Taller de Sistemas Cíber Físicos

Introducción

presentación basada en:

- Rodolfo Pellizzoni - ECE 720T5 – Waterloo
- Kang G. Shin EECS 571 - University of Michigan
- Edward A. Lee. "Resurrecting Laplace's Demon: The Case for Deterministic Models". Talk or presentation, 4, October, 2016; Keynote Talk: MODELS, St. Malo, France
- Introduction to Embedded Systems. Edward A. Lee. UC Berkeley. EECS 149/249A. Fall 2016

Computing evolution

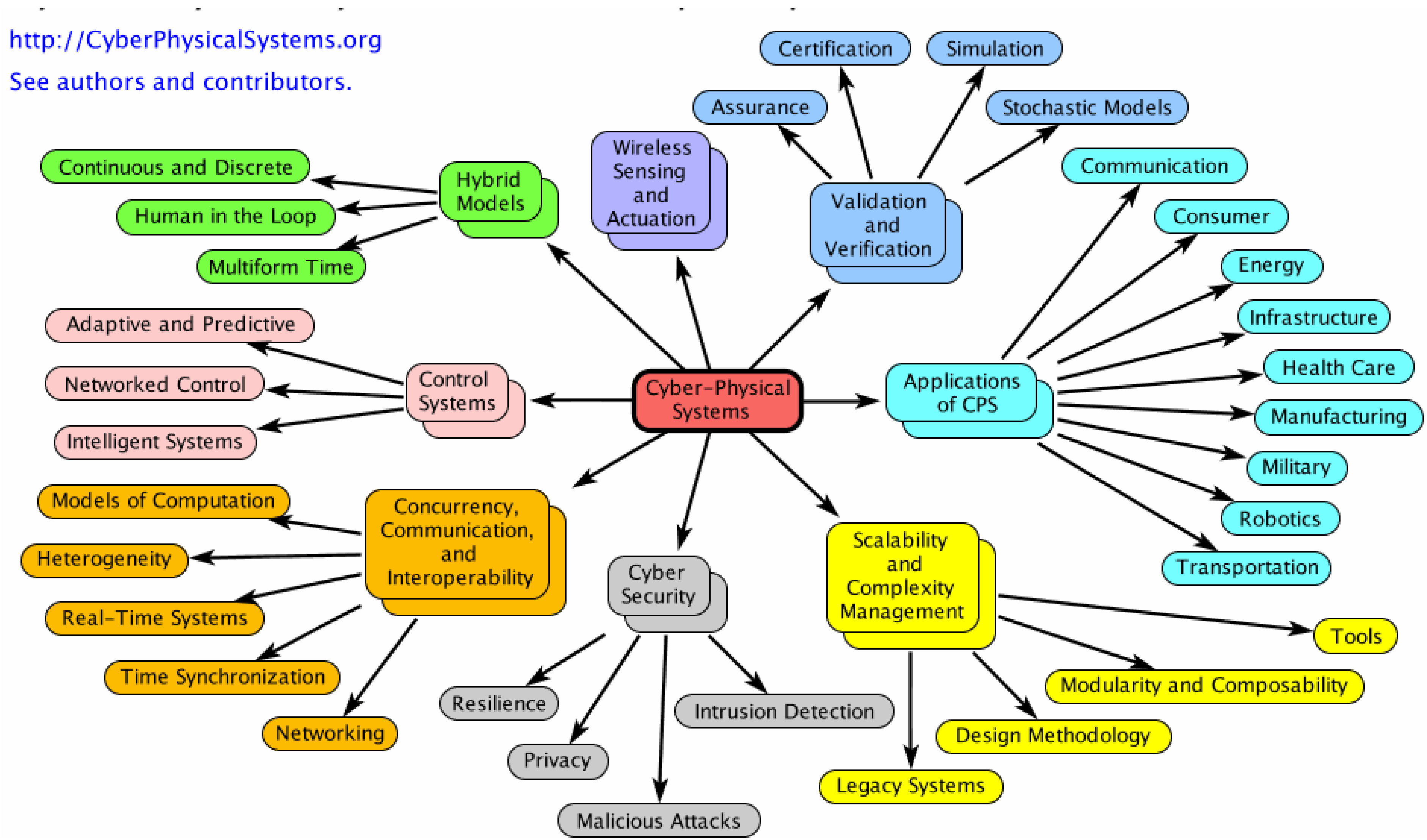
- **Mainframe computing (60's-70's)**
  - Large computers to execute big data processing applications

- **Desktop computing & Internet (80's-90's)**
  - One computer at every desk to do business/ personal activities

- **Ubiquitous computing (00's)**
  - Numerous computing devices in every place/ person
  - "Invisible" part of the environment
  - Millions for desktops vs. billions for embedded processors

- **Cyber Physical Systems (10's)**

Mapa Conceptual



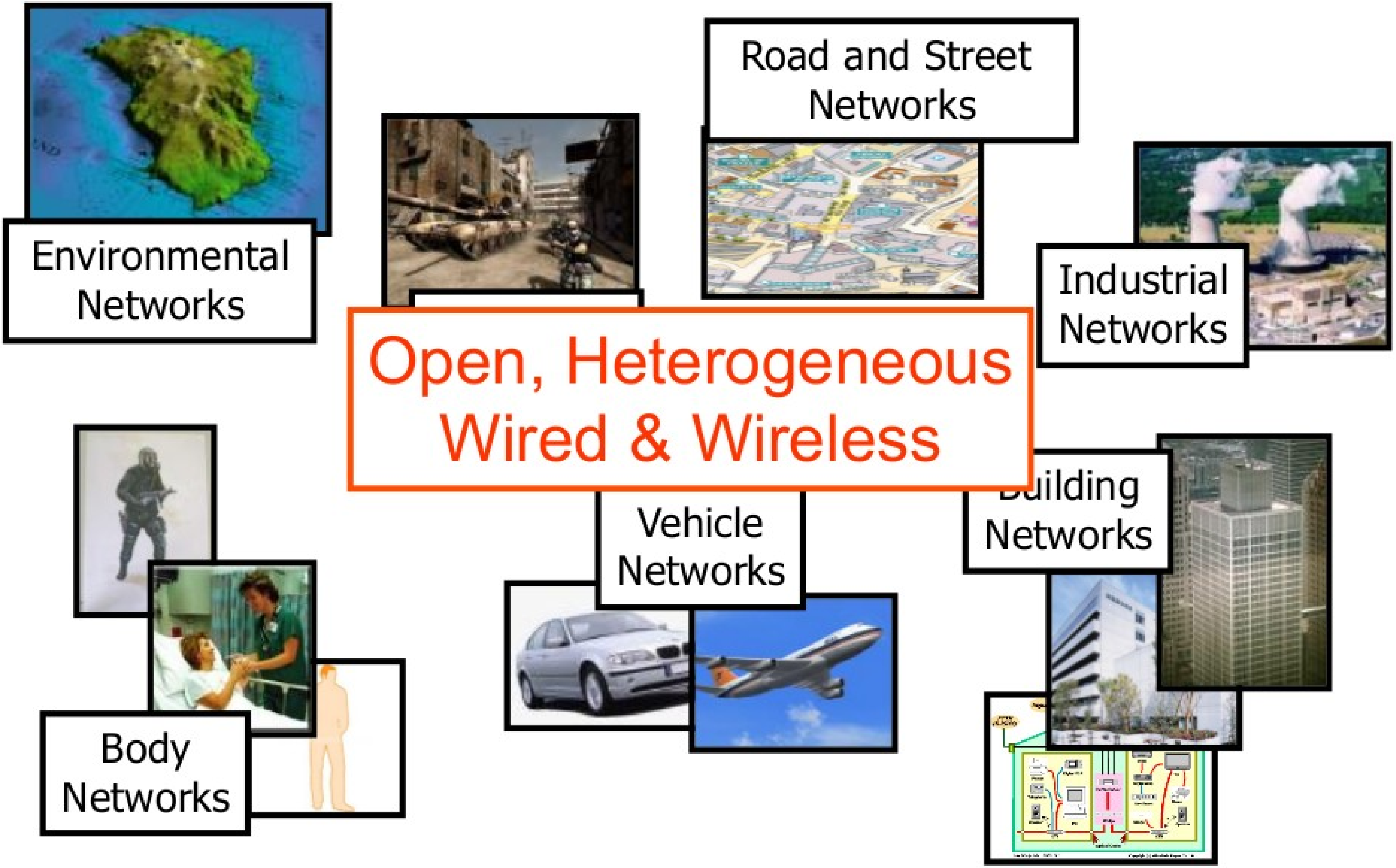
CPS: a definition?

- Cyber – computation, communication, and control that are discrete, logical, and switched
- Physical – natural and human-made systems governed by the laws of physics and operating in continuous time
- Cyber-Physical Systems – systems in which the cyber and physical systems are tightly integrated at all scales and levels
  - Change from cyber merely applied on physical
  - Change from physical with COTS “computing as parts” mindset
  - Change from ad hoc to grounded, assured development

CPS: a definition?

- Integration of physical systems and processes with networked computing
- Computations and communications are deeply embedded in, and interacting with physical processes to equip physical systems with new capabilities
- Covers a wide range of scale (pacemakers to national power grid)

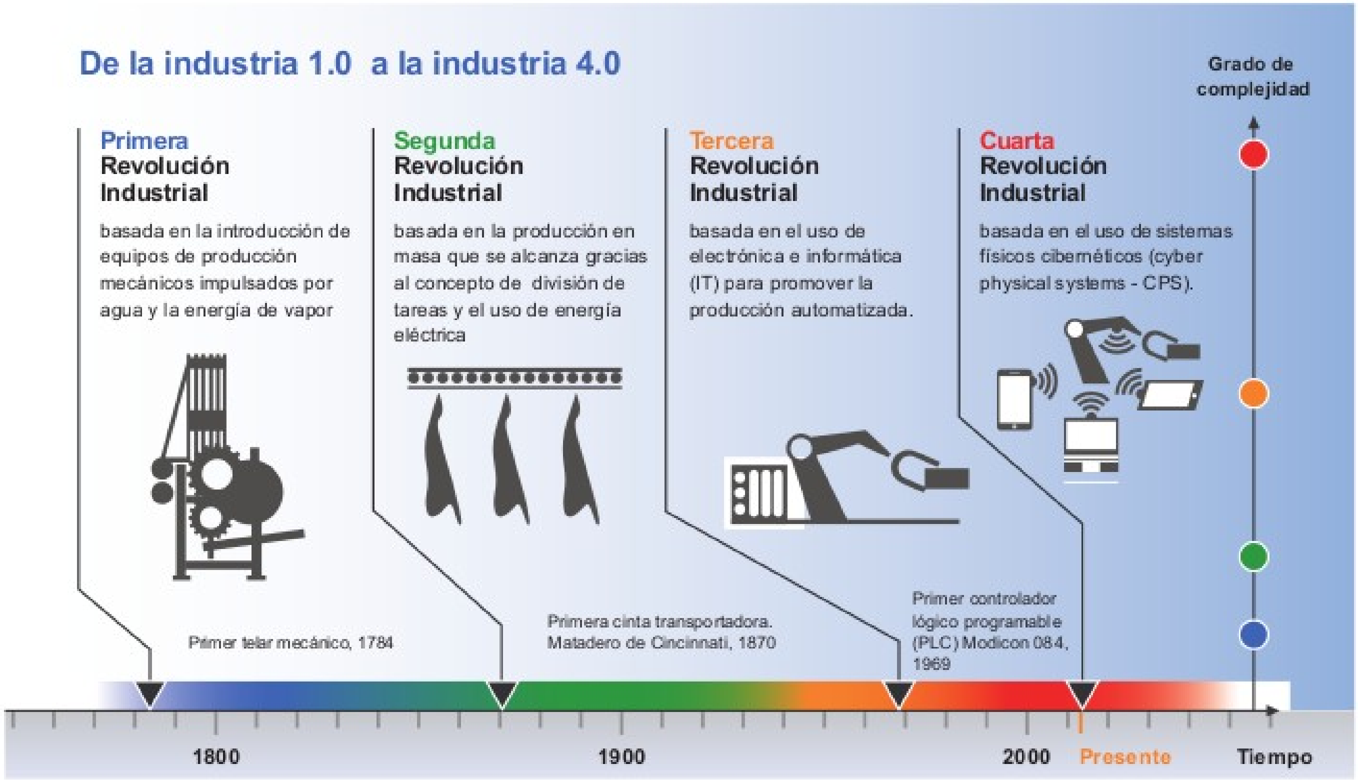
Computing in CPS



Application Domains of Cyber-Physical Systems

- Healthcare
  - Medical devices
  - Health management networks
- Transportation
  - Automotive electronics
  - Vehicular networks and smart highways
  - Aviation and airspace management
  - Avionics
  - Railroad systems
- Process control
- Large-scale Infrastructure
  - Physical infrastructure monitoring and control
  - Electricity generation and distribution
  - Building and environmental controls
- Defense systems
- Tele-physical operations
  - Telemedicine
  - Tele-manipulation

Industria 4.0



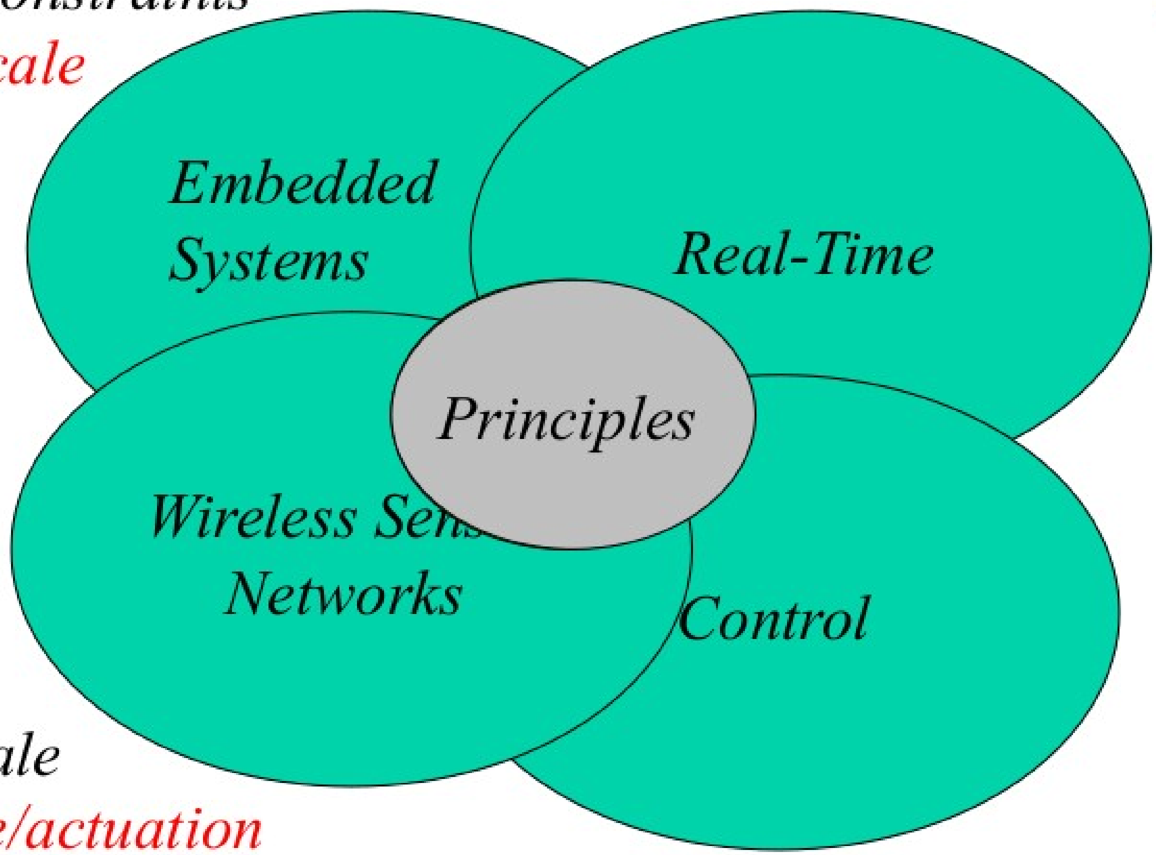
CPS characteristics

- Cyber capability in every physical component
- Networked at multiple and extreme scales
- Complex at multiple temporal and spatial scales
- Constituent elements are coupled logically and physically
- Dynamically reorganizing/reconfiguring; “open systems”
- High degrees of automation, control loops closed at many scales
- Unconventional computational & physical substrates (such as bio, nano, chem, ...)
- Operation must be dependable, certified in some cases

Confluence of diverse areas

Cost
Form factor
Severe constraints
Small-scale
Closed

Scheduling
Fault-tolerance
Wired networks



Noisy
Sensing
Large-scale
Real-time/actuation
Open

Linear
Adaptive
Distributed
Decentralized
Closed

Realistic (Integrated) Solutions

- CPS must tolerate
  - Failures
  - Noise
  - Uncertainty
  - Imprecision
  - Security attacks
  - Lack of perfect synchrony
  - Scale
  - Openness
  - Increasing complexity
  - Heterogeneity
  - Disconnectedness

Challenges Arise

- Assumptions underlying distributed systems technology has changed dramatically
  - New abstractions needed
  - Wired => wireless
  - Unlimited power => limited power
  - User interface (screen/mouse) => sensors/real world interface
  - Fixed set of resources => resources are dynamically added/deleted
  - Each node is important => aggregate behavior is important
  - Location unimportant => location is critical

New Theories

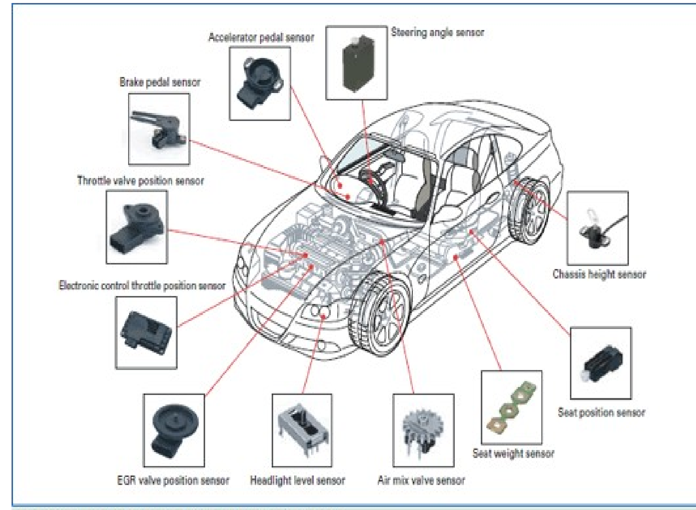
- Compositional
- Control Theory
- Optimization
- Real-Time
- Integration Issues
- Openness, Mobility, Uncertainty, Concurrency, Noise, Faults, Attacks, Self-Healing, etc.

Embedded Systems

- Embedded system: computing systems designed for a specific purpose.
- Embedded systems are everywhere!

Embedded Systems are getting more complex

- Modern high-end cars have over one hundred processors.
- Increasing number of sensors, actuators, smart control, GUI..
- Intelligent data fusion.



Alps' electric sensors have wide application in car use


F-35 Lightning II

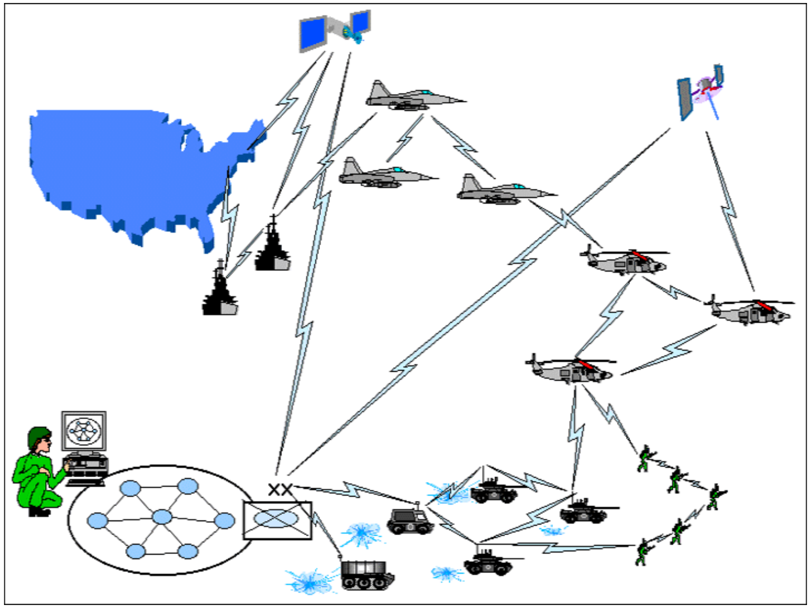
Helmet Mounted Display System



Optical Track.

… are more Interconnected

- Command-and-control network – real-time integration of vehicles, people, command.
- Geotagging: useful or scary?



+

- Many other examples
  - Power Grid
  - Medical systems
  - Transportation
  - Etc.

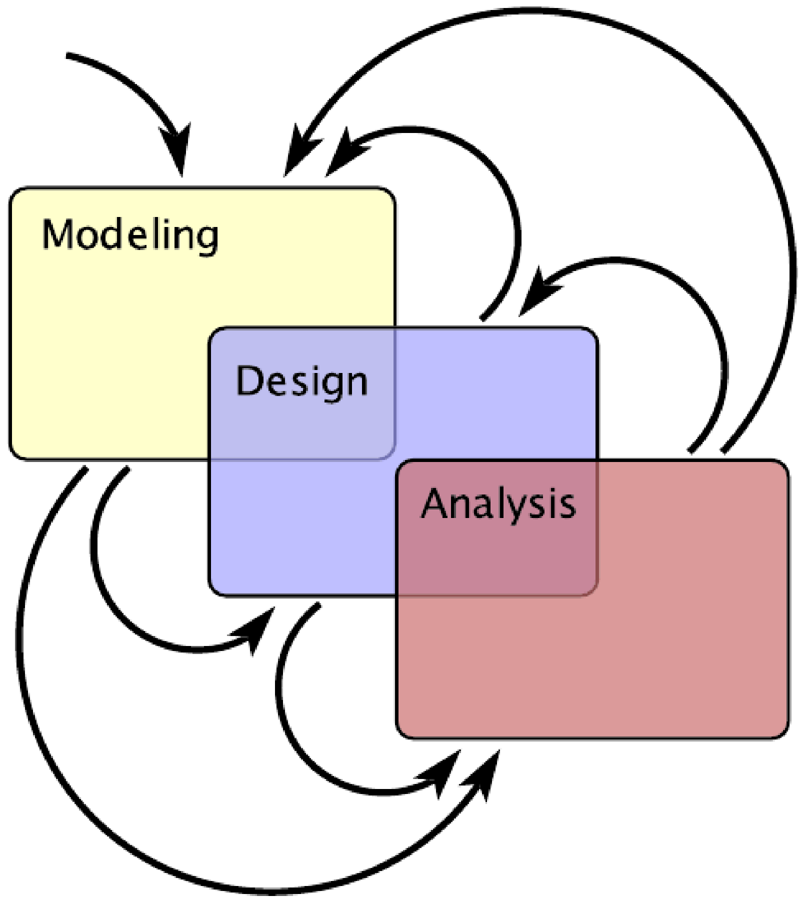
CPS – the next evolution

- Cyber-physical systems: integration of computation with physical processes.
- Still build on top of embedded computing systems.
- Interaction with the physical environment is promoted to a “first class citizen”.
- Promotes interaction and integration of subsystems
  - Classic safety-critical embedded systems: black boxes
  - CPS: white-boxes, open protocols
- Main goals:
  - Co-design the cyber and physical part of the system
  - Engineer a “system of systems”

CPS as multidisciplinary approach

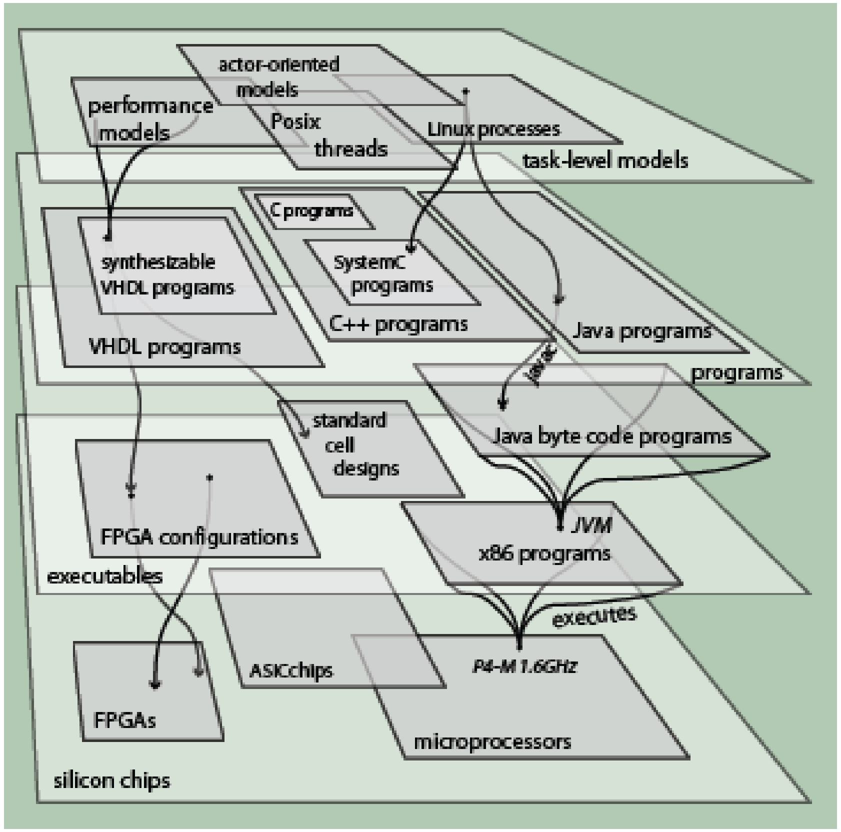
- Within ECE, CPS design requires competences in…
  - Computer Architecture
  - CAD & Embedded Design
  - Software Engineering
  - Control
  - Formal Verification
  - Real-Time Analysis
- … plus whatever engineering field(s) are related to the design of the plant/actuator.
- Problem: all such field and subfields have very different design & development conventions.
- Perhaps we need a new science of CPS design?

El proceso de diseño de un SCF



CPS Challenges – Design Abstractions

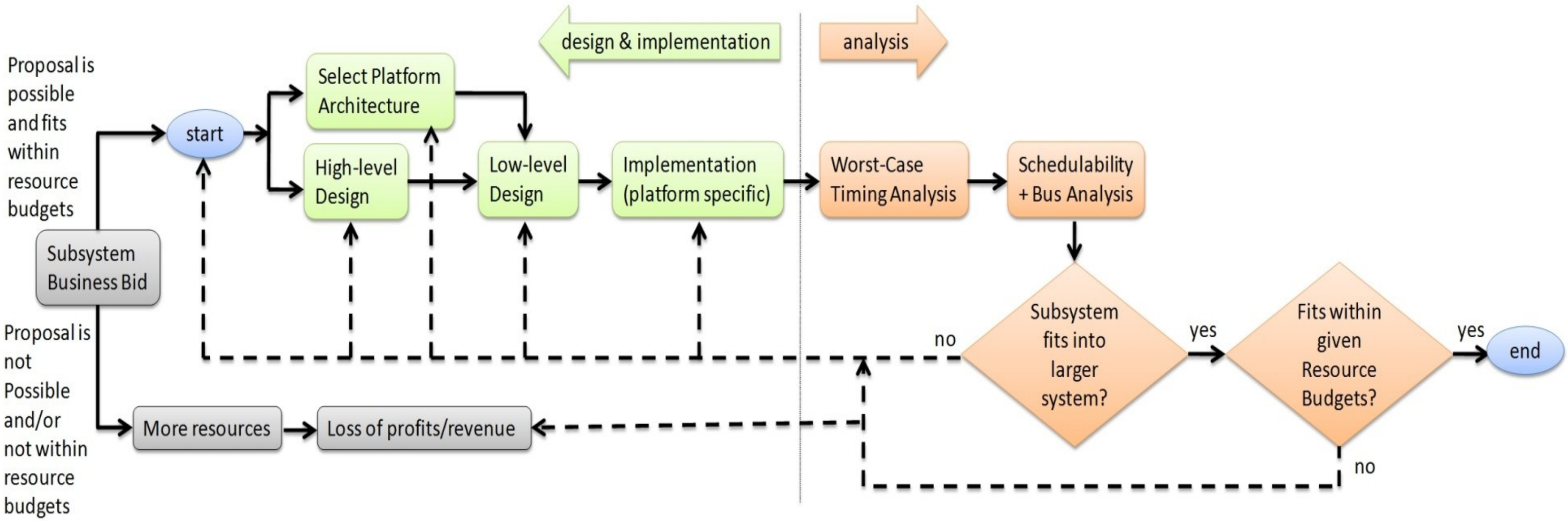
- We could argue that the biggest design challenge is in abstractions – the entire ECE design is a stack-based process.
- Unfortunately, most such abstractions do not directly encapsulate characteristics of the environment such as:
  - Concurrency
  - Criticality
  - Timing
- It is very hard to predict if the cyber part will meet the requirements of the physical part!



(from Prof. Edward Lee)

Current Design Flow

- The picture below exemplifies a typical design flow for an avionic subsystem.
- Analysis is required to verify that requirements are met.
- Analysis can only be performed after implementation.
- Recipe for disaster!



Reliable CPS: not so much!

- In 2007, 12 F-22s were going from Hawaii to Japan.
- After crossing the IDL, all 12 experienced multiple crashes.
  - No navigation
  - No fuel subsystems
  - Limited communications
  - Rebooting didn't help
- F-22 has 1.7 million lines of code.

F-22 Raptor

Example: Automotive Telematics

- In 2005, 30-90 processors per car
  - Engine control, break system, airbag deployment system
  - Windshield wiper, door locks, entertainment systems
  - Example: BMW 745i
    - 2,000,000 LOCs
    - Window CE OS
    - Over 60 microprocessors
      - 53 8-bit, 11 32-bit, 7 16-bit
    - Multiple networks
    - Buggy?

- Cars are sensors and actuators in V2V networks
  - Active networked safety alerts
  - Autonomous navigation
  - ...

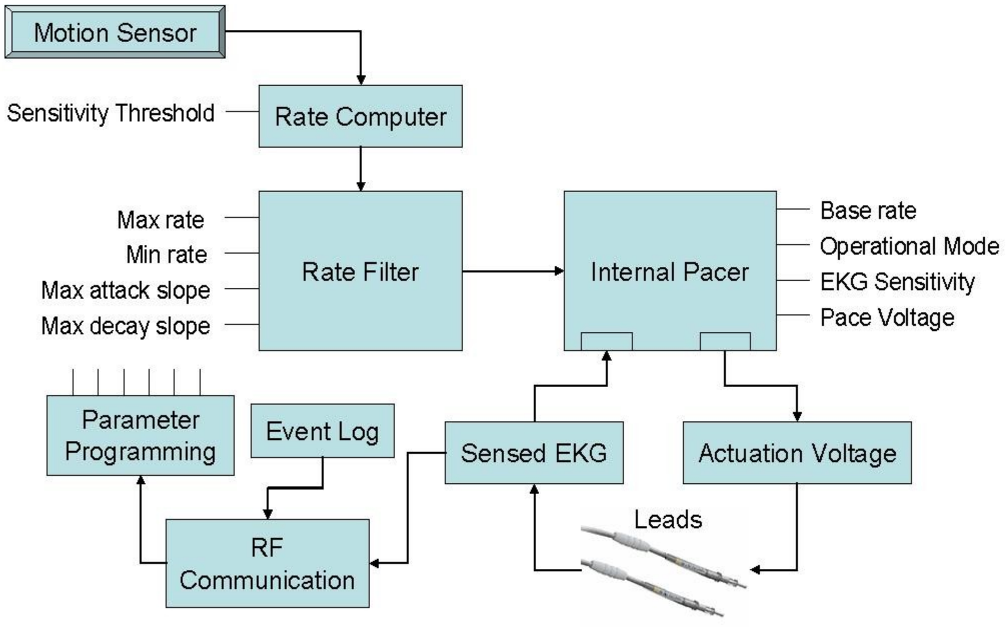
CPS Challenges - Safety

- Safety is hard to guarantee in interconnected and interdependent systems.

1. Do not trust communication channels.
   - Ex: medical plug-and-play initiative is looking to interconnect medical devices using wireless technology.
   - Problem: what happens if somebody jams the signal?
   - Each subsystem must be independently safe.
2. Do not trust the users.
   - Users are an (unfortunate) part of the systems.
   - Users are very error prone: over 90% of avionic accidents are caused by flight crew/controllers.
   - System must be protected against user mistakes

CPS Challenges - Safety

3. Do not trust lower-criticality subsystems.
   - Medical pacemaker composed of multiple subsystems.
   - Life-critical functionalities: base pacing, wiring, battery
   - Non-critical functionalities: adaptive pacing, logging, programming, RF communication.
   - Protect life-critical subsystem.



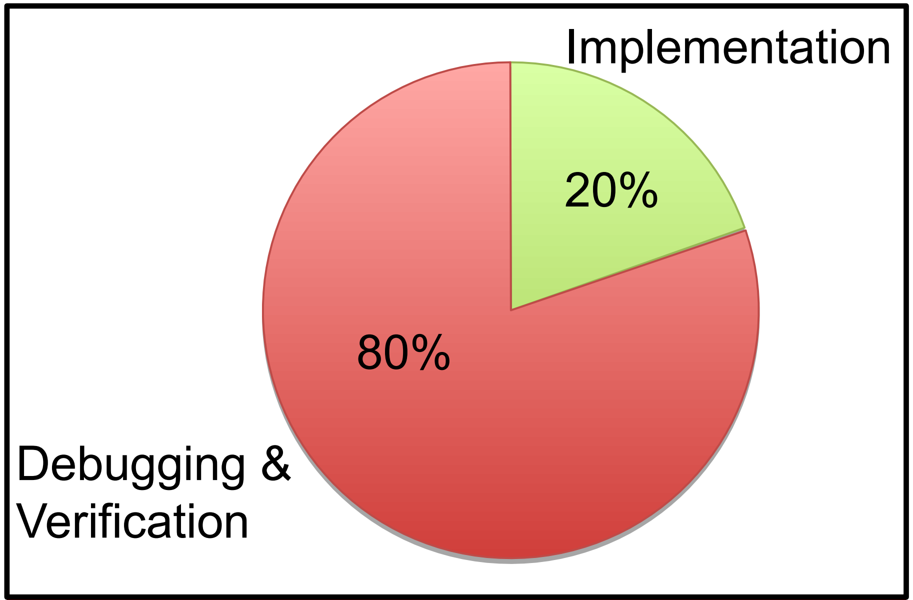
Verification & Certification

- How do we ensure safety?

1. Formal Verification
   - Build a model of the systems.
   - Prove (mathematically) that the system satisfies some safety property.
   - Problem#1: no good model for the whole system.
   - Problem#2: model is not implementation.
2. Certification
   - Usually a process-based mechanism: show that you have performed all process step according to some standard (ex: DO178a/b/c, IEC 61508).
   - Typically includes extensive testing.
   - Very expensive.

CPS Challenges - Integration

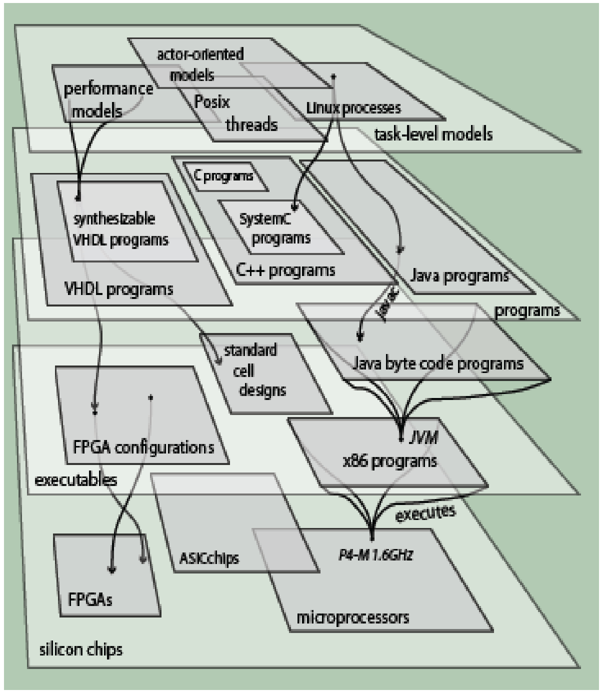
- Putting the system together is much more challenging that implementing the individual subsystems.
- Quiz (avionic systems): can you guess what % of $ goes in implementation vs debugging?
- Individual productivity for safety-critical code is reported as 6 lines/day!
  - F22: 1.7 million lines / 6 = 776 man-years
  - Perhaps the US$66.7billion program cost is not a surprise…
- Clearly the design process must be improved…



Avionic Development Cost

CPS Challenges - Timing Predictability

- The biggest architectural challenge.
- The lowest abstraction layer (transistors) is pretty deterministic – we know how to compute exact timings.
- However, higher levels lose all concept of timing.
  - Deep pipelining, caches, out-of-order and speculative execution…
  - Thread models, locking, interrupts…



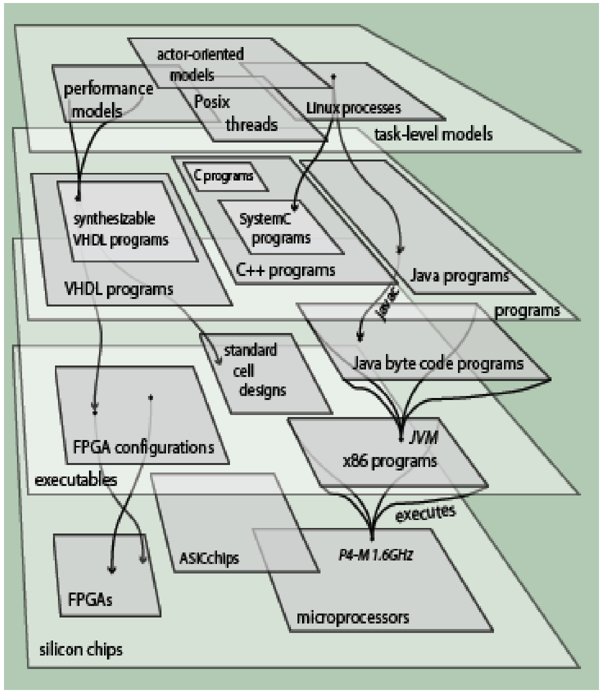
(by Prof. Edward Lee)

- This is fine for general purpose computing, but not for CPS – the physical system uses real time!

CPS Challenges - Timing Predictability

- We need to ensure that computation <u>always</u> finishes within guarantee time windows -> We are interested in worst-case performance, not average performance!

- <u>Timing predictability</u>
  – The time that the system requires to perform an operation should exhibit little variation
  – Such time should be easy to compute
  – It should not be affected by other parallel operations in the system.



(by Prof. Edward Lee)

Real-Time and Composability

- System correctness depends on:
  - Logical correctness: system produces correct results.
  - Temporal correctness: system produces results at the right time.
- Timing (real-time) analysis = verify temporal correctness.
- Ideally, we want composable analysis
  - Verify each subsystem in isolation
  - Then verify that there interaction is correct
- Unfortunately, this is very hard in practice…
- Main issue: hardware and software resources shared among multiple subsystems.

What is Required - Isolation

- Isolation: one subsystem should not affect another unrelated subsystem.
- Current architectures are pretty good at logical isolation…
  - Ex: memory protection and privilege levels in the CPU make sure that a process can not mess with the memory of another process or the OS.
- … but fairly poor at temporal isolation.

- Note #1: any and all hw isolation mechanisms are useless if not supported by the OS.
- Note #1: after the first OS was created, it took a while before hw architects started implementing protection mechanisms. So we stand a chance!

CPS Challenges – Software Models

- Current software programming models and languages are inadequate to support CPS design.
- C is by far the most popular language for embedded sys.
- C has no intrinsic support for concurrency, timing parameters, synchronization, etc.
- POSIX libraries (ex: threads) are often used, but again lack any explicit concept of timing.
- Extremely common operations in controller implementation:
  - specify that I want to execute an operation after a given amount of time
  - specify that I want to complete an operation within a given amount of time
- Why do I need to use OS constructs (times, watchdogs) for this?

Key Trends in Systems

- System complexity
  - Increasing functionality
  - Increasing integration and networking interoperability
  - Growing importance and reliance on software
  - Increasing number of non-functional constraints
- Nature of tomorrow's systems
  - Dynamic, ever-changing, dependable, high-confidence
  - Self-*(aware, adapting, repairing, sustaining)
- Cyber-Physical Systems everywhere, used by everyone, for everything
  - Expectations : 24/7 availability, 100% reliability, 100% connectivity, instantaneous response, remember everything forever, ...

R&D needs

- Development of high-confidence CPS requires
  - Engineering design techniques and tools
    - Modeling and analysis, requirements capture, hybrid systems, testing ...
    - Capture and optimization of inter-dependencies of different requirements
    - Domain-specific model-based tools
  - Systems Software and Network Supports
    - Virtualization, RTOS, Middleware, ...
    - Predictable (not best-effort) communication with QoS, predictable delay & jitter bounds, ...
    - Trusted embedded software components
      - To help structured system design and system development
      - To reduce the cost of overall system development and maintenance efforts
      - To support the reuse of components within product families
  - Validation and Certification
    - Metrics for certification/validation
    - Evidence-based certification, Incremental certification

Scientific challenges

- Computations and Abstractions
  - Computational abstractions
  - Novel Real-time embedded systems abstractions for CPS
  - Model-based development of CPS
- Compositionality
  - Composition and interoperation of cyber physical systems
  - Compositional frameworks for both functional, temporal, and non-functional properties
  - Robustness, safety, and security of cyber physical systems
- Systems & Network Supports
  - CPS Architecture, virtualization
  - Wireless and smart sensor networks
  - Predictable real-time and QoS guranattees at multiple scales
- New foundations
  - Control (distributed, multi-level in space and time) and hybrid systems - cognition of environment and system state, and closing the loop
  - Dealing with uncertainties and adaptability - graceful adaptation to applications, environments, and resource availability
  - Scalability, reliability, robustness, stability of system of systems
  - Science of certification - evidence-based certification, measures of verfication, validation, and testing

Sensado y actuación

Que es un sensor? Y un actuador?

- Un sensor es un dispositivo que mide una cantidad/magnitud física
  - Es una entrada
  - “Leer desde el mundo físco”
- Un actuador es un dispositivo que modifica una cantidad/magnitud física
  - Es una salida
  - “Escribir en el mundo físico”
- **Conectan el mundo físico con el mundo computacional**

Sensores y actuadores

- Sensores:
  - Cámaras
  - Acelerómetros
  - Giroscopios
  - Extensiómetro
  - Micrófonos
  - Magnetómetros
  - Radar/Lidar
  - Sesnores químicos
  - Sensores de presión
  - Interruptores
  - ...
- Actuadores:
  - Motores
  - Soleoides
  - LEDs, lasers
  - LCD
  - Parlantes
  - Interruptores
  - Válvulas
  - ...

Problemas de diseño con sensores

- **Calibración**
  - Relacionar medidas con el fenómeno físico
  - Puede aumentar los costos de producción dramáticamente
- **No-linearidad**
  - Mediadas pueden no ser proporcionales al modelo físico
  - Se puede requerir corrección
  - Retroalimentación puede ser usada para mantener el punto de operación en la región de linearidad
- **Muestreo**
  - Aliasing
  - Pérdida de eventos
- **Ruido**
  - *Signal conditioning*
  - Filtrado digital introduce latencia
- **Fallas**
  - Redundancia (problema de fusión de sensores)
  - Ataques

Redes

Sopa de tecnologías

- 1588
- 6LoWPAN
- 802.15.4
- 802.1(AS)
- 802.11
- AVB
- BLE
- CAN
- CoAP
- CSMA/CA
- GSM
- HART
- HTTP
- IoT
- IPv6
- LTE
- MAC
- PAN
- PTP
- QoS
- REST
- TDMA
- TSMP
- TSN
- TTEthernet
- TTP
- WAN
- WLAN
- WPAN

Redes cableadas

- Ethernet
- CAN: Controller Area Network (Bosch, 1983)
- TTP: Time-Triggered Protocol (Vienna U. of Tech.)
- FlexRay (Automotive industry, deployed 2006...)
- TTEthernet (Time-triggered Ethernet)
- TSN (Time-sensitive networks)

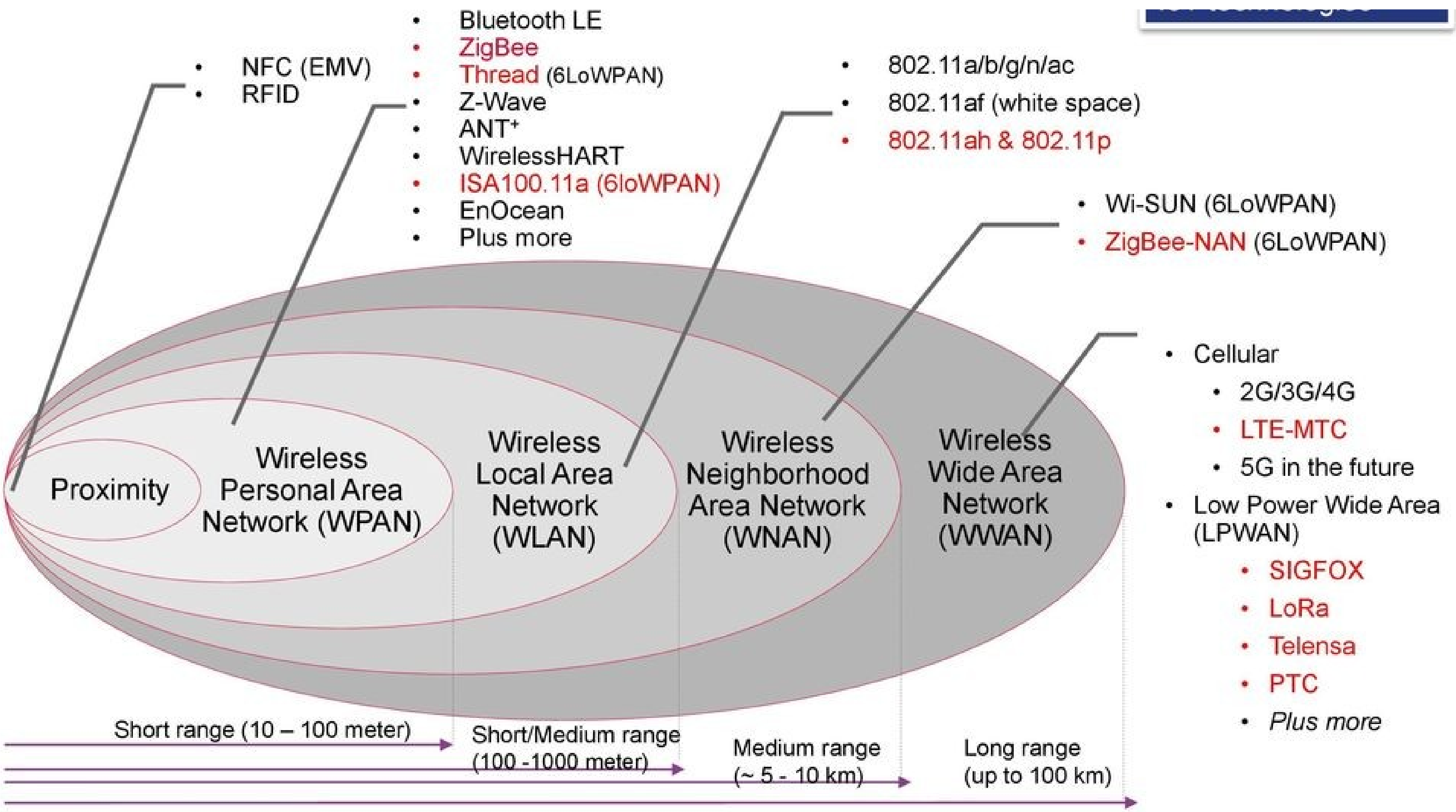
- Problemas en SCF: Control sobre latencia y timing, ancho de banda garantizado, redundancia, tolerancia a errores

Redes cableadas

- Control de acceso al medio:
  - CSMA/CA
  - Time Slotted (TDMA)
- Routing
  - Buffering, pérdida de paquetes
  - Enrutamiento
  - QoS, Prioridad

Redes inalámbricas

- Personal Area Networks (PANs)
  - Bluetooth, BLE
- Local Area Networks (LANs)
  - WiFi (IEEE 802.11.*)
  - Zigbee, et al. (IEEE 802.15.4*)
- Wide Area Networks (WANs)
  - GSM (for voice, some data)
  - LTE and 5G (for audio, video)
  - Sigfox, Lora, LTE-M (for Machine-to-Machine, M2M, IoT)

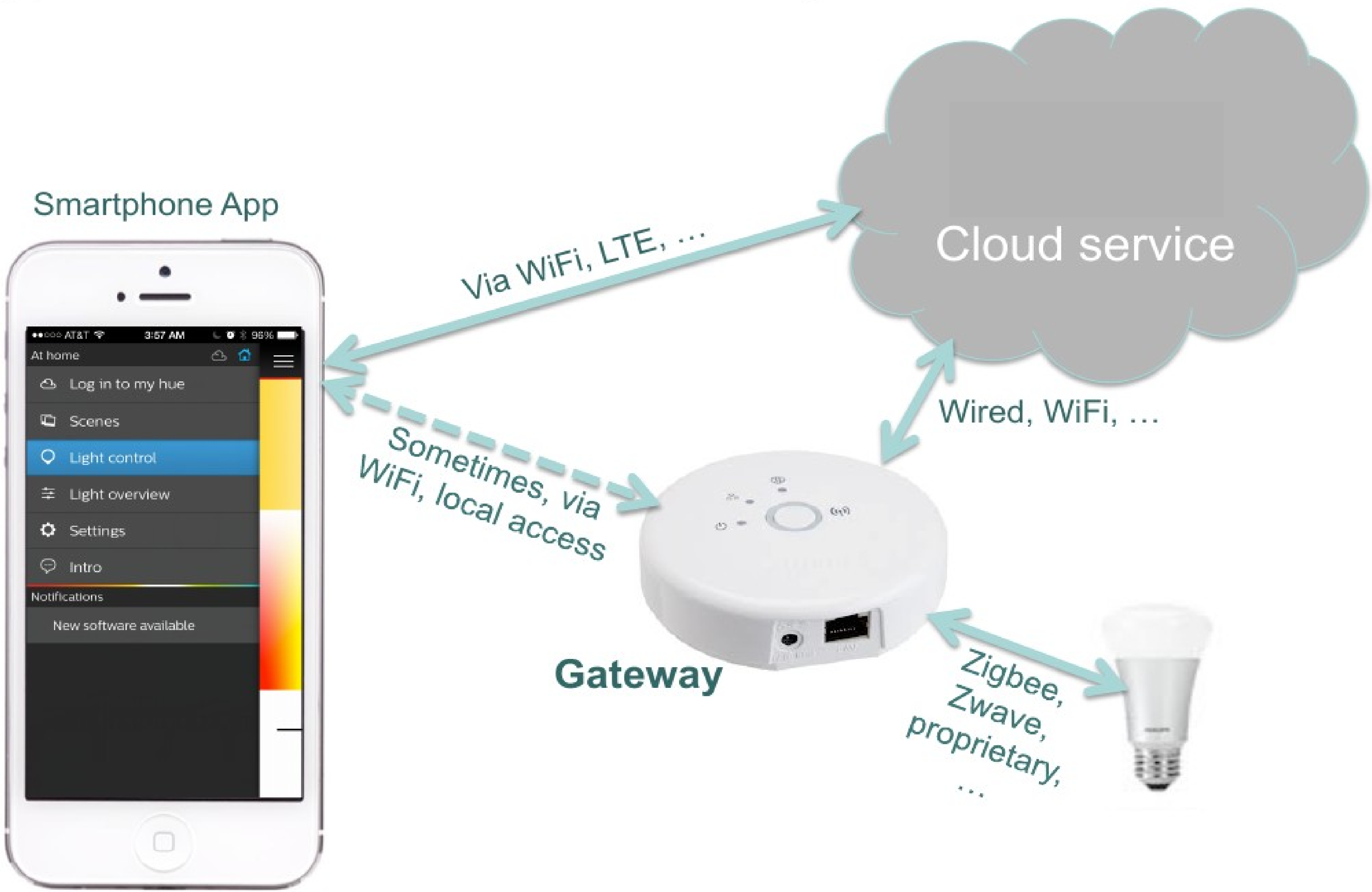


Note 1: No stringent definition of what is considered WPAN, WLAN, WWAN.
Note 2: What is shown is not a complete list of radio formats

Redes inalámbricas

- ¿Que tecnología uso?
- Eficiencia energética
- Topología
- Alcance
- Costo
- Accesibilidad
- QoS

Arquitectura de red



La nube

- Complejidad de los Data Centers
- SDN
- Enrutamiento específico
- Big Data