

# Safety Critical Systems

---



Mostly from: Douglass, “Doing Hard Time, developing Real-Time Systems with UML, Objects, Frameworks And Patterns”, Addison-Wesley. ISBN 0-201-49837-5

# Definitions

---

- ◆ **channel** – a set of devices (including processors and software running on processor) that handle a related, cohesive set of data control flows from incoming event to ultimate system response.
- ◆ **error** - (ISO) A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.
- ◆ **error detection** - Techniques used to identify errors in data transfers. See: check summation, cyclic redundancy check [CRC], parity check, longitudinal redundancy.
- ◆ **error seeding** - (IEEE) The process of intentionally adding known faults to those already in a computer program for the purpose of monitoring the rate of detection and removal, and estimating the number of faults remaining in the program. Contrast with mutation analysis.

- ◆ **fail-safe** - (IEEE) A system or component that automatically places itself in a safe operational mode in the event of a failure.
- ◆ **failure** - (IEEE) The inability of a system or component to perform its required functions within specified performance requirements.
- ◆ **failure analysis** - Determining the exact nature and location of a program error in order to fix the error, to identify and fix other similar errors, and to initiate corrective action to prevent future occurrences of this type of error.
- ◆ **Failure Modes and Effects Analysis – [FMEA]** (IEC) A method of reliability analysis intended to identify failures, at the basic component level, which have significant consequences affecting the system performance in the application considered.

- ◆ **Failure Modes and Effects Criticality Analysis – [FMECA] (IEC)**  
A logical extension of FMEA which analyzes the severity of the consequences of failure.
- ◆ **fault** – an incorrect step, process, or data definition in a computer program which causes the program to perform in an unintended or unanticipated manner.
- ◆ **fault seeding** - See: error seeding.
- ◆ **Fault Tree Analysis – [FTA] (IEC)** The identification and analysis of conditions and factors which cause or contribute to the occurrence of a defined undesirable event, usually one which significantly affects system performance, economy, safety or other required characteristics.

- ◆ **hazard** - (DOD) A condition that is prerequisite to a mishap.
- ◆ **hazard analysis**. A technique used to identify conceivable failures affecting system performance, human safety or other required characteristics. See: FMEA, FMECA, FTA, software hazard analysis, software safety requirements analysis, software safety design analysis, software safety code analysis, software safety test analysis, software safety change analysis.
- ◆ **hazard probability** - (DOD) The aggregate probability of occurrence of the individual events that create a specific hazard.
- ◆ **hazard severity** - (DOD) An assessment of the consequence of the worst credible mishap that could be caused by a specific hazard.

- ◆ **mean time between failures – [MTBF]** A measure of the reliability of a computer system, equal to average operating time of equipment between failures, as calculated on a statistical basis from the known failure rates of various components of the system.
- ◆ **mean time to failure – [MTTF]** A measure of reliability, giving the average time before the first failure.
- ◆ **mean time to repair – [MTTR]** A measure of reliability of a piece of repairable equipment, giving the average time between repairs.
- ◆ **reliability - (IEEE)** The ability of a system or component to perform its required functions under stated conditions for a specified period of time. See: software reliability.
- ◆ **reliability assessment - (ANSI/IEEE)** The process of determining the achieved level of reliability for an existing system or system component.

# Definitions

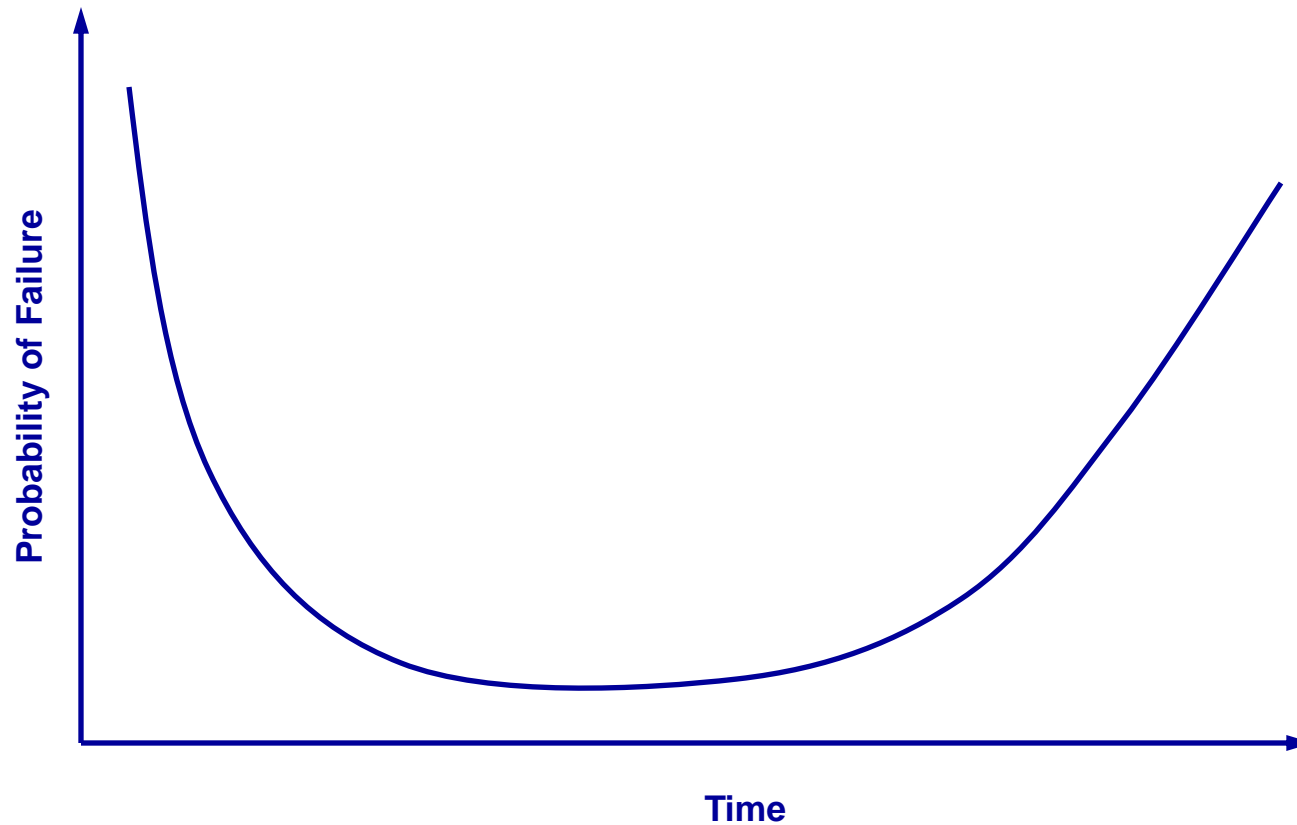
(cont)

- ◆ **risk** - (IEEE) A measure of the probability and severity of undesired effects. Often taken as the simple product of probability and consequence.
- ◆ **risk assessment** - (DOD) A comprehensive evaluation of the risk and its associated impact.
- ◆ **Safety** - (DOD) Freedom from those conditions that can cause death, injury, occupational illness, or damage to or loss of equipment or property, or damage to the environment.
- ◆ **safety critical** - (DOD) A term applied to a condition, event, operation, process or item of whose proper recognition, control, performance or tolerance is essential to safe system operation or use; e.g., safety critical function, safety critical path, safety critical component.

- ◆ **safety critical computer software components - (DOD)**  
Those computer software components and units whose errors can result in a potential hazard, or loss of predictability or control of a system.
- ◆ **(computer system) security - (IEEE)** The protection of computer hardware and software from accidental or malicious access, use, modification, destruction, or disclosure. Security also pertains to personnel, data, communications, and the physical protection of computer installations.

# Failure curve for HW Components

---



# Failures vrs. Faults

---

- ◆ Failures are caused by Faults (it is difficult to distinguish between Fault and Error). There are two types of faults:
  - **Random Faults** – faults that occur in a random way, described by a probability function (e.g. a HW component that was working suddenly stops functioning)
  - **Systematic Faults** – faults deriving from a systematic cause, e.g. a design error. All SW errors are systematic faults.

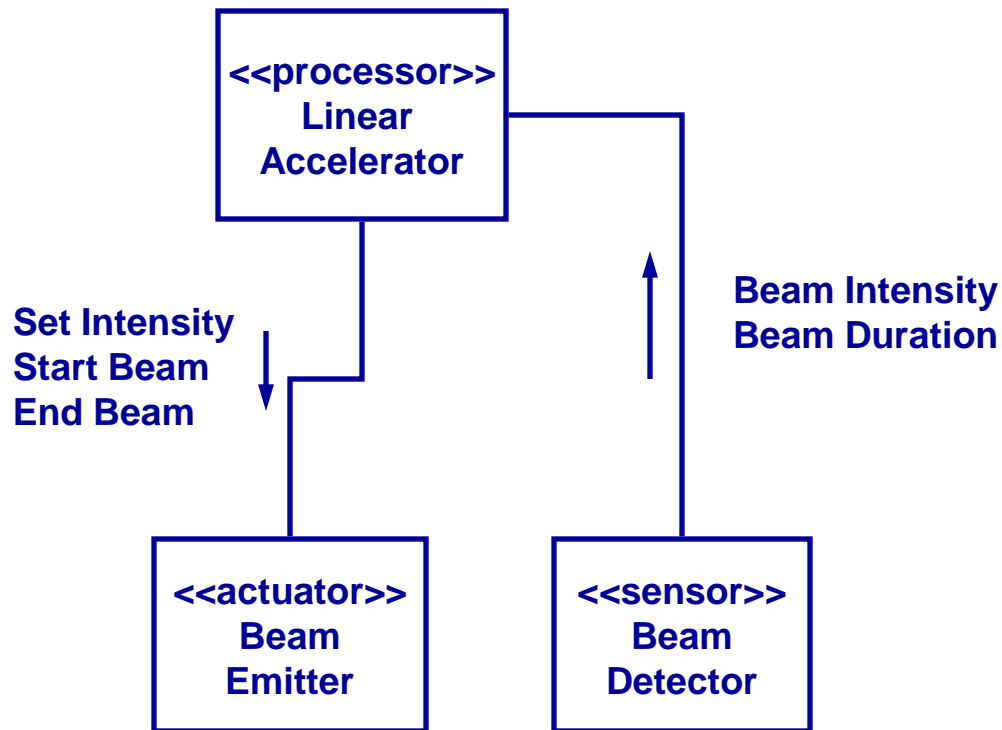
# Single-Point Failures

---

- ◆ Devices ought to be safe when there are no faults and the device is used properly.
- ◆ Most experts however consider a device safe when any single point failure cannot lead to an incident.

# Single-Point Failures

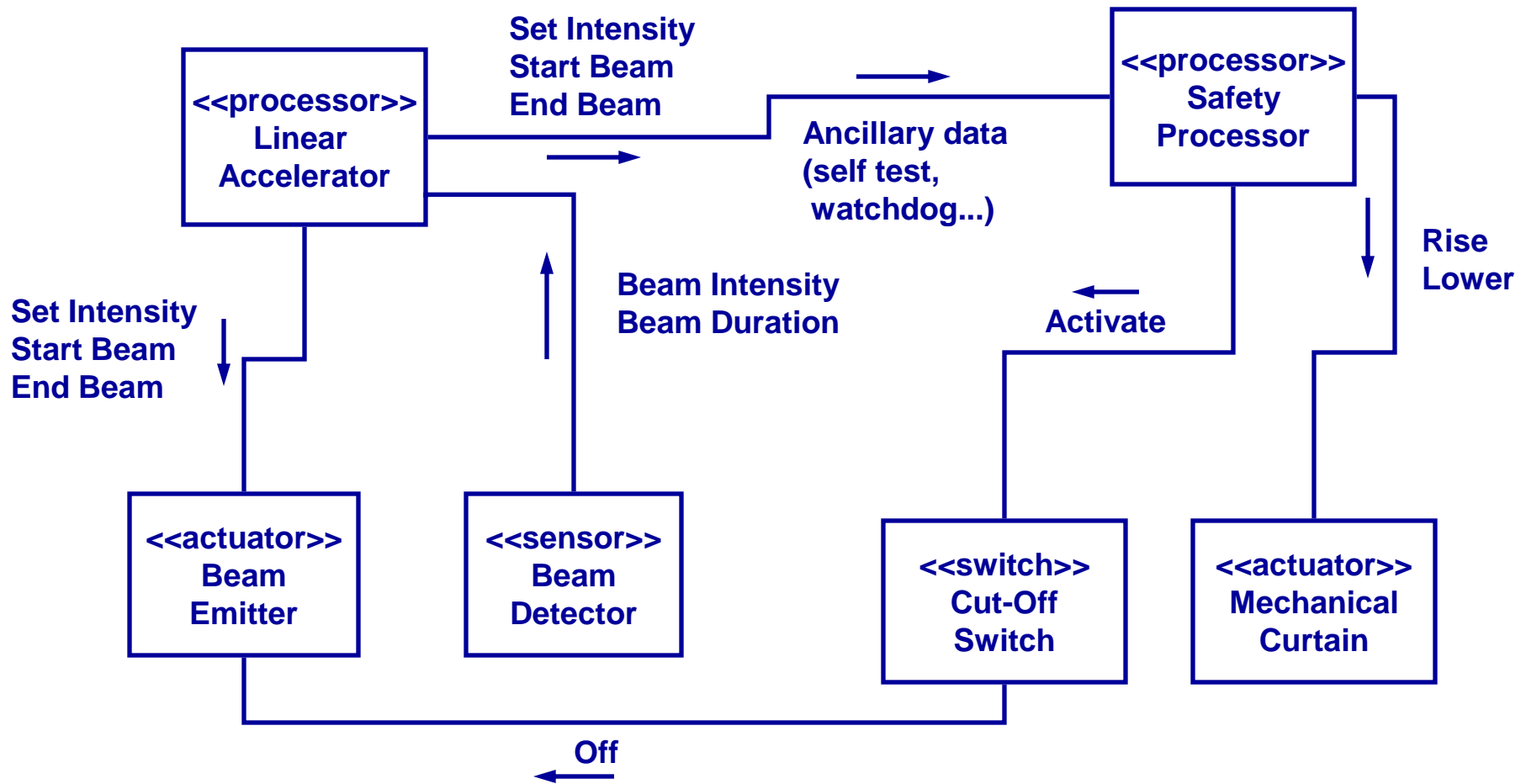
(cont)



**UNSAFE**

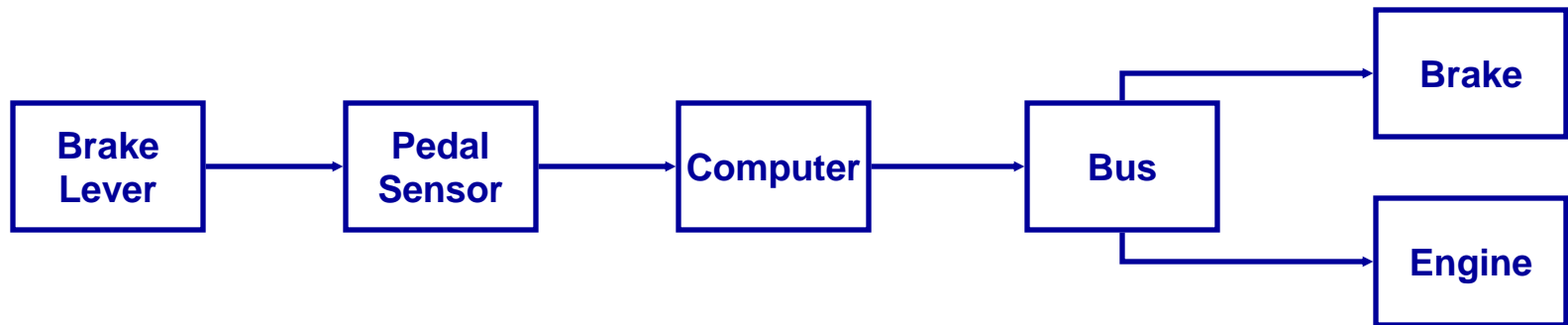
# Single-Point Failures

(cont)



# Safety Architectures

- ◆ Single Channel Protected Design (SCPD)
- ◆ In a SCPD architecture, a single channel exists for the control of some process. However, this single channel can be made by applying sufficient hazard control within the channel. E.g.

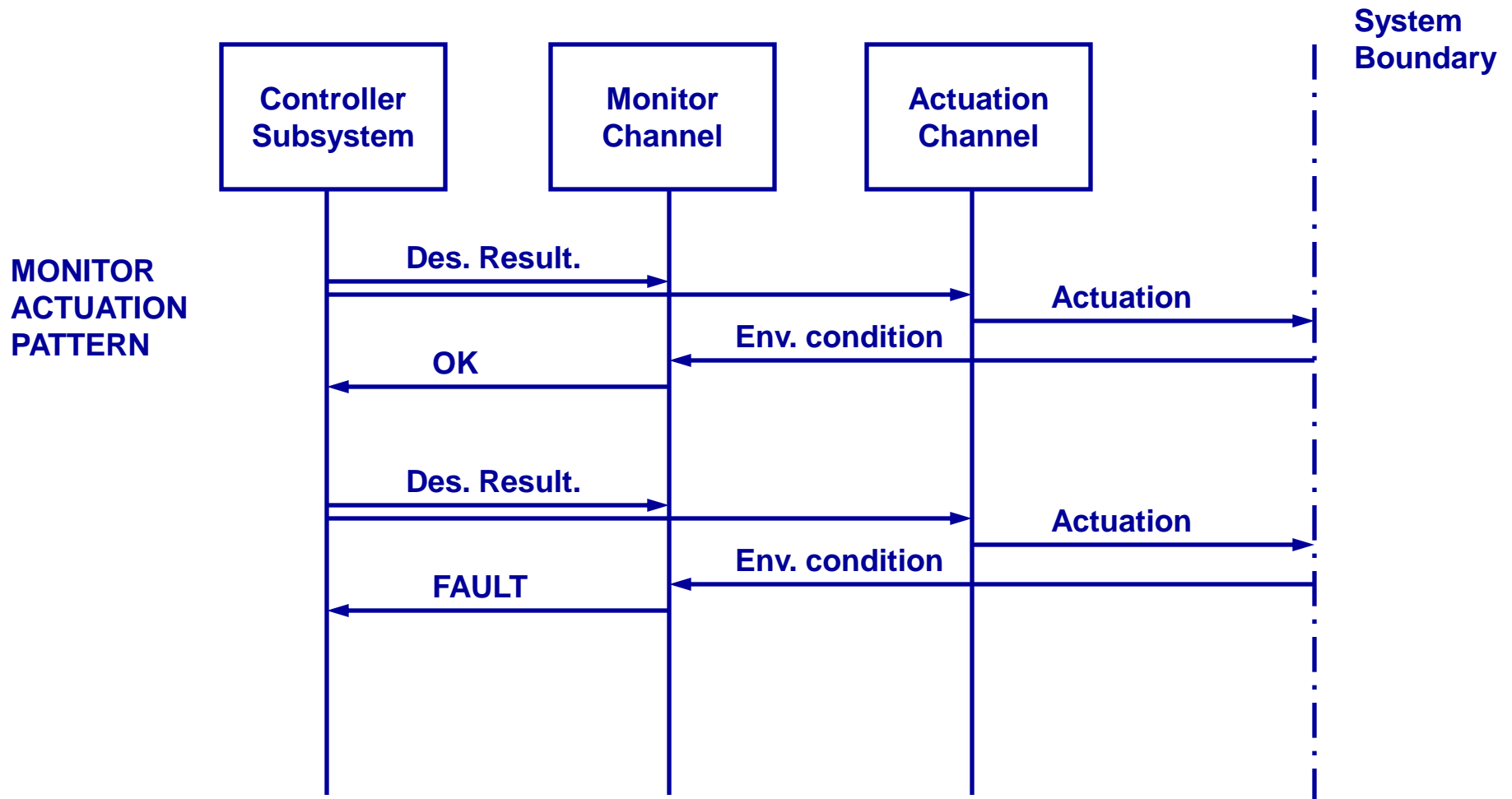


- ◆ Homogeneous Redundancy Pattern
- ◆ The HRP architecture use identical channels to increase reliability. All the redundant channels are run in parallel and the output of the channels is compared. If an oo number of channels is used, than a “majority wins” policy can be implemented that can detect and correct failures in the minority channels.

- ◆ Diverse Redundancy Pattern
- ◆ The DRP architecture provides redundant channels that are implemented by different means. This can be achieved in several ways:
  - Different but equal
  - Lightway redundancy
  - Separation of monitoring and actuation
- ◆ Watchdog pattern

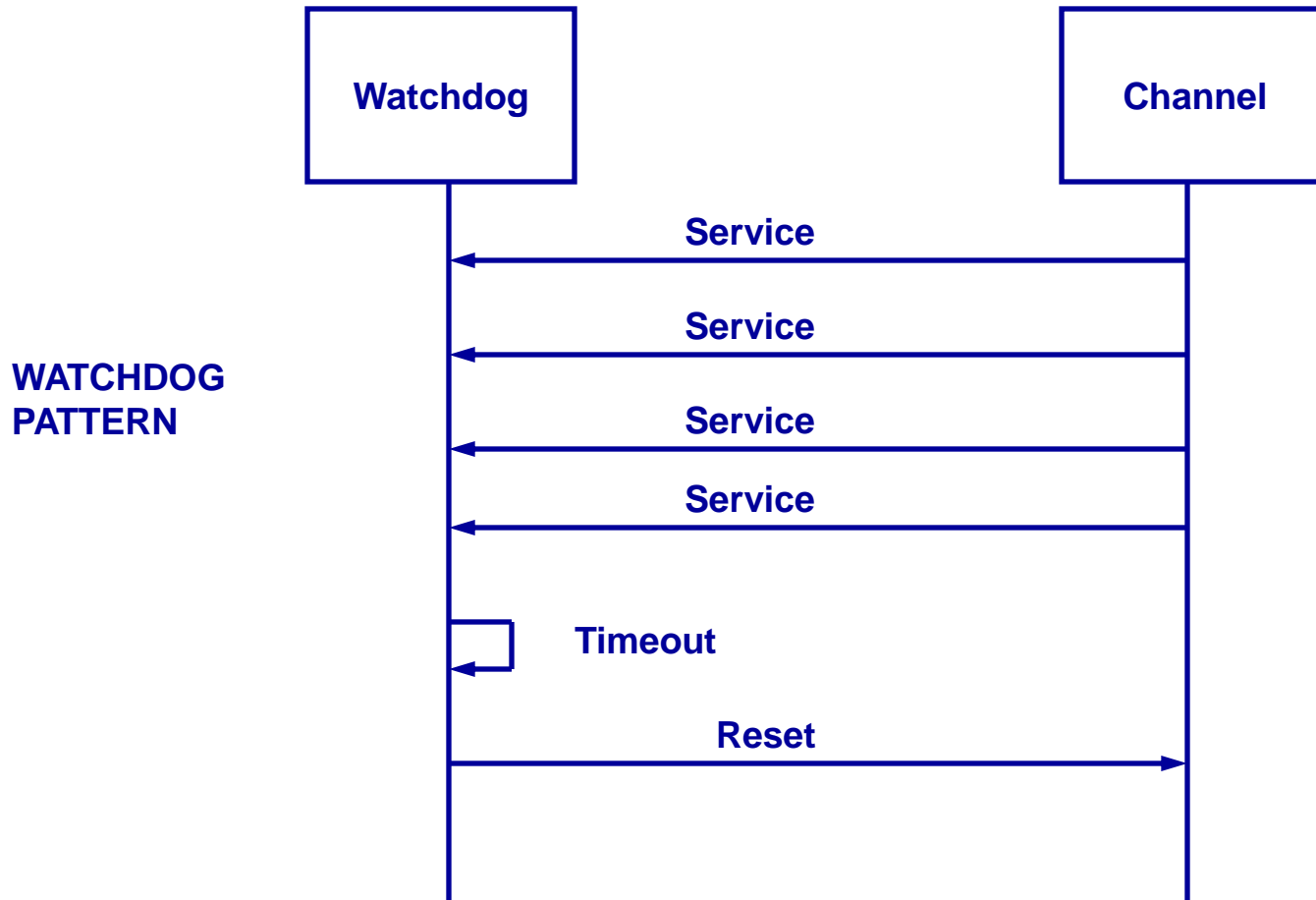
# Safety Architectures

(cont)



# Safety Architectures

(cont)



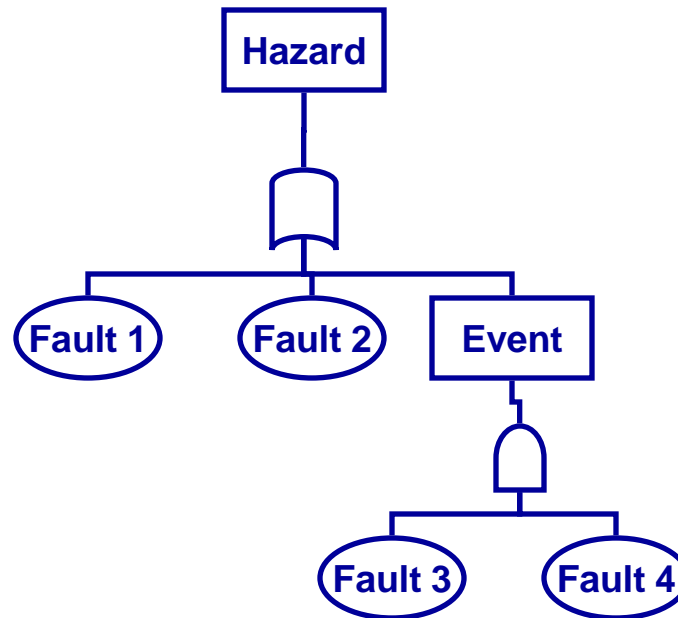
# Eight Steps to Safety

---

- ◆ Identify hazards
- ◆ Determine the risks
- ◆ Define the safety measures
- ◆ Create safe requirements
- ◆ Create safe designs
- ◆ Implement safety
- ◆ Assure the safety process
- ◆ Test, test, test

# Identify the hazards

- ◆ List the possible hazards
- ◆ For each hazard identify the faults leading to it (Fault Tree Analysis)



# Determine the risks

---

- ◆ Organizations like TÜV have defined risk requirement categories that take into account:
  - Severity of the risk
  - Duration of the period of exposure to the risk
  - Prevention of the danger
  - Probability of occurrence of the danger

Etc...

# Define the safety measures

---

- ◆ A safety measure is a behaviour added to a system to handle hazard. There are several ways to handle a hazard:
  - Obviation – making the hazard physically impossible
  - Education – educate the users so that they won't cause hazards
  - Alarming – announce the hazard to the user so that he can take proper recovery actions
  - Interlocks – the hazard is removed by making some other devices, logic, intervene when the hazard presents itself
  - Internal checking – the hazard is prevented from happening because the system is able to detect a malfunction prior to an incident
  - Safety equipment – gloves, goggles, etc...
  - Restriction of access – let only knowledgeable users access the system
  - Labelling - ...
  - Etc..

# Create safe requirements

---

- ◆ Safety has to be introduced at requirements level.

# Create safe designs

---

- ◆ Work from safe requirements
- ◆ Adopt a fundamentally safe architecture
- ◆ Periodically, during design, revisit the hazard analysis to add hazards due to faults specific to design
- ◆ Select programming in the small measures that provide the appropriate level of detection and correction
- ◆ Ensure that independent channels truly lack common modes failures
- ◆ Adopt a consistent and appropriate set of strategies for handling faults once they are identified
- ◆ Build in power on-on and periodic run-time tests to identify latent faults.

# Implement safety

---

- ◆ Language selection
- ◆ Subset selection
- ◆ Use the tools (i.e. Methodologies, programming languages, etc..) in a selected, predefined and constrained way.

# Process / Test

---

- ◆ Make sure safety is a concern in all phases of your development process.
  
- ◆ Test / Test / Test