





























Software Legacy of the Project	
<ul> <li>Gabriel (1986-1991)         <ul> <li>Written in Lisp</li> <li>Aimed at signal processing</li> <li>Synchronous dataflow (SDF) block</li> <li>Parallel schedulers</li> <li>Code generators for DSPs</li> <li>Hardware/software co-simulators</li> </ul> </li> <li>Ptolemy Classic (1990-1997)         <ul> <li>Written in C++</li> <li>Multiple models of computation</li> <li>Hierarchical heterogeneity</li> <li>Dataflow variants: BDF, DDF, PN</li> <li>C/VHDL/DSP code generators</li> <li>Optimizing SDF schedulers</li> <li>Higher-order components</li> </ul> </li> <li>Ptolemy II (1996-2022)         <ul> <li>Written in Java</li> <li>Domain polymorphism</li> <li>Multithreaded</li> <li>Network integrated</li> <li>Modal models</li> <li>Sophisticated type system</li> <li>CT, HDF, CI, GR, etc.</li> </ul> </li> </ul>	diagrams Each of these served us, first-and- foremost, as a laboratory for investigating design. PtPlot (1997-??) - Java plotting package Tycho (1996-1998) - Itcl/Tk GUI framework Diva (1998-2000) - Java GUI framework













## Examples of Actor-Oriented Component Frameworks

- Easy5 (Boeing)
- Simulink (The MathWorks)
- Labview (National Instruments)
- Modelica (Linkoping)
- OCP, open control platform (Boeing)
- GME, actor-oriented meta-modeling (Vanderbilt)
- SPW, signal processing worksystem (Cadence)
- System studio (Synopsys)
- ROOM, real-time object-oriented modeling (Rational)
- Port-based objects (U of Maryland)
- I/O automata (MIT)
- VHDL, Verilog, SystemC (Various)
- Polis & Metropolis (UC Berkeley)
- Ptolemy & Ptolemy II (UC Berkeley)
- ...

## Models of Computation Principles of Model Driven Architecture • Continuous-time models • Dataflow • synchronous dataflow • boolean/integer dataflow • dynamic dataflow • heterochronous dataflow • Push/pull models • Discrete-event models • Synchronous/reactive models • CSP models • Discrete-time models • Discrete-time models (TTA, Giotto) • Modal models are possible in all cases

Chess, UC Berkeley, E. A. Lee 23

























Chess, UC Berkeley, E. A. Lee 36











Conclusion – What to Remember	
<ul> <li>A new systems science <ul> <li>physical + computational</li> </ul> </li> <li>Actor-oriented design <ul> <li>concurrent components interacting via ports</li> </ul> </li> <li>Models of computation <ul> <li>principles of component interaction</li> </ul> </li> <li>Hierarchical heterogeneity <ul> <li>principled mixing of models of computation</li> </ul> </li> <li>Behavioral types <ul> <li>a practical approach to verification and interface definition</li> </ul> </li> <li>Domain polymorphism <ul> <li>defining components for use in multiple contexts</li> </ul> </li> <li>http://ptolemy.eecs.berkeley.edu</li> <li>http://chess.eecs.berkeley.edu</li> </ul>	
Chess, UC Berkeley, E. A. Lee 42	