Freedom from Choice and the Power of Models

*In Honor of Alberto Sangiovanni-Vincentelli*

*Edward A. Lee*

*UC Berkeley*

*Keynote, International Symposium on Physical Design (IPSN)*

San Francisco, CA, April 16, 2019
Lee, Berkeley
Platform-Based Design

Application Space
Application Instance

Platform Mapping
Platform Design-Space Export

Architectural Space
Platform Instance

Lee, Berkeley
If this is not clear to you...

O ye possessed of sturdy intellects, observe the teaching that is hidden here beneath the veil of verses so obscure.

*Dante*

*Divine Comedy, Inferno, Canto 9*
Abstraction Rules

Requirement Formalization

Reﬁnement Rules

Contracts

Abstraction Rules

Composition Rules

Behavioral and Non-Functional Models

System Requirements

Performance

Reliability

Safety

Synthesis (Optimization)

Architectural Space:

Platform Library

Sensors

Actuators

Networks

Processors

Controllers
Swarmlets

Arbitrary platforms and middleware

Applications

Swarm-OS

Resources

Lee, Berkeley
Layered Platforms

Abstraction

Refinement

The “Thing In Itself”
back of the envelope

functional concurrency

abstract executable model

mixed signals

timing accurate analog

physical device

power

geometry

A draft “abstraction map”

Lee, Berkeley
more abstract

object models

performance models

real-time tasks

embedded software

functional concurrency

abstract executable model

mixed signals

power

geometry

timing accurate

analog

RTL

ODEs+events

physical device

A draft “abstraction map”
back of the envelope

more concrete

object models

performance models

real-time tasks

embedded software

functional concurrency

abstract executable model

mixed signals

timing accurate analog

RTL ODEs+events

physical device

data power

geometry

A draft “abstraction map”
input/output relations

object models

performance models

real-time tasks

eMBEDDED SOFTWARE

proc, mem, bus architecture

RT

ODEs + events

physical device

back of the envelope

functional concurrency

abstract executable model

mixed signals

timing accurate analog

power geometry
control structure, synchronization

back of the envelope

functional concurrency

object models

performance models

real-time tasks

eMBEDDED SOFTWARE

proc, mem, bus architecture

timing accurate

analog

abstract executable model

mixed signals

power

geometry

RTL

ODEs+events

physical device

A draft “abstraction map”
A draft “abstraction map”

- timing without function
- back of the envelope
- functional
  - concurrency
  - abstract
    - executable
      - model
- mixed signals
- timing accurate
- analog
- RTL
- ODEs+events
- power
- geometry
- proc, mem, bus architecture
- embedded software
- real-time tasks
- performance models
- object models
- physical device
- A draft
  - “abstraction map”

Lee, Berkeley
function + concurrency without timing

back of the envelope

functional concurrency

abstract executable model

mixed signals

timing accurate analog

proc, mem, bus architecture RTL

ODEs+events

physical device

power

geometry

real-time tasks

performance models

object models

eMBEDDED SOFTWARE

Lee, Berkeley

A draft “abstraction map”
Lee, Berkeley

Sample hardware flow

- back of the envelope
- functional
  - concurrency
  - abstract executable model
  - mixed signals
  - analog
  - timing accurate
  - RTL
  - ODEs+events
  - physical device

- object models
- performance models
- real-time tasks
- embedded software
- proc, mem, bus architecture
- power
- geometry
Concurrency is not considered. Synchronous circuit design provides the narrow waist.
The analog properties of the circuits are not considered. A particular process is taken as a given, and is modeled using gate delays.
Note simultaneous evolution of two different views of the design.

Sample software flow
back of the envelope

CPS Design Flow

object models

functional

concurrency

abstract executable model

mixed signals

power

timing accurate

geometry

real-time tasks

performance models

proc, mem, bus architecture

RTL

ODEs+events

physical device

embedded software

Lee, Berkeley
System integration occurs at the greatest lower bound. What we want is to work at the least upper bound.
Platforms

The purpose for an abstraction is to hide details of the implementation below and provide a platform for design from above.
Life Without Platforms

Every abstraction layer has failed in the fly-by-wire scenario.

The design is the implementation.
Modeling Framework Platform
“Freedom From Choice”
A model is any description of a system that is not the thing-in-itself. (das Ding an sich in Kantian philosophy).
A Modeling Framework

\[ x(t) = x(0) + \int_0^t v(\tau) d\tau \]
\[ v(t) = v(0) + \frac{1}{m} \int_0^t F(\tau) d\tau \]

In this example, the *modeling framework or platform* is calculus and Newton’s laws in a time and space continuum.

Image by Dominique Toussaint, GNU Free Documentation License, Version 1.2 or later.
The Map and the Territory

A few things we need to model to explain this behavior:
- Plastic deformation
- Acoustic propagation
- Stretching of strings
- ...
The Value of Models

• In *science*, the value of a *model* lies in how well its behavior matches that of the physical system.

• In *engineering*, the value of the *physical system* lies in how well its behavior matches that of the model.

A scientist asks, “Can I make a model for this thing?”
An engineer asks, “Can I make a thing for this model?”
Model Fidelity

- To a *scientist*, the model is flawed.
- To an *engineer*, the realization is flawed.

I’m an engineer...
Changing the Question

Is the question whether we can build models describing the behavior of our systems?

Or

Is the question whether we can build systems whose behavior matches that of our models?
Consider Chip Design

A piece of silicon that doesn’t behave like the model is just beach sand.

Intel Haswell, each with 1.4 billion transistors

Lee, Berkeley
Useful Models and Useful Things

“Essentially, all models are wrong, but some are useful.”


“Essentially, all system implementations are wrong, but some are useful.”

Lee and Sirjani, “What good are models,” FACS 2018.
The Value of Simulation

“Simulation is doomed to succeed.”

Could this statement be confusing engineering and scientific models?

Figure 1: Three scenes generated from a single ~20-line SCENIC scenario representing bumper-to-bumper traffic.


Lee, Berkeley
Towards *Engineering*-Model-Based Design

Per Barry Boehm:
- Am I building the right product? (validation)
- Am I building the product right? (verification)
Raphael: The School of Athens