

Design Languages in 2010

Edward A. Lee
Professor
UC Berkeley

Panel Position Statement
Forum on Design Languages (FDL)
Frankfurt, Sept. 26, 2003



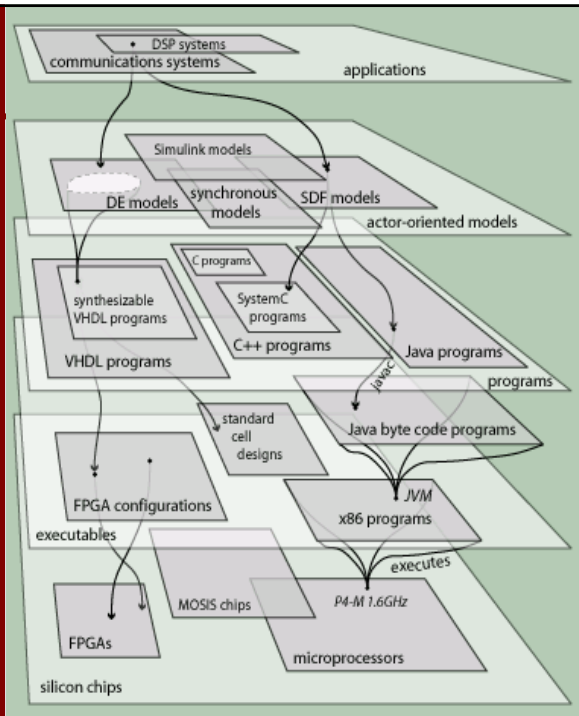
Chess:
Center for Hybrid and Embedded Software Systems



Platforms

A *platform* is a set of designs (the rectangles at the right, e.g., the set of all x86 binaries).

Model-based design is specification of designs in platforms with useful modeling properties (e.g., Simulink block diagrams for control systems).

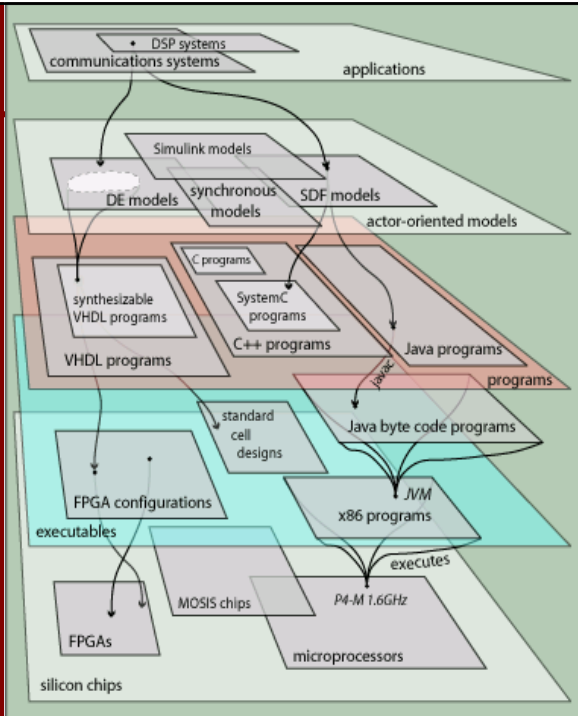


Platforms

Where the
Action Has Been:

Giving the red
platforms useful
modeling properties
(e.g. UML, MDA)

Getting from red
platforms to blue
platforms.

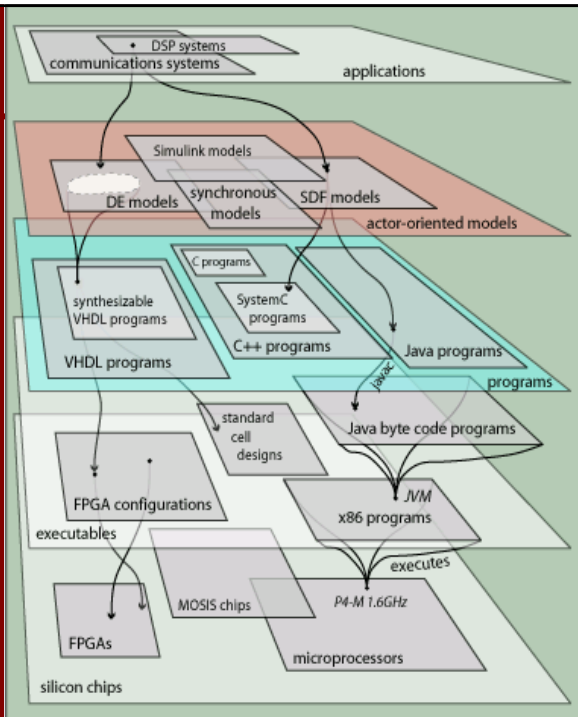


Platforms

Where the
Action Will Be:

Giving the red
platforms useful
modeling properties
(via models of
computation)

Getting from red
platforms to blue
platforms.



Design Framework

A *design framework* is a collection of platforms and *realizable relations* between platforms where at least one of the platforms is a set of *physically realizable designs*, and for any design in a user platform, the transitive closure of the relations from that design includes at least one physically realizable design.

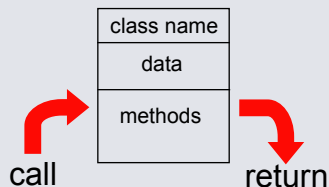
In *model-based design*, a *specification* is a point in a platform with useful modeling properties.

UC Berkeley, Edward Lee 5

Focus on Actor-Oriented Design

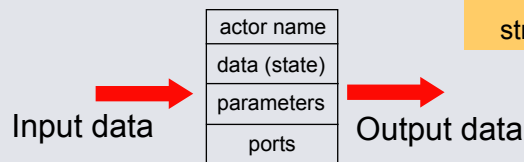
(vs. trying to give useful modeling properties to program-level designs)

- Object orientation:



What flows through an object is sequential control

- Actor orientation:



What flows through an object is streams of data

UC Berkeley, Edward Lee 6

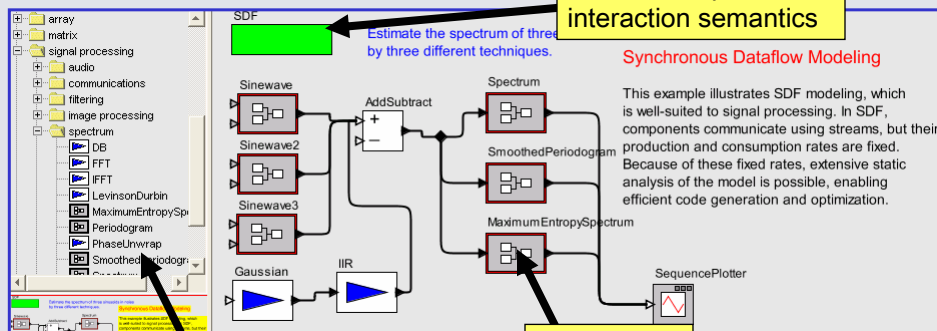
Examples of Frameworks with Actor-Oriented Mechanisms

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

UC Berkeley, Edward Lee 7

Example of Actor-Oriented Design (in this case, with a visual syntax)

Ptolemy II example:



Key idea: The model of computation is part of the framework within which components are embedded rather than part of the components themselves. Thus, components need to declare behavioral properties.

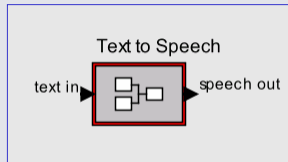
Model of Computation:

- Messaging schema
- Flow of control
- Concurrency

UC Berkeley, Edward Lee 8

Contrast Actor Orientation with Object Orientation

Actor oriented



actor-oriented interface definition says "Give me text and I'll give you speech"

Object oriented

TextToSpeech
initialize(): void
notify(): void
isReady(): boolean
getSpeech(): double[]

OO interface definition gives procedures that have to be invoked in an order not specified as part of the interface definition.

- Identified problems with object orientation:
 - Says little or nothing about concurrency and time
 - Concurrency typically expressed with threads, monitors, semaphores
 - Components tend to implement low-level communication protocols
 - Re-use potential is disappointing
- Actor orientation offers more potential for useful modeling properties, and hence for **model-based design**.

UC Berkeley, Edward Lee 9

Actor Orientation vs. Object Orientation

- Object Orientation
 - procedural interfaces
 - a class is a type (static structure)
 - type checking for composition
 - separation of interface from implementation
 - subtyping
 - polymorphism
- Actor Orientation
 - concurrent interfaces
 - a behavior is a type
 - type checking for composition of behaviors
 - separation of behavioral interface from implementation
 - behavioral subtyping
 - behavioral polymorphism

This is a vision of the future: Few actor-oriented frameworks fully offer this view. Eventually, all will.

UC Berkeley, Edward Lee 10

Will Actor-Oriented Design Yield Better Designs?

Not necessarily.

"Why isn't the answer UML, or XML, or IP, or something like that?"

Direct quote from a high-ranking decision maker at a large embedded systems company with global reach.

"New" is not better than "good"



The Box, Eric Owen Moss

Has the hardware design community been lead astray by the software engineering community?

Mandating use of the wrong platform is far worse than tolerating the use of multiple platforms.

UC Berkeley, Edward Lee 11

Source: *Contemporary California Architects*, P. Jodidio, Taschen, 1995

Better Architecture is Enabled but not Guaranteed by Actor-Oriented Design

Source: *Kaplan McLaughlin Diaz*, R. Rappaport, Rockport, 1998



Two Rodeo Drive, Kaplan, McLaughlin, Diaz

- Understandable concurrency
- Systematic heterogeneity
- More re-usable component libraries

UC Berkeley, Edward Lee 12