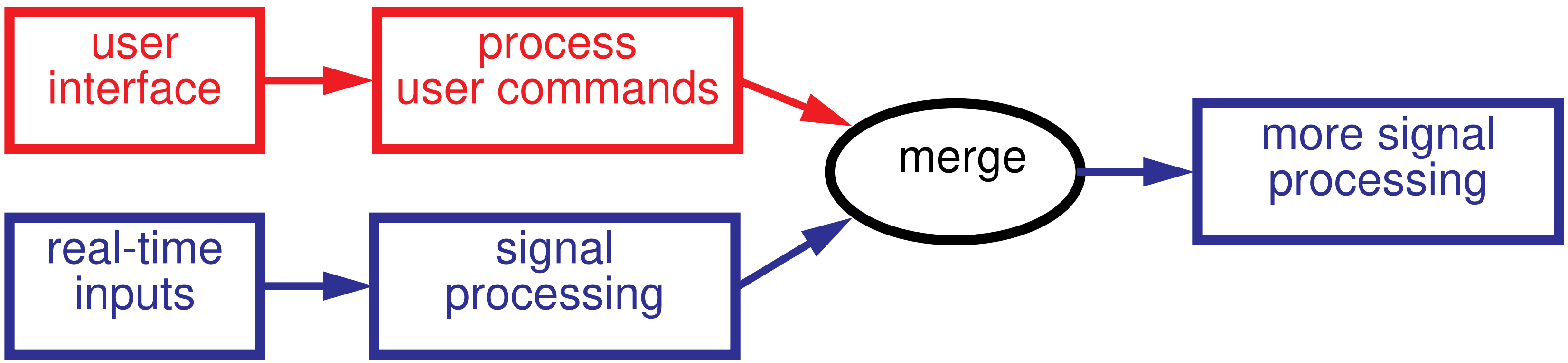
The system will integrate the software development environment (version control, class browsing, context-sensitive editors, compilation, debugging) with the end environment itself.

The system is named after Tycho Brahe, a Danish astronomer who improved on the Ptolemaic system.

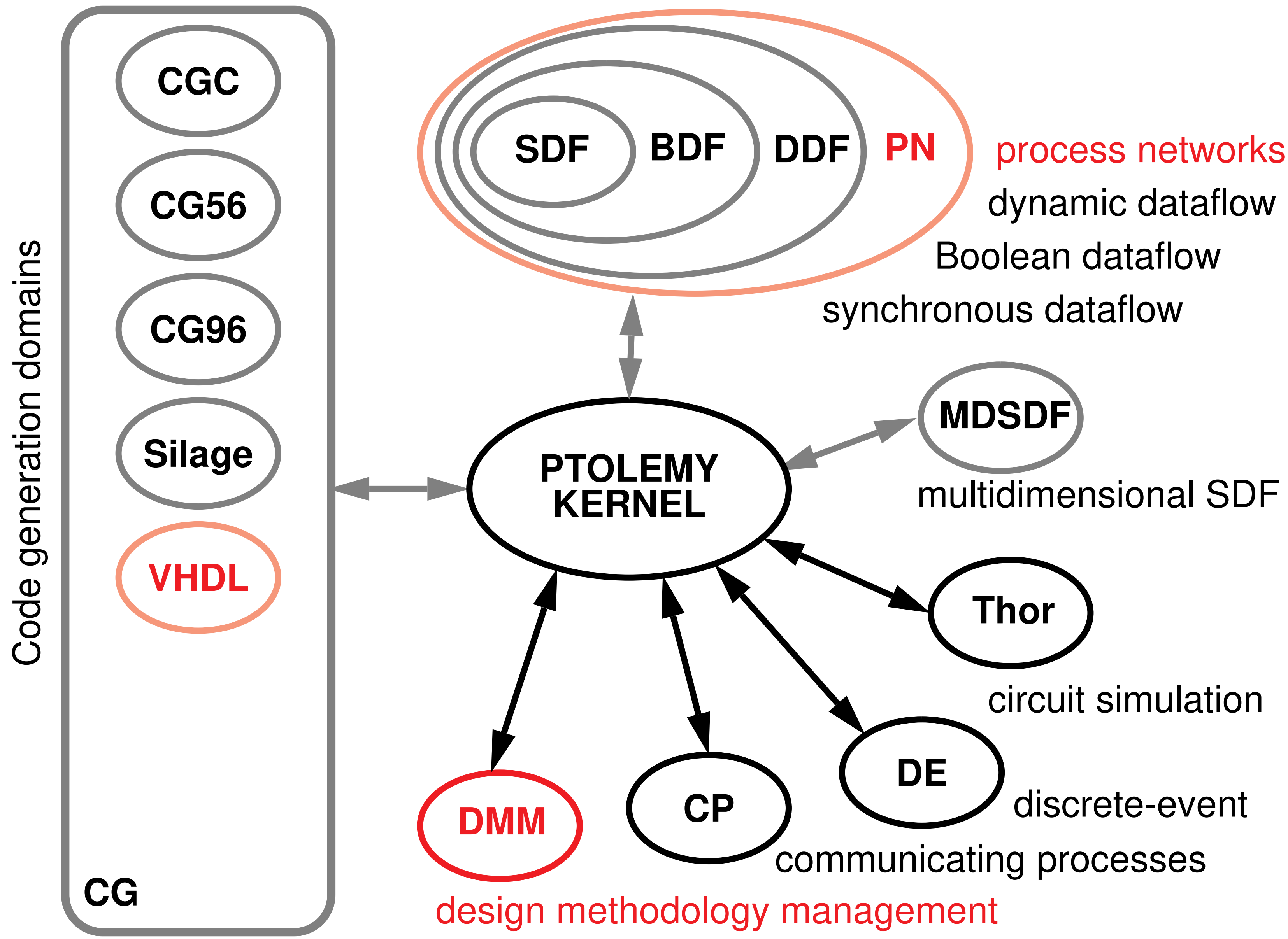
Extensions to Dataflow to Model Nondeterminism

- **The Kahn model maintains determinism by not allowing processes to test an input for the presence of data.**
- **Determinism is usually, but not always, desirable.**

Example scenario where nondeterminism is needed:



New Domains



Code generation domains
CGC
CG56
CG96
Silage
VHDL
CG
SDF
BDF
DDF
PN
process networks
dynamic dataflow
Boolean dataflow
synchronous dataflow
PTOLEMY KERNEL
MDSDF
multidimensional SDF
Thor
circuit simulation
DE
discrete-event
CP
communicating processes
DMM
design methodology management

Preview of Coming Attractions

Shuvra Bhattacharyya
Joseph T. Buck
Wan-Teh Chang
Brian L. Evans
Christopher Hylands
Asawaree Kalavade
Joel King
Bilung Lee
Edward A. Lee
David G. Messerschmitt
Praveen K. Murthy
Thomas M. Parks
José Luis Pino
Farhana Shiekh
S. Sriram
Patrick J. Warner
Michael C. Williamson