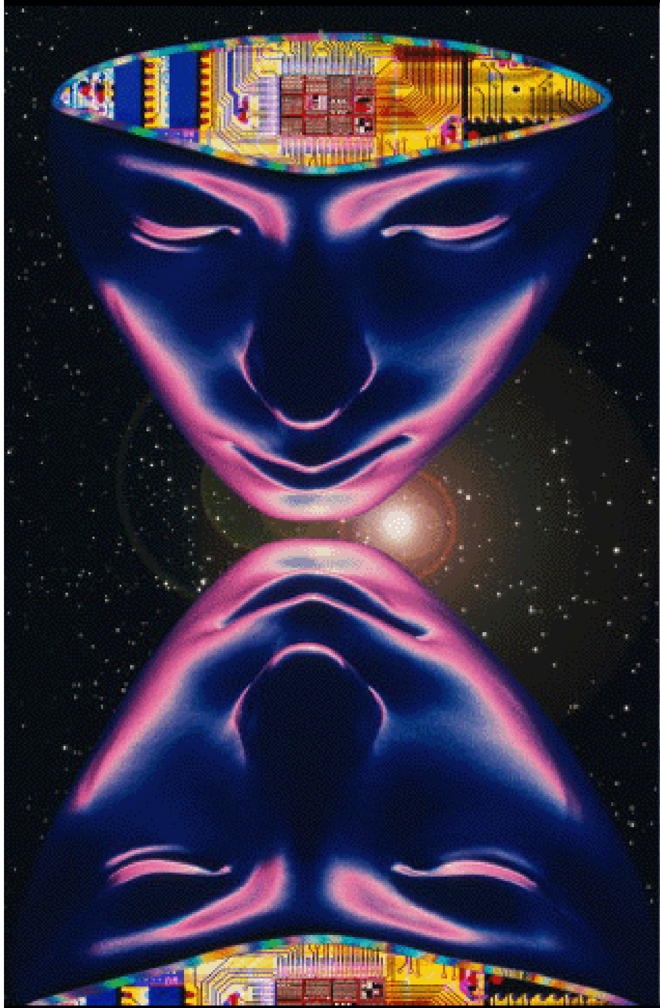


EE249
Embedded System Design: Models, Validation and Synthesis

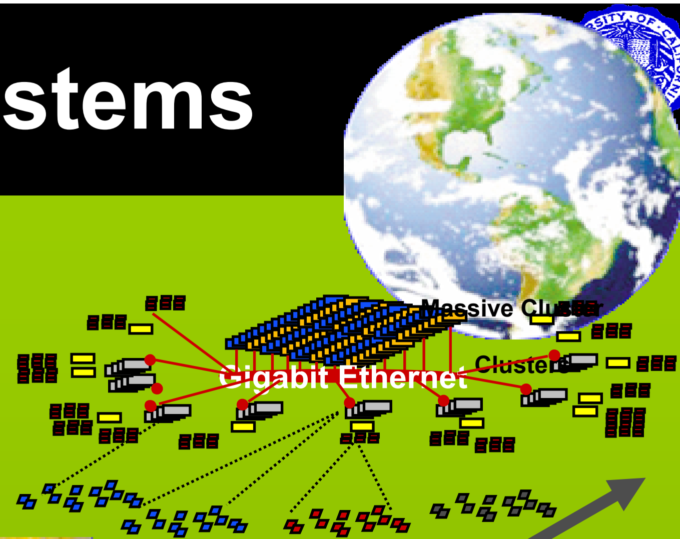
Alberto Sangiovanni-Vincentelli

"I believe we are now entering the Renaissance phase of the Information Age, where creativity and ideas are the new currency, and invention is a primary virtue, where technology truly has the power to transform lives, not just businesses, where technology can help us solve fundamental problems."

Carly Fiorina, CEO, Hewlett Packard Corporation

eMerging Societal-Scale Systems

New System Architectures
New Enabled Applications
Diverse, Connected, Physical,
Virtual, Fluid



Embedded Software Systems

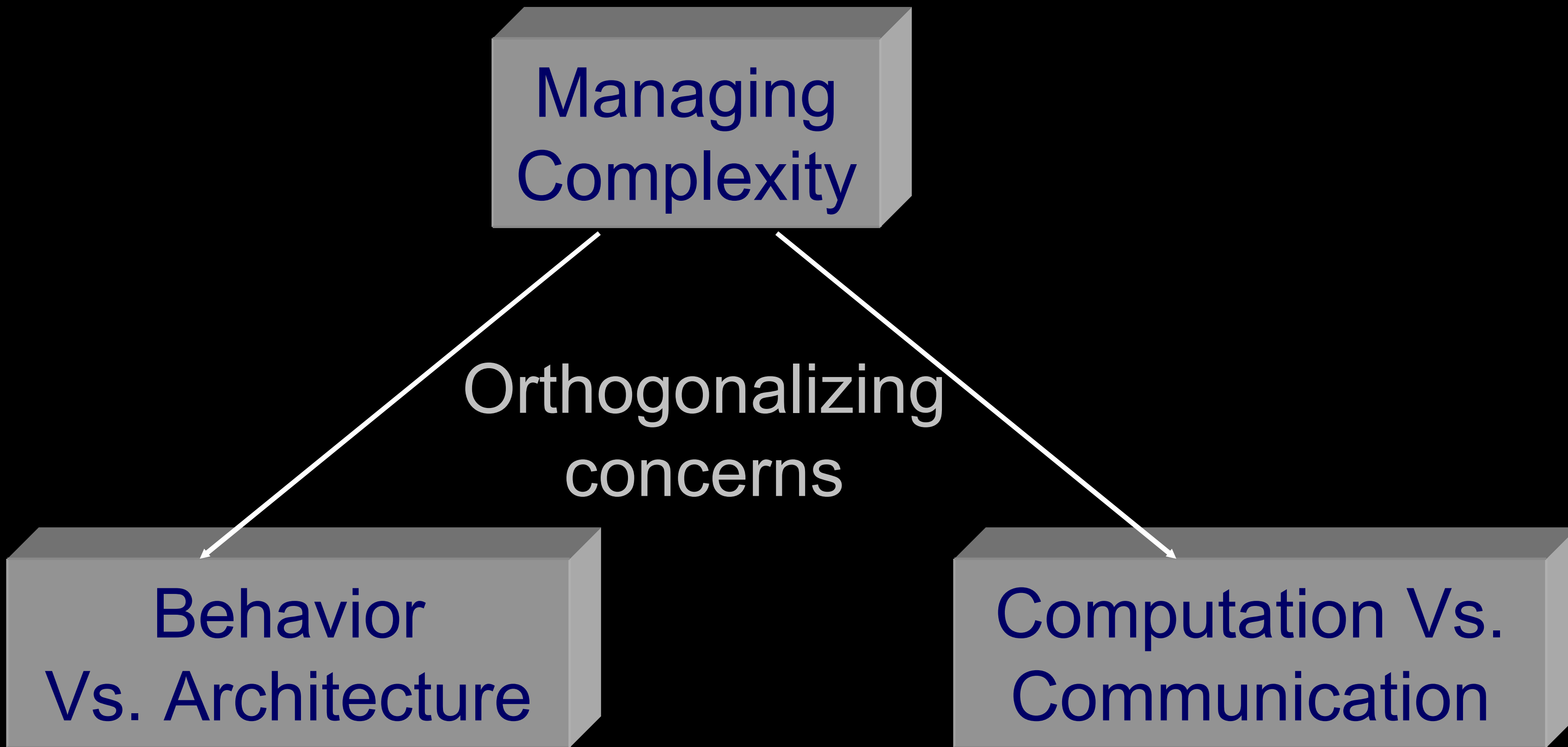
- Computational
  - but not first-and-foremost a computer
- Integral with physical processes
  - sensors, actuators
- Reactive
  - at the speed of the environment
- Heterogeneous
  - hardware/software, mixed architectures
- Networked
  - shared, adaptive

Source: Edward A. Lee

Observations

- We are on the edge of a revolution in the way electronics products are designed
- System design is the key (also for IC design!)
  - Start with the highest possible level of abstraction (e.g. control algorithms)
  - Establish properties at the right level
  - Use formal models
  - Leverage multiple “scientific” disciplines

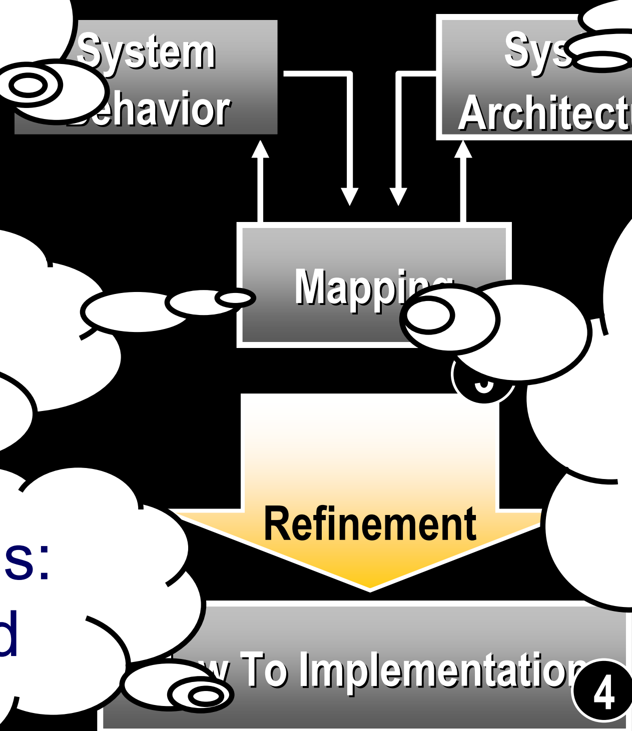
Course overview



Behavior Vs. Architecture

Models of Computation

Comm. and comp. resources



Quantity estimation

Synthesis: HW and SW

Assign functionality to arch elements HW/SW partitioning, Scheduling

- Polis (1990-1996)
- VCC (1996-2003)

Behavior Vs. Communication

- Clear separation between functionality and interaction model
- Maximize reuse in different environments, change only interaction model

Administration

- Office hours: *Alberto : Tu-Th 12:30pm-2pm or (better) by appointment (2-4882)*
- Teaching Assistant:
  - Alessandro PINTO, apinto@eecs.berkeley.edu

Grading

- Grading will be assigned on:
  - Homework (~30%)
  - Project (~50%)
  - Reading assignments (~10%)
  - Labs (10%)
- There will be approx. 7 homework (due 2 weeks after assignment) and 6 reading assignments

Discussion sections

- Lab section (Th. 4-6):
  - tool presentations
- Discussion Session (Tu. 5-6)
  - students' presentation of selected papers
    - Each student will be required to fill in a questionnaire in class for each discussion session
    - Each student (in groups of 2-3 people) will have to make an oral presentation once during the class
- Auditors are OK but please register as P-NP

Week	Lab Sections	Homeworks
1	- - -	- - -
2	Tool presentation	HW1
3	Discussion	
4	Tool presentation	HW2
5	Discussion	
6	Tool presentation	HW3
7	Discussion	
8	Tool presentation	HW4
9	Discussion	
10	Tool presentation	HW5
11	Discussion	
12	Tool presentation	HW6
13	Discussion	
14		HW7
15		

Links

- Class
  - http://www-cad.eecs.berkeley.edu/~polis/class/
- On the left frame, go to Start EE249!
  - Subscribe to the mailing list
  - Fill up the questionnaire
- We will set up a message board for communication

Outline of the course

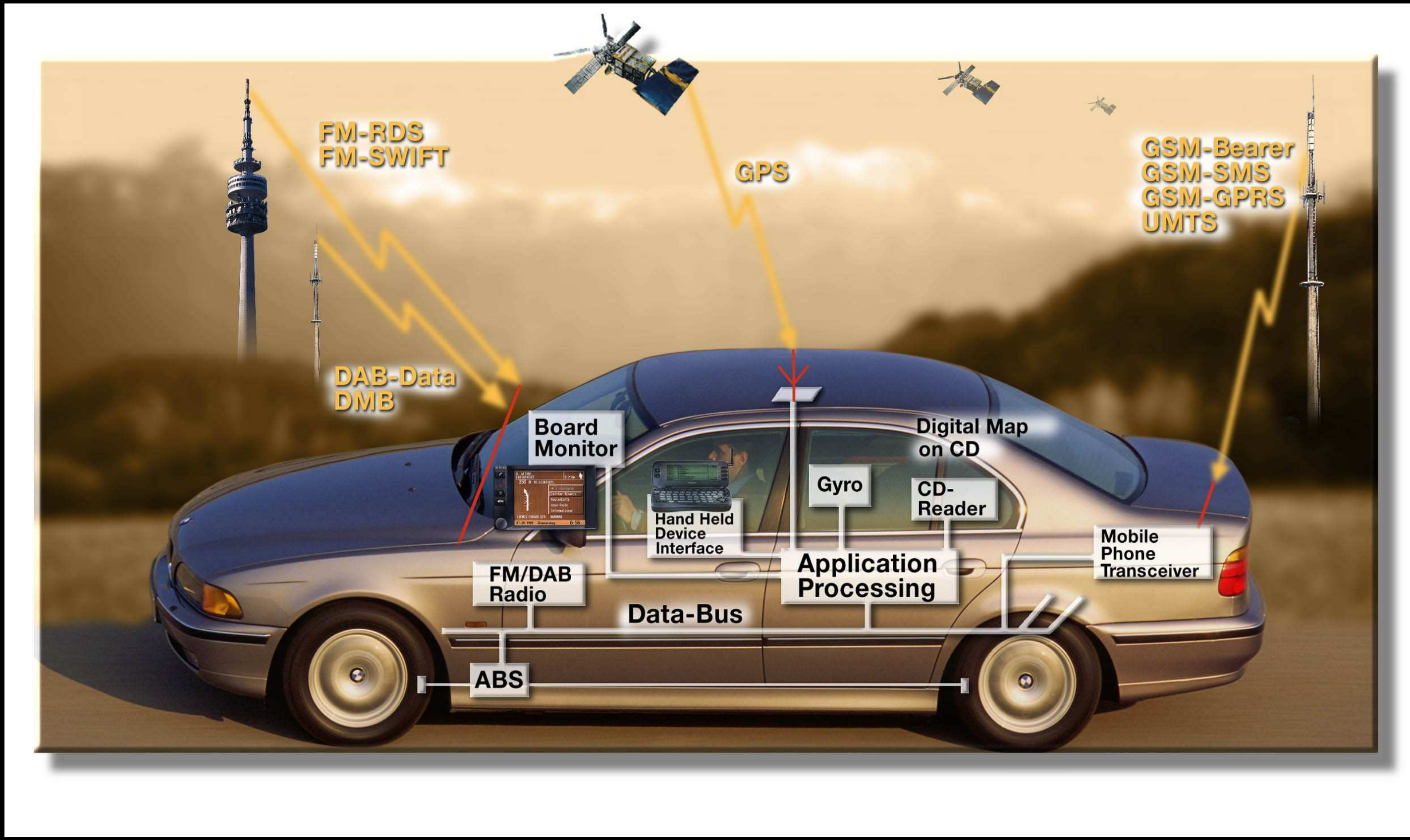
- Part 1. Introduction: Embedded Systems: what are they? What are the important questions? What is the state of the art?
- Part 2. Design Methodology (Platform-based Design)
- Part 3. Functional Design: Models of Computation
- Part 4. Architecture Design: Capture and Modeling
- Part 5. Exploration and Mapping
- Part 6. Implementation Verification and Synthesis, Hardware and Software

Outline for the Introduction

- Examples of Embedded Systems
- Their Impact on Society
- Design Challenges
- Embedded Software and Control

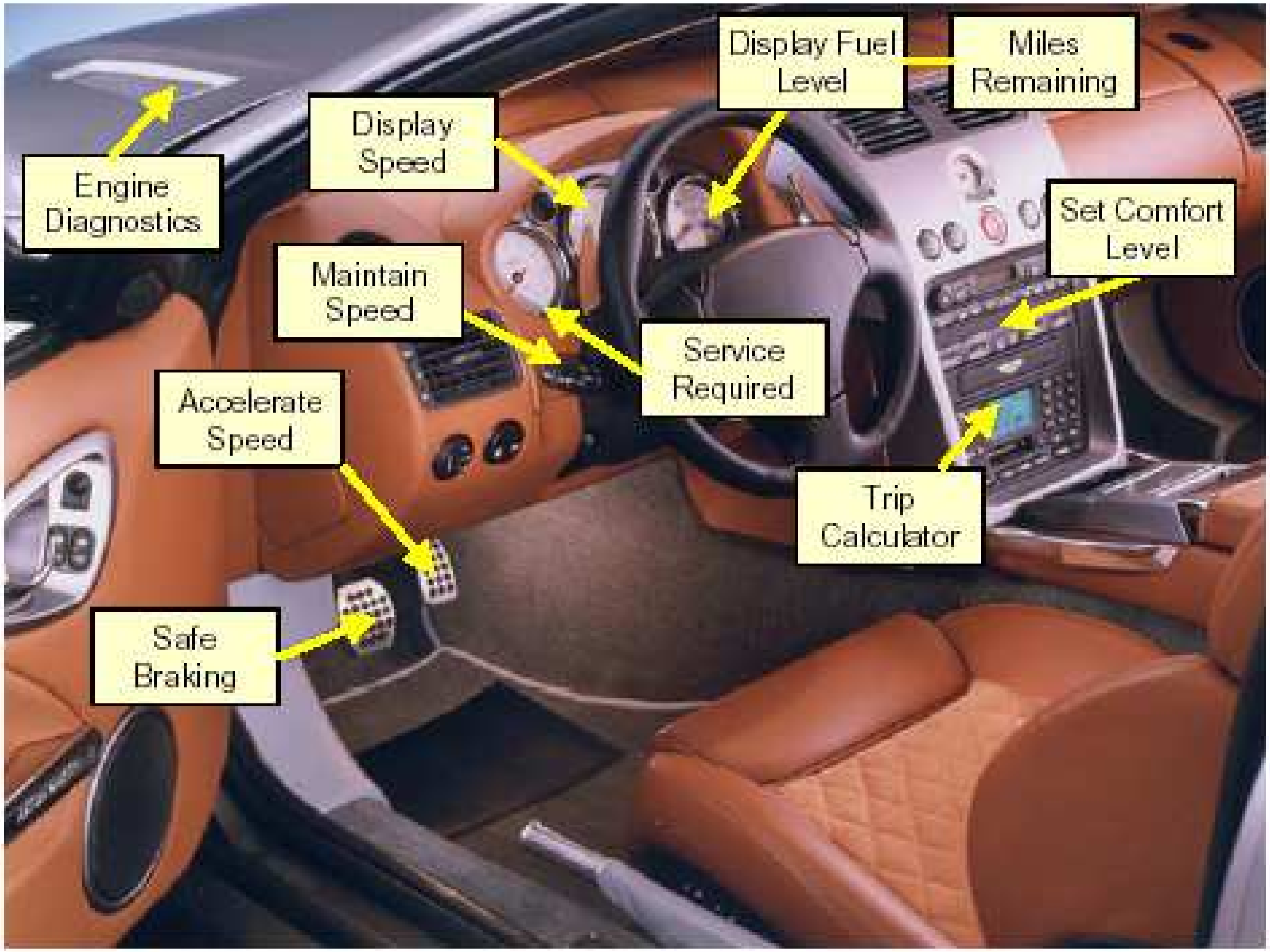
Electronics and the Car

- More than 30% of the cost of a car is now in Electronics
- 90% of all innovations will be based on electronic systems



FUNCTION OF CONTROLS

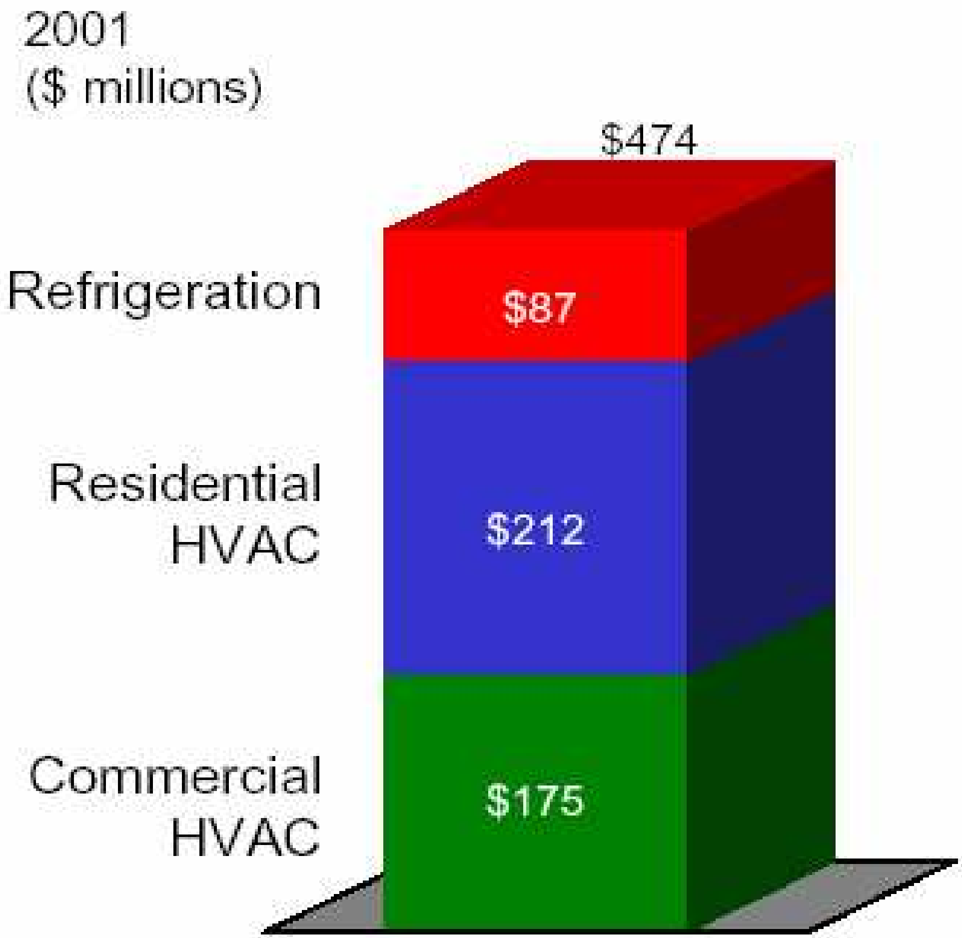
Typical minivan application



- Configure
- Sense
- Actuate
- Regulate
- Display
- Trend
- Diagnose
- Predict
- Archive

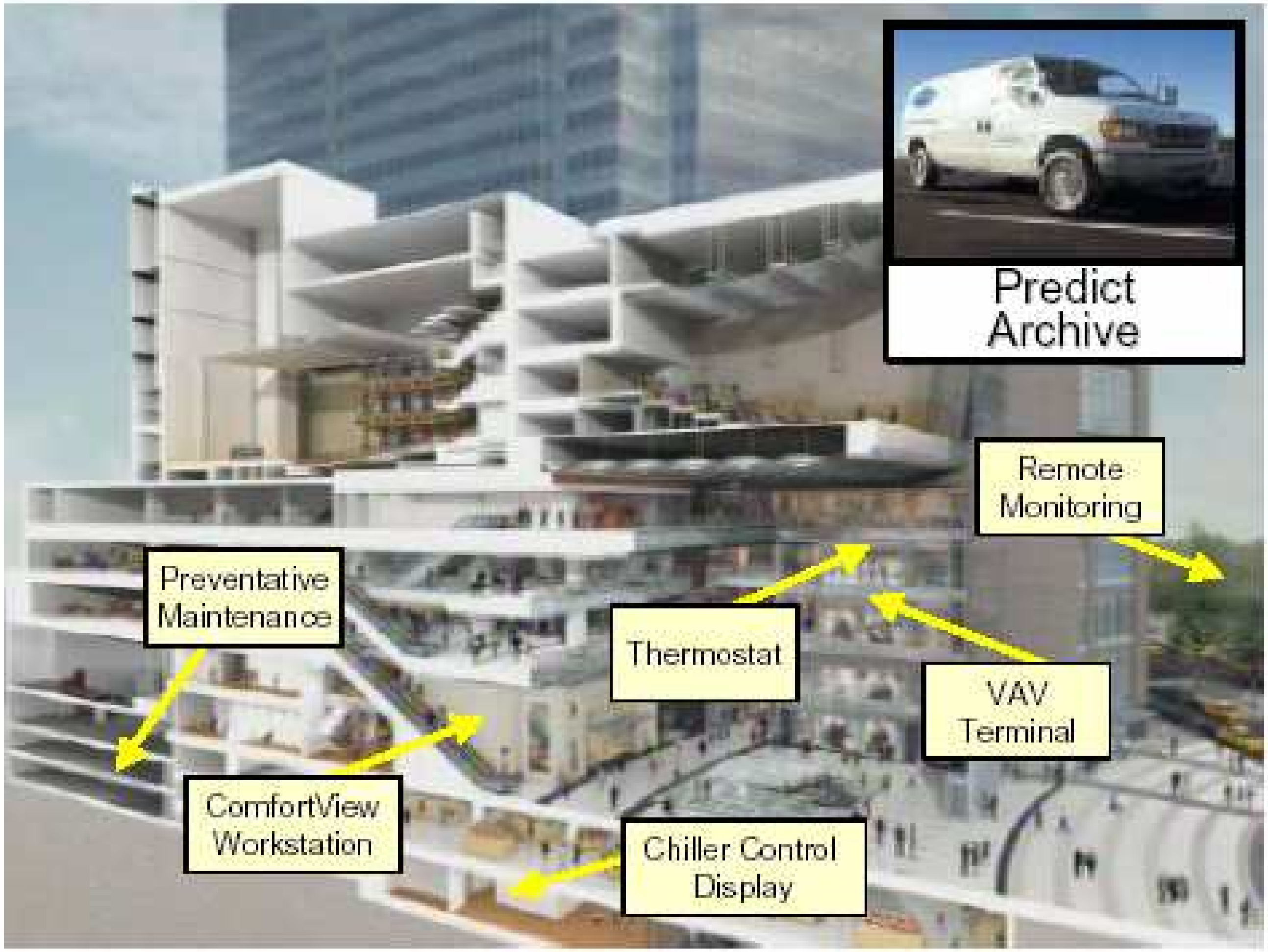
CARRIER CONTROLS BUSINESS

Market segments



FUNCTION OF CONTROLS

Typical commercial HVAC application



Configure

Sense

Actuate

Regulate

Display

Trend

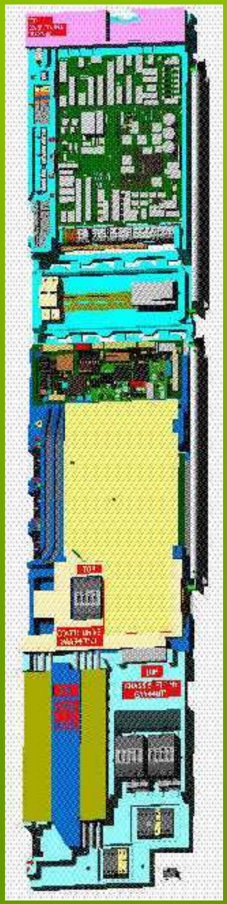
Diagnose

Predict

Archive

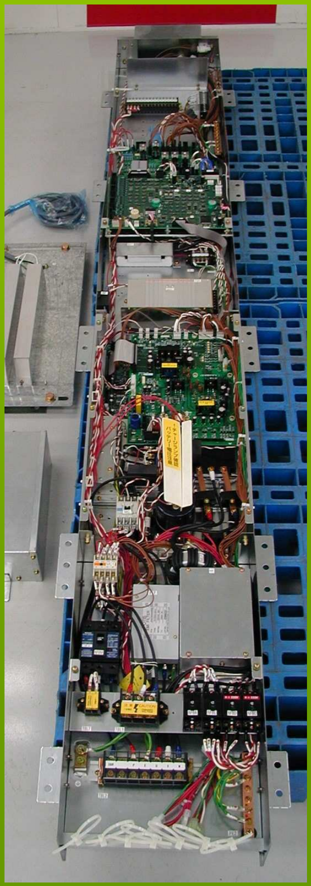
OTIS Elevators

1. EN: GeN2-Cx



2. ANSI: Gen2/GEM

3. JIS: GeN2-JIS



Segments

Attribute	Type 1	Type 2	Type 3
Stops/Rise	< 20 stops Opportunity: < 6 stops (20m)	< 64 stops	< 128 stops
Group Size	Simplex	1 – 8 cars	1 – 8 cars
Speed	< 4m/s <= .75 m/s (ANSI)	< 4 m/s	< 15 m/s
Op Features	Basic	Advanced	Hi-End Dispatch
Motion Features	Basic Perf. Basic FM	Limited Perf. Advanced FM	Advanced Perf. Advanced FM
Code	EN, ANSI, JIS	EN, ANSI, JIS	EN, ANSI, JIS
Remote Service	Yes	Yes	Yes
Price Sensitivity	High	High, Med	Med
Market	Utility	Utility, Design	Design

Embedded Systems

- Computational
  - but not first-and-foremost a computer
- Integral with physical processes
  - sensors, actuators
- Reactive
  - at the speed of the environment
- Heterogeneous
  - hardware/software, mixed architectures
- Networked
  - shared, adaptive

Source: Edward A. Lee

Common Situation in Industry

- Different hardware devices and architectures
- Increased complexity
- Non-standard tools and design processes
- Redundant development efforts
- Increased R&D and sustaining costs
- Lack of standardization results in greater quality risks
- Customer confusion

Outline for the Introduction

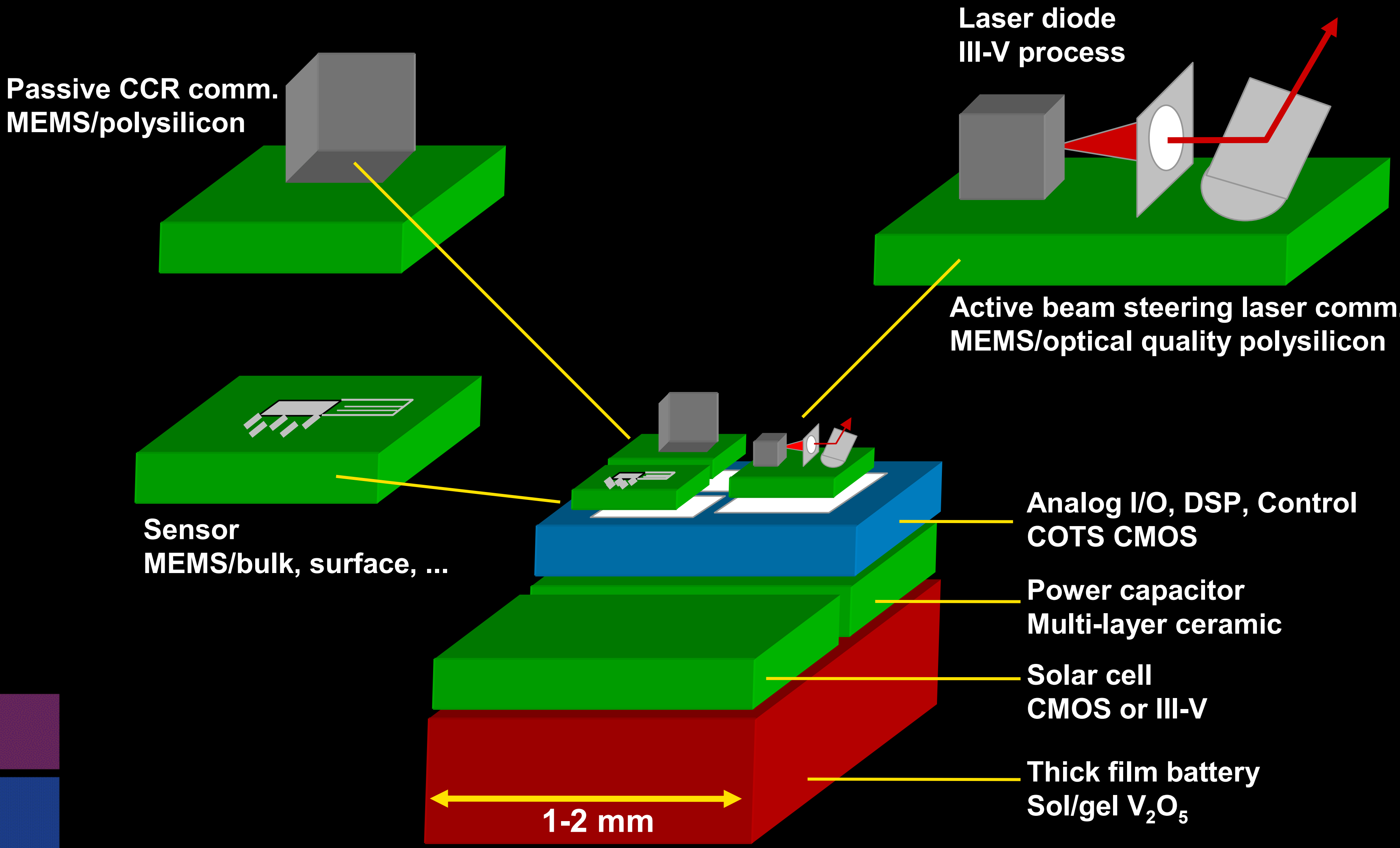
- Examples of Embedded Systems
- Their Impact on Society
- Design Challenges
- Embedded Software and Control

The Killer Applications?

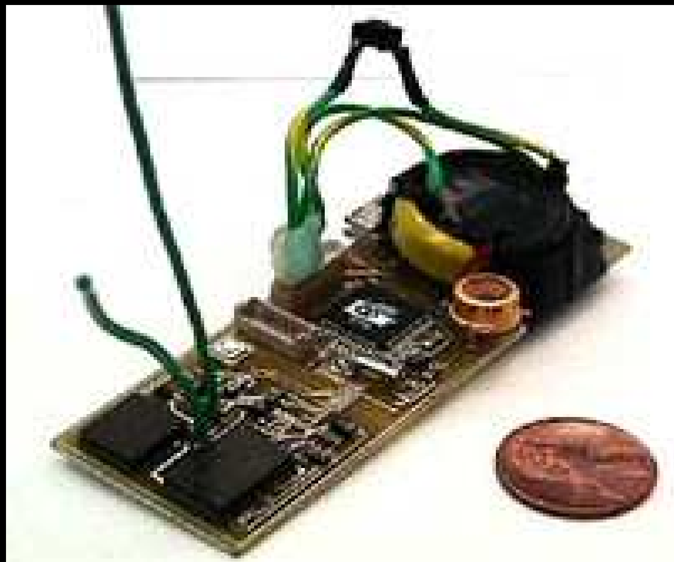



- Energy Conservation
- Emergency Response and Homeland Defense
- Transportation Efficiency
- Monitoring Health Care
- Land and Environment
- Education


Smart Dust

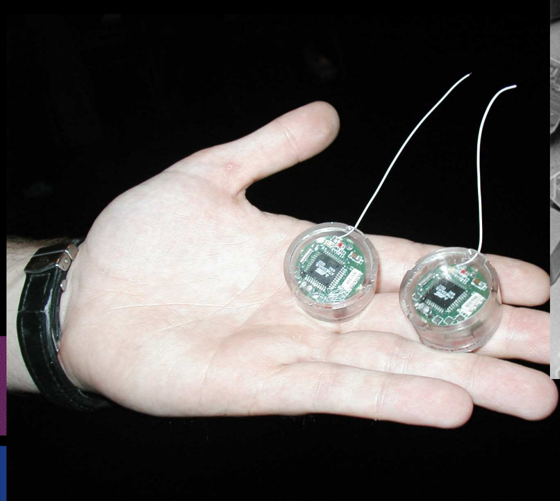


Source: K. Pister, Berkeley

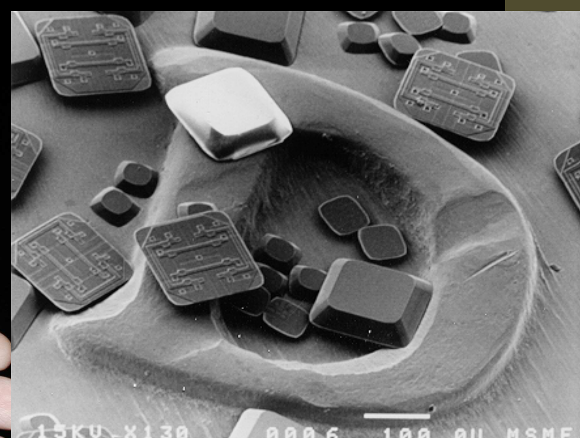


February 2000

February 2001

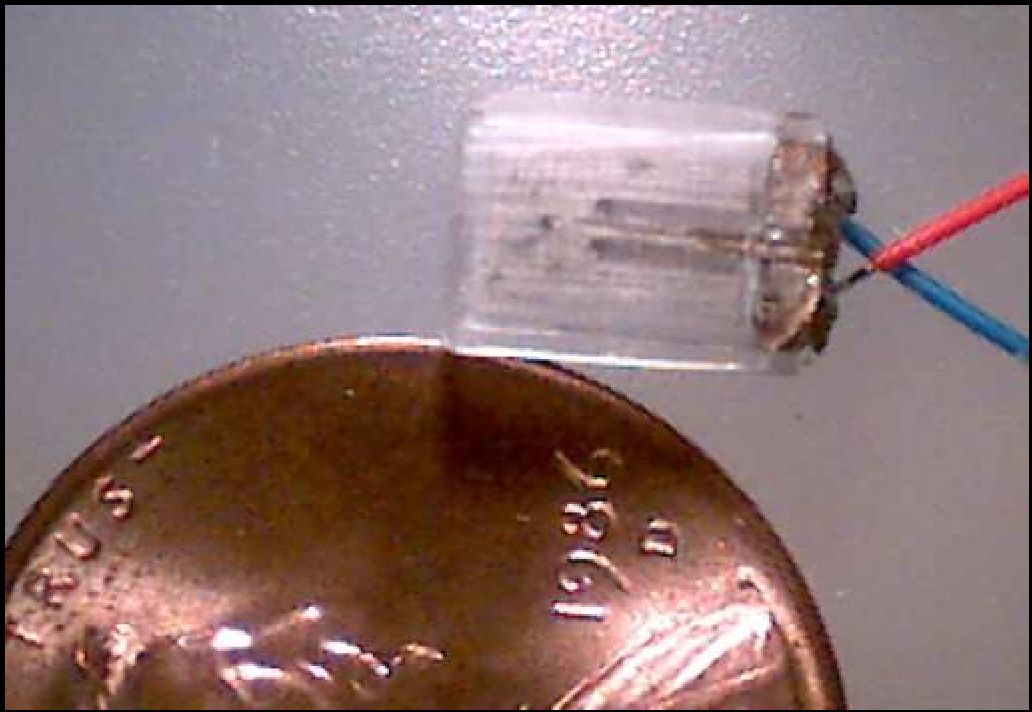


August 2001



February 2002

Energy Scavenging: Vibration

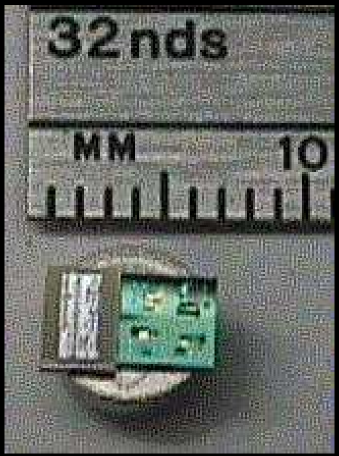


Source: P. Wright, Berkeley

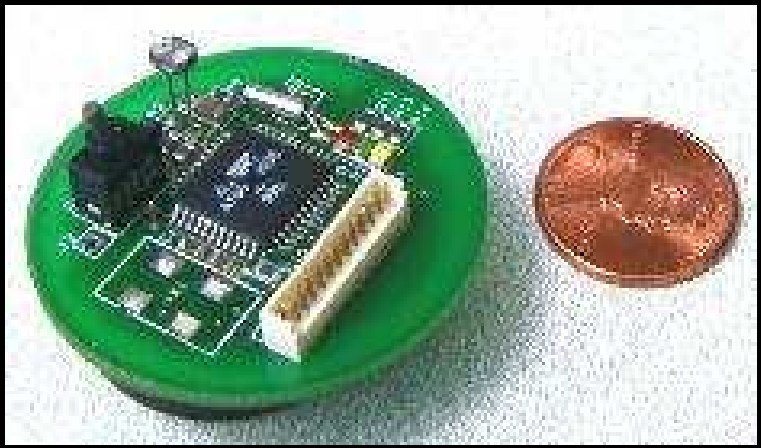
Wireless Sensor Networks

The use of wireless networks of embedded computers "could well dwarf previous milestones in the information revolution" - National Research Council Report: Embedded, Everywhere", 2001.

Berkeley Dust Mote[1]



Berkeley Mote[1]



[1]From Pister *et al., Berkeley Smart Dust Project*

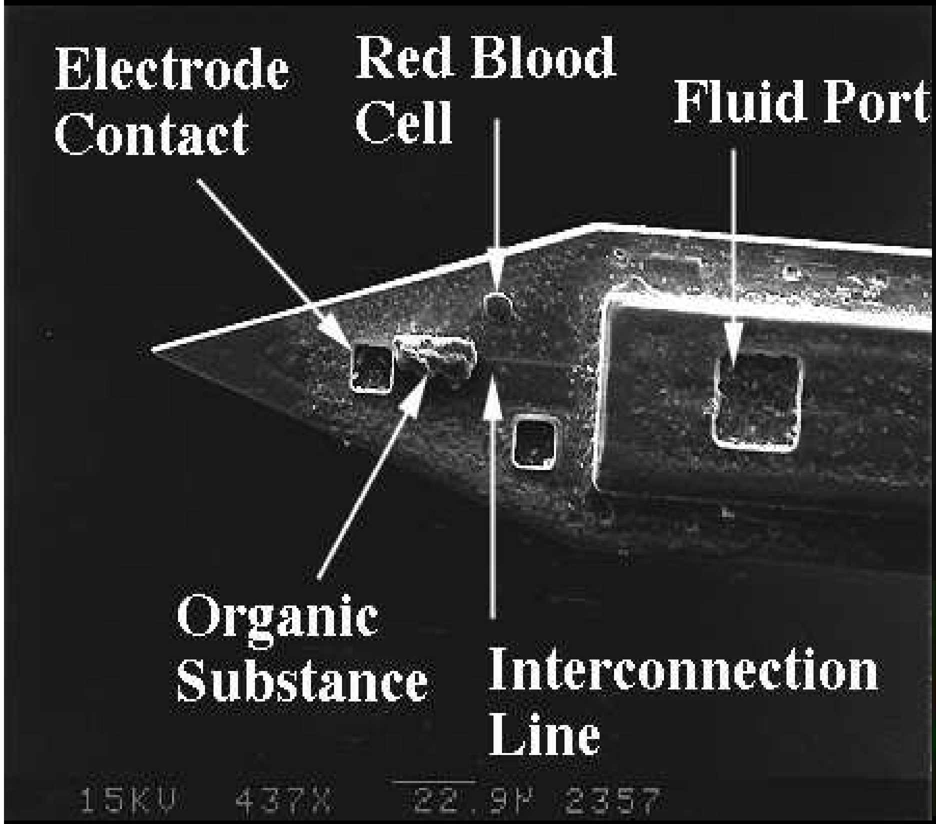
Applications

Distributed Bio-monitoring

- Wristband bio-monitors for chronic illness and the elderly
- Monitored remotely 24x7x365
- Emergency response and potential remote drug delivery
- Cardiac Arrest
  - Raise out-of-hospital survival rate from 6% to 20% => save 60K lives/year

Silicon-Processed Micro-needles

bio-medical



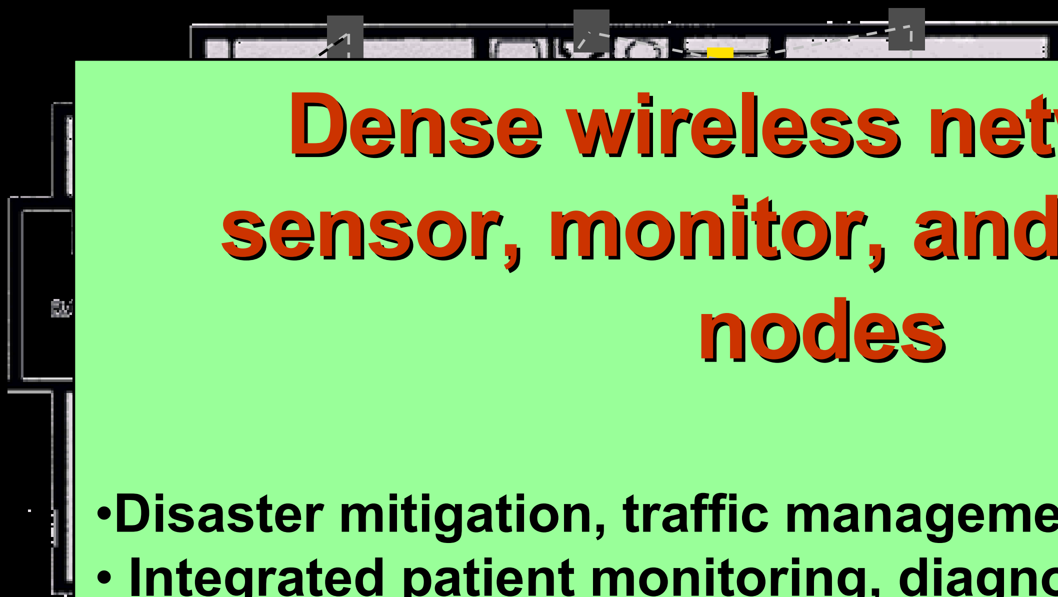
porterhouse (New-York)

Applications

Saving Energy

- Smart Buildings that adjust to inhabitants
- Make energy deregulation work via real-time metering and pricing
- Large potential savings in energy costs: for US commercial buildings
  - Turning down heat, lights saves up to $55B/year, 35M tons C emission/year
  - 30% of energy bill is from “broken systems”

Smart Buildings



Dense wireless network of sensor, monitor, and actuator nodes

- •Disaster mitigation, traffic management and control
- • Integrated patient monitoring, diagnostics, and drug administration
- • Automated manufacturing and intelligent assembly
- • Toys, Interactive Musea

•T… ng ir… building occupants , creating "micro-climates within a building"

• Other functions: security, identification and personalization, object tagging, seismic monitoring

Additional Applications

- Environmental Monitoring
  - Monitor air quality near highways to meet Federal Guidelines
  - Mutual impact of urban and agricultural areas
  - Monitor water shed response to climate events and land use changes

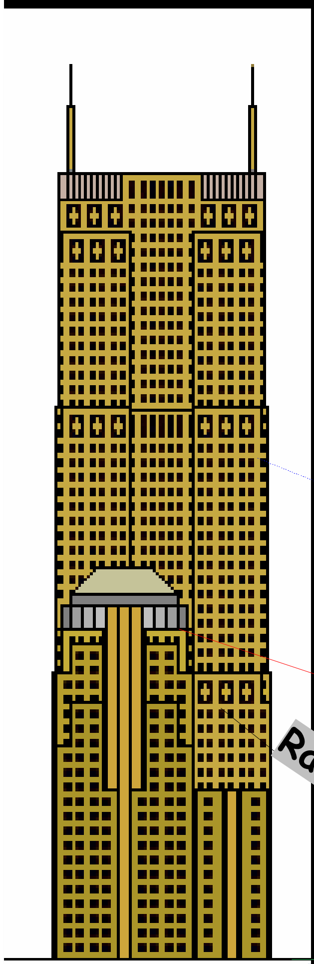
Applications

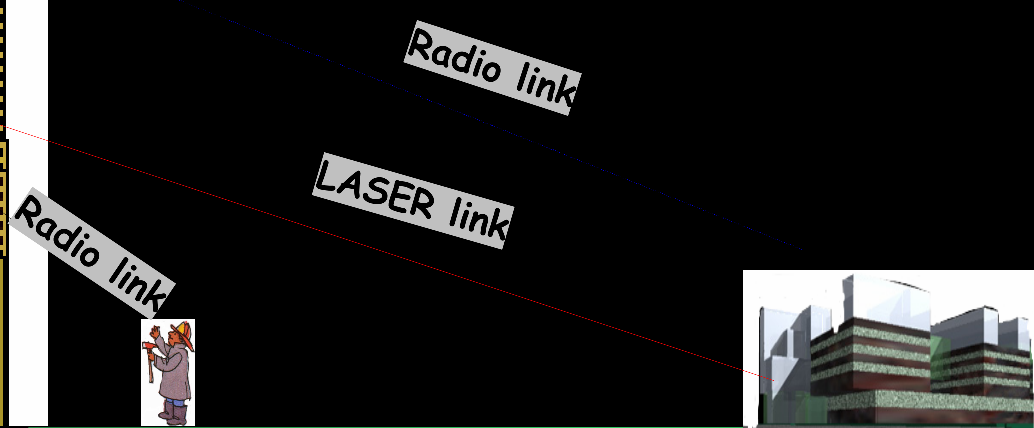
Disaster Mitigation (natural and otherwise)

- Monitor buildings, bridges, lifeline systems to assess damage after disaster
- Provide efficient, personalized responses
- Must function at maximum performance under very difficult circumstances

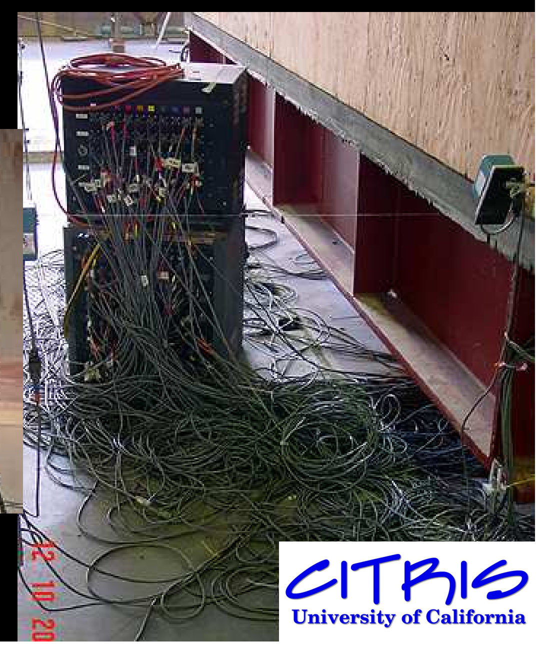
What is Disaster Response?

- **Sensors installed near critical structural points**
- **Sensor measure motion, distinguish normal deterioration and serious damage**
- **Sensors report location, kinematics of damage during and after an extreme event**
  - Guide emergency personnel
  - Assess structural safety without deconstructing building





Seismic Monitoring of Buildings: Before CITRIS



Seismic Monitoring of Buildings: With CITRIS Wireless Motes

CITRIS University of California

Stability of Masada North Face:
The Foundations of King Herod's Palace

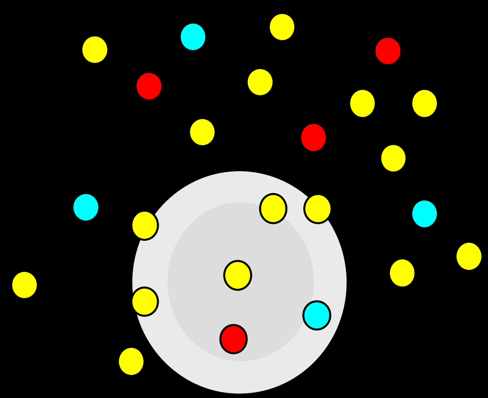
Additional Applications

- Transportation Systems
  - Use SISs to improve the efficiency and utility of highways while reducing pollution
  - Improve carpooling efficiency using advanced scheduling
  - Improve freeway utilization by managing traffic flows
  - Large potential savings in commuter time, lost wages, fuel, pollution: for CA
    - 15 minutes/commuter/day => $15B/year in wages
    - $600M/year in trucking costs, 150K gallons of fuel/day
- Distributed Education
  - Smart Classrooms
  - Lifelong Learning Center for professional education

Discussion

- What are the most challenging aspects of these applications (and how does a company make money) ?
  - Interaction mechanisms: sensors, actuators, wireless networks
  - Reliability and survivability
  - Infrastructure
  - Services
  - Legislation
  - ……

Picoradio Sensor Networks (BWRC)



- Control Environmental parameters (temperature, humidity…)
- Minimize Power consumption
- Cheap (<0.5$) and small (< 1 cm^3)
- Large numbers of nodes — between 0.05 and 1 nodes/m^2
- Limited operation range of network — maximum 50-100 m
- Low data rates per node — 1-10 bits/sec average
- Low mobility (at least 90% of the nodes stationary)

- Key challenges
  - Satisfy tight performance and cost constraints (especially power consumption)
  - Identify Layers of Abstraction (Protocol Stack)
  - Develop distributed algorithms (e.g. locationing, routing) for ubiquitous computing applications
  - Design Embedded System Platform to implement Protocol Stack efficiently


Government
Operations
Gas & Oil Storage
and Delivery
Emergency
Services
Water Supply
Systems
Critical
Infrastructures
Telecommunications
Banking &
Finance
Electrical
Energy
Transportation

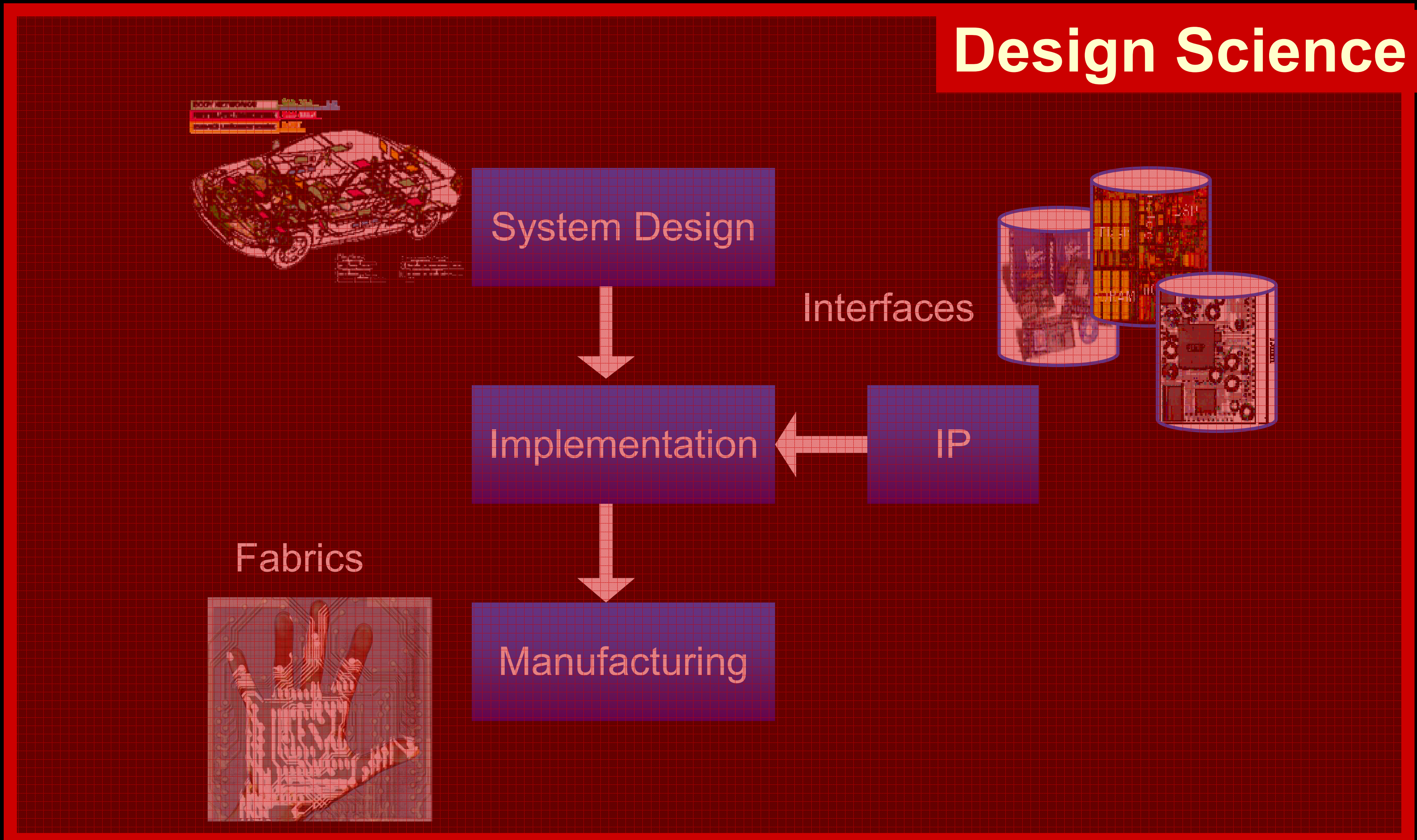
Secure Network Embedded SystEms (SENSE)

- Networked embedded systems and distributed control creates a new generation of future applications: new infrastructures
- We need to think about how to prevent the introduction of vulnerabilities via this exciting technology
- Security, Networking, Embedded Systems

Outline for the Introduction

- Examples of Embedded Systems
- Their Impact on Society
- Design Challenges
- Embedded Software and Control

Opportunity:
Electronic Systems Design Chain



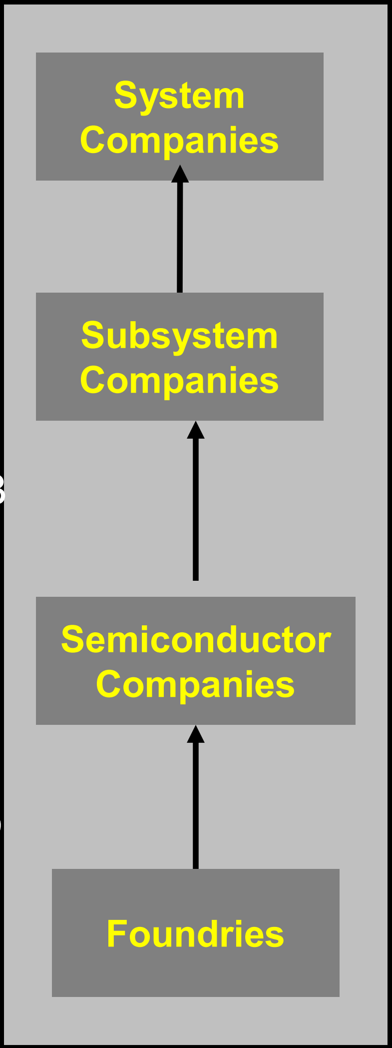
Disaggregation:
Complex Design Chain Management

Supply Chain

- Movement of tangible goods from sources to end market
- Supply Chain Management is $3.8B market projected to be $20B in 2005

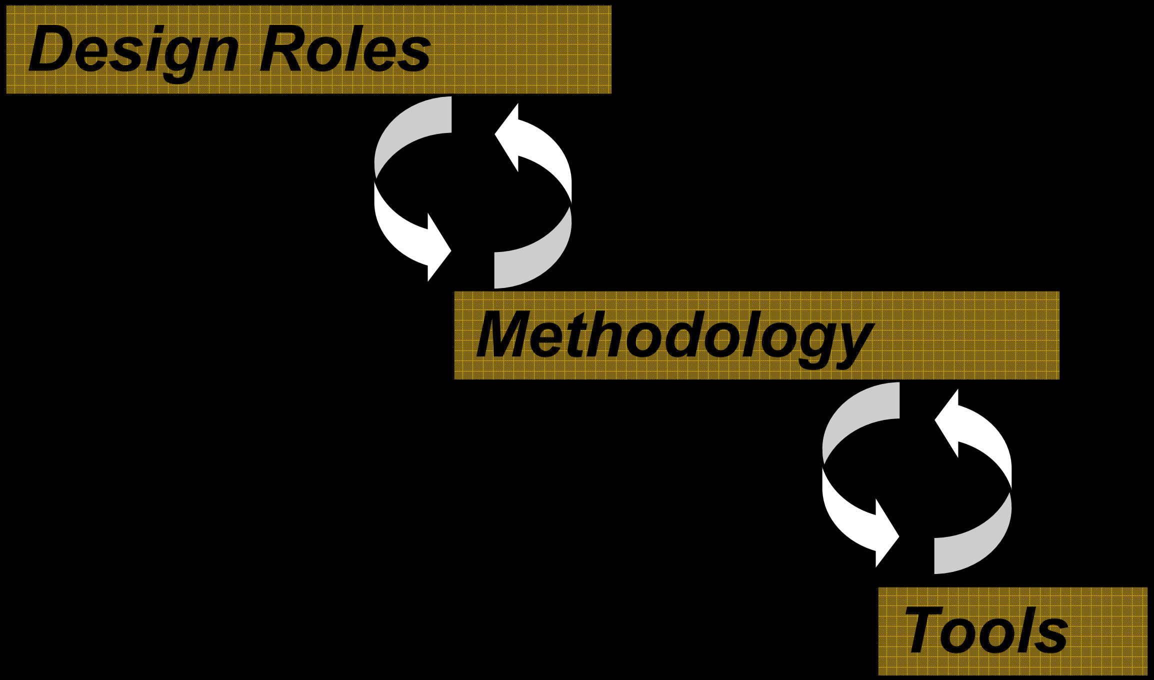
Design Chain

- Movement of technology (IP and knowledge) from sources to end market
- Design Chain Management is an untapped market

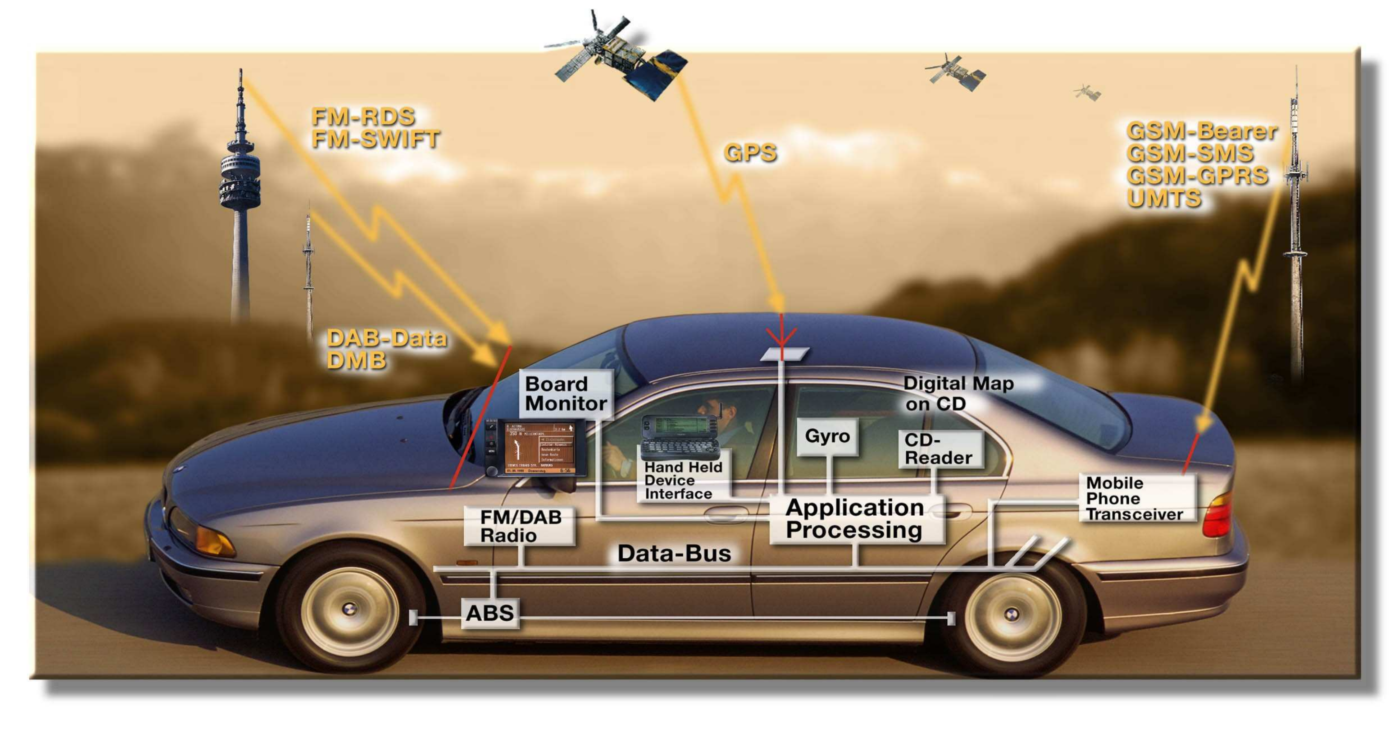


- EDA
- Design Services
- Embedded Software
- Software Development Tools
- Processor & Hardware IP
- IC Packaging & Test
- Process & Yield Services
- Mechanical CAD
- Contract Manufacturing
- System Test Equipment
- Fabrication Equipment

Supply Chain:
Design Roles-> Methodology->Tools

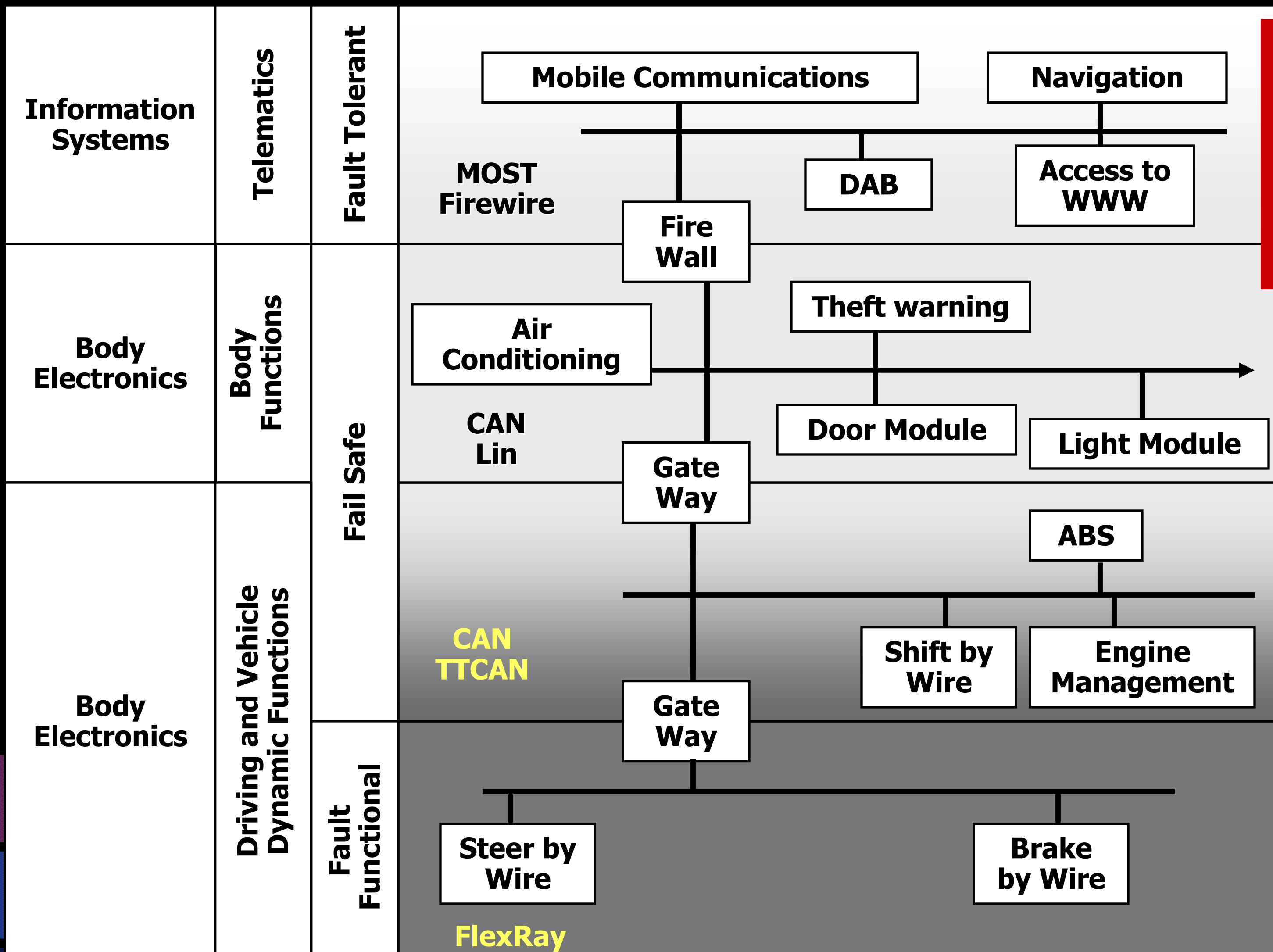


Automotive Supply Chain:
Car Manufacturers



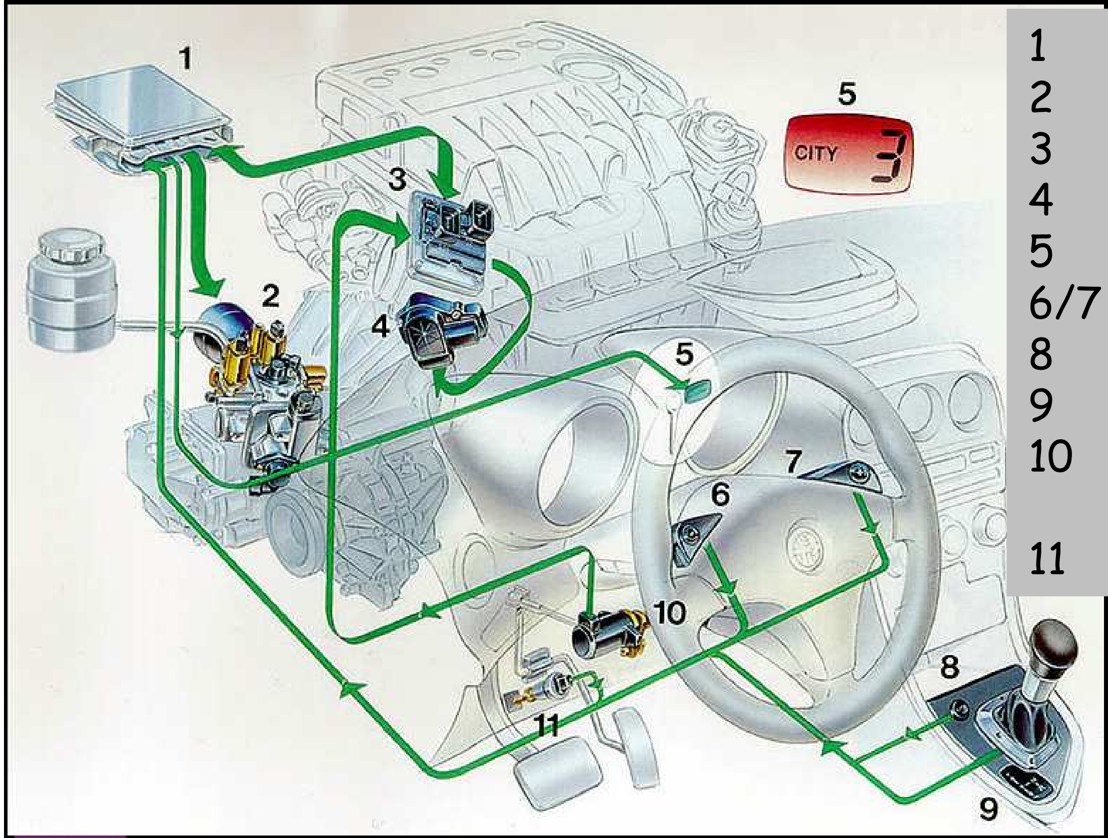
- Product Specification & Architecture Definition (e.g., determination of Protocols and Communication standards)
- System Partitioning and Subsystem Specification
- Critical Software Development
- System Integration

Electronics for the Car: A Distributed System



Today, more than 80 Microprocessors and millions of lines of code

Automotive Supply Chain: Tier 1 Subsystem Providers



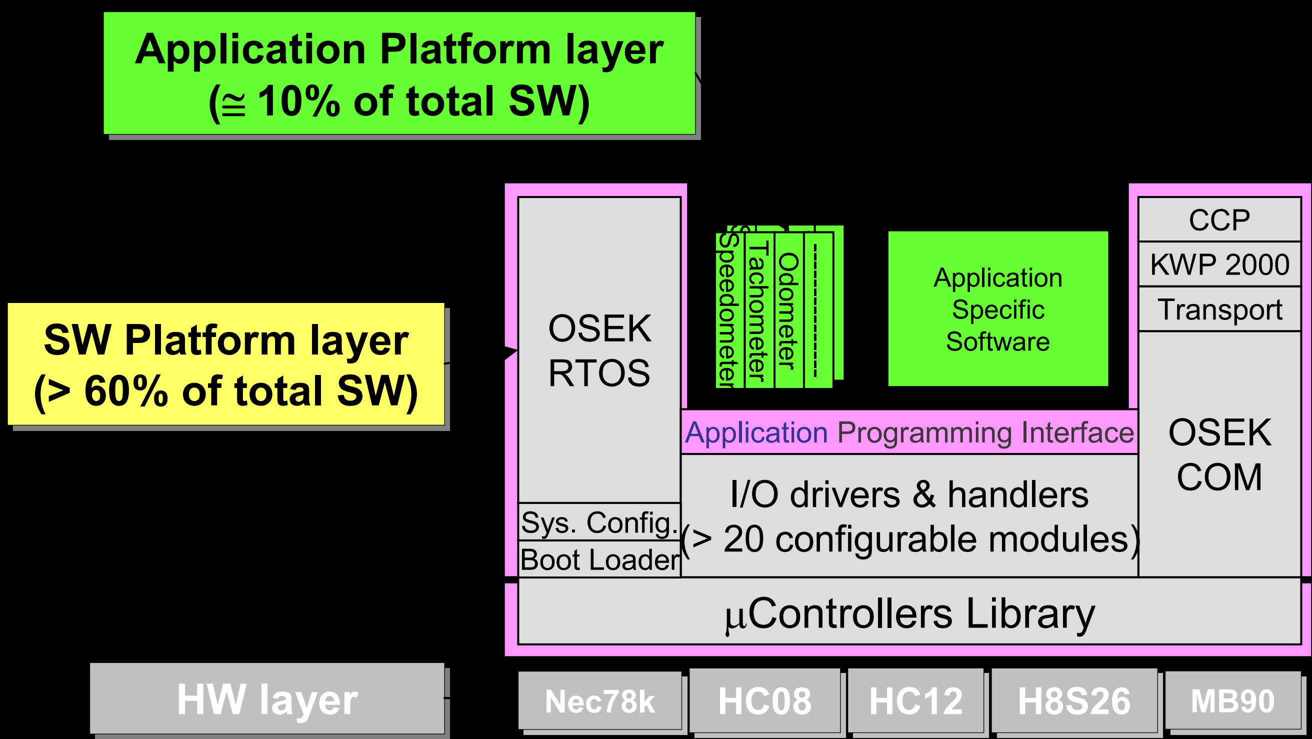
1 Transmission ECU
2 Actuation group
3 Engine ECU
4 DBW
5 Active shift display
6/7 Up/Down buttons
8 City mode button
9 Up/Down lever
10 Accelerator pedal position sensor
11 Brake switch

BOSCH

MAGNETI MARELLI

- Subsystem Partitioning
- Subsystem Integration
- Software Design: Control Algorithms, Data Processing
- Physical Implementation and Production

Automotive Supply Chain: Subsystem Providers

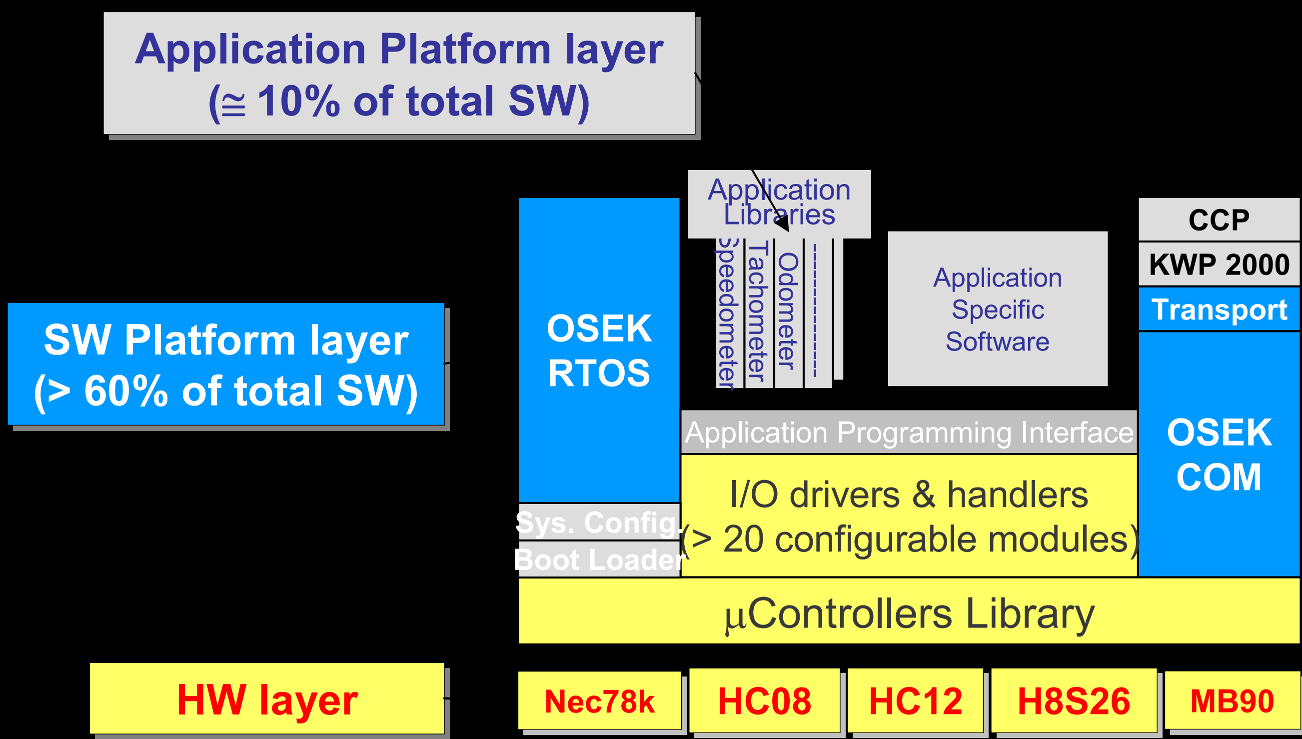


Platform Integration
Software Design

"firmware" and "glue software"

"Application"

Automotive Supply Chain: Platform & IP Providers



- "Software" platform
- "Hardware" platform

RTOS and communication layer

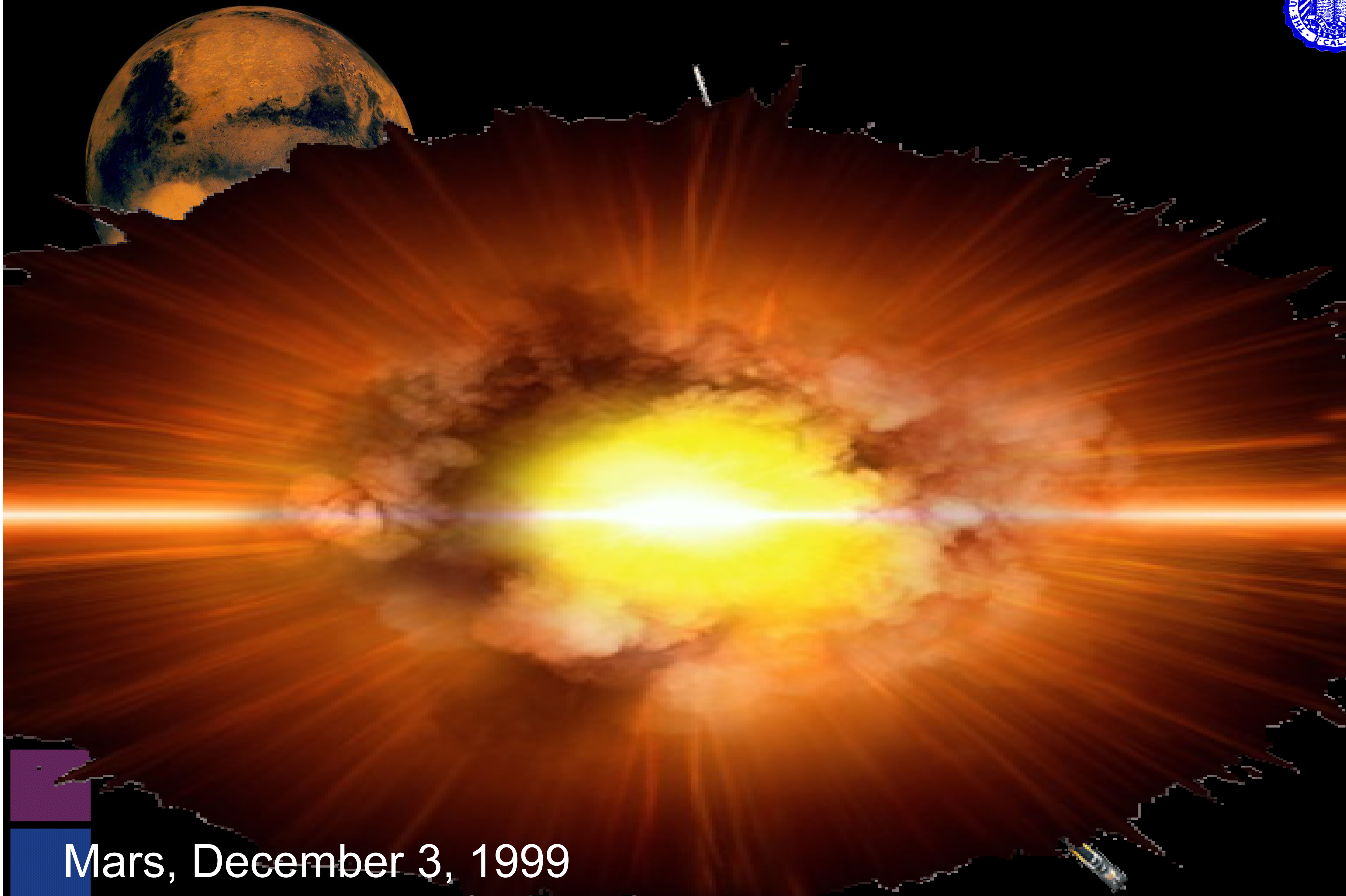
Hardware and IO drivers

Outline for the Introduction

- Examples of Embedded Systems
- Their Impact on Society
- Design Challenges
- Embedded Software and Control

How Safe is Our Real-Time Software?

Computing for Embedded Systems



Mars, December 3, 1999
Crashed due to un-initialized variable


777
BOEING 777-300
$4 billion development effort
40-50% system integration & validation cost

Complexity, Quality, & Time To Market today

	PWT UNIT 	BODY GATEWAY 	INSTRUMENT CLUSTER 	TELEMATIC UNIT 
Memory	256 Kb	128 Kb	184 Kb	8 Mb
Lines Of Code	50.000	30.000	45.000	300.000
Productivity	6 Lines/Day	10 Lines/Day	6 Lines/Day	10 Lines/Day*
Residual Defect Rate @ End Of Dev	3000 Ppm	2500 ppm	2000ppm	1000 ppm
Changing Rate	3 Years	2 Years	1 Year	< 1 Year
Dev. Effort	40 Man-yr	12 Man-yr	30 Man-yr	200 Man-yr
Validation Time	5 Months	1 Month	2 Months	2 Months
Time To Market	24 Months	18 Months	12 Months	< 12 Months

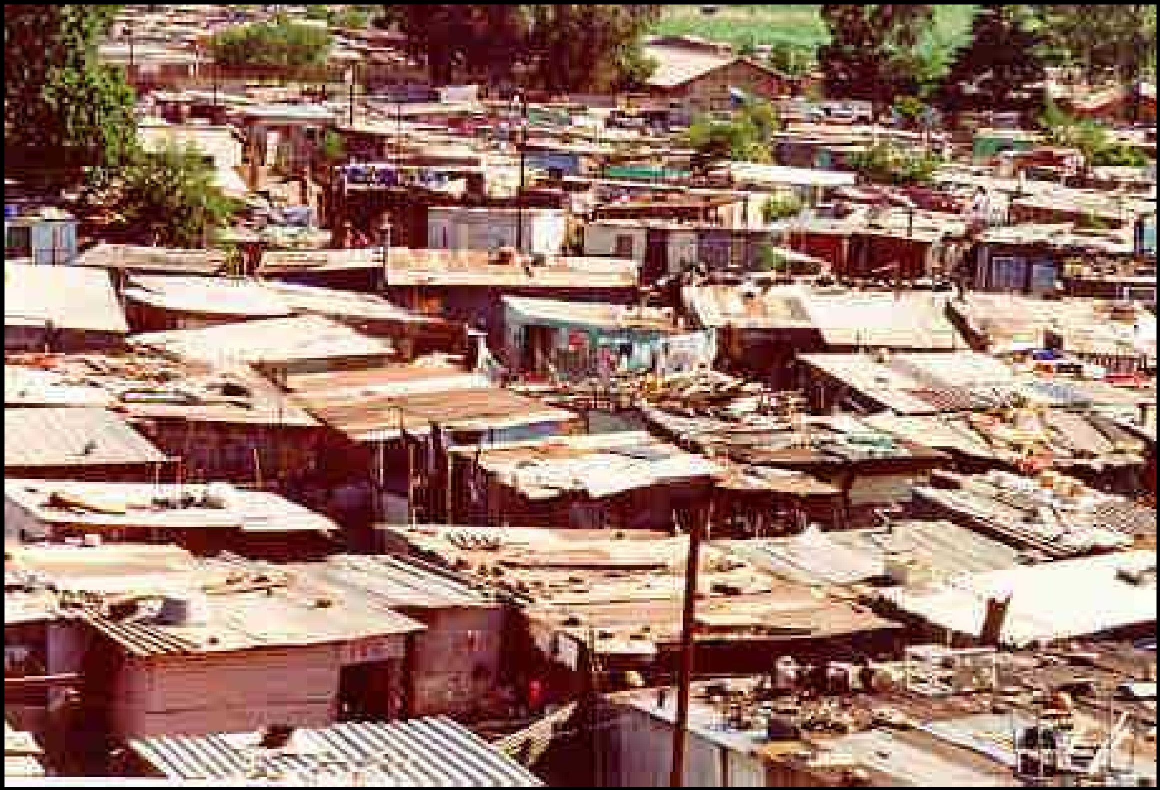
*** C++ CODE**

FABIO ROMEO, Magneti-Marelli
DAC, Las Vegas, June 20th, 2001

What About Real Time?

"Make it faster!"

Software Architecture Today



Design “Practice”

Design Science:
Build upon Solid Foundations

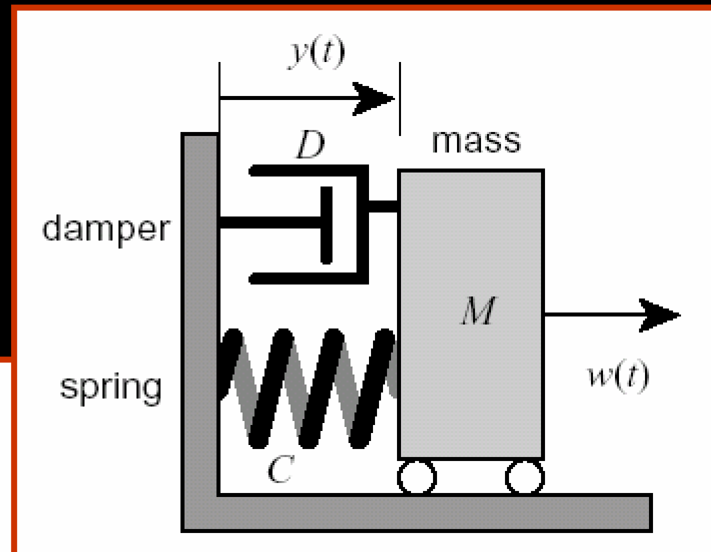
Software Architecture Tomorrow?

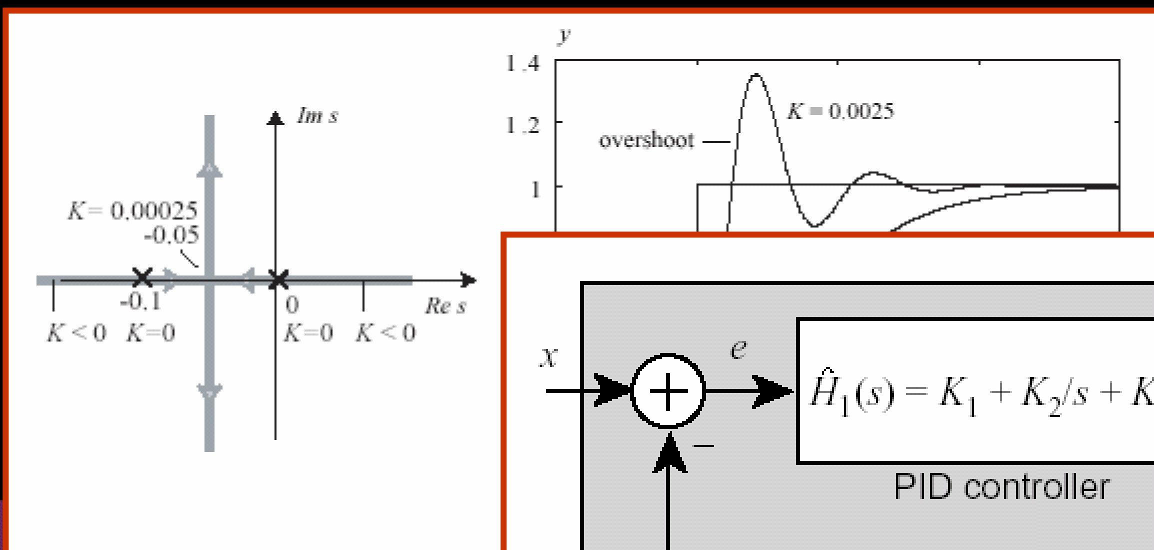
The Goal (CHESS Project)

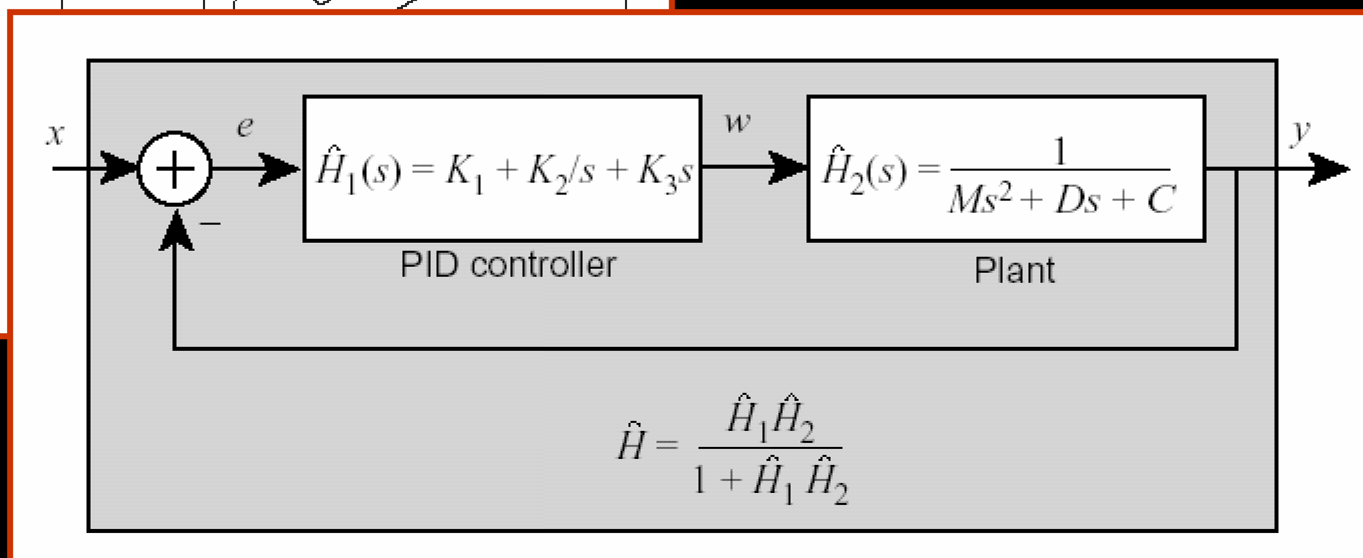
- To create a modern computational systems science and systems design practice with
  - Concurrency
  - Composability
  - Time
  - Hierarchy
  - Heterogeneity
  - Resource constraints
  - Verifiability
  - Understandability

A Traditional Systems Science – Feedback Control Systems

- **Models of continuous-time dynamics**
- **Stability analysis**
- **But not accurate for software controllers**

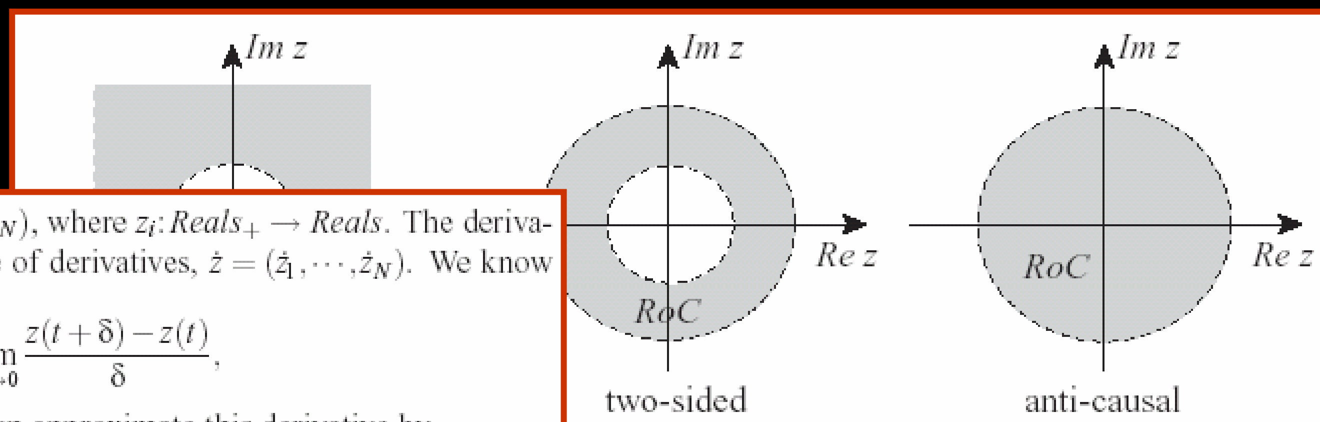






Discretized Model – A Step Towards Software

- **Numerical integration techniques provided ways to get from the continuous idealizations to computable algorithms.**
- **Discrete-time signal processing techniques offer the same sophisticated stability analysis as continuous-time methods.**
- **But it's *still* not accurate for software controllers**



In general, z is an N-tuple, $z = (z_1, \cdots, z_N)$, where $z_i\colon Reals_+ \to Reals$. The derivative of an N-tuple is simply the N-tuple of derivatives, $\dot{z} = (\dot{z}_1, \cdots, \dot{z}_N)$. We know from calculus that

$$\dot{z}(t) = \frac{dz}{dt} = \lim_{\delta \to 0} \frac{z(t+\delta) - z(t)}{\delta},$$

and so, if $\delta > 0$ is a small number, we can approximate this derivative by

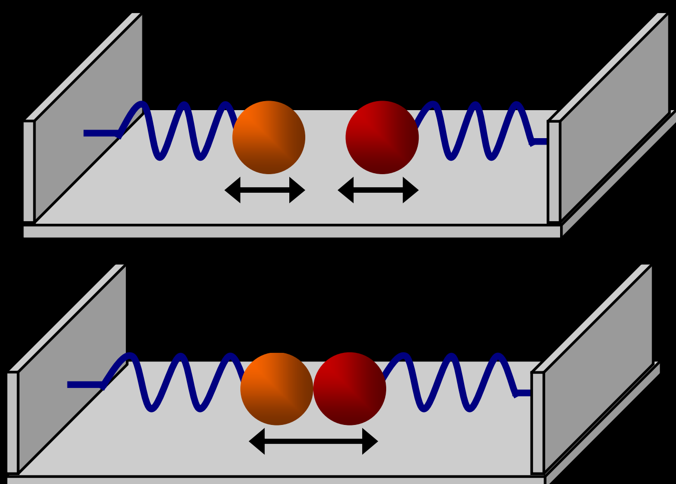
$$\dot{z}(t) \approx \frac{z(t+\delta) - z(t)}{\delta}.$$

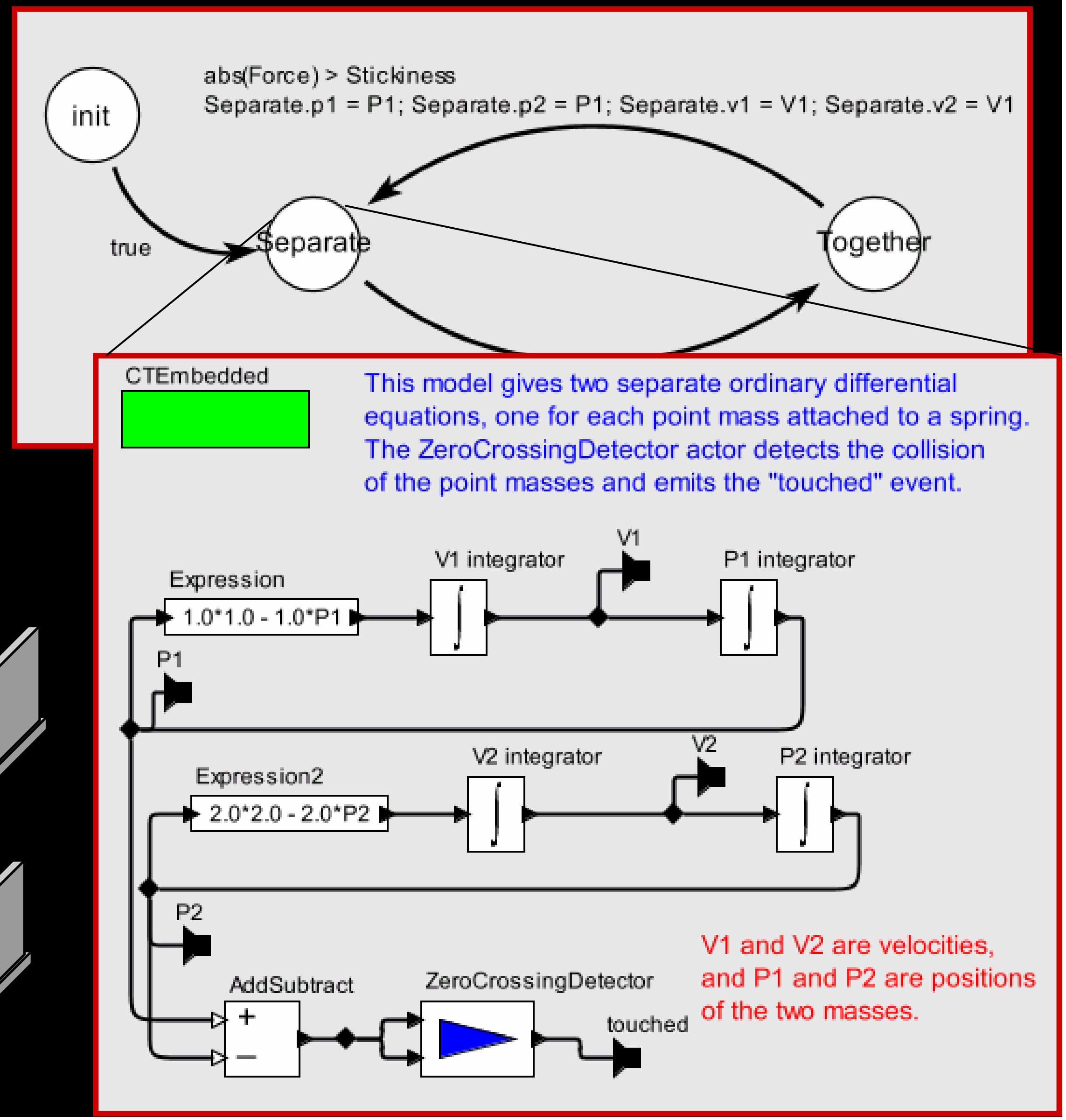
Using this for the derivative in the left-hand side of (5.50) we get

$$z(t+\delta) - z(t) = \delta g(z(t), v(t)). \tag{5.51}$$

Hybrid Systems –
Reconciliation of Continuous & Discrete

- But it's ***still*** not accurate for software controllers

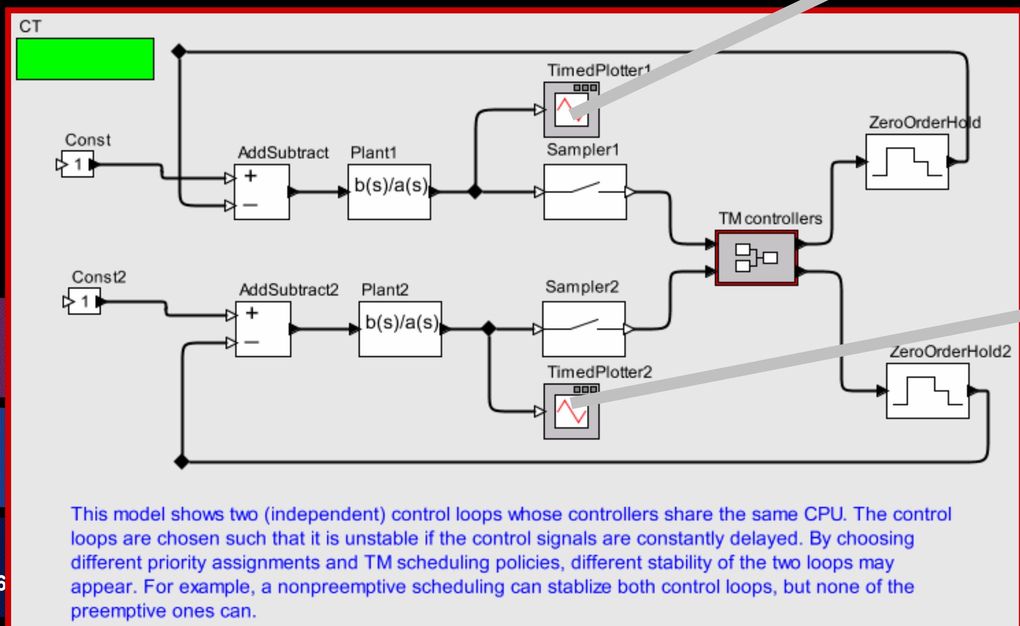


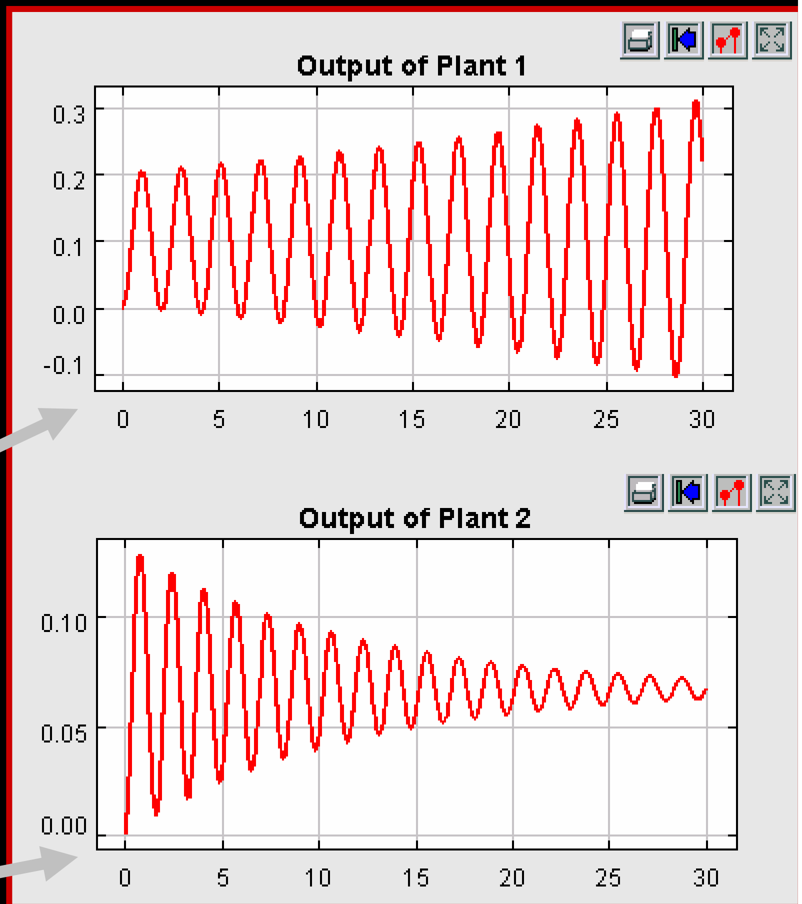


Timing in Software is More Complex Than What the Theory Deals With

An example (Jie Liu) models two controllers sharing a CPU under an RTOS. Under preemptive multitasking, only one can be made stable (depending on the relative priorities). Under non-preemptive multitasking, both can be made stable.

Where is the theory for this?





Foundational Theory Research …

- The science of computation has systematically abstracted away the physical world. The science of physical systems has systematically ignored computational limitations.
  Embedded software systems, however, engage the physical world in a computational manner.

- It is time to construct a Hybrid Systems Science that is simultaneously computational and physical.
  Time, concurrency, robustness, continuums, and resource management must be remarried to computation.