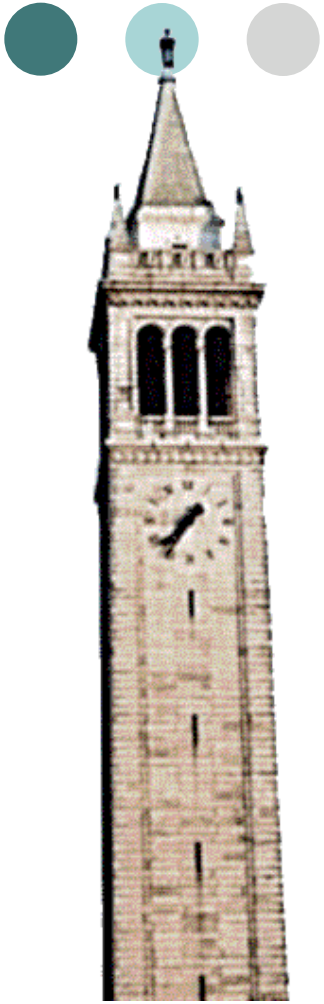


Multimodeling

Edward A. Lee
Robert S. Pepper Distinguished Professor
UC Berkeley

ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2008)

Panel:
Addressing the Challenges of Multi-Modeling for Domain-Specific Modeling Languages

October 3, 2008
Toulouse, France

Multimodeling

Simultaneous use of multiple modeling techniques.

- **hierarchical multimodeling:** hierarchical compositions of distinct modeling styles, combined to take advantage of the unique capabilities and expressiveness of each style.

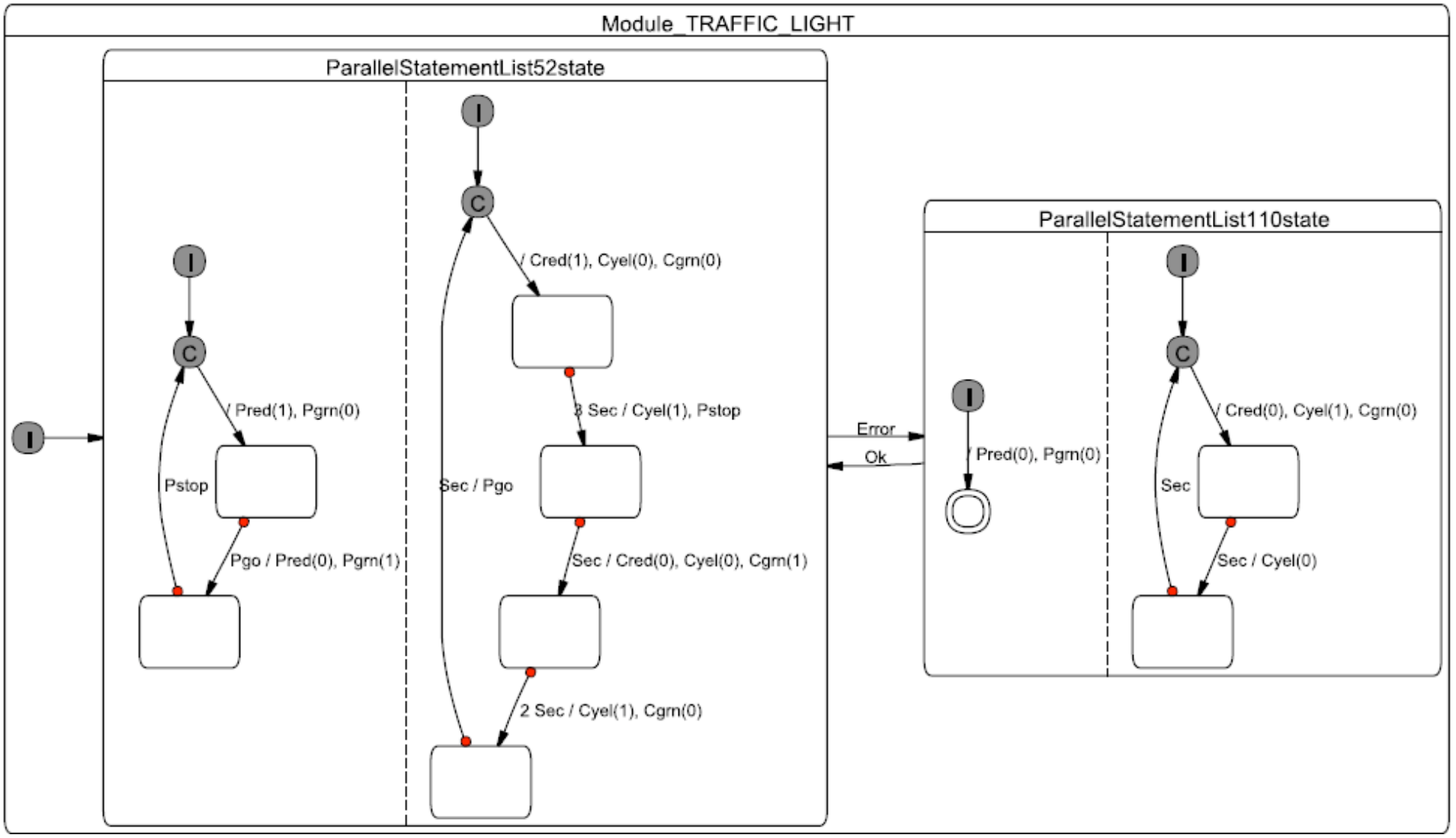
- **multi-view modeling:** distinct and separate models of the same system are constructed to model different aspects of the system.

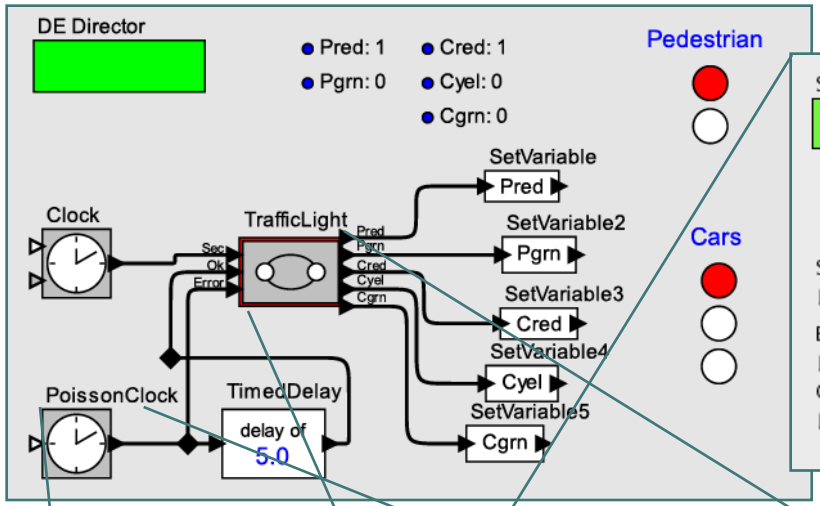
Some Hierarchical Multimodeling techniques that have influenced us:

- Statecharts [Harel 87]
- Ptolemy Classic [Buck, Ha, Lee, Messerschmitt 94]
- SyncCharts [André 96]
- *Charts [Girault, Lee, Lee 99]
- Colif [Cesario, Nicolescu, Guathier, Lyonnard, Jerraya 01]
- Metropolis [Goessler, Sangiovanni-Vincentelli 02]
- Ptolemy II [Eker, et. al. 03]
- Safe State Machine (SSM) [André 03]
- SCADE [Berry 03]
- ForSyDe [Jantsch, Sander 05]
- ModHelX [Jantsch, Sander 07]
- …

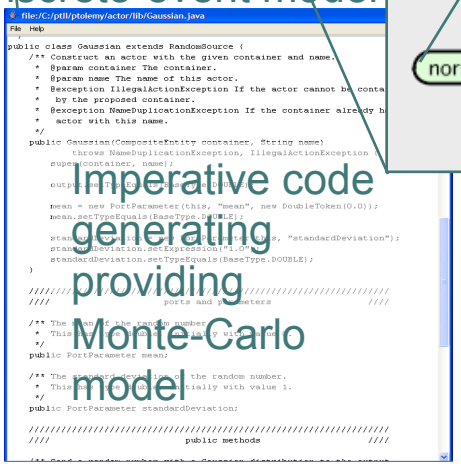
Hierarchical Multimodeling: Statecharts

- *Pred: pedestrian red signal*
- *Pgrn(0): turn pedestrian green off*
- *Cgrn: car green*
- *Sec: one second time*
- *2 Sec: two seconds time*
- *Pgo/Pstop: pedestrian go/stop*

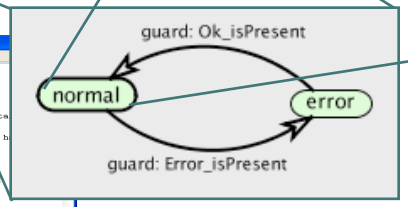




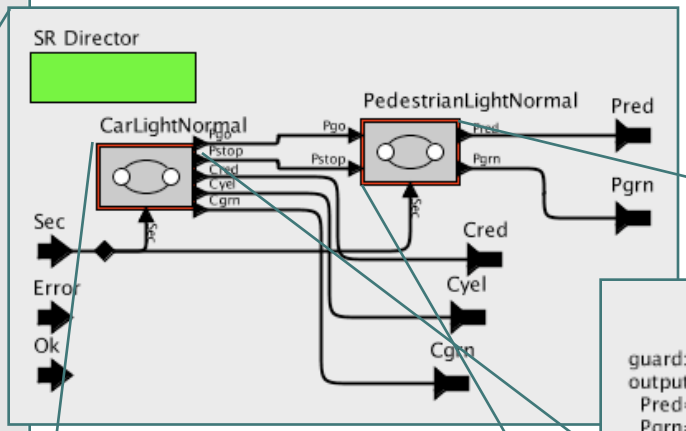
Discrete-event model



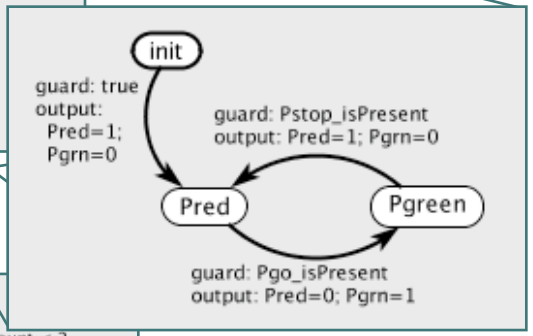
Imperative code generating providing Monte-Carlo model

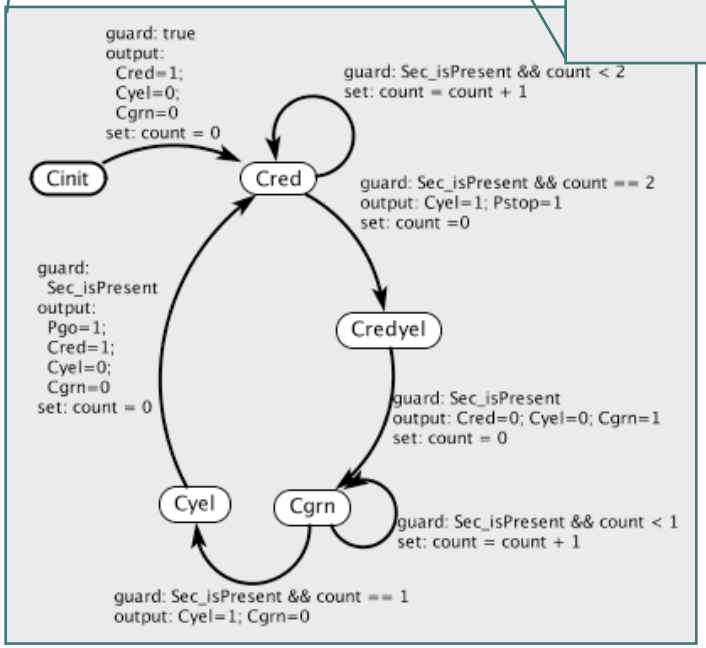


State machine model



Synchronous/ reactive concurrent composition





State machine model

Hierarchical Multimodeling in Ptolemy II

The example shows a discrete-event fault model of simple traffic light system at the top level, and a model of the traffic light system itself that hierarchically combines state machines and synchronous/ reactive concurrent composition.

Many distinct models of computation can be hierarchically combined to create executable models.

Multimodeling

Simultaneous use of multiple modeling techniques.

- **hierarchical multimodeling:** hierarchical compositions of distinct modeling styles, combined to take advantage of the unique capabilities and expressiveness of each style.

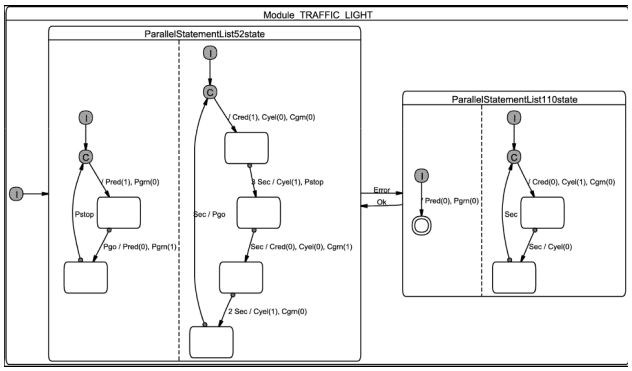
- **multi-view modeling:** distinct and separate models of the same system are constructed to model different aspects of the system.

Some Multi-View Modeling techniques that have influenced us:

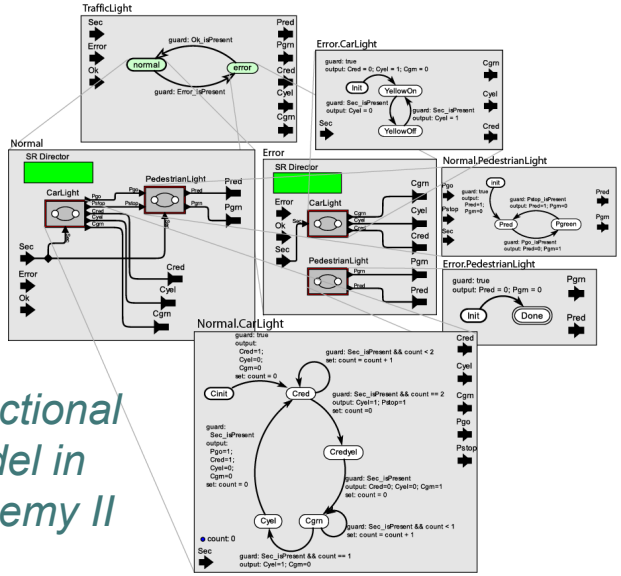
- UML [Various, 90s]
- Ptolemy Classic [Buck, Ha, Lee, Messerschmitt 94]
- Model-integrated computing [Sztipanovits, Karsai, Franke 96]
- Colif [Cesario, Nicolescu, Guathier, Lyonnard, Jerraya 01]
- Metropolis [Goessler, Sangiovanni-Vincentelli 02]
- KIEL [Prochnow, von Hanxleden 07]
- …

Multi-View Modeling:

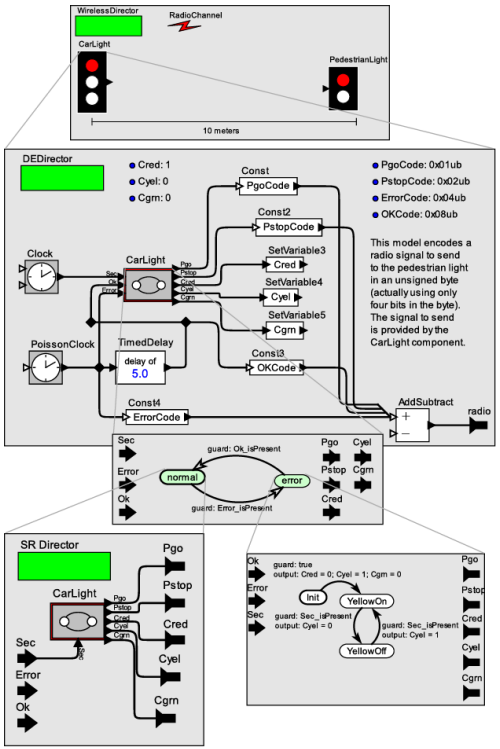
Distinct and separate models of the same system are constructed to model different aspects of the system.



Functional model in Statecharts



Functional model in Ptolemy II



Deployment model in Ptolemy II

```
MODULE CarLightNormal( Sec_isPresent )
   VAR
      state : {Cyel,Credyel,Cred,Cinit,Cgrn};
      count : { ls,0,1,2,gt };
   ASSIGN
      init(state) := Cinit;
      next(state) :=
         case
            state=Cinit & count=ls :{ Cred };
            ---
            Sec_isPresent & state=Cred
                          & count=ls :{ Cred };
            ---
            1             : state;
         esac;
      init(count) := 0;
      next(count) :=
         case
            state=Cinit & count=ls :{ 0 };
            ---
            Sec_isPresent & state=Cred & count=ls :{ ls };
            ---
            1             : count;
         esac;
   DEFINE
      Pstop_isPresent :=  ( Sec_isPresent
                          & state=Cred & count=2 ) ;
      Cred_isPresent :=  ...
      Cgrn_isPresent :=  ...
      Pgo_isPresent :=  ...
      Cyel_isPresent :=  ...
MODULE PedestrianLightNormal( Pstop_isPresent,Pgo_isPresent )
   VAR
      state : {Pinit,Pgreen,Pred};
   ASSIGN
      init(state) := Pinit;
      next(state) :=
         case
            state=Pinit :{ Pred };
            Pgo_isPresent & state=Pred :{ Pgreen };
            Pstop_isPresent & state=Pgreen :{ Pgreen };
            1             : state;
         esac;
   DEFINE
      Pred_isPresent :=  ...
      Pgrn_isPresent :=  ...
MODULE main
   VAR
      CarLightNormal: CarLightNormal( 1);
      PedestrianLightNormal: PedestrianLightNormal(
        CarLightNormal.Pstop_isPresent,
        CarLightNormal.Pgo_isPresent );
   SPEC
      ! EF (CarLightNormal.state = Cgrn
          & PedestrianLightNormal.state = Pgreen)
```

Verification model in SMV

	Project Name: Process-Based Software Components for Embedded Systems	Current Official Milestone Completion Date	Does your project support this milestone? (Y or N)	Identify your s support(e
2				
7	4. Demonstrate ability of propagating constraints among views	2QFY02	Y	Ptolemy II - hierar
8	5. Demonstrate ability to integrate different models of concurrency	2QFY02	Y	Ptolemy II - multi-
9	6. Demonstrate ability to integrate domain specific modeling tools	2QFY02	Y	Ptolemy II - multi-
10	7. Demonstrate ability to compose multiple view models	4QFY02	Y	Ptolemy II - multi-
11	8. Demonstrate ability to verify multiple-view models	4QFY03	Y	Ptolemy II - simul
12	**Task 2: Model-Based Generation Technology**			
13	1. Demonstrate ability to mathematically model generators	4QFY01	N	

Reliability model in Excel

Maintaining Consistency Across Models?

- For hierarchical multimodeling:
  - Abstract semantics
  - Behavioral type systems
- For multi-view modeling:
  - Model synthesis
  - Model transformation
  - Model ontologies

All of these require substantial research (theory, methods, and tools)!

Teaching Multimodeling?

A basic principle:

We should teach things we understand…