



iCyPhy



Software Design for Cyber-Physical Systems

Edward A. Lee

Module 1: Introduction to CPS

Technical University of Vienna
Vienna, Austria, May 2022



University of California, Berkeley

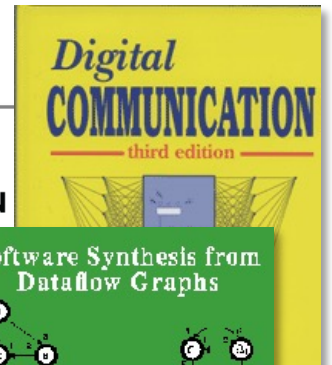
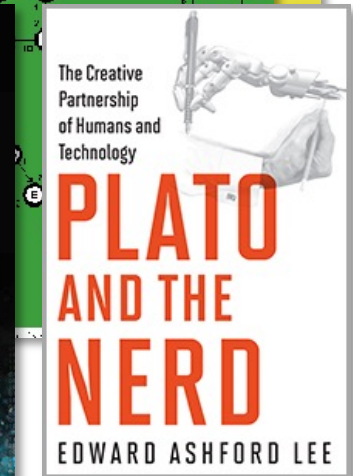
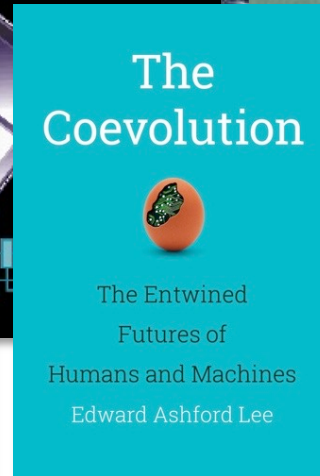
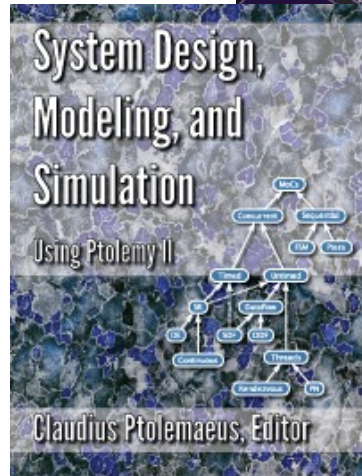


Introducing Edward A. Lee



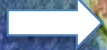
- BS (Yale), SM (MIT), PhD (Berkeley)
- Bell Labs in the early 1980s
- Berkeley EECS faculty since 1986
- Working on embedded software since 1978
- Director of iCyPhy, Industrial Cyber-Physical Systems Research Center
- Director of the Ptolemy project
- Former Chair of EECS, Berkeley
- Co-founder of BDTI, Inc.
- Lead on Lingua Franca
- Books...

<http://ptolemy.org/~eal>
eal@berkeley.edu





Location





The University of California at Berkeley





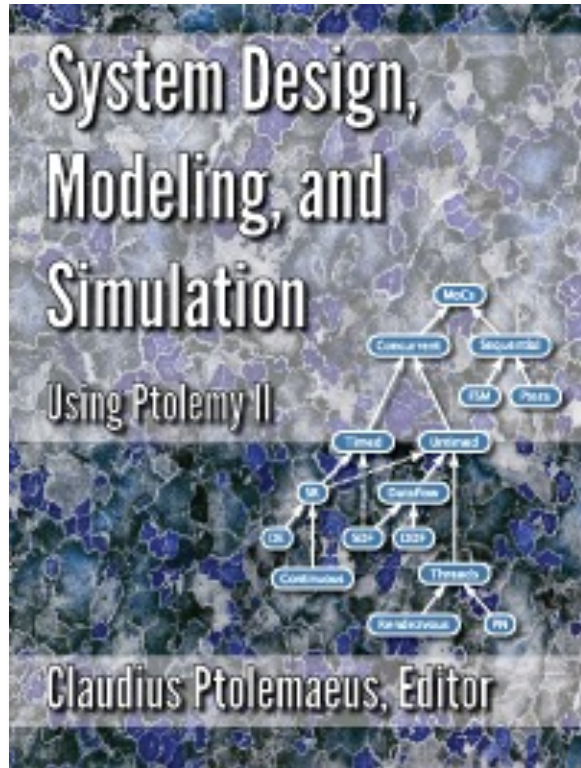
Disclaimer

This is not a survey of the field.

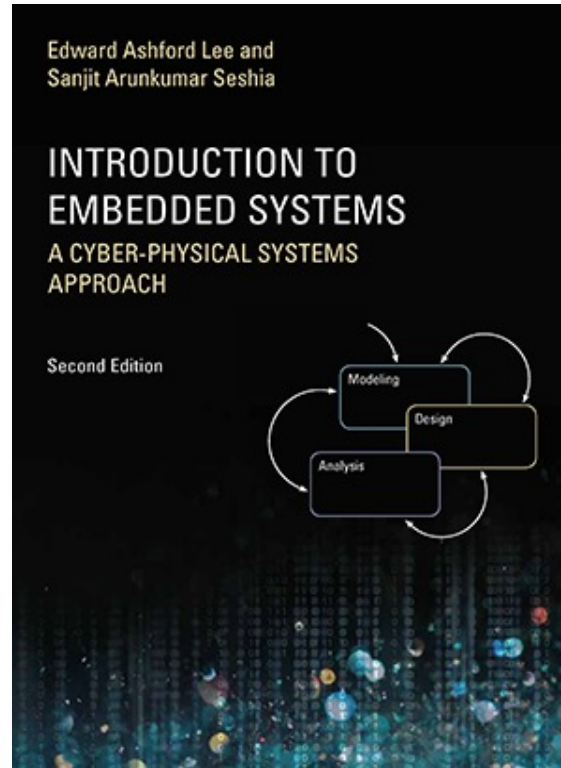
I will give you a narrow Berkeley view with a lot of opinions and personal perspectives.



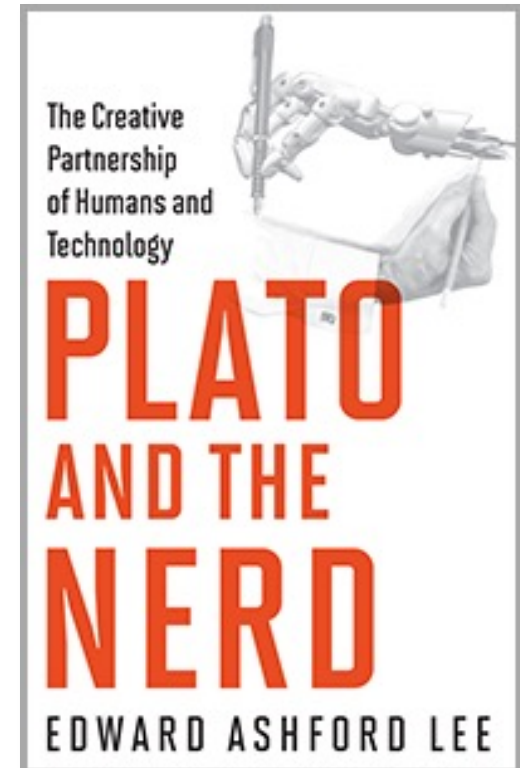
Resources



<http://ptolemy.org/systems>



<http://leeseshia.org>



<http://platoandthenerd.org>



Class Website



<https://ptolemy.berkeley.edu/~eal/cps/>



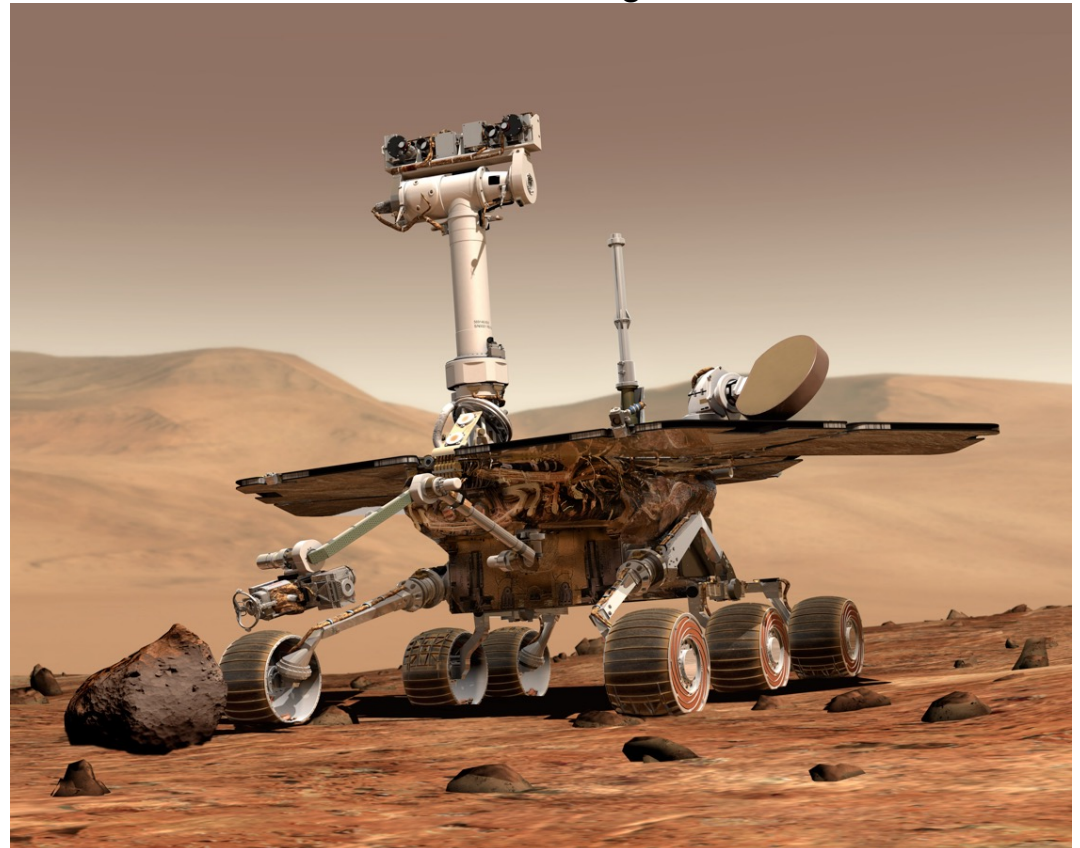
Cyber-Physical Systems

Orchestrating networked computational resources and physical systems.

Image: Wikimedia Commons

Roots:

- Term coined around 2006 by Helen Gill at the National Science Foundation in the US.
- **Cyberspace**: attributed William Gibson, who used the term in the novel Neuromancer.
- **Cybernetics**: coined by Norbert Wiener in 1948, to mean the conjunction of control and communication.

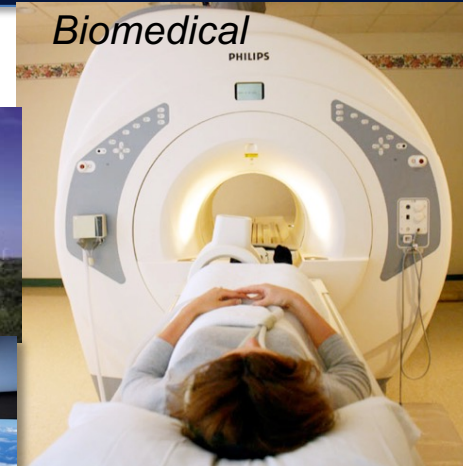
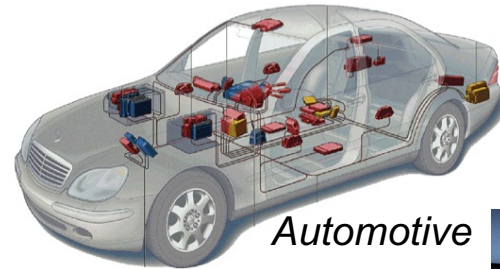




Cyber-Physical Systems

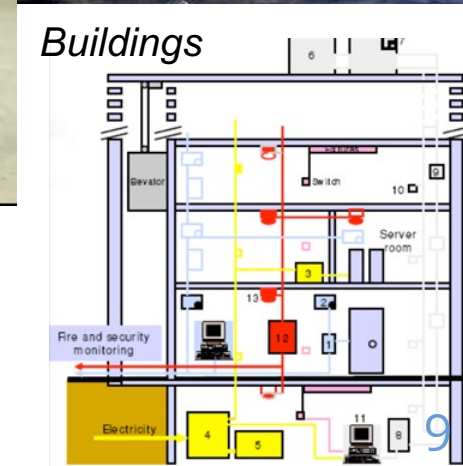
Not just information technology:

- Cyber + Physical
- Computation + Dynamics
- Security + Safety



Properties:

- Highly dynamic
- Safety critical
- Uncertain environment
- Physically distributed
- Sporadic connectivity
- Resource constrained



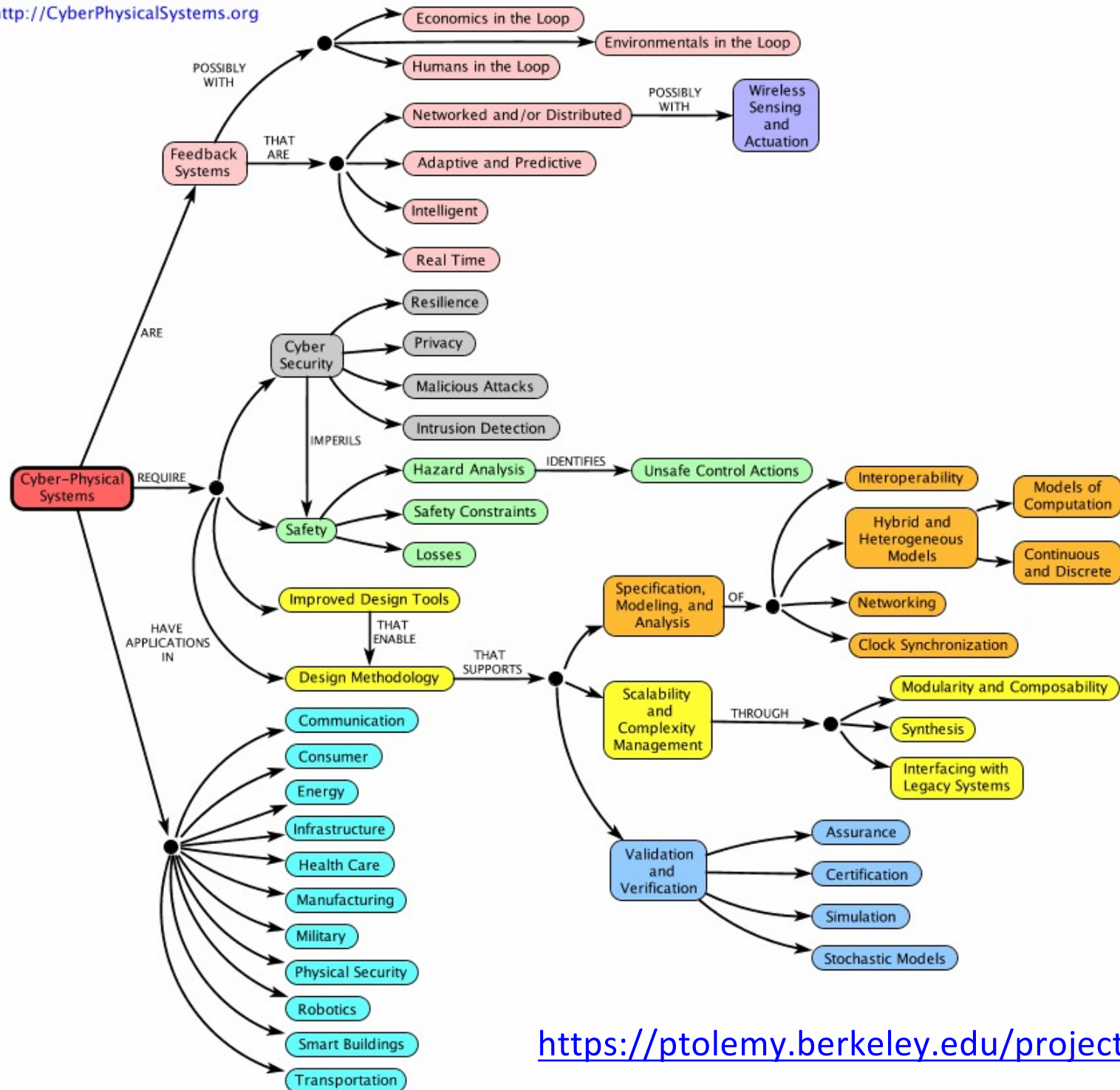
We need engineering **models** and **methodologies** for dependable cyber-physical systems.



Cyber-Physical Systems - a Concept Map

See authors and contributors.

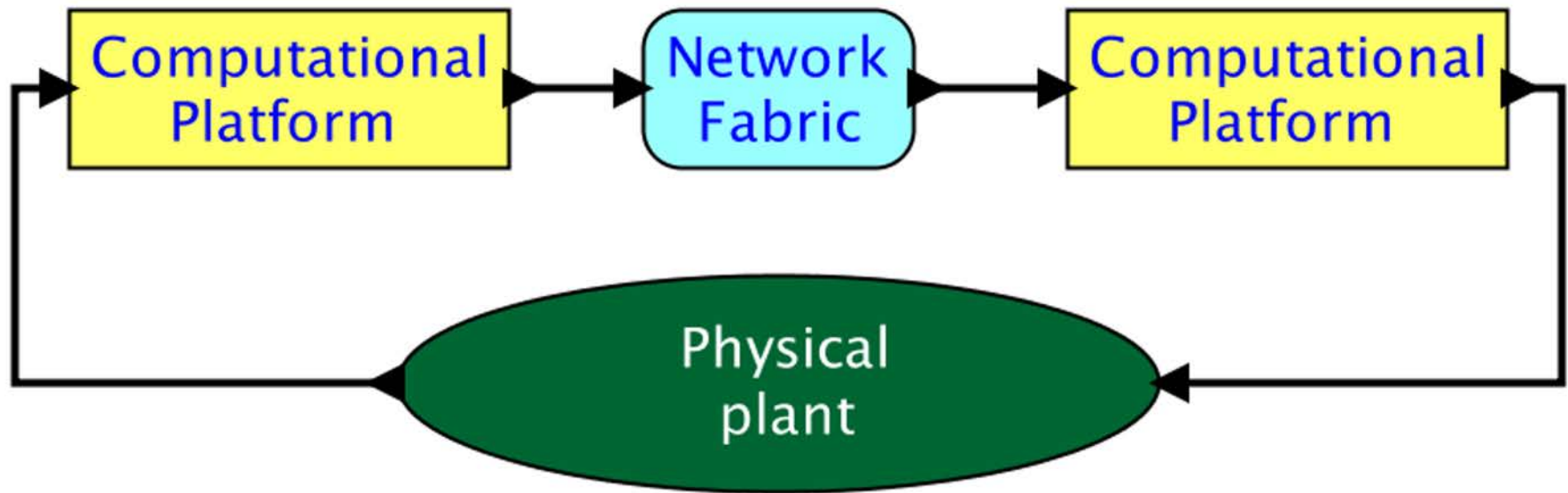
<http://CyberPhysicalSystems.org>



<https://ptolemy.berkeley.edu/projects/cps/>



Cyber-Physical Systems Pattern



Often safety critical, real time, and resource constrained.



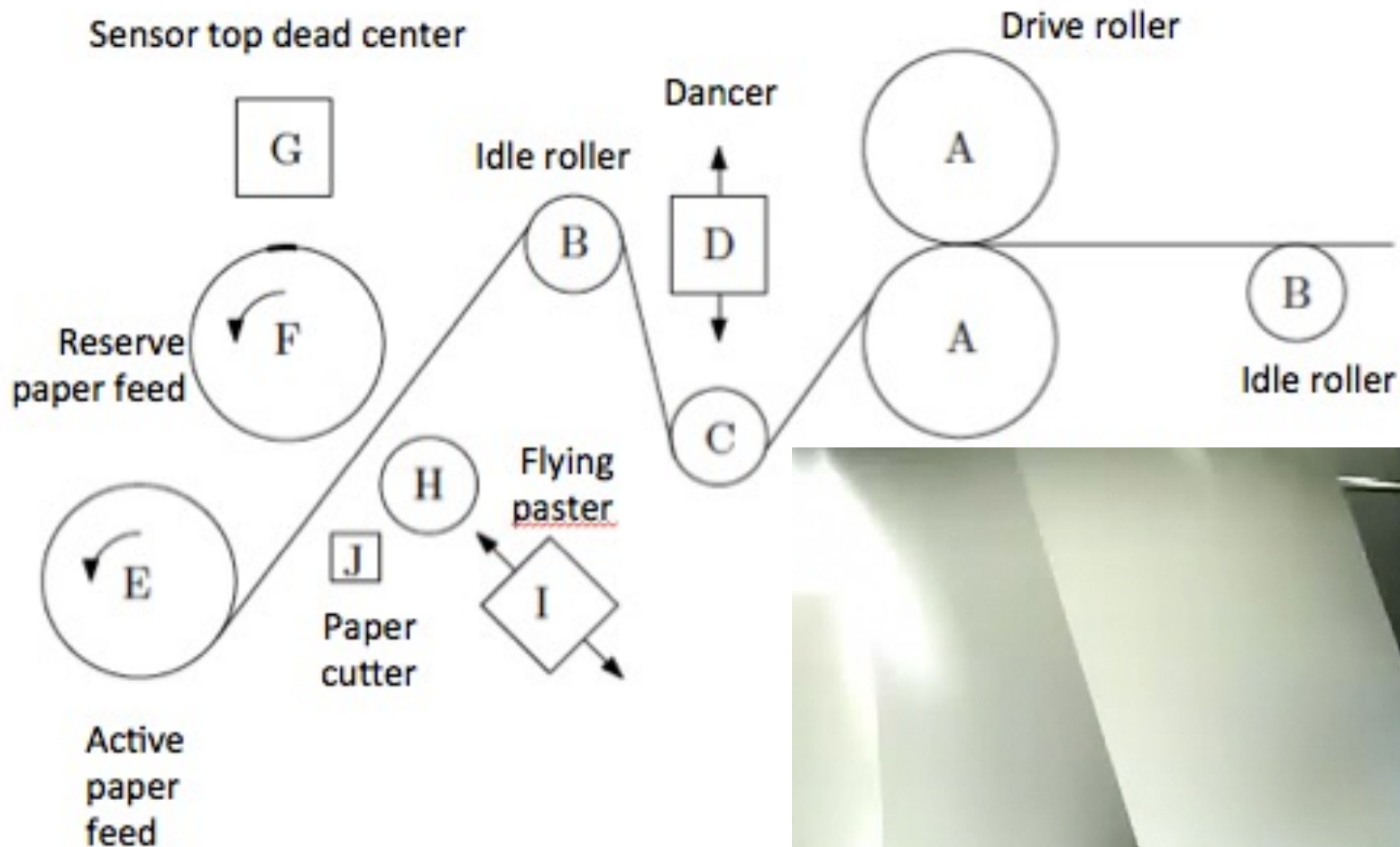
Example

Hundreds of microcontrollers orchestrating depositing ink and slicing paper flying through the machine at 100 km/hr.





Example – Flying Paster



Source: <http://offsetpressman.blogspot.com/2011/03/how-flying-paster-works.htm>



Example – Flying Paster



Source: <http://offsetpressman.blogspot.com/2011/03/how-flying-paster-works.html>



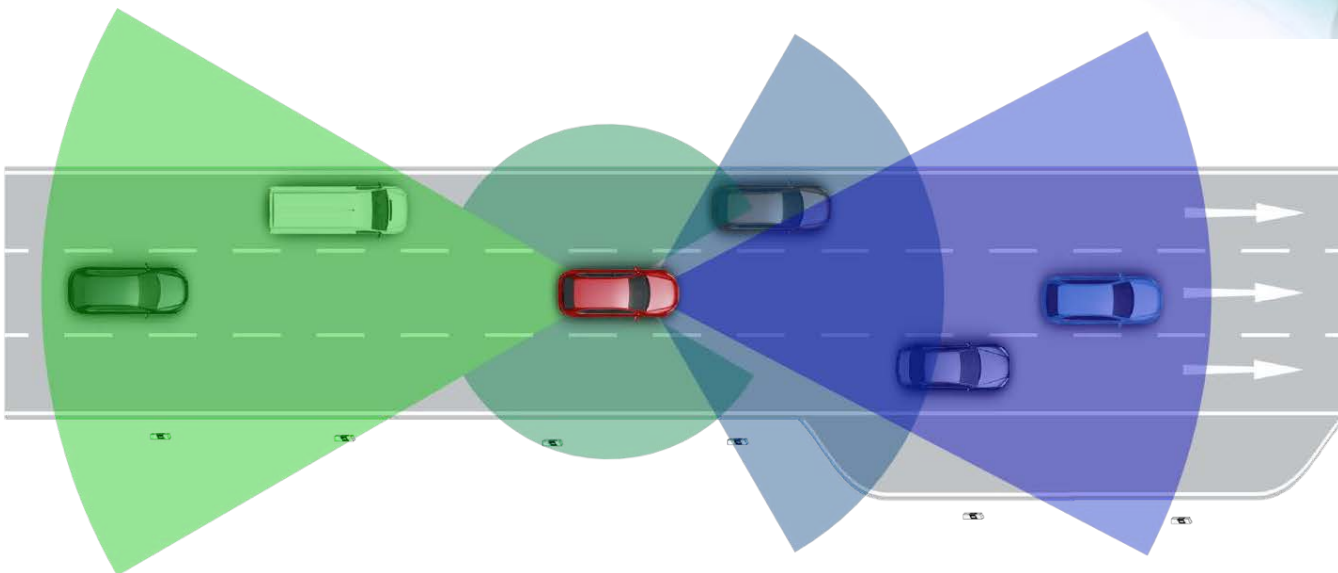
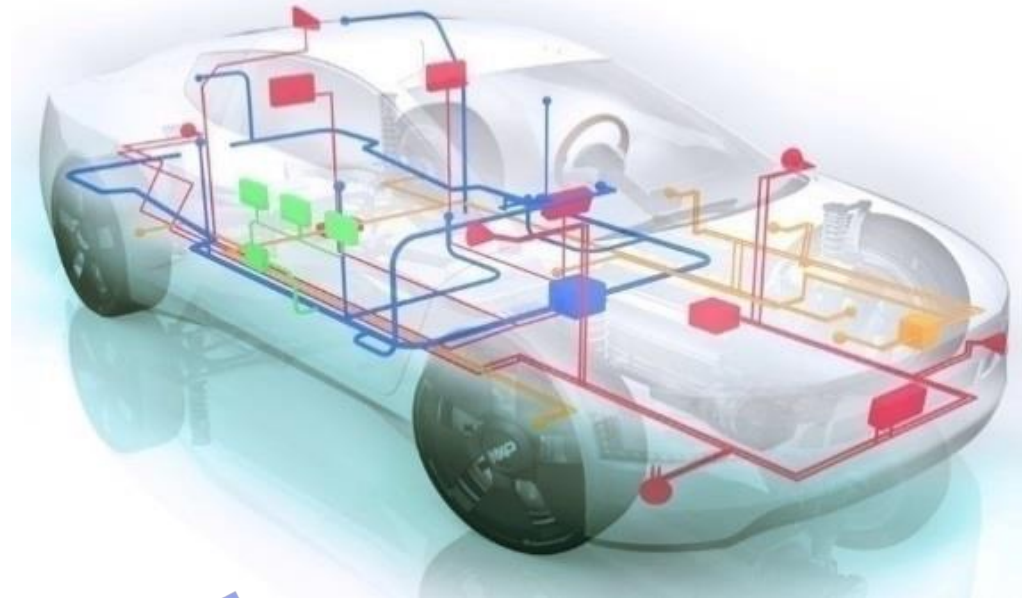
CPS Challenge Problem: *Prevent This*





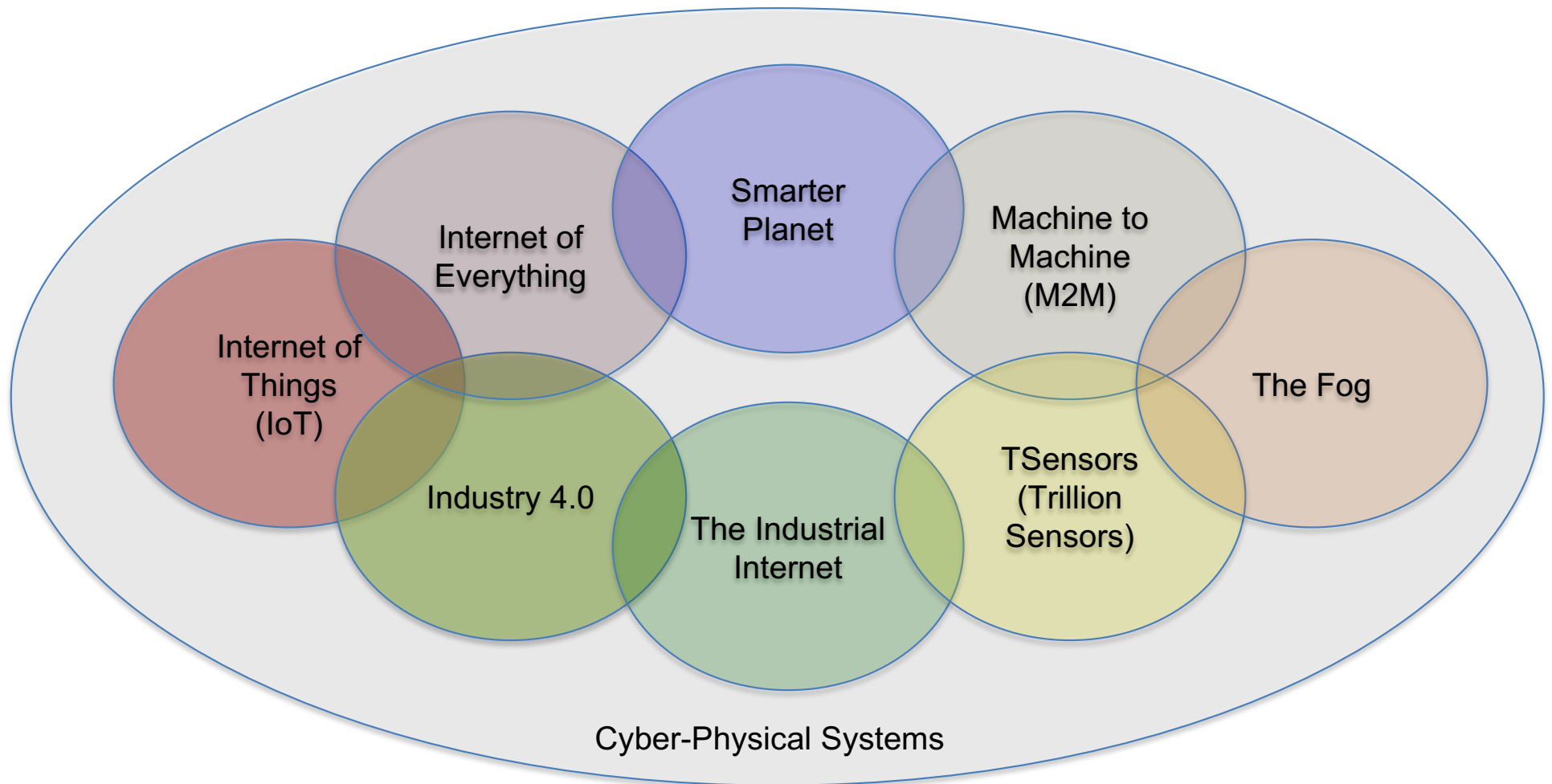
Automotive CPS and Societal Challenges

- Safer Transportation
- Reduced Emissions
- Smart Transportation
- Energy Efficiency
- Climate Change
- Human-Robot Collaboration





CPS-related terms






Challenges



Software Problems

A close-up photograph of a smartwatch on a person's wrist. The watch face is illuminated and shows a white error message on a dark background. The message reads "Unfortunately, Clock has stopped." with an "OK" button below it. The watch has a dark strap and a circular case.

Unfortunately, Clock
has stopped.

OK



Bad Design



Matt Haughey

@mathowie



Follow

If you need cheering up today, know that using my new IoT bike lock that only works with a phone took me 10min to unlock my bike after lunch



My egg tray doesn't like my Wi-Fi network. That may sound like a Mad Lib, but I'm serious. It took me 15 minutes to correctly pair Quirky's \$15 Egg Minder with the iPhone app, which gives you a count of remaining eggs. Yet when I removed eggs from the tray to make breakfast, one of them remained virtually present. I guess you could say the app was... scrambled.





Updates



Lightbulb firmware update

“My bulbs are at 7E. I keep getting prompted to update every once in a while. About 70% of the time I get an upgrade failed message. The rest of the time I get the update completed message, but bulbs still show 7E. “



Lifespan



Segal Lock.
Lifespan: ~100 years

Lee, Berkeley, with thanks to Bjoern Hartmann



August Bluetooth Lock.
Lifespan:?



Irreproducible Results



This would eventually become a recurring theme with my thermostat. In the middle of winter it began disconnecting, frequently overnight — even when there was a solid internet connection — and didn't have a backup mode. I'd wake up seeing my own breath, then spend hours rebooting the thermostat, boiler, and router to get it working again. The only way to control the gadget is via the app, so when it breaks you're really screwed.

The thermostat company later released a second version of its device with a wall control to avoid that no-backup-when-app-breaks situation, but it was another \$150 on top of what I'd already spent trying to bring smarts to my heating. Out of frustration, I got it anyway.



Lee, Berkeley, with thanks to Bjoern Hartmann



Security Risk

New Weapons Used in Attack On the Internet

The New York Times

© 2016 The New York Times Company NEW YORK, SATURDAY, OCTOBER 22, 2016

By NICOLE PERLROTH

SAN FRANCISCO — Major websites were inaccessible to people across wide swaths of the United States on Friday after a company that manages crucial parts of the internet's infrastructure said it was under attack.

Users reported sporadic problems reaching several websites, including Twitter, Netflix, Spotify, Airbnb, Reddit, Etsy, SoundCloud and The New York Times.

The company, Dyn, whose servers monitor and reroute internet traffic, said it began experiencing what security experts called a distributed denial-of-service attack just after 7 a.m. Reports that many sites were inaccessible started on the East Coast, but spread westward in three waves as the day wore on and into the evening.



Lee, Berkeley



Problems are not just annoying

NASA's Toyota Study Released by Dept. of Transportation released in 2011 found that Toyota software was “untestable.”

Possible
victim of
unintended
acceleration.





Avionics



What is assurance?

- Software is correct?
- Compiler is correct?
- Microprocessor is correct?

Correct execution of correct software provides little assurance.



A Simple Challenge Problem

A software component on a microprocessor in an aircraft door provides two network services:

1. “open”
2. “disarm”

Assume state is closed and armed.

What should it do when it receives a request “open”?



Image by Christopher Doyle from Horley, United Kingdom - A321 Exit Door, CC BY-SA 2.0



A Simple Challenge Problem

A software component on a microprocessor in an aircraft door provides two network services:

1. “open”
2. “disarm”

Assume state is closed and armed.

What should it do when it receives a request “open”?

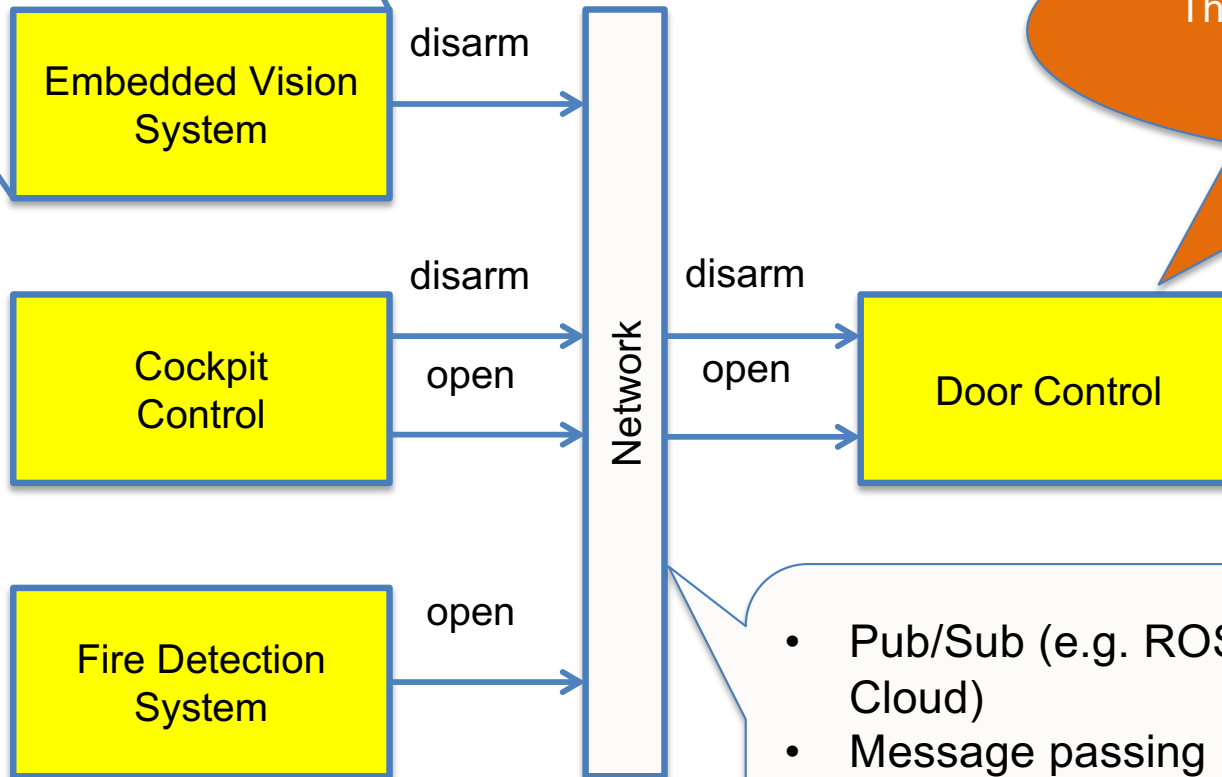


Image from *The Telegraph*, Sept. 9, 2015

Possible Architectures



Realized with an NI



The question: What to do upon receiving "open"?

- Pub/Sub (e.g. ROS, MQTT, Azure, Google Cloud)
- Message passing (e.g. Akka, Erlang)
- Service-oriented architecture (e.g. gRPC)
- Shared memory (e.g. Linda)



Some Solutions (?)

1. **Just open the door.**

How much to test? How much formal verification? How to constrain the design of other components? The network?

2. **Send a message “ok_to_open?” Wait for responses.**

How many responses? How long to wait? What if a component has failed and never responds?

3. **Wait a while and then open.**

How long to wait?

Better go read all of
Lamport's papers.





Fix with formal verification?

One possibility is to formally analyze the system.

Properties to verify:

1. If Door receives “open,” it will eventually open the door, even if all other components fail.
2. If any component sends “disarm” before any other component sends “open,” then the door will be disarmed before it is opened.

Can these be satisfied?



Fix with formal verification?

One possibility is to formally analyze the system. Properties to verify:

1. If Door receives “open,” it will eventually be opened, even if all other components fail.
2. If any component sends “disarm” before any other component sends “open,” then the door will be disarmed before it is opened.

Makes a distributed-consensus solution challenging.

Can these be satisfied?

Requires comparing times of events on distributed platforms in a model of computation that lacks time.



Can these properties be satisfied?

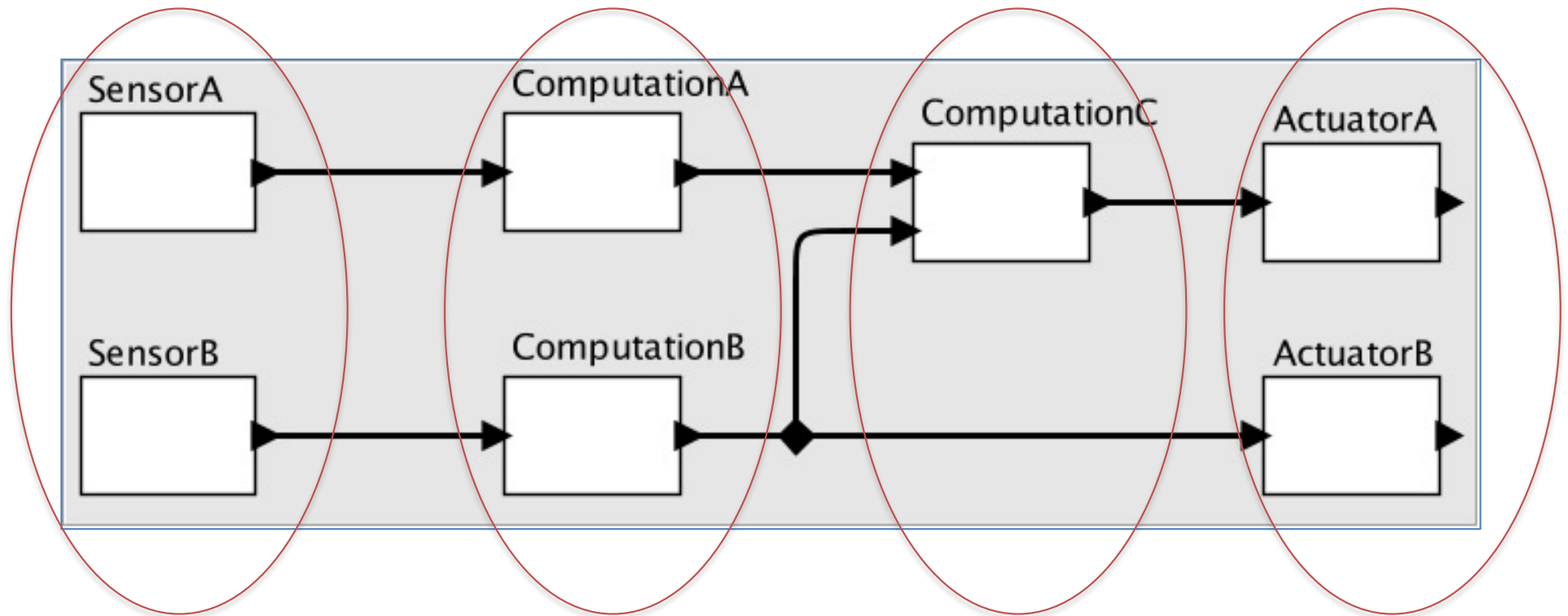
Properties to verify:

1. If Door receives “open,” it will eventually open the door, even if all other components fail.
2. If any component sends “disarm” before any other component sends “open,” then the door will be disarmed before it is opened.

My claim: These two cannot be satisfied without additional assumptions (e.g. bounds on network latency and/or clock synchronization).



A Broader Set of Questions



What combinations of periodic, sporadic, arrival behaviors are manageable?

How do execution and communication times affect feasibility? How can we know these times?

How do we get repeatable and testable behavior even when communication is across networks?

How do we specify, ensure, and enforce deadlines?



Conclusions

Cyberphysical systems represent a challenging design space requiring new techniques.