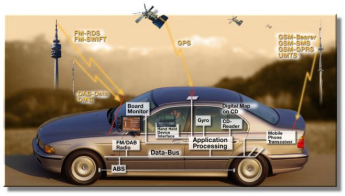
Cyber Physical Systems: The Next Computing Revolution

Insup Lee
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
www.cis.upenn.edu/~lee/

CIS 541, Spring 2010

Example Embedded Systems

Automobiles



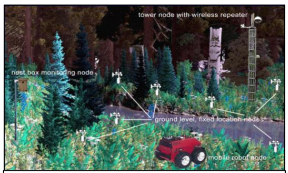
Medical

Entertainment

Handheld

Airplanes

Military

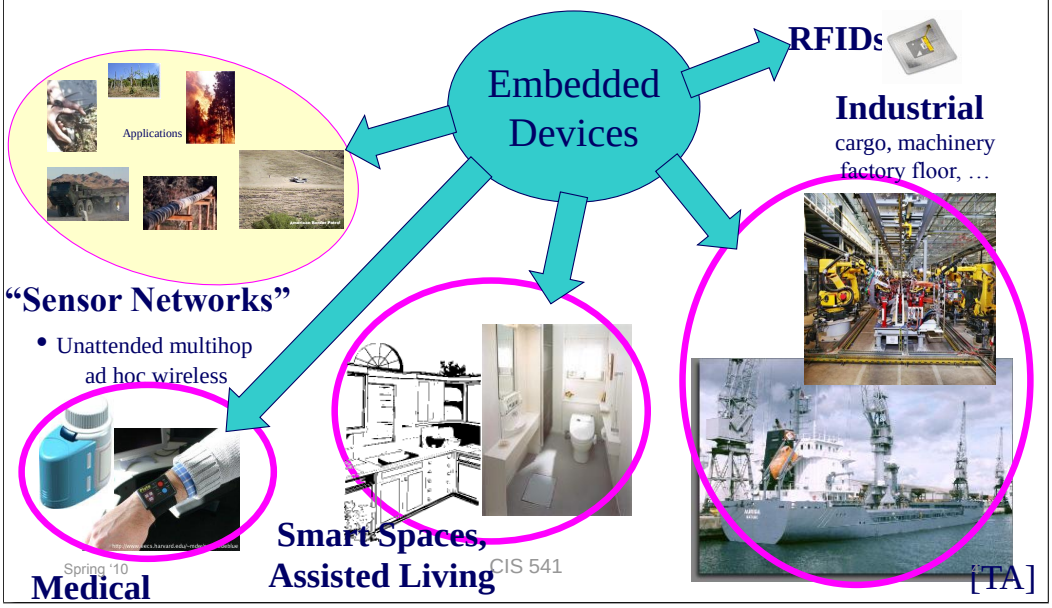


Environmental Monitoring

The Next Computing Revolution

- 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)

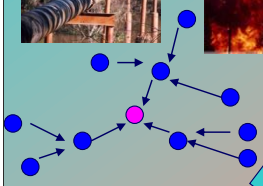
Cyber-Physical Systems:
Trend 1: Proliferation (By Moore's Law)



[TA]

Cyber-Physical Systems:
Trend 2: Integration at Scale (Isolation has cost!)

- Low end: ubiquitous embedded devices
  - Large-scale networked embedded systems
  - Seamless integration with a physical environment
- High end: complex systems with global integration
  - Examples: Global Information Grid, Total Ship Computing Environment

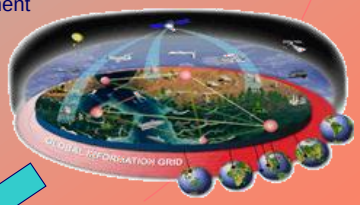


World Wide Sensor Web
(Feng Zhao)

Picture courtesy of
Patrick Lardieri

Total Ship Computing Environment
(TSCE)

Global Information Grid

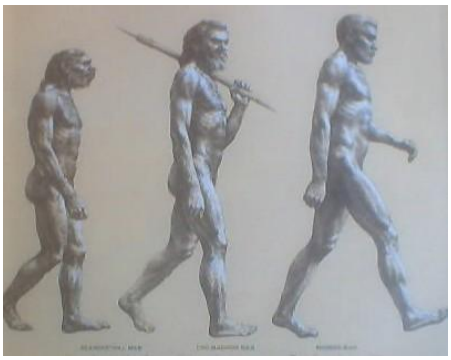


Future Combat System
(Rob Gold)

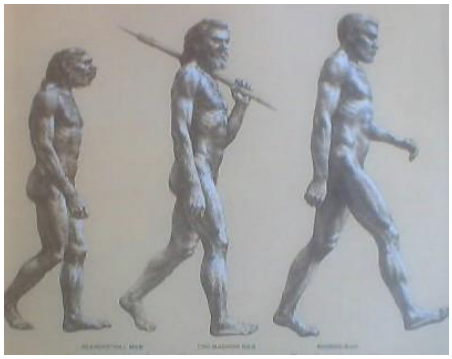
Integration and Scaling Challenges

Low End → High End

Cyber-Physical Systems:
Trend #3: Biological Evolution



Cyber-Physical Systems:
Trend #3: Biological Evolution

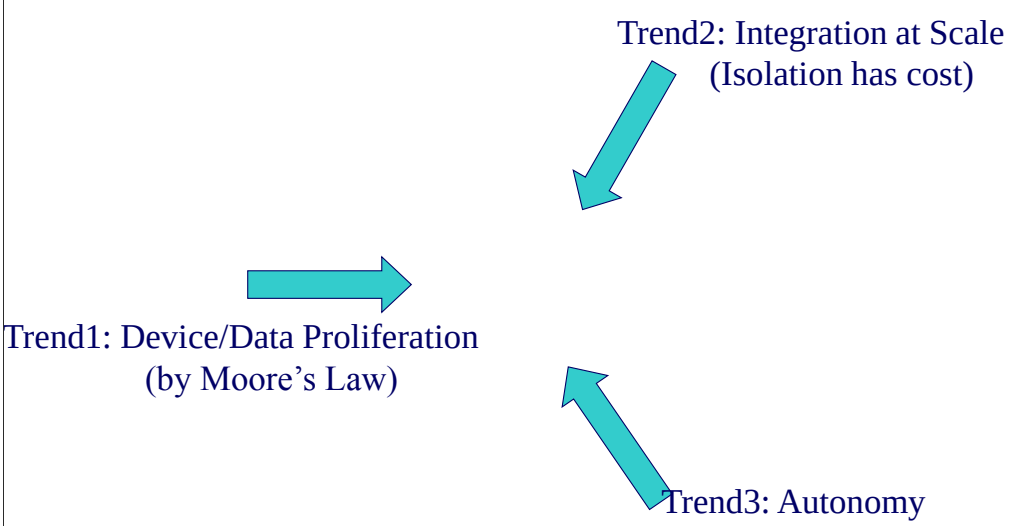


- **It's too slow!**
  - The exponential proliferation of embedded devices (afforded by Moore's Law) is ***not*** matched by a corresponding increase in human ability to consume information!

→ Increasing autonomy (human out of the loop), direct world access

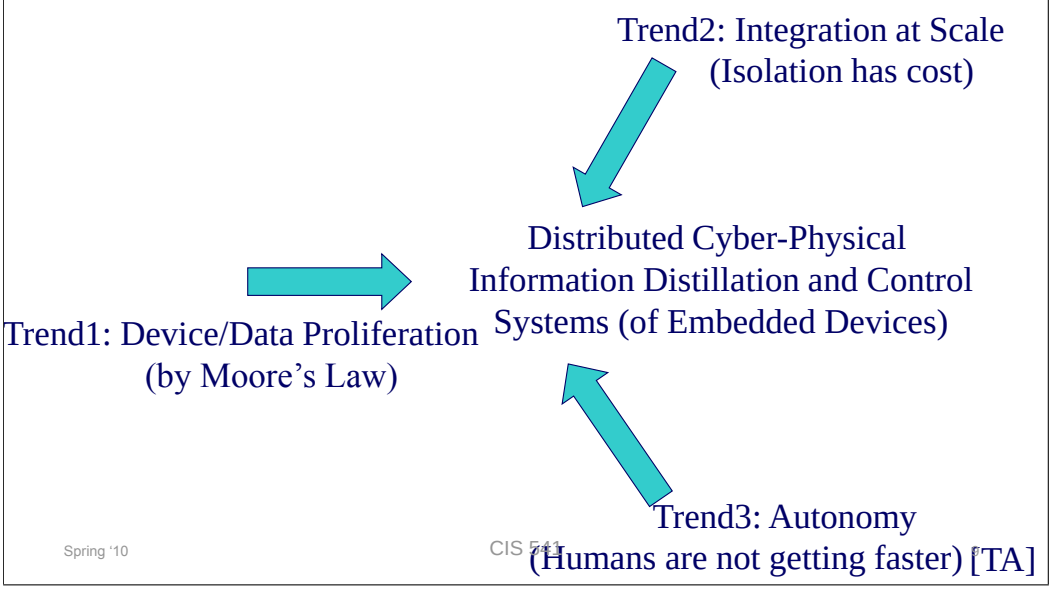
[TA]

Confluence of Trends
The Overarching Challenge

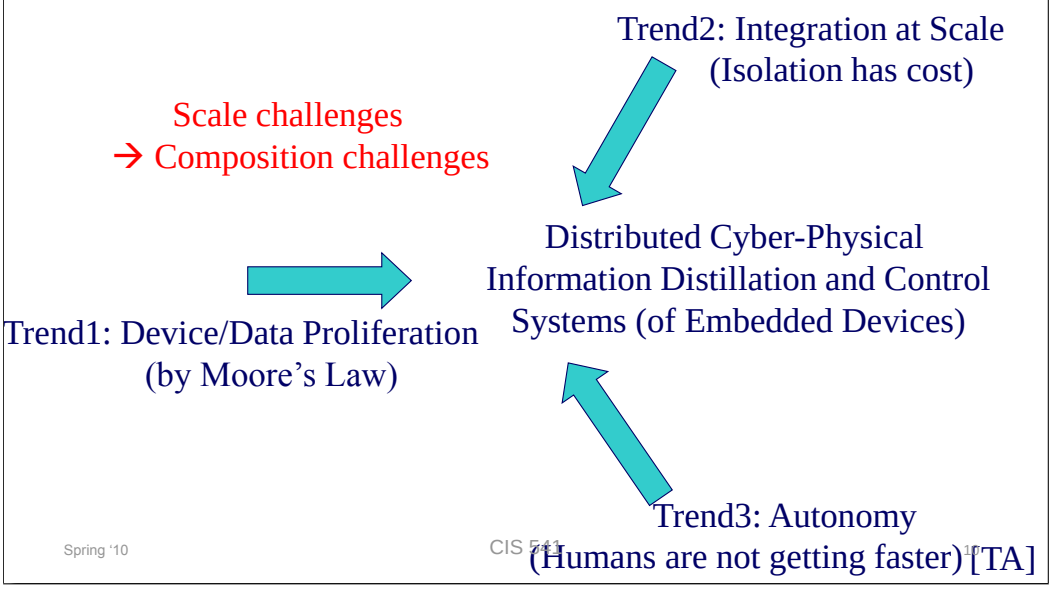


[TA]

Confluence of Trends
The Overarching Challenge

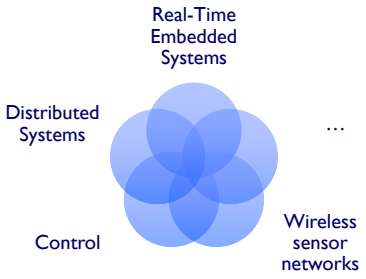


Confluence of Trends
The Overarching Challenge



What are Cyber Physical Systems?

- *Cyber-physical systems* (CPSs) are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core.
- A cyber-physical system integrates computing, communication, and storage capabilities with the monitoring and/or control of entities in the physical world
  - from the nano-world to large-scale wide-area systems of systems
  - dependably, safely, securely, efficiently and in real-time
- Convergence of computation, communication, and control



Characteristics of CPS

- Some defining characteristics:
  - Cyber – physical coupling driven by new demands and applications
    - Cyber capability in every physical component
    - Large scale wired and wireless networking
    - Networked at multiple and extreme scales
  - Systems of systems
    - New spatial-temporal constraints
    - Complex at multiple temporal and spatial scales
    - Dynamically reorganizing/reconfiguring
    - Unconventional computational and physical substrates (Bio? Nano?)
  - Novel interactions between communications/computing/control
    - High degrees of automation, control loops must close at all scales
    - Large numbers of non-technical savvy users in the control loop
  - Ubiquity drives unprecedented security and privacy needs
  - Operation must be dependable, certified in some cases
- Tipping points/phase transitions
  - Not desktop computing, Not traditional, post-hoc embedded/real-time systems, Not today's sensor nets
  - Internet as we know now, stampede in a moving crowd, …

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 LOC
    - 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
  - …

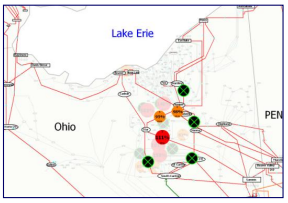
Example: Health Care and Medicine

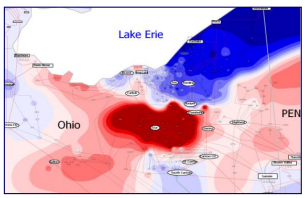
- National Health Information Network, Electronic Patient Record initiative
  - Medical records at any point of service
  - Hospital, OR, ICU, …, EMT?
- Home care: monitoring and control
  - Pulse oximeters (oxygen saturation), blood glucose monitors, infusion pumps (insulin), accelerometers (falling, immobility), wearable networks (gait analysis), …
- Operating Room of the Future
  - Closed loop monitoring and control; multiple treatment stations, plug and play devices; robotic microsurgery (remotely guided?)
  - System coordination challenge
- Progress in bioinformatics: gene, protein expression; systems biology; disease dynamics, control mechanisms

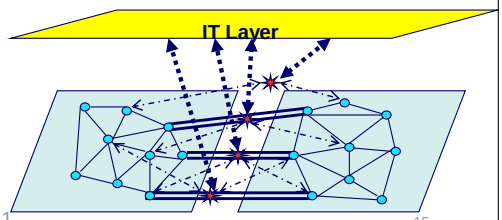
Images thanks to Dr. Julian Goldman, Dr. Fred Pearce

Example: Electric Power Grid

- **Current picture:**
  - Equipment protection devices trip locally, reactively
  - Cascading failure: August (US/Canada) and October (Europe), 2003
- **Better future?**
  - Real-time cooperative control of protection devices
  - Or -- self-healing -- (re-)aggregate islands of stable bulk power (protection, market motives)
  - Ubiquitous green technologies
  - Issue: standard operational control concerns exhibit wide-area characteristics (bulk power stability and quality, flow control, fault isolation)
  - Technology vectors: FACTS, PMUs
  - Context: market (timing?) behavior, power routing transactions, regulation



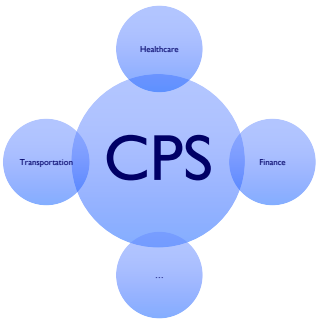




Images thanks to William H. Sanders, Bruce Krogh, and Marija Ilic

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



In general, any "*X* by wire(less)" where *X* is anything that is physical in nature.

Grand Visions and Societal Impact

- Near-zero automotive traffic fatalities, injuries minimized, and significantly reduced traffic congestion and delays
- Blackout-free electricity generation and distribution
- Perpetual life assistants for busy, older or disabled people
- Extreme-yield agriculture
- Energy-aware buildings
- Location-independent access to world-class medicine
- Physical critical infrastructure that calls for preventive maintenance
- Self-correcting and self-certifying cyber-physical systems for "one-off" applications
- Reduce testing and integration time and costs of complex CPS systems (e.g. avionics) by one to two orders of magnitude

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, …
  - Classes: young to old, able and disabled, rich and poor, literate and illiterate, …
  - Numbers: individuals, special groups, social networks, cultures, populations, …

Societal Challenge

- How can we provide people and society with cyber-physical systems that they can trust their lives on?

Trustworthy: reliable, secure, privacy-preserving, usable, etc.

- Partial list of complex system failures
  - Denver baggage handling system ($300M)
  - Power blackout in NY (2003)
  - Ariane 5 (1996)
  - Mars Pathfinder (1997)
  - Mars Climate Orbiter ($125M,1999)
  - The Patriot Missile (1991)
  - USS Yorktown (1998)
  - Therac-25 (1985-1988)
  - London Ambulance System (£9M, 1992)
  - Pacemakers (500K recalls during 1990-2000)
  - Numerous computer-related Incidents wth commer aircraft (http://www.rvs.uni-bielefeld.de/publications/compendium/incidents_and_accidents/index.html)

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

Software, the Great Enabler

- Good news: anything is possible in software!
- Bad news: anything is possible in software!

- It is the software that affects system complexity and also cost.
  - Software development stands for 70-80 % of the overall development cost for some embedded systems.

Interaction Complexity

- We know how to design and build components.
- Systems are about the interactions of components.
  - Some interactions are unintended and unanticipated
    - Interoperability
    - Emerging behaviors
- "Normal Accidents", an influential book by Charles Perrow (1984)
  - One of the Three Mile Island investigators
  - And a member of recent NRC Study "Software for Dependable Systems: Sufficient Evidence?"
  - A sociologist, not a computer scientist
- Posits that sufficiently complex systems can produce accidents without a simple cause due to
  - interactive complexity and tight coupling

Potential Accidental Systems

- Many systems created without conscious design by interconnecting separately designed components or separate systems.
  - Unsound composition: the interconnects produce desired behaviors most of the time
  - Feature interactions: promote unanticipated interactions, which could lead to system failures or accidents
- Modes of interactions
  - Among computation components
  - Through share resources
  - Through the controlled plant (e.g., the patient)
  - Through human operators
  - Through the larger Environment
- E.g., Medical Device PnP could facilitate the construction of accidental systems
  - blood pressure sensor connected to bed height, resulting in the criticality inversion problem

Unexpected interactions

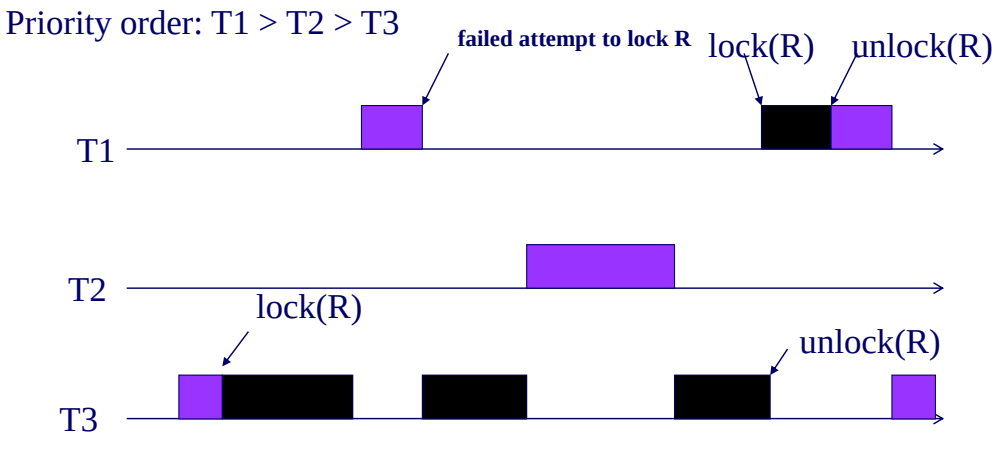
- Landed on the Martian surface on July 4th, 1997
- Unconventional landing – boucing into the Martian surface
- A few days later, not long after Pathfinder started gathering meteorological data, the spacecraft began experiencing total system reset, each resulting in losses of data

Incompatible Cross Domain Protocols

Pathological Interaction between RT and synchronization protocols Pathfinder caused repeated resets, nearly doomed the mission

[Sha]

The Priority Inversion Problem



T2 is causing a higher priority task T1 wait !

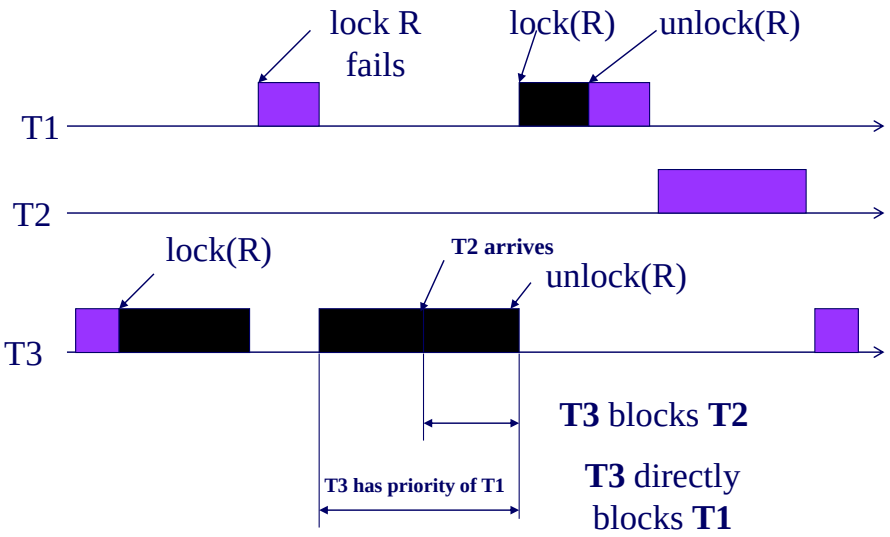
Priority Inversion

1. T1 has highest priority, T2 next, and T3 lowest
2. T3 comes first, starts executing, and acquires some resource (say, a lock).
3. T1 comes next, interrupts T3 as T1 has higher priority
4. But T1 needs the resource locked by T3, so T1 gets blocked
5. T3 resumes execution (this scenario is still acceptable so far)
6. T2 arrives, and interrupts T3 as T2 has higher priority than T3, and T2 executes till completion
7. In effect, even though T1 has priority than T2, and arrived earlier than T2, T2 delayed execution of T1
8. This is "priority inversion" !! Not acceptable.

Priority Inversion and the MARS Pathfinder

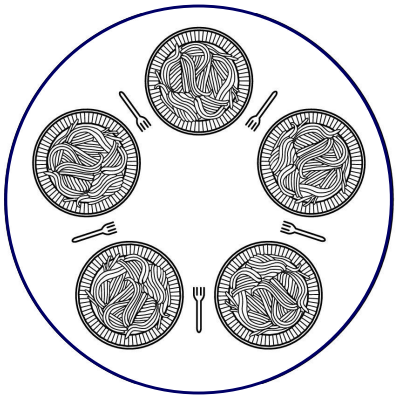
- What happened:
  - Pathfinder has an "information bus" thread [very critical – used by navigation, etc. – high priority]
  - The meteorological data gathering thread ran as an infrequent, low priority thread, and used the information bus to publish its data (while holding the mutex on bus).
  - A communication task that ran with medium priority.
  - It is possible for an interrupt to occur that caused (medium priority) communications task to be scheduled during the short interval of the (high priority) information bus thread was blocked waiting for the (low priority) meteorological data thread.
  - After some time passed, a watch dog timer goes off, noticing that the data bus has not been executed for some time, it concluded that something had gone really bad, and initiated a total system reset.

Priority Inheritance Protocol



Dining Philosophers

- Philosophers eat/think
- Eating needs 2 forks
- Pick one fork at a time
- How to prevent deadlock



The Dining Philosopher Problem

- Five philosopher spend their lives thinking + eating.
- One simple solution is to represent each fork by a semaphore.
- Down (i.e., P) before picking it up & up (i.e., V) after using it.
- `var fork: array[0..4] of semaphores=1`
  `philosopher i`
-
```
repeat
  down( fork[i] );
  down( fork[i+1 mod 5] );
     ...
     eat
     ...
  up( fork[i] );
  up( fork"[i+1 mod 5] );
     ...
     think
     ...
forever
```
- Is deadlock possible?

Number of possible states

- 5 philosophers
- Local state (LC) for each philosoper
  - thinking, waiting, eating
- Glabal state = (LC 1, LC 2, …, LC5)
  - E.g., (thinking, waiting, waiting, eating, thinking)
  - E.g., (waiting, eating, waiting, eating, waiting)
- So, the number of global states are 3 ** 5 = 243
- Actually, it is a lot more than this since waiting can be
  - Waiting for the first fork
  - Waiting for the second fork

Number of possible behaviors

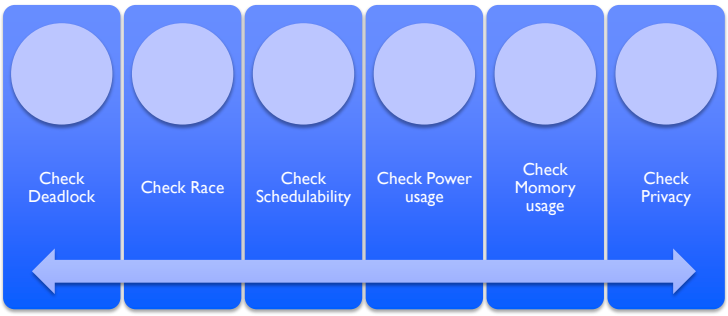
- Sequence of states
- Initial state: (thinking,thinking,thinking,thinking,thinking)
- The number of possible behaviors = 5 x 5 x 5 x …
- Deadlock state: (waiting,waiting,waiting,waiting, waiting)
- Given the state transition model of your implementation, show that it is not possible to reach the deadlock state from the initial state.

What is Formal Methods?

- These are ways of checking whether a property of a computational system holds for all possible executions
- As opposed to testing or simulation
  - These just sample the space of behaviors
  - X^2 − y^2 = (x − y)(x + y) vs. 5*5-3*3 = (5-3)*(5+3)
- Formal analysis uses automated model checking, theorem proving, static analysis, run-time verification
- Exponential complexity:
  - works best when property is simple
    - static analysis for runtime errors
    - run-time verification for run-time monitoring and checking
  - Or computational system is small or abstract
    - a specification or model rather than C-code
    - E.g, finite state models of device drivers, operator mental models, etc.

A (Research) Vision

- To provide CPS application engineers with lightweight "push-button" tools, each checking a specific application-specific property [Wing].



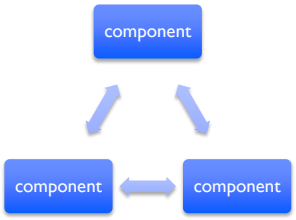
Sources of difficulties

- *Unsound compositionality*
  - incompatible abstractions, incorrect or implicit assumptions in system interfaces.
  - incompatible real time, fault tolerance, and security protocols.
  - combination of components do not preserve functional and para-functional properties; *unexpected feature interactions.*
- *Inadequate development infrastructure*
  - the lack of domain specific-reference architectures, tools, and design patterns with known and parameterized real time, robustness, and security properties.
- *System instabilities*
  - faults and failures in one component cascade along complex and unexpected dependency graphs resulting in catastrophic failures in a large part or even an entire system.

Compositionality

- Compositionality
  - system-level properties can be established by composing independently analyzed component-level properties
- Modeling and verification of combined behaviors of interacting systems
  - E.g., Assume/guarantee reasoning
    - If component C1 guarantees P1, assuming C2 ensures P2, and
    - component C2 guarantees P2, assuming C1 ensures P1
    - Then, we can conclude that C1 || C2 guarantees P1 and P2.

    Looks circular but it is sound…
  - Can be extended to many components
  - Can be used informally or formally, using formal methods.



Assurance and Certification

- How do we provide assurance that we've done so?
  - All assurance is based on arguments that purport to justify certain claims, based on documented evidence
- There are two approaches to assurance: implicit (standards based), and explicit (goal-based)
- Science of Certification
  - Certification is ultimately a judgment that a system is adequately safe/secure/whatever for a given application in a given environment
  - But the judgment should be based on as much explicit and credible evidence as possible
  - Incremental Certification
  - A Science of Certification would be about ways to develop that evidence