Dependability

Objectives

- To explain what is meant by a critical system where system failure can have severe human or economic consequence.
- To explain four dimensions of dependability - availability, reliability, safety and security.
- To explain that, to achieve dependability, you need to avoid mistakes, detect and remove errors and limit damage caused by failure.

Topics covered

- System dependability
- Availability and reliability
- Safety
- Security

Critical Systems

- Safety-critical systems
  - Failure results in loss of life, injury or damage to the environment;
  - Chemical plant protection system;
- Mission-critical systems
  - Failure results in failure of some goal-directed activity;
  - Spacecraft navigation system;
- Business-critical systems
  - Failure results in high economic losses;
  - Customer accounting system in a bank;

System dependability

- For critical systems, it is usually the case that the most important system property is the dependability of the system.
  - Dependability is a non-functional requirement.
- 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.

Importance of dependability

- Systems that are not dependable and are unreliable, unsafe or insecure may be rejected by their users.
- The costs of system failure may be very high.
- Undependable systems may cause information loss with a high consequent recovery cost.

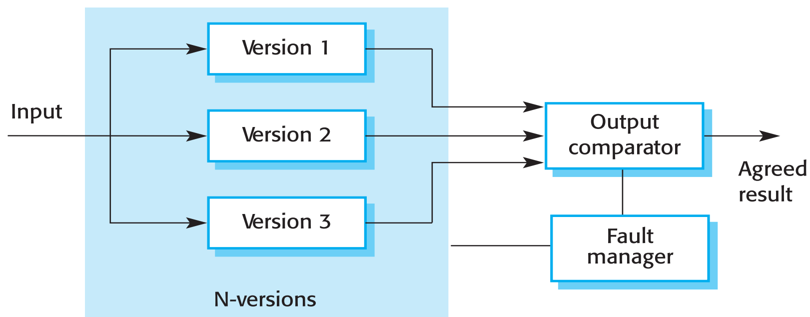
Achieving dependability

- Redundancy
  - Keep more than 1 version of a critical component available so that if one fails then a backup is available.
- Diversity
  - Provide the same functionality in different ways so that they will not fail in the same way.
- But adding diversity and redundancy adds complexity and this can increase the chances of error.
- Some engineers advocate simplicity and extensive V & V as a more effective route to software dependability.

Diversity and redundancy examples

- Redundancy. Where availability is critical (e.g. in e-commerce systems), companies normally keep backup servers and switch to these automatically if failure occurs.
- Diversity. To provide resilience against external attacks, different servers may be implemented using different operating systems (e.g. Windows and Linux)

N-version programming as a model



What's wrong with this model?

N-version programming

- The different system versions are designed and implemented by different teams. It is assumed that there is a low probability that they will make the same mistakes. The algorithms used should but may not be different.
- There is some empirical evidence that teams commonly misinterpret specifications in the same way and chose the same algorithms in their systems.

Fault-free software

- Current methods of software engineering now allow for the production of fault-free software, but for relatively small systems.
- Fault-free software means software which conforms to its specification. It does NOT mean software which will always perform correctly as there may be specification errors.
- The cost of producing fault free software is very high. It is only cost-effective in exceptional situations. It is often cheaper to accept software faults and pay for their consequences than to expend resources on developing fault-free software.

Development methods for critical systems

- The costs of critical system failure are so high that development methods may be used that are not cost-effective for other types of system.
- Examples of development methods
  - Formal methods of software development
  - Static analysis
  - External quality assurance
  - Independent verification and validation (IV&V)

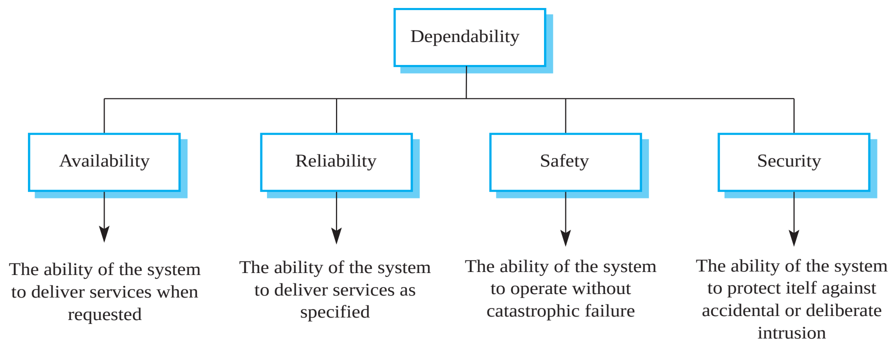
Socio-technical critical systems

- Hardware failure
  - Hardware fails because of design and manufacturing errors or because components have reached the end of their natural life.
- Software failure
  - Software fails due to errors in its specification, design or implementation.
- Operational failure
  - Human operators make mistakes. Now perhaps the largest single cause of system failures.

Dependability

- The dependability of a system equates to its trustworthiness.
- A dependable system is a system that is trusted by its users.
- Principal dimensions of dependability are:
  - Availability;
  - Reliability;
  - Safety;
  - Security
  - Others …

Dimensions of dependability



Other dependability properties

- Repairability
  - Reflects the extent to which the system can be repaired in the event of a failure
- Maintainability
  - Reflects the extent to which the system can be adapted to new requirements;
- Survivability
  - Reflects the extent to which the system can deliver services whilst under hostile attack;
- Error tolerance
  - Reflects the extent to which user input errors can be avoided and tolerated.

Maintainability

- A system attribute that 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.

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

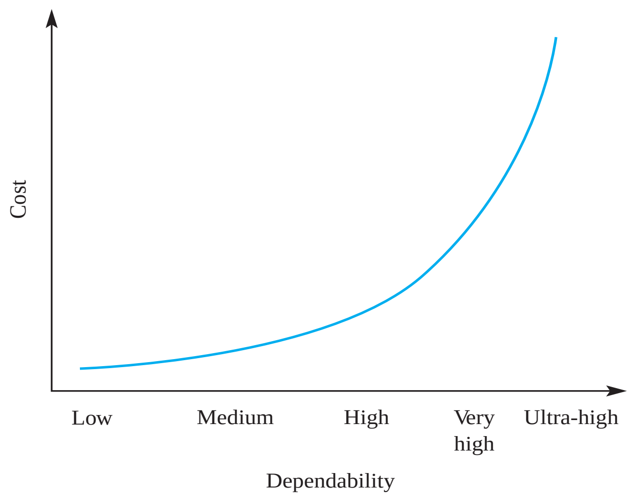
Dependability vs performance

- 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 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

Costs of increasing dependability



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
  - Reliability not the same as correctness
- 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

	Term	Description
Observation	System failure	An event that occurs at some point in time when the system does not deliver a service as expected by its users
Event	System error	An erroneous system state that can lead to system behavior that is unexpected by system users.
Cause	System fault	A characteristic of a software system that can lead to a system error. For example, failure to initialize a variable could lead to that variable having the wrong value when it is used.
	Human error or mistake	Human behavior that results in the introduction of faults into a system.

Testing consists of observing failures.
Debugging consists of finding the event that causes the failure.
Finding the ultimate cause (i.e., the fault) of the error is extremely difficult.

Faults and failures

- Failures are a usually a result of system errors that are derived from faults in the system
- However, faults do not necessarily result in system errors
    - The faulty system state may be transient and 'corrected' before an error arises
- Errors do not necessarily lead to 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
  - ❑ The system is developed in such a way that human error is avoided and thus system faults are minimized.
  - ❑ The development process is organized so that faults in the system are detected and repaired before delivery to the customer.
- Fault detection
  - ❑ Verification and validation techniques are used to discover and remove faults in a system before it is deployed.
- Fault tolerance
  - ❑ The system is designed so that faults in the delivered software do not result in system failure.

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
  - Work at NASA/GSFC showed that in reliable programs, only 70% of the code was ever tested
- 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

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

Term	Definition
Accident (or mishap)	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 injuring its operator is an example of an accident.
Hazard	A condition with the potential for causing or contributing to an accident. A failure of the sensor that detects an obstacle in front of a machine is an example of a hazard.
Damage	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.
Hazard severity	An assessment of the worst possible damage that could result from a particular hazard. Hazard severity can range from catastrophic where many people are killed to minor where only minor damage results.
Hazard probability	The probability of the events occurring which create a hazard. Probability values tend to be arbitrary but range from *probable* (say 1/100 chance of a hazard occurring) to implausible (no conceivable situations are likely where the hazard could occur).
Risk	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.

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
  - (Hardware) engineers have learned this well – 4 ways to stop a car, 3 sets of control lines to wings of an airplane.
  - Software engineers still have not learned it.
- Almost all accidents are a result of combinations of malfunctions (e.g., fault tree analysis to find multiple points of failures.)
- It is probably the case that anticipating all problem combinations, especially, in software controlled systems is impossible so achieving complete safety is impossible – but we certainly can do better!

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

The CIA of security

- Confidentiality – Only authorized individuals have access to data
- Integrity – Data cannot be altered without the knowledge of those who own it
- Availability – Data is available when needed

Security terminology

Term	Definition
Exposure	Possible loss or harm in a computing system. This can be loss or damage to data or can be a loss of time and effort if recovery is necessary after a security breach.
Vulnerability	A weakness in a computer-based system that may be exploited to cause loss or harm.
Attack	An exploitation of a system vulnerability. Generally, this is from outside the system and is a deliberate attempt to cause some damage.
Threats	Circumstances that have potential to cause loss or harm. You can think of these as a sys tem vulnerability that is subjected to an attack.
Control	A protective measure that reduces a system vulnerability. Encryption would be an example of a control that reduced a vulnerability of a weak access control system.

What causes security failure?

- Need all 3 "AAA" attributes for failure:
  - Architecture – software faults (e.g., buffer overflows), bad design (e.g., executable scripts in email)
  - Administration – poor use of features (e.g., use "default" as home wireless password, "root" as root password)
  - Attacks – Active intrusions to compromise a system.
- No intrusions, then secure. Better administration minimizes risk. Better architecture limits impact of attacks.

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 unauthorized way
- Disclosure of confidential information
  - Information that is managed by the system may be exposed to people who are not authorized 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 neutralized 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 minimized. For example, a backup policy allows damaged information to be restored

Key points

- A critical system is a system where failure can lead to high economic loss, physical damage or threats to life.
- 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
- Dependability improvement requires a socio-technical approach to design where you consider the humans as well as the hardware and software