Ubiquitous computing – The trends…

- **Devices, ever smaller**
  - Laptops, phones, PDAs, sensors…

- **Networking, wireless, wired & global**
  - Wireless & Internet everywhere

- **Systems/software**
  - Self-*$^*$
  - Mobile
  - Adaptive
  - Context-aware

- **How to design & engineer**
  - Adaptive systems and networks?

- **How to ensure**
  - Dependability and performance?
Why must we verify?

“Testing can only show the presence of errors, not their absence.”

“In their capacity as a tool, computers will be but a ripple on the surface of our culture. In their capacity as intellectual challenge, computers are without precedent in the cultural history of mankind.”

Edsger Wybe Dijkstra
1930–2002

To rule out errors must consider all possible executions – often not feasible mechanically!
But my program works!

• True, there are many successful large-scale complex computer systems...
  – Online banking, electronic commerce
  – Information services, online libraries, business processes
  – Supply chain management
  – Mobile phone networks

• Yet many new potential application domains, far greater complexity, higher expectations
  – Automotive drive-by-wire
  – Medical sensors: heart rate & blood pressure monitors
  – Intelligent buildings and spaces: WiFi hotspots, environmental sensors

• Learning from mistakes costly...
The NASA Mars space mission

Mars Climate Orbiter
Launched 11th December 1998

LOST 23rd September 1999
Conversion error from English units to metric in navigation software
Cost: $125 million

Mars Polar Lander
Launched 3rd January 1999

LOST 3rd December 1999
Engine shutdown due to spurious signals that gave false indication that spacecraft had landed

Cost: $125 million
Toyota Prius

Drive-by-wire, in car network

100s of embedded components used in modern cars

In May 2005, Toyota recalls about 75,000 cars. Some Prius drivers have reported sudden stalling or stopping at highway speeds.

According to reports “the stalling problem is due to a software glitch in its sophisticated computer system.”

Such problems are becoming more common: BMW 7 series, ... Cost $?
"Code Red" Worm
Exploiting Buffer Overflow in Microsoft IIS Indexing Service DLL

"Code Red" worm is self-replicating malicious code. It attempts to connect to TCP port 80 on a randomly chosen host, then sends a crafted HTTP GET request to the victim, attempting to exploit a buffer overflow. Web pages on victim machines may be defaced, suffer performance degradation, denial of service, etc.

Cost > $2 billion
Formal verification techniques

- **Mathematical proof**
  - manual
- **Theorem proving**
  - infinite-state models
  - computer-assisted
  - human interaction
- **Model checking**
  - finite-state models
  - fully automatic
  - exhaustive
- **Static analysis**
  - shallow properties
  - lightweight
  - fast

![Diagram showing the relationship between verification techniques and effort](image)