Lecture 8
Dependability


Tutorial Questions
• Sommerville (6th Edition) Chapter 16 & 17
  – Question 16.2
  – Question 16.3
  – Question 16.6
  – Question 16.7
  – Question 17.1
  – Question 17.2
  – Question 17.6
  – Question 17.7

The concept of dependability
• For critical systems, it is usually the case that the most important system property is the dependability of the system.
• The dependability of a system reflects the user’s degree of trust in that system. It reflects the extent of the user’s confidence that it will operate as users expect and that it will not ‘fail’ in normal use.
• Usefulness and trustworthiness are not the same thing. A system does not have to be trusted to be useful.

Dimensions of dependability

<table>
<thead>
<tr>
<th>Dependability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
</tr>
<tr>
<td>Reliability</td>
</tr>
<tr>
<td>Safety</td>
</tr>
<tr>
<td>Security</td>
</tr>
</tbody>
</table>

- The ability of the system to deliver services when requested.
- The ability of the system to deliver services as specified.
- The ability of the system to operate without catastrophic failure.
- The ability of the system to protect itself against accidental or deliberate intrusions.

- The extent to which a critical system is trusted by its users.

- Usefulness and trustworthiness are not the same thing. A system does not have to be trusted to be useful.

- For critical systems, it is usually the case that the most important system property is the dependability of the system.
- The dependability of a system reflects the user’s degree of trust in that system. It reflects the extent of the user’s confidence that it will operate as users expect and that it will not ‘fail’ in normal use.
Maintainability
• A system attribute which is concerned with the ease of repairing the system after a failure has been discovered or changing the system to include new features
• Very important for critical systems as faults are often introduced into a system because of maintenance problems
• Maintainability is distinct from other dimensions of dependability because it is a static and not a dynamic system attribute. It is not covered in this course.

Survivability
• The ability of a system to continue to deliver its services to users in the face of deliberate or accidental attack
• This is an increasingly important attribute for distributed systems whose security can be compromised
• Survivability subsumes the notion of resilience - the ability of a system to continue in operation in spite of component failures

Costs of increasing dependability

Dependability costs
• Dependability costs tend to increase exponentially as increasing levels of dependability are required
• There are two reasons for this
  – The use of more expensive development techniques and hardware that are required to achieve the higher levels of dependability
  – The increased testing and system validation that is required to convince the system client that the required levels of dependability have been achieved

Dependability vs performance
• Untrustworthy systems may be rejected by their users
• System failure costs may be very high
• It is very difficult to tune systems to make them more dependable
• It may be possible to compensate for poor performance
• Untrustworthy systems may cause loss of valuable information

Dependability economics
• Because of very high costs of dependability achievement, it may be more cost effective to accept untrustworthy systems and pay for failure costs
• However, this depends on social and political factors. A reputation for products that can’t be trusted may lose future business
• Depends on system type - for business systems in particular, modest levels of dependability may be adequate
Availability and reliability

- **Reliability**
  - The probability of failure-free system operation over a specified time in a given environment for a given purpose
- **Availability**
  - The probability that a system, at a point in time, will be operational and able to deliver the requested services
- Both of these attributes can be expressed quantitatively

Availability and reliability

- It is sometimes possible to subsume system availability under system reliability
  - Obviously, if a system is unavailable it is not delivering the specified system services
- However, it is possible to have systems with low reliability that must be available. So long as system failures can be repaired quickly and do not damage data, low reliability may not be a problem
- Availability takes repair time into account

Reliability terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System failure</td>
<td>An event that occurs at some point in time when the system does not deliver a service as expected by its users</td>
</tr>
<tr>
<td>System error</td>
<td>Error system behaviour where the behaviour of the system does not conform to its specification.</td>
</tr>
<tr>
<td>System fault</td>
<td>An incorrect system state i.e. a system state that is unexpected by the designers of the system</td>
</tr>
<tr>
<td>Human error or mistake</td>
<td>Human behaviour that results in the introduction of faults into a system.</td>
</tr>
</tbody>
</table>

Faults and failures

- **Failures** are usually a result of system errors that are derived from faults in the system
- However, faults do not necessarily result in system failures
  - The error can be corrected by built-in error detection and recovery
  - The failure can be protected against by built-in protection facilities. These may, for example, protect system resources from system errors

Perceptions of reliability

- The formal definition of reliability does not always reflect the user's perception of a system's reliability
  - The assumptions that are made about the environment where a system will be used may be incorrect
    - Usage of a system in an office environment is likely to be quite different from usage of the same system in a university environment
  - The consequences of system failures affects the perception of reliability
    - Unreliable windscreen wipers in a car may be irrelevant in a dry climate
    - Failures that have serious consequences (such as an engine breakdown in a car) are given greater weight by users than failures that are inconvenient

Reliability achievement

- **Fault avoidance**
  - Development techniques are used that either minimise the possibility of mistakes or trap mistakes before they result in the introduction of system faults
- **Fault detection and removal**
  - Verification and validation techniques that increase the probability of detecting and correcting errors before the system goes into service are used
- **Fault tolerance**
  - Run-time techniques are used to ensure that system faults do not result in system errors and that system errors do not lead to system failures
Reliability modelling

- You can model a system as an input-output mapping where some inputs will result in erroneous outputs.
- The reliability of the system is the probability that a particular input will lie in the set of inputs that cause erroneous outputs.
- Different people will use the system in different ways so this probability is not a static system attribute but depends on the system’s environment.

Input/output mapping

Reliability perception

- Possible inputs
- Erroeneous inputs
- User 1
- User 2
- User 3

Reliability improvement

- Removing X% of the faults in a system will not necessarily improve the reliability by X%. A study at IBM showed that removing 60% of product defects resulted in a 3% improvement in reliability.
- Program defects may be in rarely executed sections of the code so may never be encountered by users. Removing these does not affect the perceived reliability.
- A program with known faults may therefore still be seen as reliable by its users.

Safety

- Safety is a property of a system that reflects the system’s ability to operate, normally or abnormally, without danger of causing human injury or death and without damage to the system’s environment.
- It is increasingly important to consider software safety as more and more devices incorporate software-based control systems.
- Safety requirements are exclusive requirements i.e. they exclude undesirable situations rather than specify required system services.

Safety criticality

- Primary safety-critical systems
  - Embedded software systems whose failure can cause the associated hardware to fail and directly threaten people.
- Secondary safety-critical systems
  - Systems whose failure results in faults in other systems which can threaten people.
- Discussion here focuses on primary safety-critical systems
  - Secondary safety-critical systems can only be considered on a one-off basis.
Safety and reliability

- Safety and reliability are related but distinct
  - In general, reliability and availability are necessary but not sufficient conditions for system safety
- Reliability is concerned with conformance to a given specification and delivery of service
- Safety is concerned with ensuring system cannot cause damage irrespective of whether or not it conforms to its specification

Unsafe reliable systems

- Specification errors
  - If the system specification is incorrect then the system can behave as specified but still cause an accident
- Hardware failures generating spurious inputs
  - Hard to anticipate in the specification
- Context-sensitive commands i.e. issuing the right command at the wrong time
  - Often the result of operator error

Safety terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accident (or mishap)</td>
<td>An unplanned event or sequence of events which results in human death or injury, damage to property or to the environment. A computer controlled machine failing its primary mission is an example of an accident.</td>
</tr>
<tr>
<td>Hazard</td>
<td>A condition existing or contributing to an accident. A failure of the system which leads to an accident is also a hazard.</td>
</tr>
<tr>
<td>Damage</td>
<td>A measure of the loss resulting from a mishap. Damage can range from many people killed as a result of an accident to minor injury or property damage.</td>
</tr>
<tr>
<td>Hazard severity</td>
<td>An assessment of the worst possible damage which could result from a particular hazard. Hazard severity can range from catastrophic where many people are killed to minor where only minor damage results.</td>
</tr>
<tr>
<td>Hazard probability</td>
<td>The probability of the events occurring which create a hazard. Probability values tend to be arbitrary but range from probable (say 1/10 chance of a hazard occurring) to improbable (no conceivable situations are likely where the hazard could occur).</td>
</tr>
<tr>
<td>Risk</td>
<td>This is a measure of the probability that the system will cause an accident. The risk is assessed by considering the hazard probability, the hazard severity and the probability that a hazard will result in an accident.</td>
</tr>
</tbody>
</table>

Safety achievement

- Hazard avoidance
  - The system is designed so that some classes of hazard simply cannot arise.
- Hazard detection and removal
  - The system is designed so that hazards are detected and removed before they result in an accident
- Damage limitation
  - The system includes protection features that minimise the damage that may result from an accident

Normal accidents

- Accidents in complex systems rarely have a single cause as these systems are designed to be resilient to a single point of failure
  - Designing systems so that a single point of failure does not cause an accident is a fundamental principle of safe systems design
- Almost all accidents are a result of combinations of malfunctions
- It is probably the case that anticipating all problem combinations, especially, in software controlled systems is impossible so achieving complete safety is impossible

Security

- The security of a system is a system property that reflects the system’s ability to protect itself from accidental or deliberate external attack
- Security is becoming increasingly important as systems are networked so that external access to the system through the Internet is possible
- Security is an essential pre-requisite for availability, reliability and safety
Fundamental security

- If a system is a networked system and is insecure then statements about its reliability and its safety are unreliable
- These statements depend on the executing system and the developed system being the same. However, intrusion can change the executing system and/or its data
- Therefore, the reliability and safety assurance is no longer valid

Security terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exposure</td>
<td>Possible loss or harm in a computing system</td>
</tr>
<tr>
<td>Vulnerability</td>
<td>A weakness in a computer-based system that may be exploited to cause loss or harm</td>
</tr>
<tr>
<td>Attack</td>
<td>An exploitation of a system vulnerability</td>
</tr>
<tr>
<td>Threats</td>
<td>Circumstances that have potential to cause loss or harm</td>
</tr>
<tr>
<td>Control</td>
<td>A protective measure that reduces a system vulnerability</td>
</tr>
</tbody>
</table>

Damage from insecurity

- Denial of service
  - The system is forced into a state where normal services are unavailable or where service provision is significantly degraded
- Corruption of programs or data
  - The programs or data in the system may be modified in an unauthorised way
- Disclosure of confidential information
  - Information that is managed by the system may be exposed to people who are not authorised to read or use that information

Security assurance

- Vulnerability avoidance
  - The system is designed so that vulnerabilities do not occur. For example, if there is no external network connection then external attack is impossible
- Attack detection and elimination
  - The system is designed so that attacks on vulnerabilities are detected and neutralised before they result in an exposure. For example, virus checkers find and remove viruses before they infect a system
- Exposure limitation
  - The system is designed so that the adverse consequences of a successful attack are minimised. For example, a backup policy allows damaged information to be restored

Key points

- The dependability in a system reflects the user’s trust in that system
- The availability of a system is the probability that it will be available to deliver services when requested
- The reliability of a system is the probability that system services will be delivered as specified
- Reliability and availability are generally seen as necessary but not sufficient conditions for safety and security

Key points

- Reliability is related to the probability of an error occurring in operational use. A system with known faults may be reliable
- Safety is a system attribute that reflects the system’s ability to operate without threatening people or the environment
- Security is a system attribute that reflects the system’s ability to protect itself from external attack
Critical Systems Specification

Dependable Systems Specification
• Processes and techniques for developing a specification for system availability, reliability, safety and security

Availability and Reliability

Functional and non-functional requirements
• System functional requirements may be generated to define error checking and recovery facilities and features that provide protection against system failures.
• Non-functional requirements may be generated to specify the required reliability and availability of the system.

System reliability specification
• **Hardware reliability**
  – What is the probability of a hardware component failing and how long does it take to repair that component?
• **Software reliability**
  – How likely is it that a software component will produce an incorrect output. Software failures are different from hardware failures in that software does not wear out. It can continue in operation even after an incorrect result has been produced.
• **Operator reliability**
  – How likely is it that the operator of a system will make an error?

System reliability engineering
• Sub-discipline of systems engineering that is concerned with making judgements on system reliability
• It takes into account the probabilities of failure of different components in the system and their combinations
  – Consider a system with 2 components A and B where the probability of failure of A is $P(A)$ and the probability of failure of B is $P(B)$. 

Failure probabilities

- If there are 2 components and the operation of the system depends on both of them then the probability of system failure is
  \[ P(S) = P(A) + P(B) \]
- Therefore, as the number of components increase then the probability of system failure increases
- If components are replicated then the probability of failure is
  \[ P(S) = P(A)^n \text{ (all components must fail)} \]

Examples of Functional reliability requirements

- A predefined range for all values that are input by the operator shall be defined and the system shall check that all operator inputs fall within this predefined range.
- The system shall check all disks for bad blocks when it is initialised.
- The system must use N-version programming to implement the braking control system.
- The system must be implemented in a safe subset of Ada and checked using static analysis.

Non-functional reliability specification

- The required level of system reliability required should be expressed in quantitatively.
- Reliability is a dynamic system attribute - reliability specifications related to the source code are meaningless.
  - No more than N faults/1000 lines.
  - This is only useful for a post-delivery process analysis where you are trying to assess how good your development techniques are.
- An appropriate reliability metric should be chosen to specify the overall system reliability.

Reliability metrics

- Reliability metrics are units of measurement of system reliability.
- System reliability is measured by counting the number of operational failures and, where appropriate, relating these to the demands made on the system and the time that the system has been operational.
- A long-term measurement programme is required to assess the reliability of critical systems.

Reliability metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>POFOD</td>
<td>Probability of failure on demand</td>
</tr>
<tr>
<td>ROCOF</td>
<td>Rate of failure occurrence</td>
</tr>
<tr>
<td>MTTF</td>
<td>Mean time to failure</td>
</tr>
<tr>
<td>MTTR</td>
<td>Mean time to repair</td>
</tr>
<tr>
<td>AVAIL</td>
<td>Availability</td>
</tr>
</tbody>
</table>

Availability

- Measure of the fraction of the time that the system is available for use.
- Takes repair and restart time into account.
- Availability of 0.998 means software is available for 998 out of 1000 time units.
- Relevant for non-stop, continuously running systems.
  - Telephone switching systems, railway signalling systems.
Probability of failure on demand
- This is the probability that the system will fail when a service request is made. Useful when demands for service are intermittent and relatively infrequent.
- Appropriate for protection systems where services are demanded occasionally and where there are serious consequences if the service is not delivered.
- Relevant for many safety-critical systems with exception management components.
  - Emergency shutdown system in a chemical plant.

Rate of fault occurrence (ROCOF)
- Reflects the rate of occurrence of failure in the system.
- ROCOF of 0.002 means 2 failures are likely in each 1000 operational time units, e.g., 2 failures per 1000 hours of operation.
- Relevant for operating systems, transaction processing systems where the system has to process a large number of similar requests that are relatively frequent.
  - Credit card processing system, airline booking system.

Mean time to failure
- Measure of the time between observed failures of the system. Is the reciprocal of ROCOF for stable systems.
- MTTF of 500 means that the mean time between failures is 500 time units.
- Relevant for systems with long transactions, i.e., where system processing takes a long time. MTTF should be longer than transaction length.
  - Computer-aided design systems where a designer will work on a design for several hours, word processor systems.

Failure consequences
- Reliability measurements do NOT take the consequences of failure into account.
- Transient faults may have no real consequences but other faults may cause data loss or corruption and loss of system service.
- May be necessary to identify different failure classes and use different metrics for each of these. The reliability specification must be structured.

Failure classification
<table>
<thead>
<tr>
<th>Failure class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transient</td>
<td>Occurs only with certain inputs</td>
</tr>
<tr>
<td>Permanent</td>
<td>Occurs with all inputs</td>
</tr>
<tr>
<td>Recoverable</td>
<td>System can recover without operator intervention</td>
</tr>
<tr>
<td>Unrecoverable</td>
<td>Operator intervention needed to recover from failure</td>
</tr>
<tr>
<td>Non-corrupting</td>
<td>Failure does not corrupt system state or data</td>
</tr>
<tr>
<td>Corrupting</td>
<td>Failure corrupts system state or data</td>
</tr>
</tbody>
</table>
1. For each sub-system, analyse the consequences of possible system failures.
2. From the system failure analysis, partition failures into appropriate classes.
3. For each failure class identified, set out the reliability using an appropriate metric. Different metrics may be used for different reliability requirements.
4. Identify functional reliability requirements to reduce the chances of critical failures.

**Bank auto-teller system**
- Each machine in a network is used 300 times a day
- Bank has 1000 machines
- Lifetime of software release is 2 years
- Each machine handles about 200,000 transactions
- About 300,000 database transactions in total per day

**Examples of a reliability spec.**
<table>
<thead>
<tr>
<th>Failure class</th>
<th>Example</th>
<th>Reliability metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permanent, non-corrupting</td>
<td>The system fails to operate with any card which is input. Software must be restarted to correct failure.</td>
<td>ROCOF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 occurrence/1000 days</td>
</tr>
<tr>
<td>Transient, non-corrupting</td>
<td>The magnetic stripe data cannot be read on an undamaged card which is input.</td>
<td>POFOD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 in 1000 transactions</td>
</tr>
<tr>
<td>Transient, corrupting</td>
<td>A pattern of transactions across the network causes database corruption.</td>
<td>Unquantifiable! Should never happen in the lifetime of the system</td>
</tr>
</tbody>
</table>

**Specification validation**
- It is impossible to empirically validate very high reliability specifications.
- No database corruptions means POFOD of less than 1 in 200 million.
- If a transaction takes 1 second, then simulating one day's transactions takes 3.5 days.
- It would take longer than the system's lifetime to test it for reliability.

**Key points**
- There are both functional and non-functional dependability requirements.
- Non-functional availability and reliability requirements should be specified quantitatively.
- Metrics that may be used are AVAIL, POFOD, ROCOF and MTTF.
- When deriving a reliability specification, the consequences of different types of fault should be taken into account.

**Safety**
Safety specification

- The safety requirements of a system should be separately specified
- These requirements should be based on an analysis of the possible hazards and risks
- Safety requirements usually apply to the system as a whole rather than to individual sub-systems. In systems engineering terms, the safety of a system is an emergent property.

Safety processes

1. Hazard and risk analysis
   - Assess the hazards and the risks of damage associated with the system
2. Safety requirements specification
   - Specify a set of safety requirements which apply to the system
3. Designation of safety-critical systems
   - Identify the sub-systems whose incorrect operation may compromise system safety. Ideally, these should be as small a part as possible of the whole system.
4. Safety validation
   - Check the overall system safety

Hazard and risk analysis

- Identification of hazards which can arise which compromise the safety of the system and assessing the risks associated with these hazards
- Structured into various classes of hazard analysis and carried out throughout software process from specification to implementation
- A risk analysis should be carried out and documented for each identified hazard and actions taken to ensure the most serious/likely hazards do not result in accidents.

Hazard and risk analysis stages

- Hazard identification
  - Identify potential hazards which may arise
- Risk analysis and hazard classification
  - Assess the risk associated with each hazard
- Hazard decomposition
  - Decompose hazards to discover their potential root causes
- Risk reduction assessment
  - Define how each hazard must be taken into account when the system is designed.
Fault-tree analysis

• Method of hazard analysis which starts with an identified fault and works backward to the causes of the fault.
• Can be used at all stages of hazard analysis from preliminary analysis through to detailed software checking
• Top-down hazard analysis method. May be combined with bottom-up methods which start with system failures and lead to hazards

Fault-tree analysis

1. Identify hazard
2. Identify potential causes of the hazard. Usually there will be a number of alternative causes. Link these on the fault-tree with ‘or’ or ‘and’ symbols
3. Continue process until root causes are identified
4. Consider the following example which considers how data might be lost in some system where a backup process is running

Fault tree

Data deleted

External attack

H/W failure

S/W failure

External attack

H/W failure

S/W failure

External attack

H/W failure

S/W failure

Operator failure

Operating system failure

Backup system failure

Incorrect configuration

Incorrect operator input

Execution failure

Timing fault

Algorithm fault

Data fault

UI design fault

Training fault

Human error

Risk assessment

• Assesses hazard severity, hazard probability and accident probability
• Outcome of risk assessment is a statement of acceptability
  – Intolerable. Must never arise or result in an accident
  – As low as reasonably practical (ALARP). Must minimise possibility of hazard given cost and schedule constraints
  – Acceptable. Consequences of hazard are acceptable and no extra costs should be incurred to reduce hazard probability

Risk acceptability

• The acceptability of a risk is determined by human, social and political considerations
• In most societies, the boundaries between the regions are pushed upwards with time i.e. society is less willing to accept risk
  – For example, the costs of cleaning up pollution may be less than the costs of preventing it but this may not be socially acceptable
• Risk assessment is subjective
  – Risks are identified as probable, unlikely, etc. This depends on who is making the assessment
Risk reduction

- System should be specified so that hazards do not arise or result in an accident
- Hazard avoidance
  - The system should be designed so that the hazard can never arise during correct system operation
- Hazard detection and removal
  - The system should be designed so that hazards are detected and neutralised before they result in an accident
- Damage limitation
  - The system is designed in such a way that the consequences of an accident are minimised

Specifying forbidden behaviour

- The system shall not allow users to modify access permissions on any files that they have not created (security)
- The system shall not allow reverse thrust mode to be selected when the aircraft is in flight (safety)
- The system shall not allow the simultaneous activation of more than three alarm signals (safety)

Security specification

- Has some similarities to safety specification
  - Not possible to specify security requirements quantitatively
  - The requirements are often ‘shall not’ rather than ‘shall’ requirements
- Differences
  - No well-defined notion of a security life cycle for security management
  - Generic threats rather than system specific hazards
  - Mature security technology (encryption, etc.). However, there are problems in transferring this into general use

The security specification process

1. Asset identification and evaluation
   - The assets (data and programs) and their required degree of protection are identified. The degree of required protection depends on the asset value so that a password file (say) is more valuable than a set of public web pages.
2. Threat analysis and risk assessment
   - Possible security threats are identified and the risks associated with each of these threats is estimated.
3. Threat assignment
   - Identified threats are related to the assets so that, for each identified asset, there is a list of associated threats.
Stages in security specification

4. **Technology analysis**
   - Available security technologies and their applicability against the identified threats are assessed.

5. **Security requirements specification**
   - The security requirements are specified. Where appropriate, these will explicitly identify the security technologies that may be used to protect against different threats to the system.

Key points

- Hazard analysis is a key activity in the safety specification process.
- Fault-tree analysis is a technique which can be used in the hazard analysis process.
- Risk analysis is the process of assessing the likelihood that a hazard will result in an accident. Risk analysis identifies critical hazards and classifies risks according to their seriousness.
- To specify security requirements, you should identify the assets that are to be protected and define how security techniques should be used to protect them.