Verification of Software and Hardware

Zohar Manna

Computer Science Dept.
Stanford University

Infamous Bugs: Medical

- Therac-25 Radiation Therapy (1986)
  - Computer-controlled radiation-therapy
  - More dependent on software for safety than predecessors
  - **Cause**: "race condition"
    - Miscoordination between concurrent tasks
    - Common type of software bug
  - **Cost**: 6 massive overdoses of radiation (2 fatal)

Infamous Bugs: Space

- Ariane 5, Flight 501 (June 4, 1996)
  - European-built rocket
  - Maiden flight: exploded 40 seconds after launch
  - **Cause**: Overflow condition
    - Reused Ariane 4 software
    - Error in code that converts 64-bit floating point number to 16-bit signed integer
    - Bigger engines cause 64-bit numbers to be larger than in Arianne 4
  - **Cost**: $120 million

Infamous Bugs: Space

- Mars Polar Lander (December 3, 1999)
  - Part of *Mars Surveyor* program
  - Failure probably during entry, deployment, landing
    - Leg deployment causes **transient signal** from touchdown sensors
  - **Cause**:
    - Behavior was understood and expected
    - But software specification failed to describe event
    - Software implementation interpreted signal as touchdown
  - **Cost**: over $100 million

Infamous Bugs: Space

- Mars Climate Orbiter (September 23, 1999)
  - Part of *Mars Surveyor* program
  - Purpose: Observe climate of Mars
  - Burned up in Mars's atmosphere instead of entering orbit
  - **Cause**: metric/English unit confusion
  - **Cost**: over $100 million
- More: Mariner I, Phobos I, Titan IV B-32, ...

Infamous Bugs: Space

"Almost all software-related aerospace accidents have been related to flawed requirements...."

"The software performed exactly as the designers intended..., but the designed behavior was not safe from a system viewpoint."

—Dr. Nancy Leveson, MIT

From *The Role of Software in Space Accidents*, AIAA Journal of Spacecraft and Rockets

Infamous Bugs: Cars

- Transition from mechanical to electronic control
- Many bugs in embedded systems
  - Toyota Prius: stall at highway speeds
  - Toyota Camry: unintended acceleration
  - Mercedes-Benz: braking failure
  - Jaguar: slip into reverse gear
  - Range Rover: suspension failure at highway speeds
- All result from errors in the embedded software of the "smart cars"

Infamous Bugs: Hardware

- Intel Pentium FDIV Bug (1993)
  - In the Pentium processor's floating point unit (FPU)
  - Division error in floating-point computations:
    - 4195835.0 / 3145727.0 yields 1.333**74**, not 1.333**82**
  - **Cause**: error in table (5 entries out of 1066)
  - **Cost**:
    - Initial reaction: "no big deal"
    - Public relations nightmare
    - Finally cost Intel $475 million to replace all affected chips

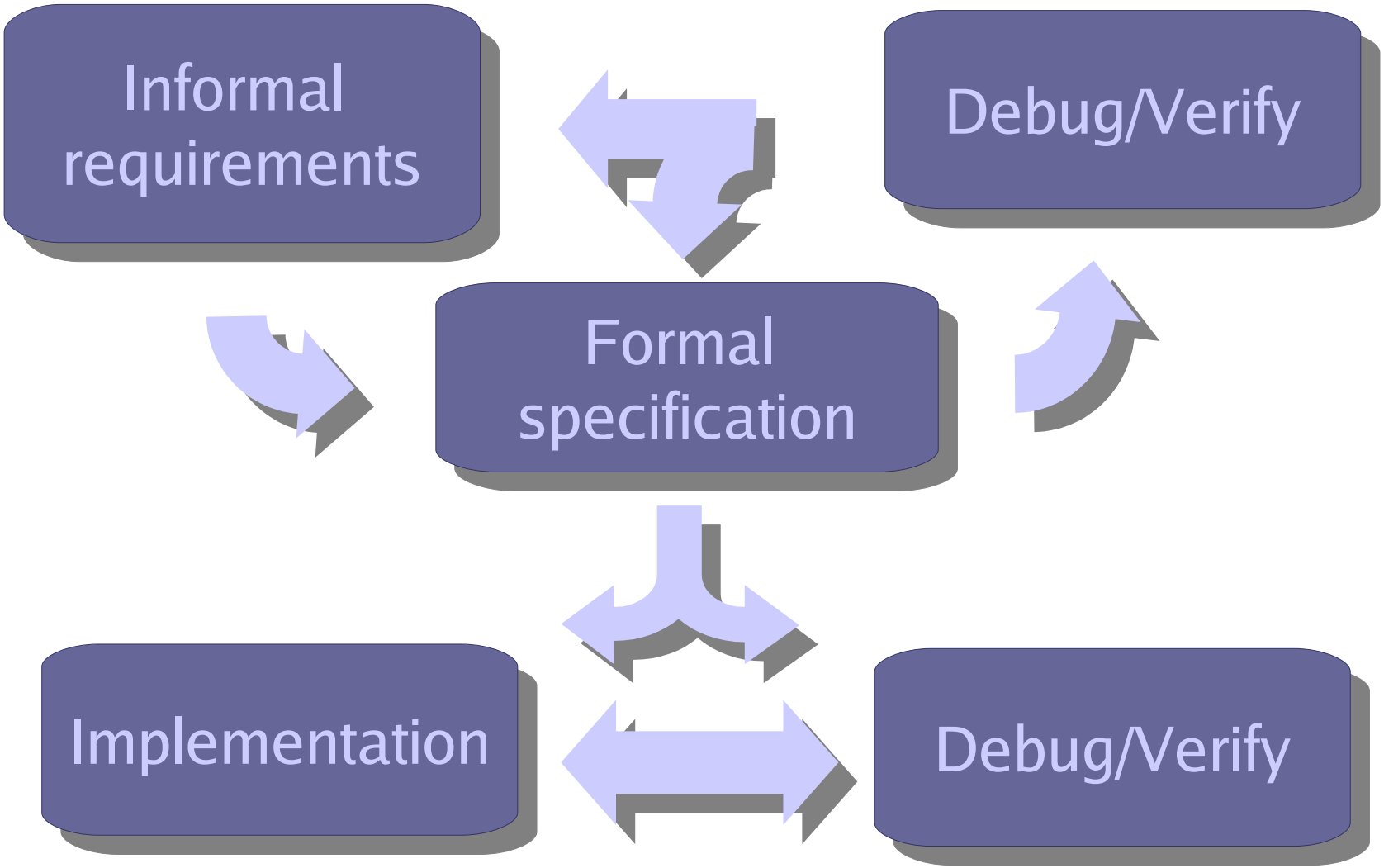
Infamous Bugs: Software

- Everyone has their favorite example
- Security is now the main concern

Formal Methods: Overview

- Goals
  - Verification:
    - **Prove** that specifications are consistent ("make sense")
    - **Prove** that implementation obeys specification
  - Debugging:
    - **find** counterexamples to specification

Engineering Lifecycle



Formal Methods: Hardware

- Finite state
- Algorithmic method
  - Model checking: over finite state graphs
    - **Prove** property, or
    - **Find** counterexample
- Applied in industry (*e.g.*, Intel, IBM, Synopsis)

Formal Methods: Software

- Infinite state
- Deductive method
  - Reduction to first-order assertions
  - Theorem proving
- Combination deductive/algorithmic method
  - Model check a finite abstraction of software
  - Applied in industry (*e.g.*, Microsoft)

U.S. Industry in Israel

- Intel: Intel Development Center (Haifa)
- IBM: IBM Research Labs (Haifa)

Cooperation with academia:

- Collaboration
- Research grants
- Student fellowships
- Student training

U.S. Industry in Israel

- Microsoft: No research center yet. Why?
  - Bill Gates (1995, FOCUS magazine): on bugs
    - "There are no significant bugs in our released software that any significant number of users want fixed."
    - "We don't do a new version to fix bugs. We don't. Not enough people would buy it."
  - Bill Gates (2002, Windows Engineering Conference): on software model checking
    - "the holy grail of computer science"
    - "Now... we're building tools that can do actual proofs... to guarantee the reliability."

U.S.-Israel Academic Collaboration

- Orna Grumberg (Technion)
  Ed Clark (CMU)
- Orna Kupferman (Hebrew U.)
  Moshe Vardi (Rice U.)
- Amir Pnueli (Weizmann Inst.)
  Zohar Manna (Stanford U.)
- Mooly Sagiv (Tel Aviv U.)
  Tom Ball (Microsoft)
- Dan Dolev (Hebrew U.)
  Joe Halpern (Cornel U.)

• • •