

The Ptolemy Project Plans for the Future



Principal Investigator

Edward A. Lee

Staff

Jennifer Basler
Christopher Hylands
Mary P. Stewart

Postdocs

Bart Kienhuis
John Reekie

Students

John Davis, II
Ron Galicia
Mudit Goel
Bilung Lee
Michael Leung
Jie Liu
Xiaojun Liu
Lukito Muliadi
Steve Neuendorffer
Neil Smyth
Jeff Tsay
William Wu
Yuhong Xiong

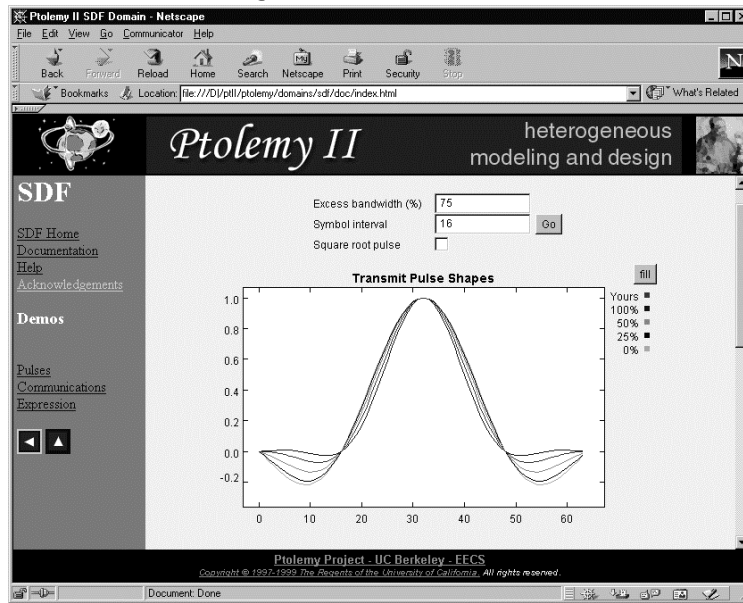
Ptolemy Miniconference - 1

Near Term Objectives

- Make current domains releasable
- New domains (DT, SR)
- Distributed interoperability (CORBA/JINI)
- XML file formats
- Domain-polymorphic actor library
- Domain-polymorphic HOF actors
- Visualization infrastructure (Tycho, Diva)
- Ptolemy II on the web
- Ptolemy II 1.0 release in 1999

Ptolemy Miniconference - 2

Ptolemy II on the Web



Ptolemy Miniconference - 3

Longer Term

- Focus on networked embedded systems
 - adaptive, updateable, real-time, safety critical
- Process-level type systems
 - communication and concurrency in the type system
- Process-level polymorphism
 - tolerant interfaces
- Run-time type inference
 - support dynamic software architectures

Ptolemy Miniconference - 4

Longer Term (cont)

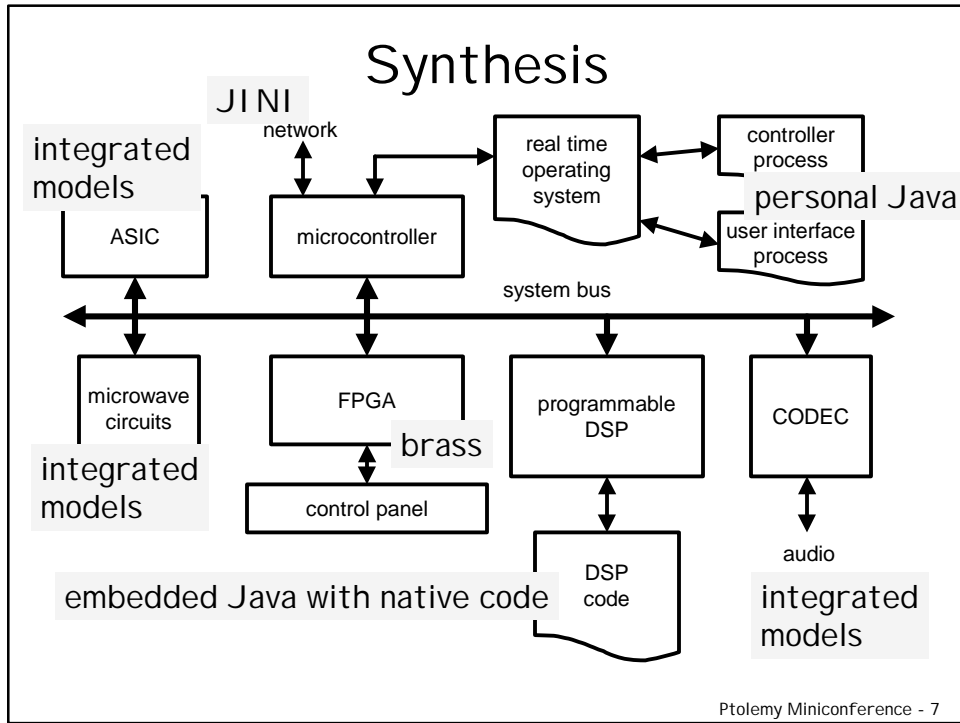
- Reflection of program dynamics
 - introspection to assure real-time, safety, liveness
- Integrated modeling of hardware
 - provide unity between HW & SW components
- Verification of finite-state controllers
 - integrated model-checking

Ptolemy Miniconference - 5

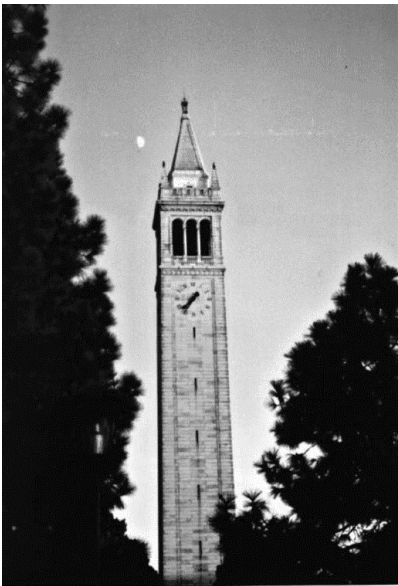
Practical Application Example

- Networked digital cellular phone/PDA
 - ODEs for RF components
 - dataflow for modem, speech coder
 - FSMs within modem, speech coder for control logic
 - SR for UI event handling
 - FSMs for call processing
 - DE for environment (network) modeling
 - Reflection for on-line adaptability
 - Introspection for real-time assurance

Ptolemy Miniconference - 6



Conclusion



The Ptolemy project continues to be a rich source of research problems pertinent to EDA and system-builders. Fundamentally, it deals with using higher levels of abstraction than HDLs or standard imperative languages such as C. Concurrency in these abstractions is their main source of novelty and expressiveness. Concepts from programming languages, such as type theories, reflection, and semantics are applicable.

More Information:
<http://ptolemy.eecs.berkeley.edu/>

Ptolemy Miniconference - 8