
# EECS 144/244
# Fundamental Algorithms for System Modeling, Analysis, and Optimization

## Lecture 2: Design approaches

Sanjit Seshia, Stavros Tripakis

UC Berkeley

Fall 2013

# State-of-the-art embedded systems

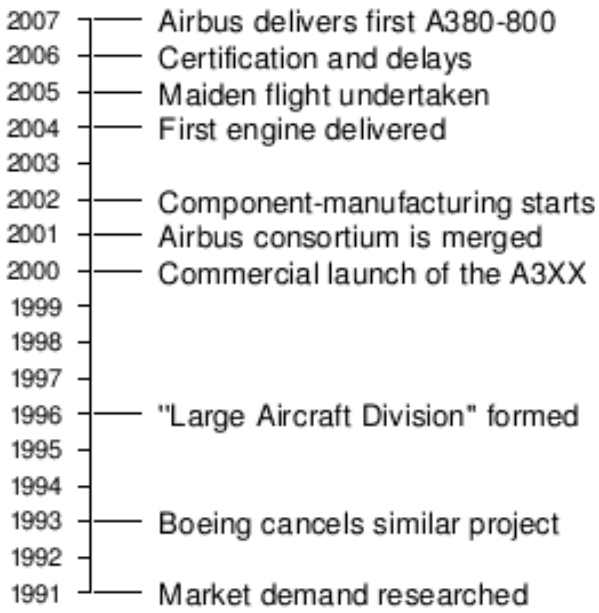
- Seem to fall under two distinct classes:

- Safety-critical (and for the most part dependable), but very expensive:
  - Satellites, airplanes, nuclear power plants, ...

- Cheap, but unreliable:
  - Consumer electronics, ...

# Safety-critical systems

- Airbus A380:
  - Development time: decades
  - Development cost: tens of billions




- Boeing 787
  - Development time: decades
    - Delays, delays … (first flight programmed initially for July 2007, finally done in December 2009)
    - Grounded for ~3 months in early 2013 due to battery problems
  - Cost: US$32 billion



Source: wikipedia

These systems are very expensive!

# Cheap systems

- Consumer electronics


These systems are not very dependable...

# Dependability & Affordability

- For future CPS we need both
- Safety critical systems => dependability
- Massively deployed => affordability

Our ultimate goal: how to achieve both?

# This course

- How can we design complex systems?
- *Fundamental Algorithms for System Modeling, Analysis, and Optimization*
- 3 elements:
  - Systems
    - System-oriented view: look at the system as a whole, then focus on its parts
  - Algorithms
    - Focus on computer-aided design techniques
  - Modeling, Analysis, and Optimization
    - Focus on **model-based design**

# SYSTEM DESIGN APPROACHES

# Approaches to system design (1)

- ***Trial-and-error*** approach:
  - Build prototype
  - Test it, find errors
  - Fix errors
  - Repeat
- Not good for dependability + affordability:
  - Too risky
  - Too costly
- Yet common...

# Design by trial-and-error

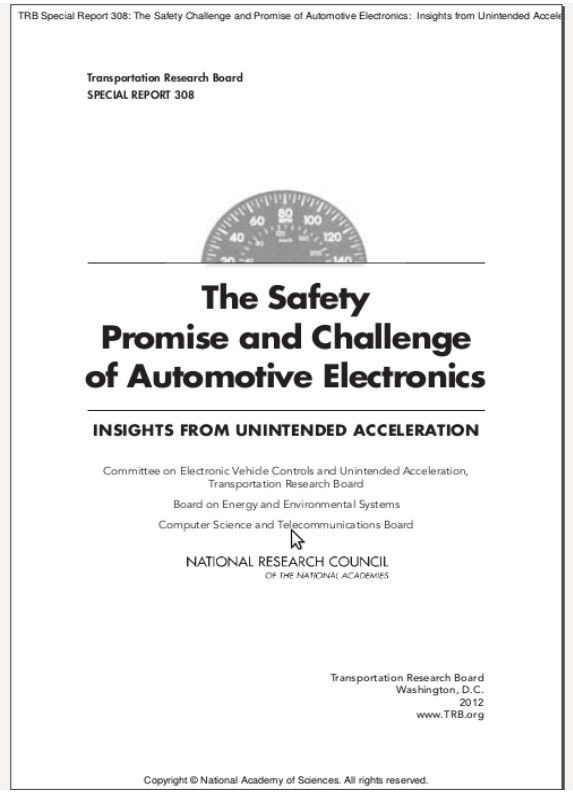
- Tacoma bridge:
http://www.youtube.com/watch?v=xox9BVSu7Ok&feature=related

# Design by trial-and-error

- Fukushima nuclear power plant:

# Design by trial-and-error

- Toyota unintended acceleration incidents
- Millions of cars recalled
- Cost: $ billions
- U.S. National Highway Transportation Safety Administration's (NHTSA) concluded that electronic throttle control systems were **not** the cause.



TRB Special Report 308: The Safety Challenge and Promise of Automotive Electronics: Insights from Unintended Accele

Transportation Research Board
SPECIAL REPORT 308

**The Safety Promise and Challenge of Automotive Electronics**

**INSIGHTS FROM UNINTENDED ACCELERATION**

Committee on Electronic Vehicle Controls and Unintended Acceleration, Transportation Research Board
Board on Energy and Environmental Systems
Computer Science and Telecommunications Board

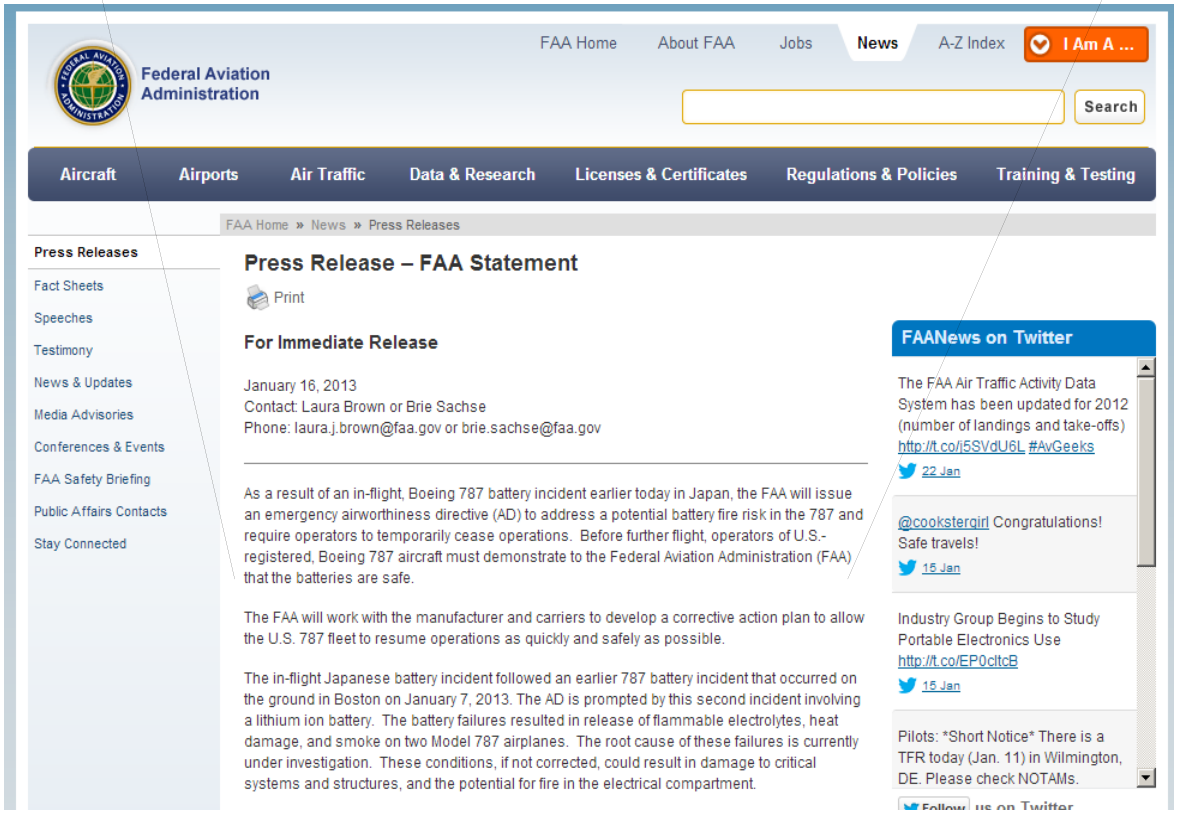
NATIONAL RESEARCH COUNCIL
OF THE NATIONAL ACADEMIES

Transportation Research Board
Washington, D.C.
2012
www.TRB.org

# Design by trial-and-error

- Boeing 787 grounded
- "*All-Nippon today announced it had canceled 320 flights, including 51 international flights, on 787s affecting a total of 46,800 passengers*" [San Jose Mercury News, 1/22/2013]
- FAA restriction finally lifted in April 2013.


*As a result of an in-flight, Boeing 787 battery incident earlier today in Japan, the FAA will issue an emergency airworthiness directive (AD) to address a potential battery fire risk in the 787 and require operators to temporarily cease operations.*
***Before further flight, operators of U.S.-registered, Boeing 787 aircraft must demonstrate to the Federal Aviation Administration (FAA) that the batteries are safe.***



# Design by trial-and-error

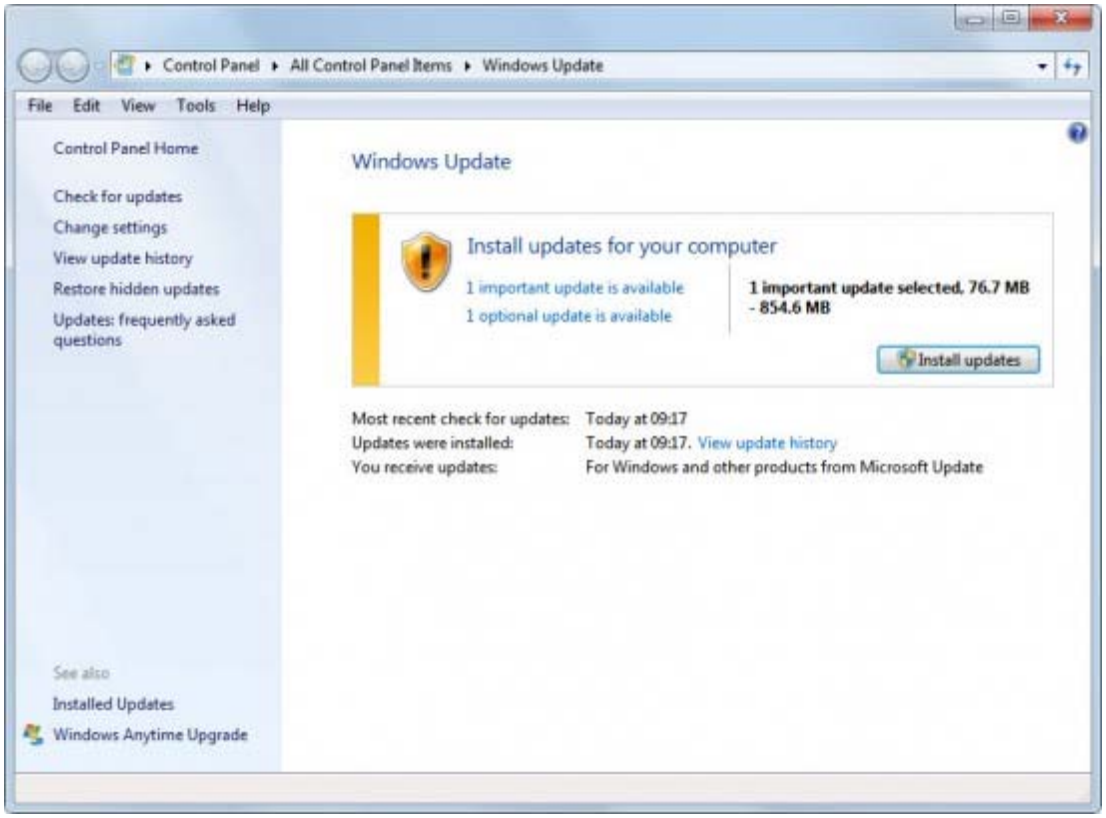
- Drugs (medical):
  - Theoretical candidates for “new chemical entity”: ~5000 – 10000
  - Of these, promising candidates on which lab/mice tests are run: ~250
  - Of these, ~10 qualify for tests on humans
  - Of these, ~20% make it to marketing
  - Source: http://en.wikipedia.org/wiki/Drug_development

# Design by trial-and-error

- Last but not least ...

# Design by trial-and-error

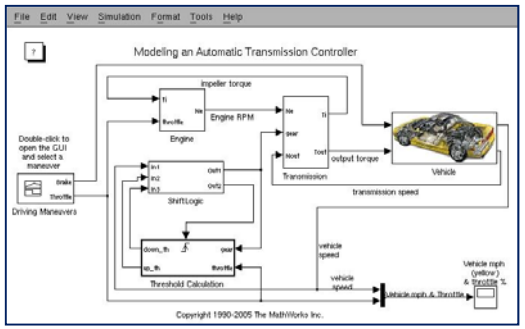
- Software!




# Approaches to system design (2)

- Rigorous, ***"model-based"*** design:
  - Build model ("executable specification") of system
    - before building the system itself
  - Analyze the model, find errors
  - Fix errors in the design (model)
  - Repeat until the design seems OK
  - Give models/specs to someone (or to a computer) to implement them
- Better for affordability:
  - Catch design errors early => easier / less costly to fix
- Better for dependability:
  - Sometimes can formally prove that design is correct
- Gaining acceptance in the industry

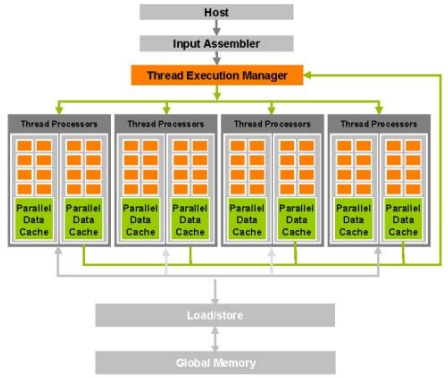
# The Elements of Model-Based Design




How to describe what we want?

Modeling

How to be sure that this is what we want?




Analysis

Implementation, Optimization

How to build it?
Automatically
Correct-by-construction

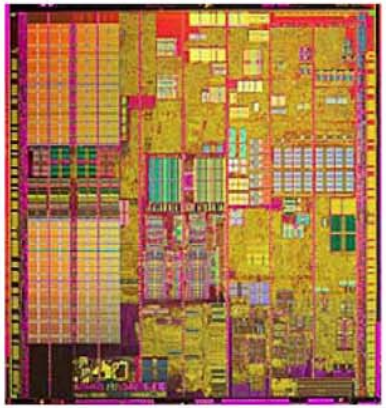
# From standard compilers ...

```
class HelloWorldApp {
  public static void
  main(String[] args) {
    System.out.println(
      "Hello World!");
  }
}
```

source code
(C, Java, …)

compiler

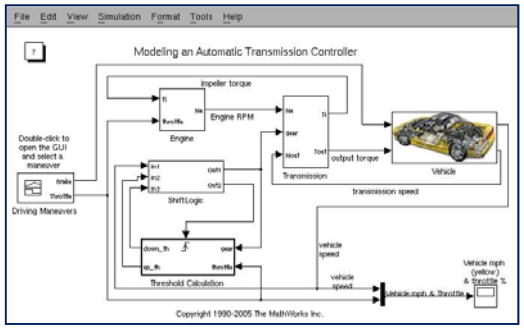
type checking, debugging, static analysis, …



machine code

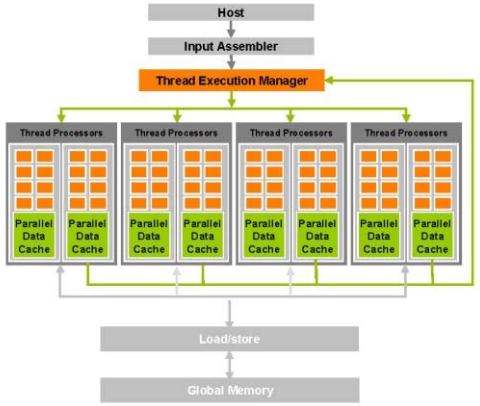
# … to system compilers

Vision: *modeling/simulation languages of today will become the **system-programming** languages of tomorrow*




Rich languages:
concurrency, time,
robustness, reliability,
energy, security, …

**system compiler**




Complex
execution platforms:
networked, distributed,
multicore, …

Powerful analyses:
model-checking, WCET analysis, schedulability,
performance analysis, reliability analysis, …

# Note

In real life, we need both MBD and trial-and-error methods.

Why?

1. We cannot trust our models 100%
2. All models are abstractions of reality. They make assumptions that need not hold.
   - E.g., road condition, weather condition, …
3. Analysis and optimization methods also have their limitations.
   - As we will see in this course.

# Model-based design is fine, but …

- There are many systems, of different kinds
- People have been designing these for decades
- Can we pretend to find a single design method that works for every kind of system?
- Of course not

# The thesis

Design is a SCIENCE:

- No matter what the application is, there are common features
- Foundational work in mathematical modeling, algorithms, methodologies

To demonstrate the value of design SCIENCE, apply the foundational work to a variety of areas from electronics, to automotive, to avionics, to intelligent buildings, to biological systems

**Raphael, The Athens School**

# Example of a successful model-based design flow

- RTL synthesis flow