Safety critical software

Patrick R.H Place Kyo C.Kang

건국대학교 컴퓨터공학과

200310323 권도윤

Purpose

- To understand the role of safety-critical software in requirement engineering.

- Bring together concepts necessary for the development of software in safety-critical systems.

- Understanding of Hazard Identification and analysis

Background

- Systems whose failure can threaten human life or cause serious environmental damage

- New Software components are replacing existing hardware component

- Hardware safety is often based on the physical properties of the hardware.

- traditional engineering techniques cannot be used with software

Definition of terms

- Mishap (or accident)

  An unplanned event or event sequence which results in human death or injury.

- Hazard

  A condition with the potential for causing or contributing to a mishap

- Hazard severity

  An assessment of the worst possible damage which could result from a particular hazard

- Hazard probability

  The probability of the events occurring which create a hazard

- Risk

  This is a complex concept which is related to the hazard severity, the hazard

Requirements Engineering and Safety

- Safety Issues must be considered from the start
- Safety concerns often conflict with performance and/or cost
- Hazards Risk Analysis must be performed independent of Performance and Cost Risk Analyses
- individual components may be safe, the integrated system may not be safe
- Customers requirements has to be organized into a coherent form that may be analyzed

Comments on Software Safety

Safety is a System Issue

- Safety is not software issue. it is a system issue.
- Software does nothing unsafe.
- **what makes system unsafe?**

- Control of systems with hazardous components
- Providing of information to people who make decision that have potentially hazardous consequences.

•Software can be considered unsafe only in the context of a particular system.

Safety is Measured as Risk

- Safety is abstract concept

- The definition of safety becomes related to risk

$$Risk = \sum_{hazard} E_{hazard} \times P_{hazard}$$

- E_{hazard} is a measure of the effect that may be caused by rticular mishap
- P_{hazard} is the probability that the mishap will occur

There is no system wholly safe. So what we have to?

- minimize the risk by containing the hazard
- reduce the probability that the hazard will occur

Reliability is Not Safety

Reliability

- measure of the rate of failure make the system unusable
- concerned with conformance to a given specification and delivery of service

Safety

- concerned with ensuring system cannot cause damage irrespective of whether or not it conforms to its specification

- measure of the absence of unsafe software conditions

Software Need Not Be Perfect

- Software need not be perfect to be safe

- if errors are masked, or ignored by the safety components, the system could still be safe.

- ex) Nuclear power plant using control room and protection software

- Developers and analyst of safe software can concentrate their most detailed check on the safety conditions and not on the operational requirements

"it is commonly assumed that other parts of the system are imperfect and may not behave as expected”

Safe Software Is Secure and Reliable

- The safety critical components of a system need to be secure since it is important that the software and data cannot be altered by external software or human).

If the safety system software is unsecure?

the data or software can be altered, then the executing components will no longer safe

If the safety system software is unreliable?

System require the software to be operational to prevent mishap
Unreliable software could fail to perform when needing avoid mishap

Software Should Not Replace Hardware

advantages of software

it is flexible and relatively easy to modify
Software reproduction costs are very low
Hardware may be quite expensive to reproduce

What is the problem if software replace hardware?

hardware fails in more predictable ways than software,
a failure may be foreseen by examining the hardware

Software does not exhibit physical characteristics that may be observed in the same way as hardware

there may be no warning of the impending failure
It is a danger that leads to unsafe systems.

Hazard identification

- No easy way to identify hazards within a given system.
- But a mishap should not be allowed to occur

How we can identify system hazards?

The only acceptable approach for hazard identification is to attempt to develop a list of possible system hazards before the system is built.

What techniques we can use?

- The obvious approach is to use “**brainstorming**,”
- Delphi Technique or Joint Application Design (JAD)

The Delphi Technique

- The basic approach is to send out a questionnaire to all members of the group that enables them to express their opinions on the topic of discussion.

- The group opinion is defined as the aggregate of individual opinions after the final round.

advantage & disadvantage

- The Delphi Technique overcomes the issue of group consensus when the group is unable to attend a meeting

- Delphi Technique makes for slow communication and it may take several weeks to arrive at consensus.

Joint Application Design(JAD)

- To help a group reach decisions about a particular topic.

- Used for any meeting where group consensus must be reached concerning a system to be deployed.

What makes JAD to be successful?

- the group must be made up of people with certain characteristics

- A JAD session is led by a facilitator who should have no vested interest in the detailed content of the design

- ideas should become owned by the group rather than individuals

Hazard Analysis

- To examine the system and determine which components of the system may lead to a mishap

- two basic strategies to analysis

Inductive

consider a particular fault in some component of the system and then attempt to reason what the consequences of that fault will be

ex) event tree analysis and failure modes and effects analysis,

Deductive

consider a system failure and then attempt to reason about the system
Component states

ex) fault tree analysis

Fault Tree Analysis.

- deductive hazard analysis technique

- Starts with a particular undesirable event and provides an approach for analyzing the causes of this event

- It is important to choose this event carefully

- A graphical representation of the various combinations of events that lead to the undesired event.

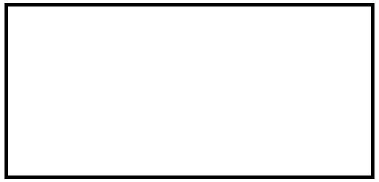

Figure 3-5: *Intermediate* Event

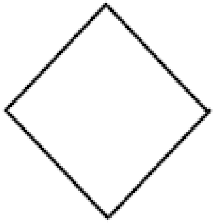

Figure 3-4: *Undeveloped* Event

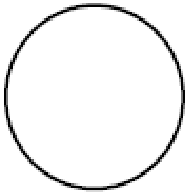

Figure 3-3: *Basic* Event

Fault Tree Analysis

- Once the undesirable event has been chosen, it is used as the top event of a fault tree diagram. ex) car his object

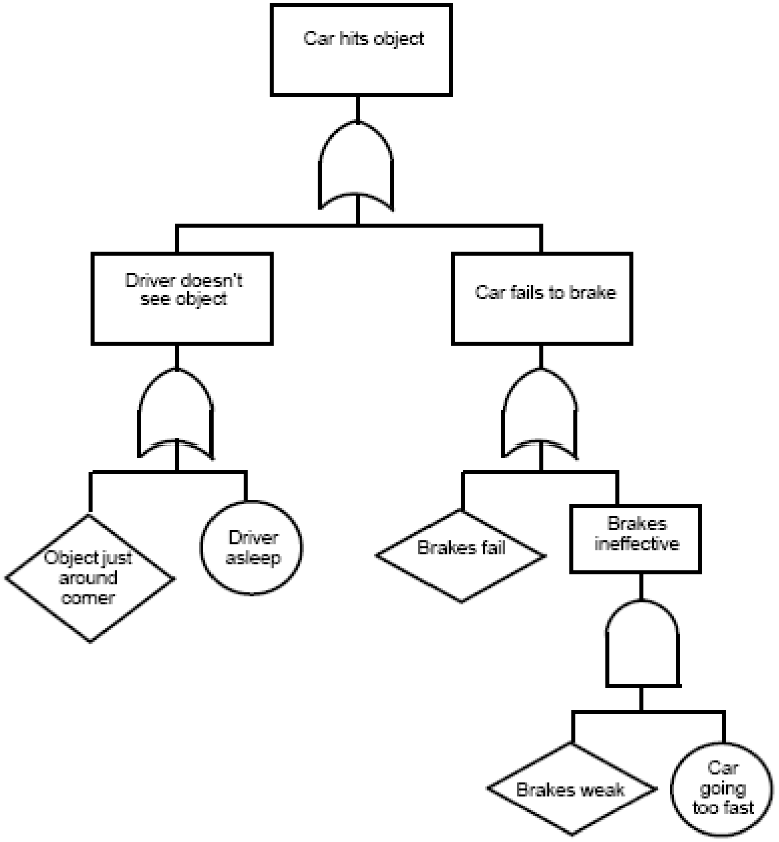

Figure 3-6: Example Fault Tree for a Car Crash

Event tree Analysis

- inductive technique using essentially the same representations as fault tree analysis

- The purpose of event tree analysis is to consider an initiating event in
  the system and consider all the consequences of the occurrence that
  lead to a mishap

What is difference between FTA and ETA?

Event tree analysis is forward looking and considers potential future problems while fault tree analysis is backward looking and considers knowledge of past problems

Event tree analysis is not as widely used as fault tree analysis.

Failure mode and Effect Analysis

- inductive technique and attempts to anticipate potential failures so that the source of those failures can be eliminated.

- consists of constructing a table based on the components of the system and the possible failure modes of each component.

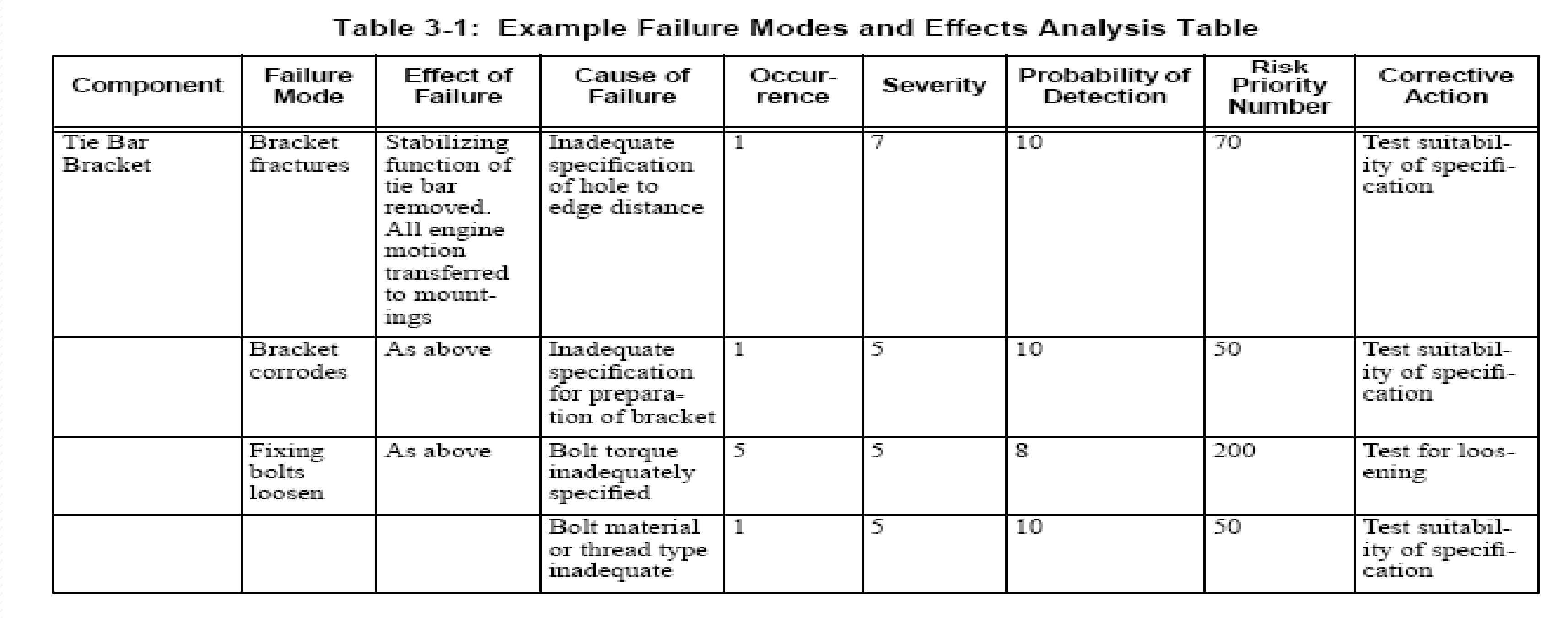

Table 3-1: Example Failure Modes and Effects Analysis Table

Component	Failure Mode	Effect of Failure	Cause of Failure	Occurrence	Severity	Probability of Detection	Risk Priority Number	Corrective Action
Tie Bar Bracket	Bracket fractures	Stabilizing function of tie bar removed. All engine motion transferred to mountings	Inadequate specification of hole to edge distance	1	7	10	70	Test suitability of specification
	Bracket corrodes	As above	Inadequate specification for preparation of bracket	1	5	10	50	Test suitability of specification
	Fixing bolts loosen	As above	Bolt torque inadequately specified	5	5	8	200	Test for loosening
			Bolt material or thread type inadequate	1	5	10	50	Test suitability of specification

Summary

- The process of performing a safety analysis of a system is time consuming and employs many techniques all of which require considerable domain expertise

- Create a list of all hazards and for those with a sufficiently high risk perform fault tree analysis indicating which components are safety critical.

- Perform an FMEA for all components of the system, potentially using fault tree and event tree analysis to determine causes and effects of a component failure respectively.