Chapter 12 – Safety Engineering

Topics covered

- ✧ Safety-critical systems
- ✧ Safety requirements
- ✧ Safety engineering processes
- ✧ Safety cases

Safety

- ✧ A property of a system
- ✧ The system's ability to operate services
  - Prevent danger causing human injury or death
  - Avoiding damage to the system's environment.
- ✧ Software safety issues become important
  - Most devices incorporate software-based control systems.
  - Control real-time, safety-critical processes.

Software in safety-critical systems

- ✧ Software-controlled systems
  - Decisions are made by the software.
  - Subsequent actions are safety-critical.
  - Software behaviour is related to safety of the system.
- ✧ Checking and monitoring safety-critical components
  - E.g., monitoring aircraft engine components for fault detection.
- ✧ Monitoring software is safety-critical
  - Other components may fail due to the failure of fault detection.

Safety and reliability

- ✧ Safety and reliability
  - Reliability and availability are not sufficient for system safety
- ✧ Reliability
  - Conformance to a given specification and delivery of service
- ✧ Safety
  - Ensuring system cannot cause damage.
- ✧ System reliability is essential for safety
  - However, reliable systems can be unsafe

Unsafe reliable systems

- ✧ Dormant system faults
  - Undetected for a number of years and only rarely arise.
- ✧ Specification errors
  - Software system behaves as specified but cause an accident.
- ✧ Hardware failures at runtime
  - E.g., generating spurious inputs
  - Hard to anticipate in the specification
- ✧ Context-sensitive commands
  - E.g., a system command is executed at the wrong time.

Safety-critical systems

Safety critical systems

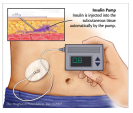
- ✧ Essential that system operation is always safe
  - Must not cause damage to people or the system's environment
- ✧ Examples
  - Process control systems in chemical manufacture
  - Automobile control systems such as braking management systems

Safety criticality

- ✧ Primary safety-critical systems
  - Embedded software systems
  - Cause associated hardware failures, directly threatening people.
  - E.g., the insulin pump control system.
- ✧ Secondary safety-critical systems
  - Result in faults in other connected systems, affecting safety consequences.
  - E.g., the Mentcare system producing inappropriate treatment being prescribed.
  - Infrastructure control systems.

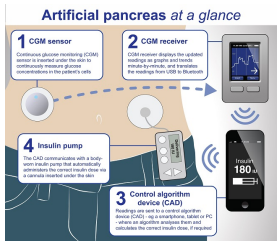
Insulin pump control system

- ✧ Collects data from a blood sugar sensor and calculates the amount of insulin required to be injected.
- ✧ Calculation based on the rate of change of blood sugar levels.



Insulin pump control system (cont.)

- ✧ Sends signals to a micro-pump to deliver the correct dose of insulin.
- ✧ Safety-critical system as low blood sugars can lead to brain malfunctioning, coma and death; high-blood sugar levels have long-term consequences such as eye and kidney damage.



Safety criticality

- ✧ Primary safety-critical systems
  - Embedded software systems
  - Cause associated hardware failures, directly threatening people.
  - E.g., the insulin pump control system.
- ✧ Secondary safety-critical systems
  - Result in faults in other connected systems, affecting safety consequences.
  - E.g., the Mentcare system producing inappropriate treatment being prescribed.
  - Infrastructure control systems.

Hazards

- Situations or events that can lead to an accident
  - Incorrect computation by software in navigation system
  - Failure to detect possible disease in medication prescribing system
- Perform accident prevention actions
  - Hazards do not inevitably lead to accidents

Safety achievement

- Hazard avoidance
  - Appling hazard avoidance design to software systems.
  - Prevent some classes of hazard.
- Hazard detection and removal
  - Detecting and removing hazard before causing accidents.
- Damage limitation
  - Protection features to minimise the damage.

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. An overdose of insulin 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 measures blood glucose is an example of a hazard.
Damage	A measure of the loss resulting from a mishap. Damage can range from many people being killed as a result of an accident to minor injury or property damage. Damage resulting from an overdose of insulin could be serious injury or the death of the user of the insulin pump.
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. When an individual death is a possibility, a reasonable assessment of hazard severity is 'very high'.
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 in which the hazard could occur). The probability of a sensor failure in the insulin pump that results in an overdose is probably low.
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 the hazard will lead to an accident. The risk of an insulin overdose is probably medium to low.

Normal accidents

- Rarely have a single cause in complex systems
- Designed to be resilient to a single point of failure
- A fundamental principle of safe systems design
  - A single point of failure does not cause an accident.
- A result of combinations of malfunctions.
- Hard to anticipate all combinations in software systems
  - Difficult to achieve complete safety.
  - Accidents are inevitable.

Software safety benefits

- Software control systems contributes to system safety
  - A large number of conditions to be monitored and controlled.
  - Reducing human efforts and time in hazardous environments.
  - Detecting and repairing safety-critical operator errors.

Safety requirements

Functional and non-functional requirements

- Functional requirements
  - Statements of services the system should provide,
  - How the system should react to particular inputs and how the system should behave in particular situations.
- Non-functional requirements
  - Constraints on the services or functions of the system.
  - Apply to the whole system rather than individual features.

Functional requirements

- Describe functionality or system services.
  - Depending on the type of software systems and users.
- Functional user requirements
  - High-level statements of what the system should do.
- Functional system requirements
  - The system services in detail.

Safety specification

- Goal
  - Identifying protection requirements.
  - Preventing injury or death or environmental damage.
- Safety requirements
  - Shall Not requirements.
  - Define situations and events that should never occur.
- Functional safety requirements
  - Checking and recovery features in a system.
  - Protection feature against failures and external attacks.

Hazard-driven analysis

- Hazard identification
- Hazard assessment
- Hazard analysis
- Risk reduction
  - Safety requirements specification

Hazard identification

- Identify the hazards threatening the system.
- Different types of hazard:
  - Physical hazards
  - Electrical hazards
  - Biological hazards
  - Service failure hazards
  - Etc.

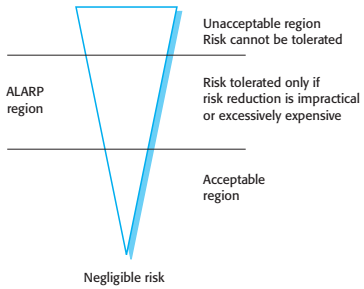
Insulin pump risks

- Insulin overdose (service failure).
- Insulin underdose (service failure).
- Power failure due to exhausted battery (electrical).
- Electrical interference with other medical equipment (electrical).
- Poor sensor and actuator contact (physical).
- Infection caused by introduction of machine (biological).
- Allergic reaction to materials or insulin (biological).

Hazard assessment

- Understanding the likelihood that a risk will arise and the potential consequences.
- Risks category:
- Intolerable.
  - Unsupportable.
- As low as reasonably practical (ALARP).
  - Minimising risk possibilities given available resources.
- Acceptable.
  - No extra costs to reduce hazard probability.

The risk triangle



Social acceptability of risk

- The acceptability of a risk.
  - Human, social and political considerations.
- Society is less willing to accept risk in most cases.
  - E.g., the costs of cleaning up or preventing pollution.
- Subjective assessment
  - Depending on evaluators making the assessment.

Hazard assessment

- The risk probability and the risk severity.
- Relative values: ‘unlikely’, ‘rare’, ‘very high’, etc.
  - Impossible to do precise measurement
- Goal:
  - Prevent or remove potential risks with the high severity.

Risk classification for the insulin pump

Identified hazard	Hazard probability	Accident severity	Estimated risk	Acceptability
1.Insulin overdose computation	Medium	High	High	Intolerable
2. Insulin underdose computation	Medium	Low	Low	Acceptable
3. Failure of hardware monitoring system	Medium	Medium	Low	ALARP
4. Power failure	High	Low	Low	Acceptable
5. Machine incorrectly fitted	High	High	High	Intolerable
6. Machine breaks in patient	Low	High	Medium	ALARP
7. Machine causes infection	Medium	Medium	Medium	ALARP
8. Electrical interference	Low	High	Medium	ALARP
9. Allergic reaction	Low	Low	Low	Acceptable

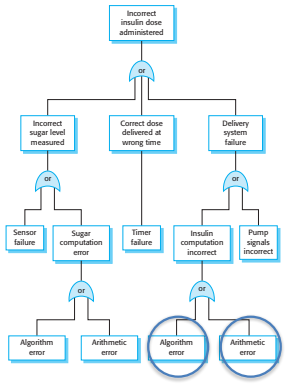
Hazard analysis

- The root causes of risks in a particular system.
- Hazard analysis techniques
  - Inductive, bottom-up techniques:
    Evaluate the hazards, starting with system failures.
  - Deductive, top-down techniques:
    Reason failure causes, starting with a hazard

Fault-tree analysis

- ✧ A deductive top-down technique.
- ✧ Hazard at the root of the tree
  - Identify states causing hazards.
- ✧ Linking conditions by relationships (e.g., ‘and’ or ‘or’)
- ✧ Goal
  - Minimizing the number of single failure causes.

An example of a software fault tree



Fault tree analysis

- ✧ Possible conditions of incorrect dose of insulin:
  - Incorrect measurement of blood sugar level
  - Failure of delivery system
  - Dose delivered at wrong time
- ✧ Root causes of these hazards:
  - Algorithm error
  - Arithmetic error

Risk reduction

- ✧ Goal:
  - Identify requirements for risk managements to avoid accidents.
- ✧ Risk reduction strategies
  - Hazard avoidance
  - Hazard detection and removal
  - Damage limitation

Strategy use

- ✧ Combining multiple risk reduction strategies
- ✧ E.g., a chemical plant control system:
  - Detecting and correcting excess pressure in the reactor.
  - Opening a relief valve as independent protection system

Insulin pump - software risks

- ✧ Arithmetic error
  - Data variable overflow or underflow during a computation.
  - Handing runtime exception.
- ✧ Algorithmic error
  - Comparison between previous and current values
  - Checking the maximum value to control dose.

Examples of safety requirements

SR1: The system shall not deliver a single dose of insulin that is greater than a specified maximum dose for a system user.

SR2: The system shall not deliver a daily cumulative dose of insulin that is greater than a specified maximum daily dose for a system user.

SR3: The system shall include a hardware diagnostic facility that shall be executed at least four times per hour.

SR4: The system shall include an exception handler for all of the exceptions that are identified in Table 3.

SR5: The audible alarm shall be sounded when any hardware or software anomaly is discovered and a diagnostic message, as defined in Table 4, shall be displayed.

SR6: In the event of an alarm, insulin delivery shall be suspended until the user has reset the system and cleared the alarm.

Safety engineering processes