Cpr E 545 Fault Tolerant Systems Dependable Computing

- Dependability
  - The quality of service such that reliance can be justifiably placed on the service
- Service
  - Delivered by a system and is the system behavior as perceived by another special system
- System failures
  - Delivered service deviates from the specified behavior
  - It occurs because the system is erroneous

Service Specification

- Abstraction of a system's behavior which is agreed upon
- Specification may change, still (re-)agreed upon
- Faults in the system may be
  - Physical faults
  - Human-made faults
  - Design faults
  - Interaction faults

Dependable Computing

- Combined utilization of
  - Fault Avoidance
    - By construction, prevent fault occurrence
  - Fault Tolerance
    - By redundancy provide service even in the presence of faults
  - Error Removal
    - By verification minimize the presence of latent errors
  - Error Forecasting
    - By evaluation estimate the presence of creation and consequence of errors

Classification

- Fault avoidance and fault tolerance are dependability procurement
- Fault removal and error forecasting are dependability validation
- Service states and transitions
  - Accomplished: service is as specified
  - Interrupted: service is different
  - Failure and restoration are the events which constitute the transition between these two states

Dependability Measures

- Reliability: A measure of continuous service
- Trustability: Indication of reliable/unreliable operation
- Availability: A measure of service accomplishment with respect to alteration
- Safety: Fail-safe operation, either correct or incorrect
- Performability: System performance at some level
- Maintainability: Restore system operation within time t
- Testability: Capability to verify system operation

Fault Avoidance

- Conservative design practices
- High reliability components use
- Careful signal routing
- Well tested and simulated design
- Proven design methods
- Adhere to design methodology
- Proper shielding
- Fault avoidance is not cheap!!
- When fault occurs, system may fail!!

Can we meet dependability goals?

- Fully? Probably not!
- All activities are related to human beings
  - Susceptible to errors
  - Fault occurs => need for removal
  - Error removal imperfect => Need for forecasting
- Building the system right => Verification
- Building the right system => Validation

Fault Tolerant System Design Issues

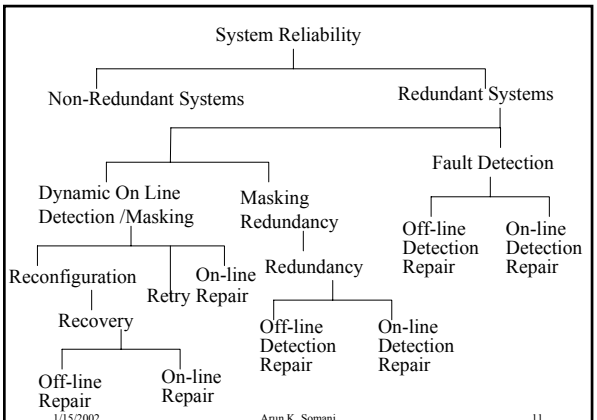
- How to achieve high dependability in
  - Long-life Applications: Space Craft
  - Critical-computation Applications: Aircraft, Weapon
  - Maintenance-postponement Applications: ESS
  - High-availability Application: Transaction Processing
- A system can be reliable without being fault tolerant
- A system can be fault tolerant without being reliable
- Probability of catastrophic failures must be minimized

Steps in Fault Tolerance

- Fault Confinement: Limit the scope of fault
  - Contamination should not occur
  - Consistency check, mutual suspicion
- Fault Detection: Detect presence of fault ASAP
  - Fault latency: time between occurrence and detection
  - On-line is expensive, Off-line needs interruption
- Fault Masking: Hide the effect of a defect
  - Usually automatic, TMR, NMR, Check before output
- Retry: Perform computation again
  - transient fault's effects can be removed

Steps in Fault Tolerance (Contd.)

- Fault Diagnosis: Locate the exact faulty component
  - Fault location may not be obvious
  - Requires additional testing
- Reconfiguration: Replace the faulty component
- Recovery: Remove the effect of fault, roll back
- Restart: If too much damage, restart the system
- Repair: Repair the faulty components
  - System may operate in degraded state until repair done
- Reintegration: replaced repaired module



Overhead

- Fault Detection
  - Up to 100%
- Dynamic Redundancy
  - Between 100% to 200%
- Masking Redundancy
  - 200% or more
- Multiple failure tolerance
  - More than 200%