Thakur Educational Trust's (Regd.)

THAKUR RAMNARAYAN COLLEGE OF ARTS & COMMERCE

ISO 21001:2018 Certified

PROGRAMME: B.Sc (I.T)

CLASS: S.Y.B.Sc (I.T)

SUBJECT NAME: SOFTWARE ENGINEERING

SEMESTER: IV

FACULTY NAME: Ms. SMRITI DUBEY

UNIT II

Chapter 2 – Critical System

Concepts:

Types of Critical Systems

Simple Safety Critical System

Dependability of Systems

Availability and Reliability

Safety and Security

Introduction to Critical System

Software failures are relatively common. In most cases these failures cause inconvenience but no serious long-term damage. However, in some systems failure can result in significant economic losses, physical damage or threats to human life. These systems are called critical system.

A critical system is a system that refers to the systems that are efficient and retain this efficiency as they change without prohibitive costs being incurred. Critical systems are technical or socio technical systems that people or business depend on. If these systems fail to deliver their services as expected then serious problems and significant losses may result.

Types of Critical System

There are three types of critical systems:

1) Safety Critical System: Any failure in these systems results in injury, death or damage to the environment. For example, chemical plant system.

2) Mission Critical Systems: Any failure in these systems result in the failure of some expected goals. For example, Spacecraft navigation system.

3) Business Critical System: Any failure in these systems result in high financial loss. For example, the Bank accounting system.

The most important emergent property of a critical system is its dependability. It covers the related system attributes of availability, reliability, safety & security.

– Systems that are unreliable, unsafe or insecure are often rejected by their users (refuse to the product from the same company).

– System failure costs may be very high. (reactor / aircraft navigation)

– Untrustworthy systems may cause information loss with a high consequent recovery cost.

The high costs of failure of critical systems means that trusted methods and techniques must be used for development. Consequently, critical systems are usually developed using well-tried techniques rather than newer techniques that have not been subject to extensive practical experience. Rather than embrace new techniques and methods, critical systems developers are naturally conservative. They prefer to use older techniques whose strengths and weaknesses are understood, rather than new techniques which may appear to be better but whose long-term problems are unknown.

Simple safety critical system

Safety-critical systems are those systems whose failure could result in loss of life, significant property damage or damage to the environment. There are many well-known examples in application areas such as medical devices, aircraft flight control, weapons and nuclear systems.

Example of simple safety critical system (An insulin pump control system)

Diabetes is a relatively common condition where the human pancreas is unable to produce sufficient quantities of a hormone called insulin. Insulin metabolizes glucose (sugar) in the blood. The conventional treatment of diabetes involves regular injections of genetically engineered insulin. Diabetics measure their blood sugar levels using an external meter and then calculate the dose of insulin that they should inject.

The problem with this treatment is that the level of insulin required does not just depend on the blood glucose level but also on the time of the last insulin injection. This can lead to very low levels of blood glucose (if there is too much insulin) or very high levels of blood sugar (if there is too little insulin). That causes the serious problem to the human body.

Current advances in developing miniaturized sensors have meant that it is now possible to develop automated insulin delivery systems. These systems monitor blood sugar levels and deliver an appropriate dose of insulin when required. A software-controlled insulin delivery system might work by using a micro-sensor embedded in the patient to measure some blood parameter that is proportional to the sugar level. This is then sent to the pump controller. This controller computes the sugar level and the amount of insulin that is needed. It then sends signals to a miniaturized pump to deliver the insulin via a permanently attached needle.

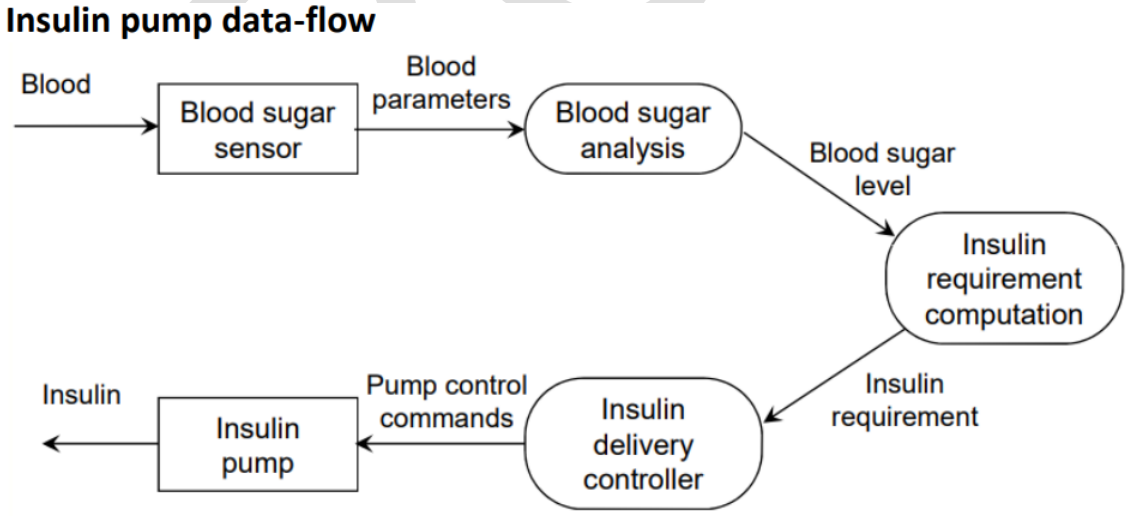


Figure shown above is a UML activity model that illustrates how the software transforms an input blood sugar level to a sequence of commands that drive the insulin pump.

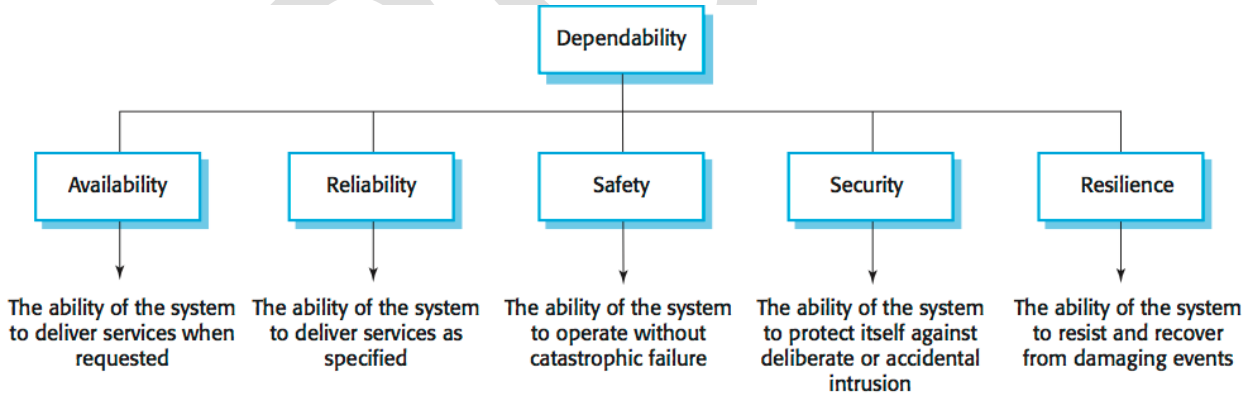
Clearly, this is a safety-critical system. If the pump fails to operate or does not operate correctly, then the user's health may be damaged or they may fall into a coma because their blood sugar levels are too high or too low. There are therefore two essentials high-level requirements that this system must meet:

1. The system shall be available to deliver insulin when required.
2. The system shall perform reliably and deliver the correct amount of insulin to counteract the current level of blood sugar.

Failure of the system could in principle cause excessive doses of insulin to be delivered and this could threaten the life of the user. It is particularly important that overdoses of insulin should not occur.

Dependability of systems

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. Dependability covers the related systems attributes of reliability, availability and security. These are all inter-dependent.



Dimensions of dependability

- **Availability**: The probability that the system will be up and running and able to deliver useful services to users.
- **Reliability**: The probability that the system will correctly deliver services as expected by users.
- **Safety**: A judgment of how likely it is that the system will cause damage to people or its environment.

- **Security**: A judgment of how likely it is that the system can resist accidental or deliberate intrusions.
- **Resilience**: A judgment of how well a system can maintain the continuity of its critical services in the presence of disruptive events such as equipment failure and cyberattacks.

Availability and Reliability

System availability and reliability are closely related properties that both can be expressed as numerical probabilities. **Reliability** is the probability of failure-free system operation over a specified time in a given environment for a given purpose. **Availability** is the probability that a system, at a point in time, will be operational and able to deliver the requested services.

We cannot say that reliable systems will always be available and vice versa. If users expect continuous service, then the availability requirements are high. If the consequences of a failure are minimal and the system can recover quickly from the failures then the same system can have low reliability requirements.

The following terminologies define the reliability issues occurred in system:

Term	Description
Human error or mistake	Human behaviour that results in the introduction of faults into a system.
System fault	A characteristic of a software system that can lead to a system error.
System error	An erroneous system state that can lead to system behaviour that is unexpected by system users.
System failure	An event that occurs at some point in time when the system does not deliver a service as expected by its users.

Complementary approaches to improve reliability:

Reliability is the probability of failure-free operation over a specified time in a given environment for a specific purpose. The definition of reliability states that the environment in which the system is used and the purpose that it is used for must be taken into account. If you measure system reliability in one environment you can't assume that the reliability will be same in another environment where the system is used in a different way.

Complementary approaches to improve reliability are as follows:

Fault avoidance

Development techniques are used that either minimize 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/or that system errors do not lead to system failures.

Safety and Security

Safety:

Safety critical systems are systems where it is essential that system operation is always safe. That is the system should never damage people or the system's environment even if the system fails. Examples of safety critical system are control and monitoring systems in aircraft, process control systems in chemical and pharmaceutical plants and automobile control systems.

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 critical software are 2 types:

Primary safety-critical systems

– Embedded software systems whose failure can cause hardware malfunction which results inhuman injury or environmental damage.

Secondary safety-critical systems

– Systems whose failure indirectly results in injury. Eg - Medical Database holding details of drugs.

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.

Ways to achieve Safety

• Hazard avoidance

– The system is designed so that hazard simply cannot arise.

– E.g., Press 2 buttons at the same time in a cutting machine to start

• Hazard detection and removal

– The system is designed so that hazards are detected and removed before they result in an accident.

– E.g., Open relief valve on detection over pressure in chemical plant.

• Damage limitation

– The system includes protection features that minimise the damage that may result from an accident

– Automatic fire safety system in aircraft.

Security:

Security is a system property that reflects the 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

Example: Viruses, unauthorised use of service/data modification

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 system 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.

Damage from insecurity

• Denial of service

– The system is forced into a state where normal services become unavailable.

• Corruption of programs or data

– The system components of the system may alter in an unauthorised way, which affect system behaviour & hence its reliability and safety

• 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 remove them before they result in an exposure. For example, virus checkers find and remove viruses before they infect a system

• Exposure limitation

– The consequences of a successful attack are minimised. For example, a backup policy allows damaged information to be restored