SAFETY-CRITICAL SOFTWARE : STATUS REPORT

컴퓨터공학부
소프트웨어
200312478노태호

CONTENTS

INTRODUCTION

- Requirement Engineering 에서의 “Safety-Critical Software”
- Safety-Critical분야에서 소프트웨어의 사용이 증가하고 있다
- 많은 분야에서 소프트웨어가 하드웨어를 대체하고 있다
- 시스템의 새로운 실패 양상을 소개한다

COMMENTS ON SOFTWARE SAFETY

- Safety Is a System Issue
- Safety Is Measured as Risk
- Reliability Is Not Safety
- Software Need Not Be Perfect
- Safe Software Is Secure and Reliable
- Software Should Not Replace Hardware
- Development Software Is Also Safety-Critical

Safety Is a System Issue

- 1991년 Leveson은 안정성이 시스템문제라고 주장
- 특정 시스템 상황에서 소프트웨어는 불안전한 것으로 간주됨
- 시스템 레벨에서 소프트웨어는 하나의 구성요소로 취급
- 이러한 상태는 사고발생을 초래할 수 있음

Safety Is Measured as Risk

- 안정성(Safety)은 추상적인 개념
- 안전한 시스템 – 사람이나 재산의 피해를 입히는 원인이 되지 않음
- 완전히 안전하게 만들 수 있는 시스템이 있지만 그 시스템은 기능의 저하를 초래함
- 안전의 정의는 위험과 관련
- $$\text{Risk} = \sum_{\text{hazard}} \text{E}_{\text{hazard}} \times \text{P}_{\text{hazard}}$$

Reliability Is Not Safety

- 신뢰성과 안전성은 다르다
- 신뢰성
  시스템을 사용할 수 없게 만드는 실패 확률의 측정
  시스템이 얼마나 기능을 잘 수행하는가
- 안전성
  시스템에서 위험한 행위가 없음
  시스템이 사고로 이어지지 않는가

Software Need Not Be Perfect

- 소프트웨어는 안전하기 위해 완벽 할 필요는 없다
- 완벽함의 개념은 모든 오류를 동등하게 고려한다. 그러므로 소프트웨어는 완벽하지 않다
  ex) 맞춤법실수, 출력오류, 심각한 기능오류
- 하지만 Failure의 원인이 되는 에러들은 중요하다

Safe Software Is Secure and Reliable

- 안전성과 보안 사이에는 차이가 있다
- 보안은 신뢰성에 의존, 안전성은 보안에 의존한다
- 안전에 중요한 소프트웨어와 데이터는 외부에 의해서 변경될 수 없기 때문에 안전할 필요가 있다

Software Should replace Hardware

- 소프트웨어의 장점 : 유연 ,쉽게 변경 가능
- 소프트웨어의 경제적 장점

➡ 개발이 완료되면 재생산 비용이 아주 낮음

- 하드웨어는 재생산이 비싸다
- 경제적 관점에서 하드웨어를 소프트웨어로 교체하는 것은 매력적이다
- 단점: 교체에는 위험이 따른다

ex)방사선 치료기계 Therac20(퓨즈폭발),Therac25(치명적인 방사선 투사)

Development Software Is Also Safety Critical

- 개발 소프트웨어도 안전해야 한다
- 1. 현재의 개발단계의 분석은 설명서에 따르면 안전하다는 것을 보여주는 것
- 2. 후속 개발단계에서는 현재의 설명서에 따름
- 소프트웨어 안전성 분석의 최저 수준은 어셈블리 언어로 되어있는 실행언어 수준에서 수행

Hazard Analysis Techniques

- Hazard Identification
  - Delphi Techniques
  - Joint Application Design(JAD)

- Hazard Analysis
  - Fault Tree Analysis
  - Event Tree Analysis
  - Failure Modes and Effects Analysis(FMEA)

Delphi Techniques

- 미국정부를 위해 Rand corporation이 만듦
- 각각의 개인에게 질문 표를 보내고 그것을 합하여서 그룹의 일치된 결정을 이끌어낸다
- 단점 – 느린 의사소통, 서로 다른 날짜에 의견들이 도착함
- 요즈음의 전자메일이나 통신의 발달로 delphi Techniques의 문제를 해결하는데 도움이 된다

Joint Application Design(JAD)

- JAD는 IBM에 의해서 소개
- 목적 – 특별한 주제의 그룹 범위 결정을 도움
- 성공적인 JAD를 위한 조건
  - 숙련된 전문가들 이어야 한다
  - 대표한 그룹의 결정권이 있어야 한다
  - 6~10명 정도의 사람이 적당하다
- 운영자는 기술적인 능력이 있어야 하고, 커뮤니케이션과 외교, 대립되는 의견을 제어할 재능이 있어야 한다

Fault Tree analysis

Basic event

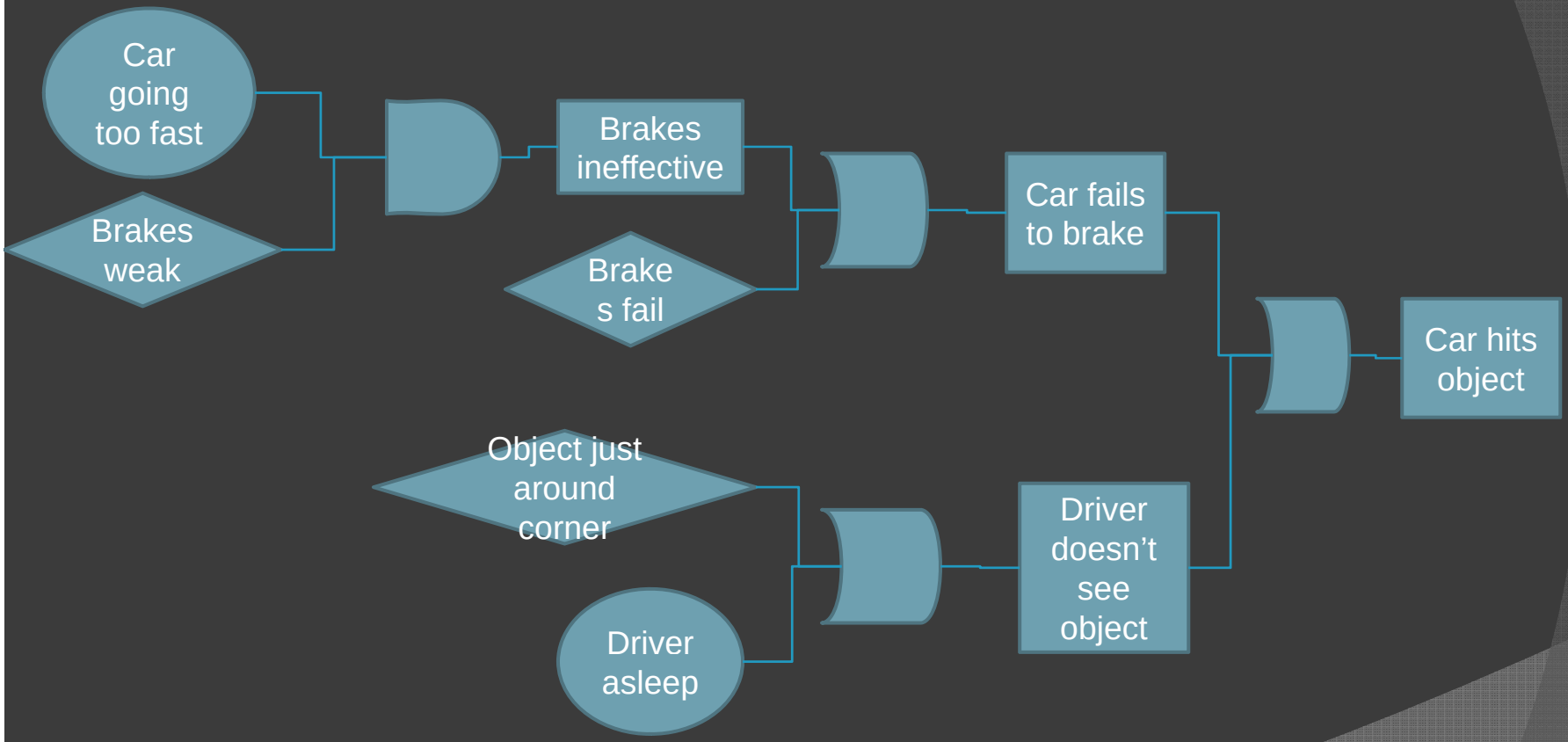
Undeveloped event

And gate

Or gate

Intermediate event

Example fault tree for a car crash



Event Tree Analysis

- Fault Tree Analysis 와 대조적
  inductive한 방법
- Fault Tree Analysis의 기호를 그대로 사용
- Fault Tree Analysis는 하드웨어 시스템의 실패를 검사하는것의 수단으로 등장
  Event Tree Analysis는 소프트웨어 시스템의 실패를 검사하기 위해서 등장
- Fault Tree Analysis 보다 널리 사용되지 못함
- 이유? 가능한 모든 결과를 고려하는 것이 힘듦
  Tree가 커지고 다루기 힘들게 될 수 있음

Failure Modes & Effect Analysis(FMEA)

- Component
- Failure mode
- Effect of failure
- Cause of failure
- Occurrence
- Severity
- Probability of detection
- Risk priority number
- Corrective action

Failure Modes & Effect Analysis(FMEA)

- Component
  프로세서 검토
- Failure mode
  고장이 일어날 가능성이 있는 모드를 나열
- Effect of failure
  고장에 의해서 야기되는 효과를 나열
- Cause of failure
  고장의 원인이 무엇인지 분석
- Occurrence
  고장의 중대성을 1~10으로 설정

Failure Modes & Effect Analysis(FMEA)

- Severity
  고장의 발생비율을 1~10으로 설정
- Probability of detection
  고장의 검출비율을 설정
- Risk priority number
  위험의 우선순위를 정하기 위해 Occurrence, Severity, Probability of detection을 곱한 값을 구하여 적음
- Corrective action
  고장을 줄이기 위한 행동을 적음

SUMMARY

- Delphi와 JAD모두 어떤 주제에서 그룹의 합의를 얻는 것에 대한 접근
- 시스템 안전분석은 많은 시간과 기술자를 요구
- Fault Tree Analysis는 deductive(연역적)이고 Event Tree Analysis와 Failure Modes & Effects Analysis는 inductive(귀납적)인 분석방법
- Safety-Critical Software 분석은 모든 위험들 중 높은 위험을 가진 요소들을 분석하는 것

THE END

감사합니다