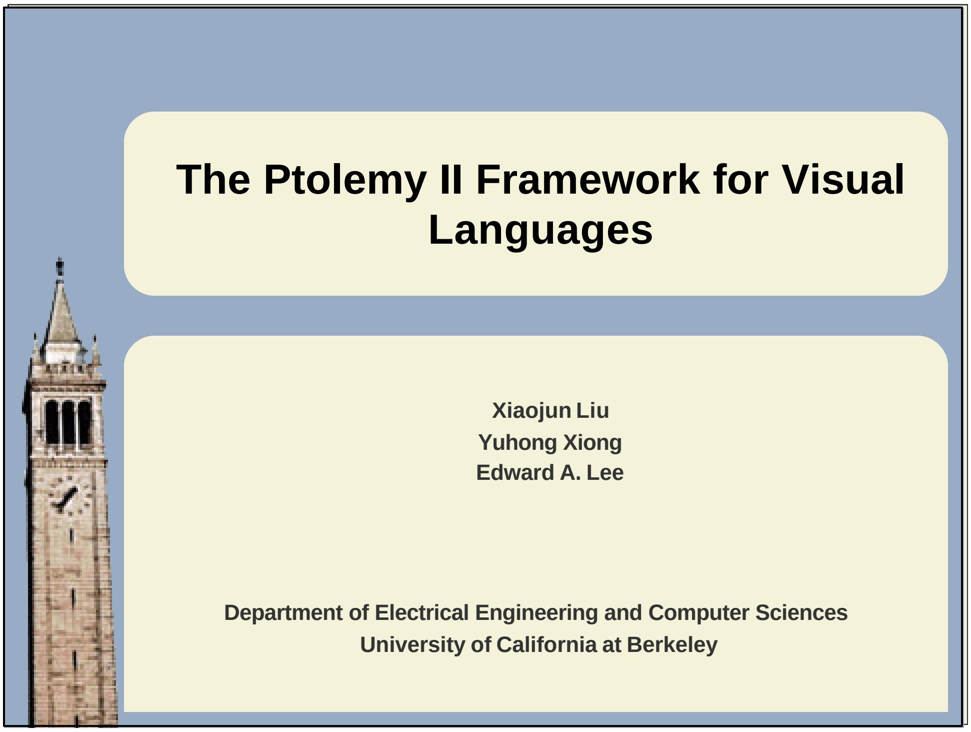
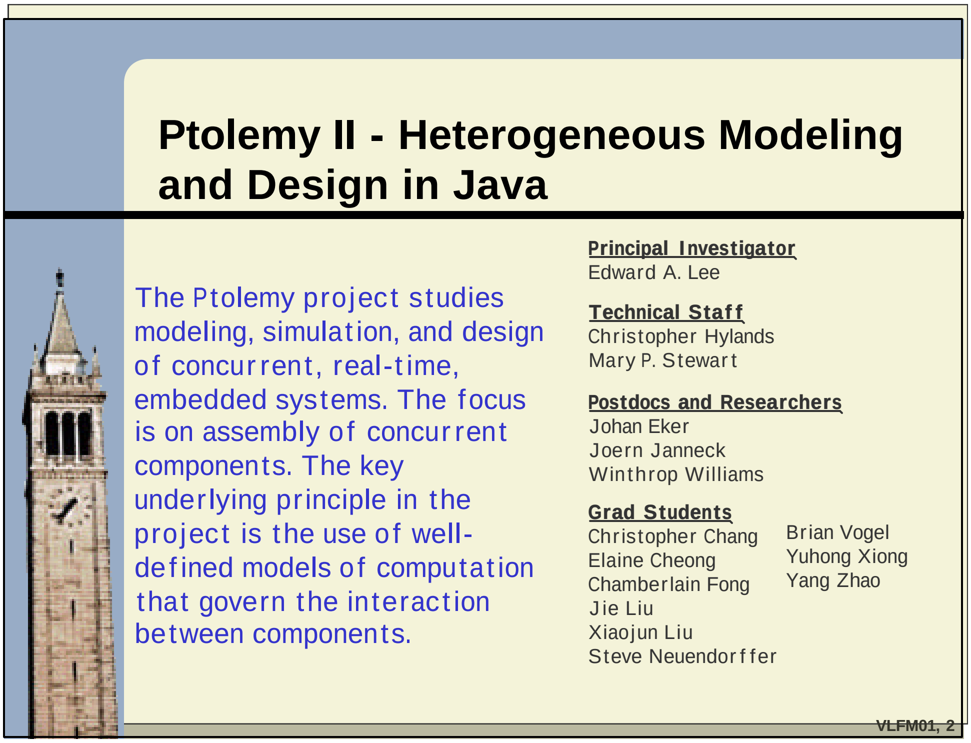


The Ptolemy II Framework for Visual Languages

Xiaojun Liu
Yuhong Xiong
Edward A. Lee

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley



Ptolemy II - Heterogeneous Modeling and Design in Java

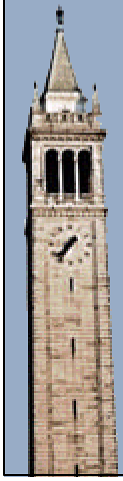
The Ptolemy project studies modeling, simulation, and design of concurrent, real-time, embedded systems. The focus is on assembly of concurrent components. The key underlying principle in the project is the use of well-defined models of computation that govern the interaction between components.

Principal Investigator
Edward A. Lee

Technical Staff
Christopher Hylands
Mary P. Stewart

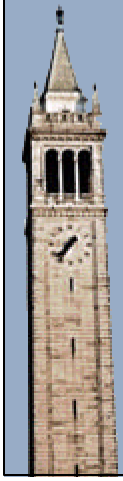
Postdocs and Researchers
Johan Eker
Joern Janneck
Winthrop Williams

Grad Students
Christopher Chang
Elaine Cheong
Chamberlain Fong
Jie Liu
Xiaojun Liu
Steve Neuendorffer
Brian Vogel
Yuhong Xiong
Yang Zhao

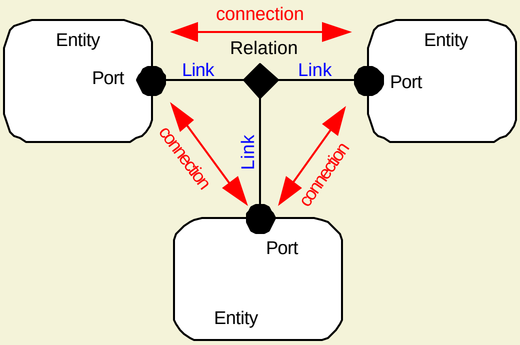


Approach to Visual Modeling

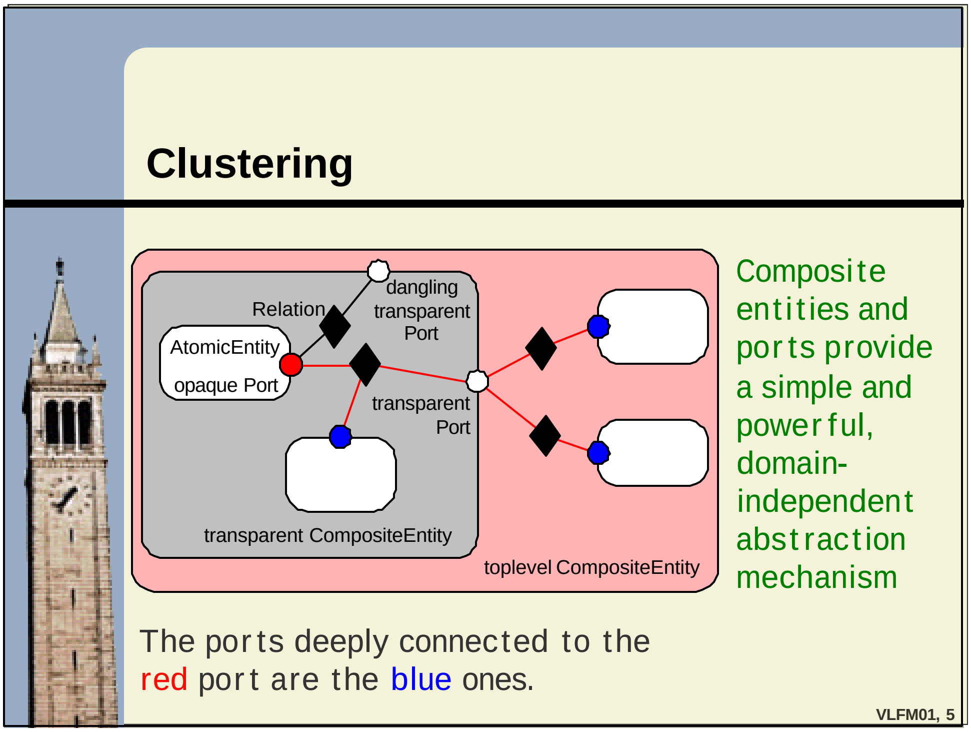
- Abstract syntax: clustered graph of *entities* and *relations.*
- Executable entities and execution control: *actors* communicate with each other through message passing under the control of a *director.*
- Domains: implementation of semantics for component interaction.
- *charts: mixing finite state machine (FSM) with other domains.
- System-level types: formal framework to study the dynamic properties of component interaction.



Abstract Syntax



Clustered graphs well-suited to a wide range of domains, ranging from state machines to process networks



Clustering

Relation

AtomicEntity

opaque Port

dangling transparent Port

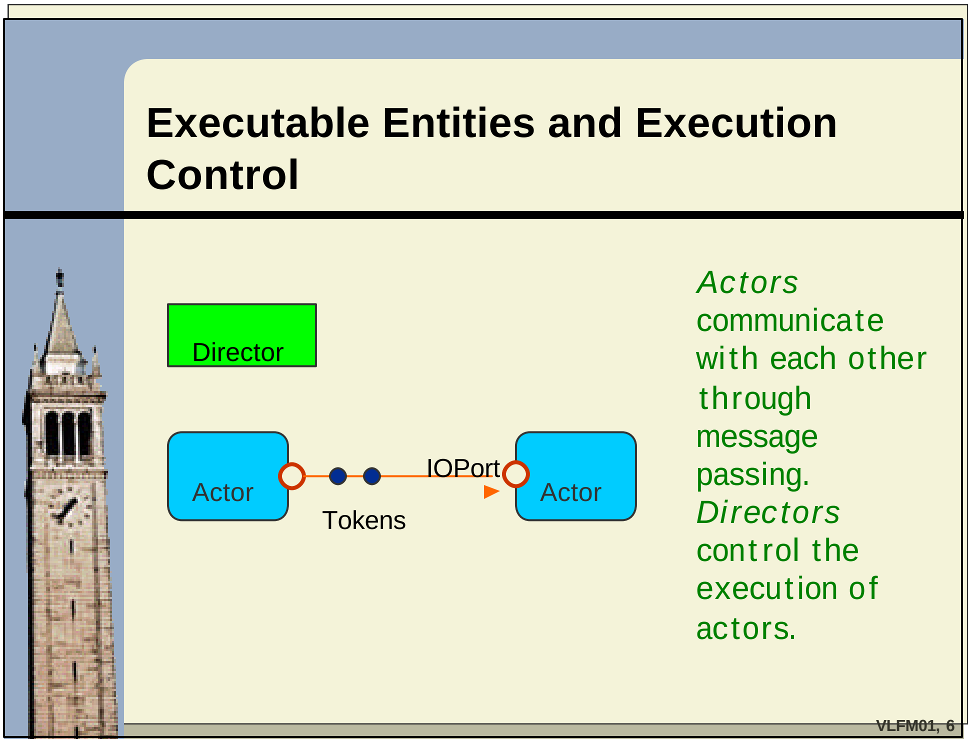
transparent Port

transparent CompositeEntity

toplevel CompositeEntity

Composite entities and ports provide a simple and powerful, domain-independent abstraction mechanism

The ports deeply connected to the red port are the blue ones.



Executable Entities and Execution Control

Director

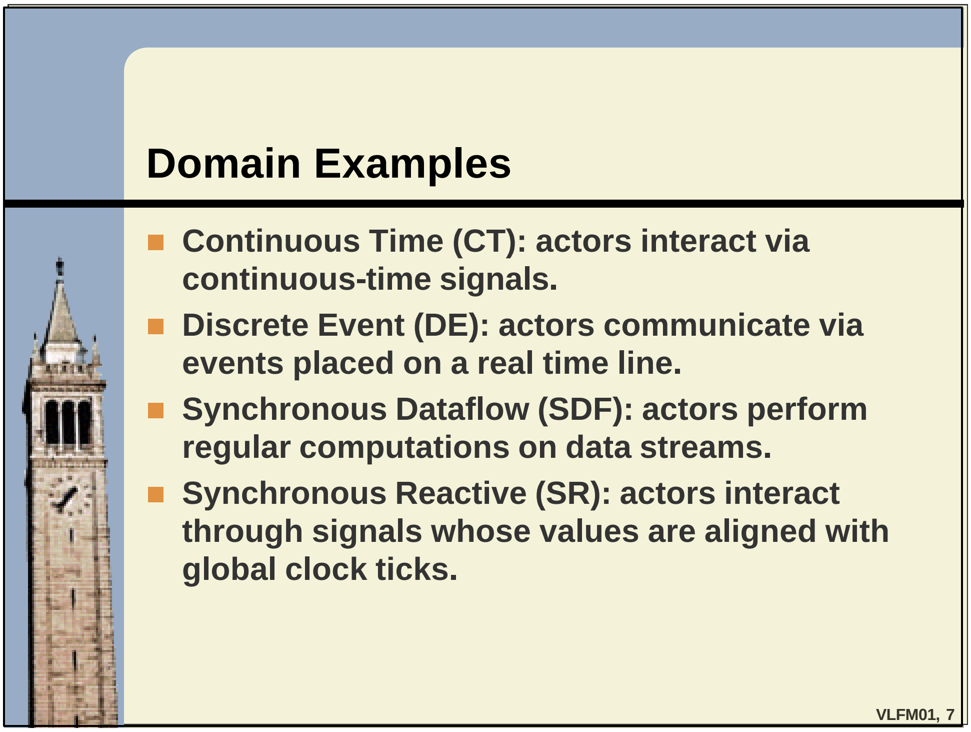
Actor

IOPort

Actor

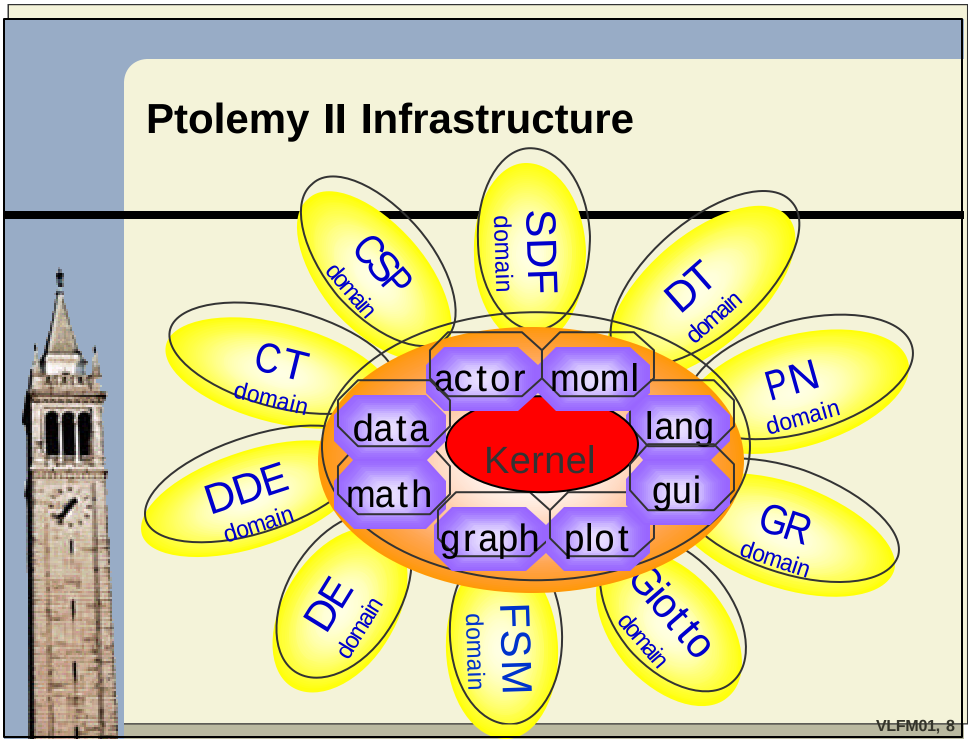
Tokens

Actors communicate with each other through message passing. *Directors* control the execution of actors.



Domain Examples

- Continuous Time (CT): actors interact via continuous-time signals.
- Discrete Event (DE): actors communicate via events placed on a real time line.
- Synchronous Dataflow (SDF): actors perform regular computations on data streams.
- Synchronous Reactive (SR): actors interact through signals whose values are aligned with global clock ticks.

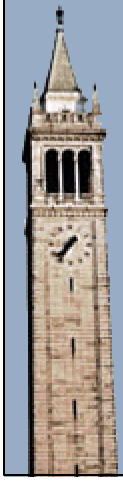


Ptolemy II Infrastructure

Kernel

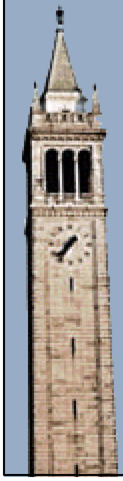
actor
moml
lang
gui
plot
graph
math
data

SDF domain
DT domain
PN domain
GR domain
Giotto domain
FSM domain
DE domain
DDE domain
CT domain
CSP domain



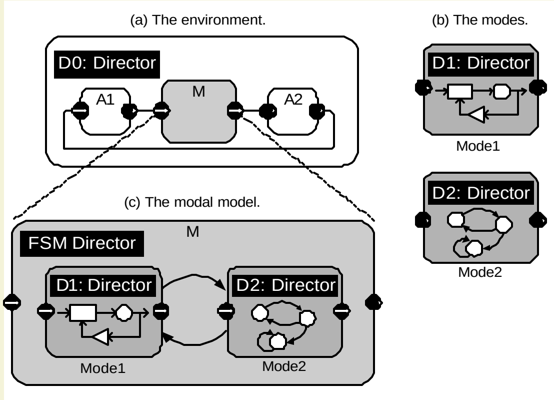
*charts (Girault, Lee and Lee)

- Motivated by Statecharts and hybrid systems.
- Allows nesting FSMs with a variety of models of computation.
- Nesting can happen at any level in a heterogeneous model.
- Decouples the concurrency model from the hierarchical FSM semantics.

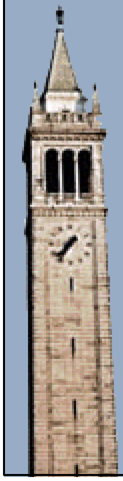


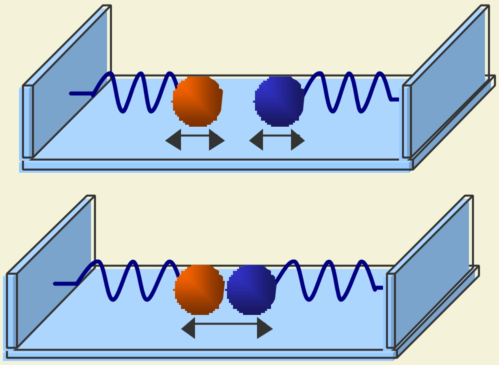
*charts – Ptolemy II Implementation

- FSMActor
- Modal Model



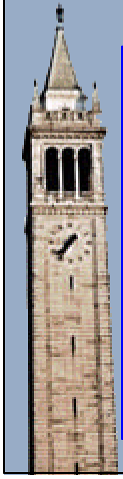
Example: Sticky Masses

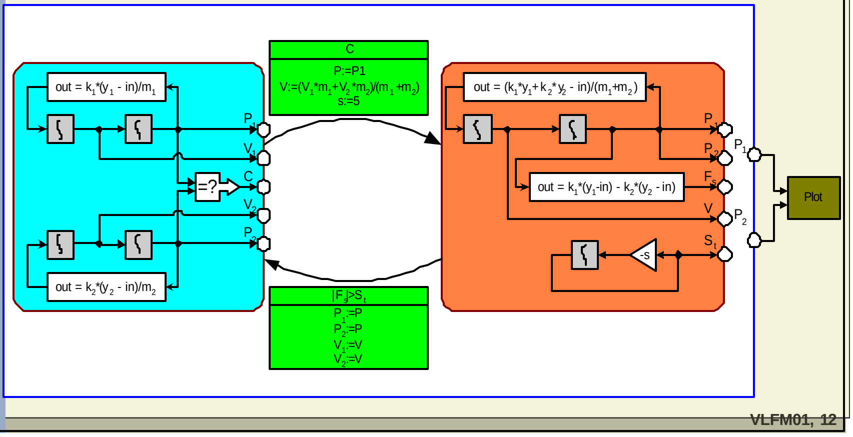




The stickiness is exponentially decaying with respect to time.

Sticky Masses: Block Diagram





out = k_1*(y_1 - in)/m_1

out = k_2*(y_2 - in)/m_2

=?

P_1 V_1 C V_2 P_2

C

P:=P1

V:=(V_1*m_1+V_2*m_2)/(m_1+m_2)

s:=5

|F_s|>S_t

P_1:=P

P_2:=P

V_1:=V

V_2:=V

out = (k_1*y_1+k_2*y_2 - in)/(m_1+m_2)

out = k_1*(y_1-in) - k_2*(y_2 - in)

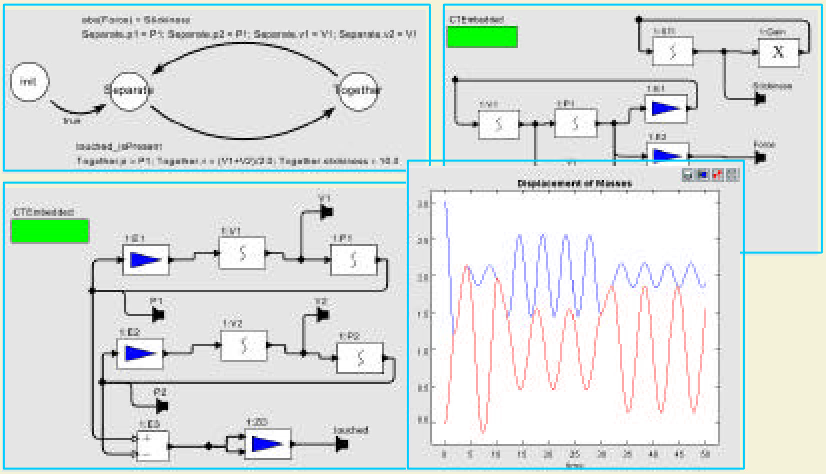
-s

P_1 P_2 F_s V S_t

P_1 P_2

Plot

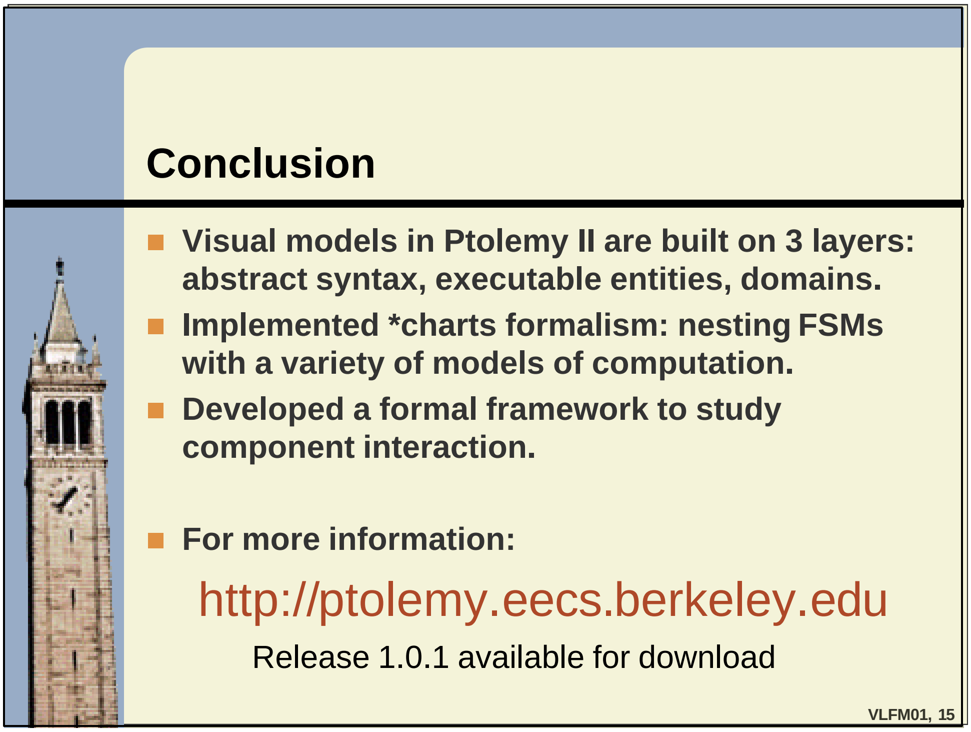
Sticky Masses: Simulation



Created by Jie Liu, Xiaojun Liu and Yuhong Xiong.

Formal Framework for Component Interaction

- Treat different communication protocols in Ptolemy II domains as types: interaction types or *system-level types.*
- Type signature and component behavior described by interface automata (de Alfaro and Henzinger).
- Compatibility of components with an interaction type checked through automata composition.
- Simulation relation captures subtyping of interaction types.
- Components may be polymorphic: compatible with multiple interaction types.



Conclusion

- Visual models in Ptolemy II are built on 3 layers: abstract syntax, executable entities, domains.
- Implemented *charts formalism: nesting FSMs with a variety of models of computation.
- Developed a formal framework to study component interaction.

- For more information:

http://ptolemy.eecs.berkeley.edu

Release 1.0.1 available for download