Engineering High-Dependability Systems (1)

CS3 / SEOC1

Note 16

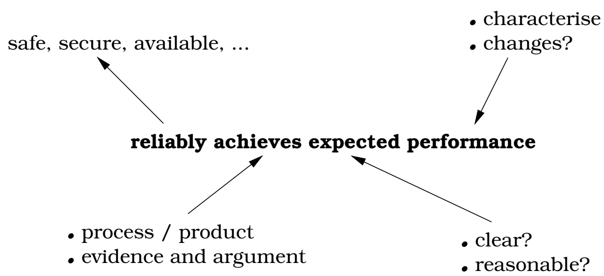
Dependability of Computer-Based Systems

Dependability – high integrity, reliable, safe, secure, available, fault tolerant, . . .

of Computer-Based – software is (significant) part of "whole machine"

Systems – involving "machine", humans, organisations, environment, . . .

What is dependability?



Dependable: justified trust in a service

(Some) Flavours of Dependable Systems

Safety-Critical: failure leads to serious injury, loss of life, or significant environmental damage

Security-Critical: access control, permissions and monitoring (potentially in the face of malicious attack) a key issue

Fault-Tolerant: system is *robust*. Can withstand errors in, or failures of, parts of the system (e.g. auto-pilots)

High-Reliability: likelihood of failure-on--demand exceptionally low (e.g. fire-safety shutdown systems)

What is undependability?

"Classic" high profile failures:

- Mars Climate Orbiter
- Ariane 5
- Therac 25 ...

What else?

- pervasiveness of computers (eg, Y2K)
- multiple low-criticality failures
- dependence of society
- service loss: "the system's down"

Impact on organisations

- NATS, healthcare, finance, ...

NASA's Mars Climate Orbiter

- part of Mars Surveyor programme (1993)
- developed at cost of $ 327.6 million (orbiter and lander)
- launched December 1998
- intended to enter Mars orbit September 1999, at 210km altitude
- September 23rd 1999, attempted orbit at 57km, burned up in Martian atmosphere

Mars Climate Orbiter: Investigation

- Phase 1 Mishap Investigation report, November 1999
  - root cause: failure to use metric units in ground software file "Small Forces"
  - team developing `SM_FORCES` used English units of pounds-seconds
  - team developing navigation software algorithm assumed metric units of Newton-seconds
  - Project SIS (Software Interface Specification) not followed
- contributing causes
  - process did not adequately address transition from development to operations
  - inadequate communication between teams
  - V & V process did not adequately address ground software

Therac 25 Radiotherapy Machine

- Therac-25 had two operating modes:
  - low intensity (electron radiation), wide spread
  - high intensity (X-ray radiation), tight focus
- software error in data entry permitted high intensity, wide spread
  - X-rays generated by placing tungsten shield as "filter" for high-intensity electron beam
  - set-up process takes considerable time
  - changes *during* set-up not validated
- 6 known accidents between June 1985 and January 1987, leading to 2 confirmed deaths
- hardware interlock in Therac-20 removed (software error present, but caused blown fuse)

(Some) Other Major S/W Failures

- London Ambulance Service
- Taurus Financial System
- CUFS (Cambridge University Financial System)
- Swanwick ATC? Proposed 1988 (for 1996), building commenced 1991, completed 1994, software working “by winter 2002”?

Safety Critical Systems

1. Variety of industrial sectors; both regulated and (relatively) unregulated
2. safety cases: “arguments” of acceptable safety of proposed system
3. focus on design for assessment
4. motivation/drivers for “safety culture”
5. “whole system” issues
6. software not necessarily susceptible to “traditional” engineering approaches

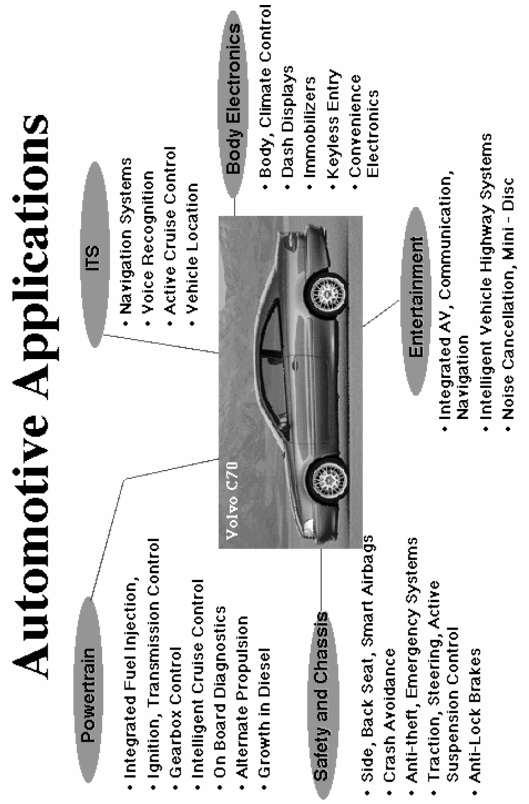
Domains of safety Critical Systems

Regulated:

- hazardous manufacturing (chemical, explosives)
- travel and transport (air, rail, sea)
- energy (nuclear, petrochemical)

(less) regulated:

- automotive (eg, engine controllers, ABS)
- medical informatics (eg, radiotherapy, anaesthetics, medical expert systems)



Regulation and Assessment

- Regulatory standards:
  - national and international
  - generic and domain-specific
  - independent assessment and regulatory authorities
- The safety case:

  pre-1990's: largely prescriptive

  1988: Piper-alpha; Cullen inquiry highly critical of "box ticking"

  post-Cullen: move to *goal-setting* standards

Motivation and drivers for safety (1)

- Economic – cost benefit analysis (one life $\approx$ £1-2 million)
- Responsibility
  - developer versus assessor versus regulator
  - in-house versus 3rd party (eg, COTS/SOUP)
- Liability; eg British Rail:

  pre-privatisation: HSE, rail regulatory authority

  post-privatisation: HSE, rail regulatory authority, TOC's, Railtrack, SPAD working party, 3rd party maintenance, strategic rail regulators, rail safety assessors, ...

Motivation and drivers for safety (2)

- History:
  - design for last 3 significant accidents
    - e.g. Clapham, Ladbrooke Grove, Selby?
  - safety culture "disaster-driven"
    - Cullen report on Piper Alpha
    - Titanic
  - no significant automotive/medical disasters
  - ...yet...

Software Engineering for Safety Critical Systems

- No "new" software engineering techniques
- adoption of traditional, physical engineering techniques:
  - for design (eg, triple modular redundancy, fault tolerance, failsafes, error recovery)
  - for analysis (hazard analysis, fault tree analysis, failure modes and effects analysis)
- ... but software unlike physical systems
  - not "convex"
  - high functional complexity
  - common mode failures
  - complex dependencies
  - software errors are all latent