Common Cause Failure (CCF)

건국대학교 컴퓨터공학과
UC Lab. 정 혁 준 & 박 경 식

Table of Contents

1. Introduction

1.1. Conditional Probability

Conditional Probability란 어떤 두 개의 사상 *A*와 *B*에 있어서 사상 *A*가 일어나는 것을 조건으로 하여 사상 *B*가 일어나는 경우에 대한 확률을 의미하며, 이 확률은 사상 *A*가 주어졌을 때 사상 *B*의 Conditional Probability, $P(B|A)$로 나타낼 수 있다.

$P(B|A)$는 $P(A)$에 대한 $P(A \cap B)$의 비로써 정의 할 수 있으며, 다음의 식 1과 같이 나타낼 수 있다.

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \qquad \text{식 1}$$

식 1에서 $P(A)$는 주어진 사상 *A*가 일어날 확률이고, $P(A \cap B)$는 주어진 사상 *A*와 제기된 사상 *B*가 동시에 일어날 확률이다.

1.2. Independent Failures

Independent Failures란 *Item 1* 과 *Item 2*가 있을 때, *Item 1*이 고장날 확률이 *Item 2*가 고장날 확률에 아무런 영향을 주지 않는 것을 의미한다. 다시 말해서 *Item 1*과 *Item 2*가 존재하고 E_i가 *Item i* 를 고장 상태에 이르게 하는 Event를 나타낼 때, *Item 1*과 *Item 2*가 모두 고장 상태일 확률은 식1의 Conditional Probability를 이용하여 다음의 식 2와 같이 나타낼 수 있다.

$$P(E_1 \cap E_2) = P(E_1|E_2) \cdot P(E_2) = P(E_2|E_1) \cdot P(E_1) \qquad \text{식 2}$$

이때 만약 식 2에서 $P(E_1|E_2) = P(E_1)$, $P(E_2|E_1) = P(E_2)$이면, 다음 식 3과 같이 되므로 E_1과 E_2은 통계적으로 Independent 하다고 할 수 있다.

$$P(E_1 \cap E_2) = P(E_1) \cdot P(E_2) \qquad \text{식 3}$$

1.3. Dependent Failures

Dependent Failures란 Independent Failures와 다르게 *Item 1*과 *Item 2*가 있을 때, *Item 1*이 고장날 확률이 *Item 2*가 고장날 확률에 영향을 끼치는 것을 의미한다. 다시 말해서 *Item 1*과 *Item 2*가 존재하고 E_i가 *Item i*를 고장 상태에 이르게 하는 Event를 나타낼 때, $P(E_1|E_2) \neq P(E_1)$, $P(E_2|E_1) \neq P(E_2)$이면 E_1

과 E_2은 통계적으로 Dependent 하다고 할 수 있다.

- Positive Dependence : *Item 1*과 *Item 2*에 대하여 $P(E_1 \mid E_2) > P(E_1)$, $P(E_2 \mid E_1) > P(E_2)$이면 $P(E_1 \cap E_2) > P(E_1) \bullet P(E_2)$. 이 때 *Item 1*과 *Item 2*는 Positive Dependence를 갖는다고 말한다.

- Negative Dependence : *Item 1*과 *Item 2*에 대하여 $P(E_1 \mid E_2) < P(E_1)$, $P(E_2 \mid E_1) < P(E_2)$이면 $P(E_1 \cap E_2) < P(E_1) \bullet P(E_2)$. 이 때 *Item 1*과 *Item 2*는 Negative Dependence를 갖는다고 말한다.

- Intrinsic Dependency : 시스템의 한 컴포넌트의 기능적 상태가 시스템의 또 다른 컴포넌트의 기능적 상태에 의하여 영향을 받는 상황을 Intrinsic Dependency라 한다.
  - Functional Requirement Dependency
  - Functional Input Dependency
  - Cascading Failure

- Extrinsic Dependency : Intrinsic Dependency가 아닌 경우, 즉, 시스템 내부의 영향이 아닌 외부의 요인으로 인해 영향을 받는 상황을 Extrinsic Dependency라 한다.
  - Physical or Environment Stresses
  - Human Intervention

Cascading Failures

Cascading Failures는 다음 1.4절에서 설명할 Common Cause Failure (CCF)의 범주에 속한다는 견해와 혹은 Dependent Failures의 범주에서 Common Cause Failure (CCF)와 동등한 위치에 있다는 견해가 있으나 이 문서에서는 전자를 따른다.

Cascading Failures는 일련의 연쇄적인 고장이라 할 수 있다. 즉, 첫 번째 Item의 고장이 그 다음 Item들의 부하를 증가시켜 연쇄적으로 고장을 일으키는 것을 말한다. 때문에 Cascading Failures를 Domino Effect라고도 한다.

1.4. Definition of Common Cause Failure (CCF)

Common Cause Failure (CCF)란 그림 1과 같이 하나의 공통 원인으로 인하여 둘 또는 그 이상의 컴포넌트들이 동시에 혹은 짧은 시간 간격으로 고장 상태에 이르는 Dependent Failures를 말한다.

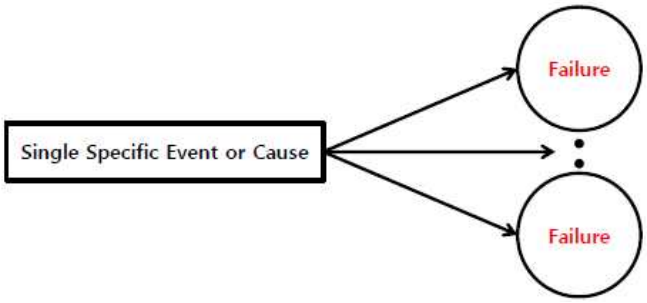

그림 1. Common Cause Failure (CCF)

예를 들어, 원자력 발전소 설계 시 원자력 발전소가 정상운전 상태를 벗어날 때 사고로 전개되는 것을 예방하거나, 사고 피해를 최소화하기 위해 중요 안전 기능을 수행하는 설비를 필요 수량보다 더 많이 설치하게 되는데, 설치된 설비들의 작동 원리가 같은 경우에는 한 가지 Common Cause에 의하여 모든 설비가 기능을 상실 할 수 있다.

CCF는 다음 그림 2와 같이 Root Cause와 Coupling Factor라는 두 가지 주요 속성을 가진다.

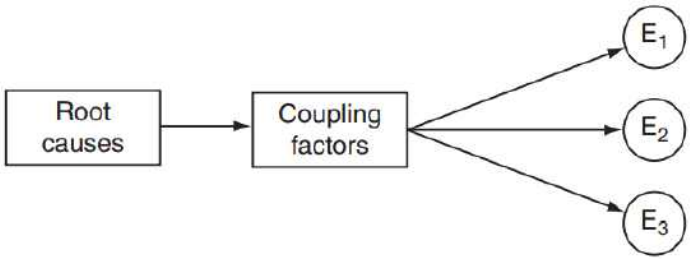

그림 2. Main Attributes of CCF

- Root Cause : Item 고장의 가장 근본적인 원인을 말한다. (Why did the item fail?)

- Coupling Factor : 다수의 Item들이 공통의 Root Cause에 의해 영향을 받

는 속성을 말한다. Coupling Factor는 Coupling Mechanism이라고도 한다. (Why were several items affected?)

Root Cause는 다음과 같이 Pre-Operational Root Cause와 Operational Root Cause로 구분할 수 있다.

- Pre-Operational Root Causes : Operation 전 요인들로 인한 Root Cause를 말한다.
  - Design, Manufacturing, Construction, Installation, Commissioning Errors

- Operational Root Causes : Operation 중 요인들로 인한 Root Cause를 말한다.
  - Operation and Maintenance-Related : Inadequate Maintenance, Operational Procedures, Execution, Competence and Scheduling
  - Environmental Stresses : Earthquake, Fire, Floodind, ETC.

Coupling Factor는 일반적으로 다음과 같이 *Item*들 간의 유사성을 찾음으로써 발견할 수 있다

- Same Design
- Same Hardware
- Same Function
- Same Software
- Same Installation Staff
- Same Maintenance and Operational Staff
- Same Procedures
- Same System/Item Interface
- Same Environment
- Same Physical Location

1.5. Attributes of a CCF Definition

1980년 Smith와 Watson이 제안한 CCF 속성의 정의는 다음과 같다.

- 고장은 하나의 결함 또는 물리적 현상, 즉, Common Cause에 기인한다.
- Common Cause에 의해 영향을 받은 *Item*들은 요구되는 기능을 수행할 수 없다.
- 다수 *Item*들의 고장은 *Item*들의 중복 구성에 의해 발생한다.
- 고장은 특정 중요한 기간/시간 내에 발생한다.
- 고장의 영향은 수행되어야하는 시스템의 주요 기능을 마비시킨다.

1.6. Some Different Definitions

시스템의 Common Cause Failure (CCF)에 대한 연구는 1988년 미국의 NUREG/CR-4780을 작성하면서 시작되었다.

위험한 사건을 예방하고 인간과 환경, 재산에 미치는 영향을 경감하기 위한 안전 제어시스템(Safety Instrumented System, SIS)의 기능안정성(Functional Safety)에 대한 요구조건을 정립하는 IEC-61508이 2000년에 공표된 이래 매우 높은 안전 방호시스템을 요구하는 산업부문들에서 Common Cause에 의한 Failure를 다루고 있으며, 다음의 표 1은 각 산업부문별 CCF의 정의를 보여준다.

Space Industry (NASA PRA guide, 2002)
The failure (or unavailable state) of more than one component due to a shared cause during the system mission.

Process Industry (IEC-61511, 2003)
Failure, which is the result of one or more events, causing failures of two or more separate channels in a multiple channel system, leading to system failure.

Nuclear Industry (NEA, 2004)
A dependent failure in which two or more component fault states exist simultaneously or within a short time interval, and are a direct result of a shared cause.

Lundteigen and Rausand (2007) - Related to Safety Instrumented Systems -
• The CCF event comprises complete failures of two or more redundant components or two or more Safety Instrumented Functions (SIFs) due to a shared cause • The multiple failures occur within the same inspection or function test interval • The CCF event may lead to failure of a single SIF or loss of several SIFs

표 1. Some Different Definitions

1.7. Common Cause Component Group (CCCG) & CCF Event

- Common Cause Component Group (CCCG) : 동일한 CCF 모드를 가지는 시스템 *Item*들의 집합을 한다.

- CCF Event : Common Cause로 인한 특정 컴포넌트들의 집합의 고장을 포함하는 Event를 의미한다.
  - CCF Event는 둘 또는 그 이상의 *Item*들의 고장을 포함한다.
  - CCF Event의 *Item*들의 고장은 동시에 혹은 짧은 시간 간격으로 발생한다.
  - *Item*들의 고장은 Common Cause에 의하여 발생하거나 발생하지 않는다.
  - CCF Event는 Common Cause Basic Event (CCBE)라고도 불린다.

1.8. CCF Modeling

CCF를 예방하기 위하여 Common Cause를 찾아내고, 분석하기 위한 CCF Modeling 과정은 다음과 같다.

1) 시스템 로직 모델을 개발한다. (Fault Tree 또는 Reliability Block Diagram)
2) 관련 있는 CCCG를 식별한다.
3) 관련 있는 Root Causes와 Coupling Factors를 식별한다.
4) CCF 예방의 효율성을 평가한다.
5) Explicit Model을 수립한다.
6) Implicit Model을 포함한다.

7) 신뢰도를 정량화하고 결과를 분석한다.

CCF Modeling은 다음과 같이 Explicit Modeling과 Implicit Modeling으로 구분할 수 있다.

- Explicit Modeling : Common Cause가 별도의 기본 Event/Element로 식별되어진다. 그림 3은 Explicit Modeling의 예를 보여준다.
  - Human Errors
  - Utility Failures(Power Failure, Cooling/Heating Failure, Loss of Hydraulic Power)
  - Shared Equipment
  - Environmental Events(Lightning, Flooding, Storm)

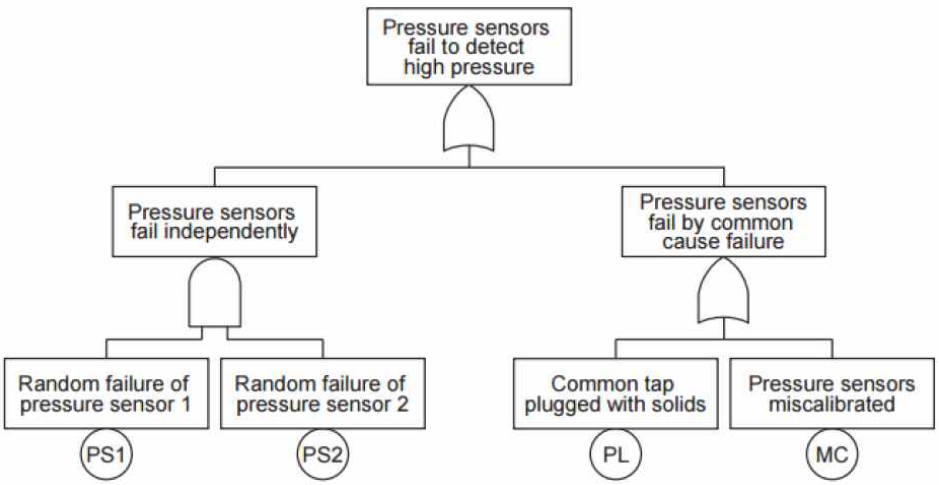

그림 3. Example of Explicit Modeling (Two Pressure Sensors)

- Implicit Modeling : *Item*들의 집합이 다수의 Root Cause와 Coupling Factor를 공유하며, Explicit Modeling으로 다루기 어려운 것들을 포함한다. Common Cause는 결합된 Event/Element로 모델링 되며, Implicit Modeling은 CCF Modeling의 사용 방식을 의미한다.
  - 2-out-of-3 System
  - β-Factor Model
  - MGL(Multiple Greek Letter) Model
  - BFR(Binominal Failure Rate) Model

1.9. Multiplicity

Multiplicity란 실제 CCF Event에 실패한 그룹의 *Item* 수를 의미하며, 다음과 같이 Complete(Lethal) Failure와 Partial(Non Lethal) Failure로 구분할 수 있다.

- Complete(Lethal) Failure : 그룹의 모든 *Item*이 고장난 경우를 의미한다.
- Partial(Non Lethal) Failure : 하나 이상의 *Item* 고장이지만, 전체 *Item*의 고장은 아닌 경우를 의미한다.

예를 들어, 컴포넌트 1, 2, 3이 있고, E_i가 컴포넌트 *i* 의 고장 상태를, E_i^*가 컴포넌트 *i* 의 정상 상태를 나타낼 때, 다음과 같이 3가지의 다른 Multiplicity를 가진다.

- 하나의 컴포넌트가 고장나는 3가지 경우

$$(E_1 \cap E_2^* \cap E_3^*),\ (E_1^* \cap E_2 \cap E_3^*),\ (E_1^* \cap E_2^* \cap E_3)$$

- 두 개의 컴포넌트가 고장나는 3가지 경우

$$(E_1 \cap E_2 \cap E_3^*),\ (E_1 \cap E_2^* \cap E_3),\ (E_1^* \cap E_2 \cap E_3)$$

- 세 개의 컴포넌트가 고장나는 경우

$$(E_1 \cap E_2 \cap E_3)$$

고장 컴포넌트와 정상 컴포넌트의 특정 조합 확률은 다음과 같이 $g_{k,m}$으로 나타낼 수 있다. (*k* : 고장 컴포넌트의 수, *m* : 전체 컴포넌트 수)

- $g_{1,3} = P(E_1 \cap E_2^* \cap E_3^*) = P(E_1^* \cap E_2 \cap E_3^*) = P(E_1^* \cap E_2^* \cap E_3)$
- $g_{2,3} = P(E_1 \cap E_2 \cap E_3^*) = P(E_1 \cap E_2^* \cap E_3) = P(E_1^* \cap E_2 \cap E_3)$
- $g_{3,3} = P(E_1 \cap E_2 \cap E_3)$

*m*개의 컴포넌트를 갖는 시스템의 CCF Event 확률은 다음과 같이 $Q_{k:m}$으로 나타 낼 수 있다. (*k* : 고장 컴포넌트의 수, *m* : 전체 컴포넌트 수)

- $Q_{1:3} = \binom{3}{1} \cdot g_{1,3} = 3 \cdot g_{1,3}$
- $Q_{2:3} = \binom{3}{2} \cdot g_{2,3} = 3 \cdot g_{2,3}$
- $Q_{3:3} = \binom{3}{3} \cdot g_{3,3} = g_{3,3}$

예를 들어, 3개의 컴포넌트 중 2개 이상의 컴포넌트가 고장 났을 때 전체 시스템이 고장에 이르는 2-out-of-3 시스템의 고장 확률은 다음과 같이 구할 수 있다.

$$P(System\,Failure) = Q_{2:3} + Q_{3:3} = 3 \cdot g_{2,3} + g_{3,3}$$

*m*개의 컴포넌트를 갖는 시스템에서 특정 한 컴포넌트가 고장 상태임을 알 때의 CCF Event 조건부 확률은 $f_{k,m}$으로 나타낼 수 있다. (*k* : 고장 컴포넌트의 수, *m* : 전체 컴포넌트 수)

예를 들어, 2-out-of-3 시스템에서 컴포넌트 1이 고장난 상태임을 알 때의 조건부 확률은 다음과 같다. *Q*는 컴포넌트 1이 고장날 확률을 의미한다.

- 세 개의 컴포넌트가 고장나는 경우 : 컴포넌트 1이 고장난 상태에서, 컴포넌트 2와 컴포넌트 3이 고장나는 경우

$$f_{3,3} = P(E_1 \cap E_2 \cap E_3 \mid E_1) = \frac{P(E_1 \cap E_2 \cap E_3)}{P(E_1)} = \frac{g_{3,3}}{Q}$$

- 두 개의 컴포넌트가 고장나는 경우 : 컴포넌트 1이 고장난 상태에서, 컴포넌트 2가 고장나는 경우 또는 컴포넌트 3이 고장나는 경우

$$f_{2,3} = \frac{g_{2,3}}{Q} + \frac{g_{2,3}}{Q} = \frac{2 \cdot g_{2,3}}{Q}$$

- 하나의 컴포넌트가 고장나는 경우 : 컴포넌트 1의 고장

$$f_{1,3} = P(E_1 \cap E_2^* \cap E_3^* \mid E_1) = \frac{g_{1,3}}{Q}$$

- $f_{1,3} + f_{2,3} + f_{3,3} = 1$

2. Types of CCF

가능한 Common Cause의 유형을 표 2와 표 3에 간략하게 나타내었다.

CCF	Some Possible Common Causes of CCF
Specification or Design Failure	Failure to Recognize within the Specification the Full Range of Circumstances in which the Plant must Operate
	Wrong/Inadequate Standards used
	Inadequate Management of Change (Control of Plant Modifications)
	Common Ageing Processes on Redundant Channels
Construction, Installation, Inspection, Commissioning failure	Poor Quality Control of Components and Sub Systems during Manufacture
	Lack of Physical/Electrical Separation during Installation
	Improper Installation
	Commissioning Testing : Failure to Test Adequately all Credible Circumstances
Maintenance or Operations Failure	Failure to Repair Defective Equipment in a Timely Manner
	Maladjustment of set Points, Limit Switches, ETC
	Improper or Inadequate Maintenance or Test Procedures
	Failure to Follow Maintenance Procedures
	Poor Control of Overrides or Interlock Defeats
	Poor Housekeeping
	Poor Quality Spare Components
Environmental Aspects	Temperature
	Humidity
	Vibration
	Stress
	Corrosion
	Contamination (Abrasive Material, Chemical Agent, ETC)
	Radio Frequency Interference (RFI)
	Radiation
	Static Charge
	Extreme Weather (Rain, Snow, Hail, Ice, Wind)
	Seismic Event, Tsunami
Other External and Internal Hazards	Fire
	Flood
	Explosion
	Air crash
	Terrorism

표 2. Some Possible Common Causes of CCF

표 3은 다음 3장에서 CCF Examples의 Common Cause를 구분하는데 사용된다.

System or Component Requirements (may Ignore Several CCF Factors)	
Wear Out (If all similar Items are old, They may be reaching the End of Life Together)	
Contamination (Foreign Object, Chemical Degradation, Internal Generated Debris, ETC)	
Corrosion (Inter-Granular, Corrosion Fatigue, Stress Corrosion Cracking)	
Environment	Weather (Ice, Rain, Winds)
	Lightning/Electromagnetic Interference
	Earthquake
	Thermal Conditions
Loss of Power	
Software (The Hardware may be redundant but the Software is the same Version)	
Saturation of Signals (under sizing the data handling system)	
Design Deficiency	
Lack of Process Control/Manufacturing Deficiency (All the items in the lot used are defective)	
Transportation / Shipping	
Human Error/System Complexity (e.g. Maintenance or Installation Errors)	
Cascading (Multi-Channel Systems with Load Sharing)	
Single Physical Point where Redundant Items Meet	

표 3. Several contributing Factors or Causes for a CCF

3. CCF Examples

3.1. Power Grid (Cascading)

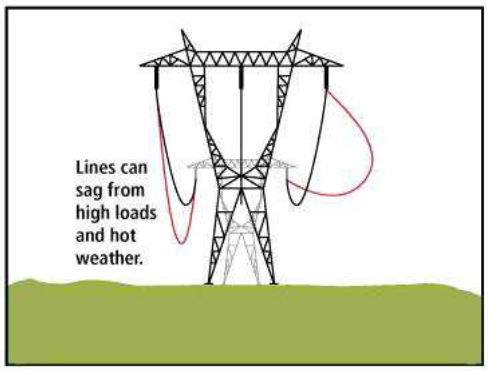

그림 4. CCF Example : Power Grid (Cascading)

더운 여름날 전력 사용량의 증가로 송전탑의 부하가 증가함에 따라 송전선의 송전량 또한 증가하게 된다. 송전량의 증가는 송전선에서 발생하는 열의 증가를 야기 시키며, 이는 송전선을 처지게 하여 결국에는 송전선의 기능을 상실하게 만든다. 하나의 송전선이 상실되면 그 송전선이 전송하던 전력을 다른 송전선을 통해 전송하기 때문에 다른 송전선의 부하를 더욱 증가시켜 연쇄적으로 송전선의 상실을 야기한다. (Cascading)

3.2. Apollo 13 Explosion (Single Physical Point)

그림 5. CCF Example : Apollo 13 Explosion (Single Physical Point)

Apollo 13호에 산소를 공급하는 산소 탱크의 고장을 대비하기 위하여 여러 개의 산소 탱크를 중복 설치하였으나, 설치 된 산소 탱크들의 위치가 인접해 있어 하나의 산소 탱크 폭발로 모든 산소 탱크의 상실을 야기하였다. (Single Physical

Point)

3.3. Airlines Flight 232 (Single Physical Point)

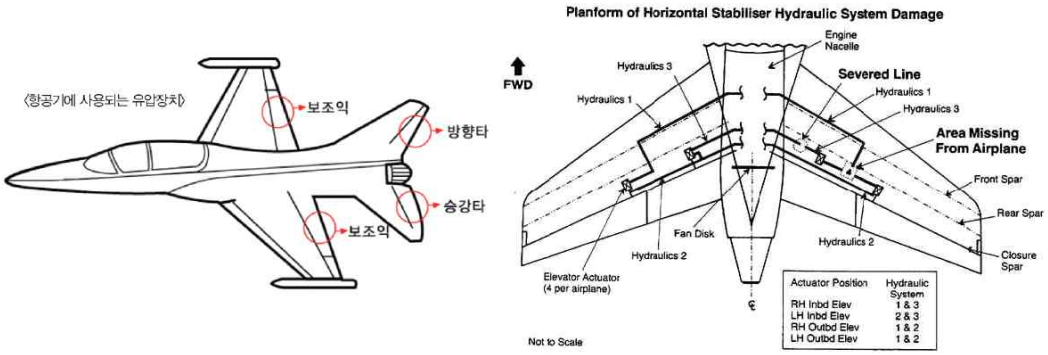

그림 6. CCF Example : Airlines Flight 232 (Single Physical Point)

비행기의 방향과 고도를 조절하는 보조익, 승강타, 방향타는 유압 시스템에 의하여 조종되어진다. McDonnell Douglas DC-10 항공기의 경우 유압 시스템의 액체가 흐르는 튜브가 하나로 연결되어 있어, 파편에 의하여 튜브가 손상됨에 따라 튜브 내의 액체가 유실되어 전체 유압 시스템의 마비를 야기하였고, 이는 비행기 조종 불가 상태에 이르게 하였다. (Single Physical Point)

3.4. Japan's Fukushima Daiichi Power Plant (Environmental)

그림 7. CCF Example : Japan's Fukushima Daiichi Power Plant (Environmental)

원자력 발전소를 가동하는 전력을 생성하기 위한 발전기의 고장은 원자로에 이상이 있을 때 원자로를 보호하는 원자로 제어 시스템의 마비를 야기한다. 일본의 후쿠시마 원자력 발전소의 경우 발전기의 고장을 대비하여 보조 발전기를 설치하였으나, 쓰나미에 의하여 발전기들이 모두 침수되어 원자로 제어시스템의 마비로

원자로 폭발이 발생하였다. (Environmental)

3.5. RAID System

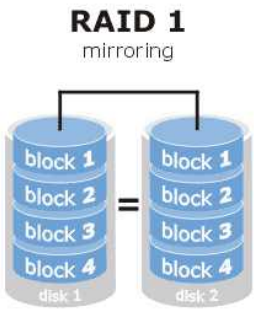

그림 8. CCF Example : RAID System

RAID 시스템을 구성하기 위하여 온라인에서 디스크를 구매하고 컴퓨터에 설치하였을 때, 다양한 Common Cause에 의한 CCF가 발생할 수 있다.

- 디스크들이 같은 제조회사의 같은 모델일 경우, 동일한 설계를 갖기 때문에 Design Deficiency에 의한 CCF가 발생할 수 있다.

- 디스크들이 유사한 시리얼 넘버를 가질 경우, 동일한 제조 공정 흐름을 갖기 때문에 Manufacturing Deficiency에 의한 CCF가 발생할 수 있다.

- 디스크들이 같이 배송되었을 경우, 배송 중 동일한 충격 때문에 Transportation/Shipping에 의한 CCF가 발생할 수 있다.

- 디스크들이 같은 파워 서플라이에 연결되어 설치되었을 경우, 동일한 파워 서플라이의 고장 때문에 Loss of Power에 의한 CCF가 발생할 수 있다.

- 디스크들이 같은 케이스에 장착되었을 경우, 동일한 과열 현상 때문에 Thermal Conditions에 의한 CCF가 발생할 수 있다.

- 디스크들이 같은 프로그램에 의하여 운용될 경우, 동일한 버그나 바이러스 때문에 Software에 의한 CCF가 발생할 수 있다.

- 디스크들이 같은 액세스 패턴, 같은 부하 등, 동일한 스트레스 때문에 Wear Out에 의한 CCF가 발생할 수 있다.

4. Examples of Reducing CCF

4.1. Environmental Control Fan (Cascading, Loss of Power)

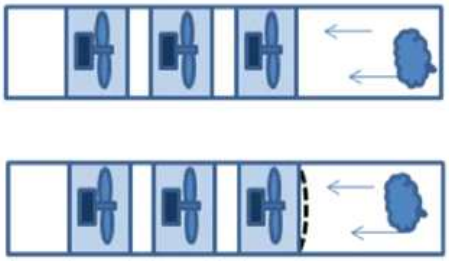

그림 9. Examples of Reducing CCF : Environmental Control Fan (Cascading) 1

그림 9와 같이 3개의 환풍기가 설치되어 있을 때, 환풍기들은 외부로부터 유입되는 먼지나 파편에 의하여 연쇄적으로 고장에 이를 수 있다. 이를 방지하기 위하여 그림 9와 같이 제일 첫 번째 환풍기에 스크린을 설치하여 이를 예방할 수 있다.

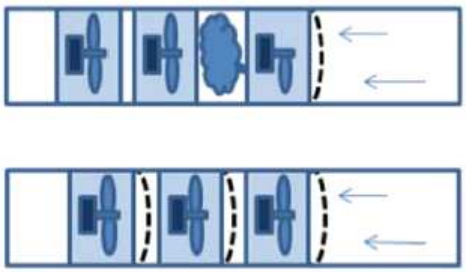

그림 10. Examples of Reducing CCF : Environmental Control Fan (Cascading) 2

하지만, 그림 10과 같이 첫 번째 환풍기의 고장으로 인해 발생한 먼지나 파편에 의하여 나머지 환풍기들이 연쇄적으로 고장에 이를 수 있다. 이를 방지하기 위하여 그림 10과 같이 모든 환풍기에 스크린을 설치하여 이를 예방할 수 있다.

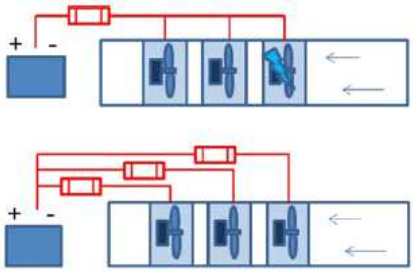

그림 11. Examples of Reducing CCF : Environmental Control Fan (Loss of Power) 1

그림 11과 같이 환풍기들이 하나의 퓨즈에 연결되어 있을 때, 하나의 환풍기의 쇼트로 인하여 퓨즈가 단절되어 모든 환풍기의 전원이 상실될 수 있다. 이를 방지하기 위하여 그림 11과 같이 모든 환풍기에 퓨즈를 설치하여 이를 예방할 수 있다.

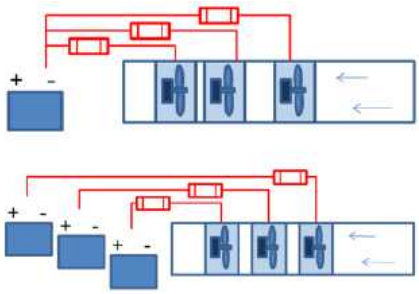

그림 12. Examples of Reducing CCF : Environmental Control Fan (Loss of Power) 2

하지만, 그림 12와 같이 환풍기들이 하나의 배터리에 연결되어 있을 때, 배터리의 고장으로 인하여 모든 환풍기의 전원이 상실될 수 있다. 이를 방지하기 위하여 그림 12와 같이 각각의 환풍기에 각각의 배터리를 연결하여 이를 예방할 수 있다.

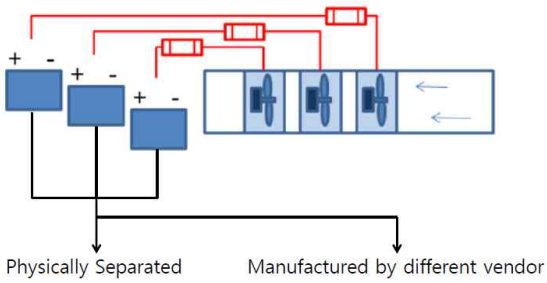

그림 13. Examples of Reducing CCF : Environmental Control Fan

하지만, 배터리가 같은 제조사에 의하여 제조되었을 경우, 같은 설계를 갖기 때

문에 CCF를 발생 시킬 수 있으며, 물리적으로 같은 공간에 위치할 경우에도 CCF를 발생 시킬 수 있다. 이를 방지하기 위하여 각각의 배터리를 다른 제조사에서 제조된 배터리를 사용하고, 물리적으로 독립된 공간에 위치시켜 CCF를 예방할 수 있다.

4.2. Closely Located Hardware Device (Single Physical Point)

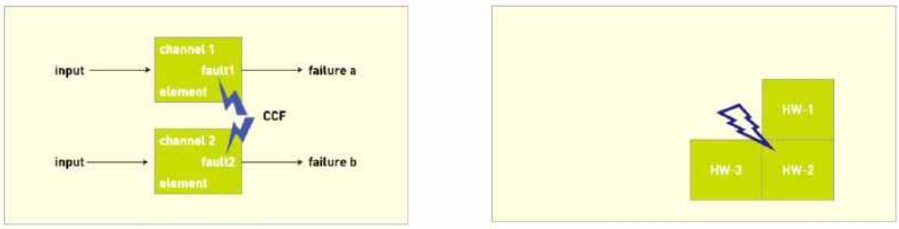

그림 14. Examples of Reducing CCF : Closely Located Hardware Device

그림 14와 같이 물리적으로 가까이 위치한 장치들은 동일한 Common Cause에 의하여 CCF가 발생할 수 있다.

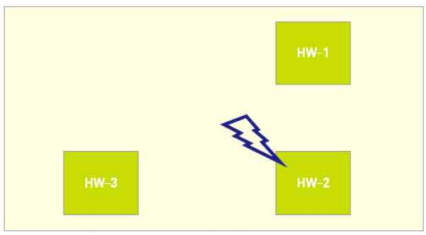

그림 15. Examples of Reducing CCF : Separately Located Hardware Device

이를 방지하기 위하여 그림 15와 같이 물리적으로 분리되도록 장치들을 위치하면 CCF를 예방할 수 있다.

4.3. Clock Tree & Clock Monitoring (Design Deficiency)

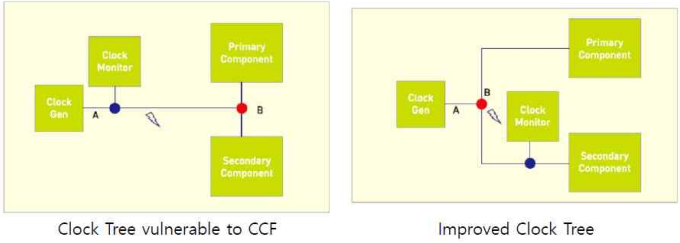

그림 16. Examples of Reducing CCF : Clock Tree & Clock Monitoring (Design Deficiency)

그림 16의 왼쪽과 같이 클록 생성기 다음에 바로 클록 모니터를 배치할 경우 (A), 클록 모니터 이후부터 컴포넌트까지의 경로 (B)에 있는 고장을 발견할 수 없다. 하지만, 그림 16의 오른쪽과 같이 클록 모니터를 컴포넌트 바로 앞에 배치하면, 경로에서 발생하는 고장을 탐지할 수 있다.

5. Analysis for Reducing CCF

5.1. Use a Common Cause Failure List (Check List, IEC-61508)

CCF를 감소시키기 위한 방법 중 하나로, 표 4와 같은 Check List를 이용하여 항목들을 하나씩 점검하면서 CCF를 감소시킬 수 있다.

Hardware	Software	ASICs and FPGAs
During design and implementation 1. Robust project management and documentation (throughout) 2. Structured specification, design 3. Observance of guidelines and standards 4. Functional testing, analysis 5. Operation and maintenance instructions, user- and maintenance-friendly 6. Interference testing 7. Fault insertion testing During operation 1. Program sequence monitoring and on-line monitoring or testing 2. Power supply monitoring and protection 3. Spatial separation 4. Ambient temperature protection 5. Modification protection	1. **Functional safety assessment**: checklists, truth tables, failure analysis, CCF analysis, reliability block diagrams 2. **Software requirements specification** – formal or semi-formal methods, traceability, software tools 3. Fault detection, error detecting codes 4. Diverse monitoring techniques 5. Recovery mechanisms or graceful degradation 6. Modular design 7. Trusted/verified software elements 8. **Forwards/backwards traceability at all stages** 9. Structured or semi-formal or formal methods, auto-code generation 10. Software tools 11. Guaranteed maximum cycle time, time-triggered architecture, maximum response time 12. Static resource allocation, synchronisation 13. Language selection, suitable tools 14. Defensive programming, modular approach, coding standards, structured programming 15. **Testing**: dynamic, functional, black box, performance, model-based, interface, probabilistic 16. Process simulation, modelling 17. **Modification/change control**: impact analysis, re-verification, revalidation, regression testing, configuration management, data recording and analysis 17. **Verification**: Formal proof, static analysis, dynamic analysis, numerical analysis	1. Structured description, VHDL design description and simulation, Boolean design description 2. Proven in use VHDL simulators and design environment 3. Functional testing on module and top levels, and embedded in system environment 4. Avoid asynchronous constructs, synchronised primary inputs 5. Design for testability; modularisation 6. Code guidelines adherence, code checker, defensive programming 7. Documentation of simulation results 8. Code inspection, walk-through 9. Validation of soft-cores 10. Internal consistency checks 11. Simulation of gate netlist to check timing constraints; static timing analysis of propagation delay 12. Verification of gate netlist 13. Check ASIC vendor requirements and constraints 14. Documentation of synthesis constraints, results and tools; use of proven in use tools and target libraries 15. Script based procedures 16. Test insertion and test pattern generation 17. Placement, routing, layout generation 18. Proven in use chip technology and manufacturing, QA, QC 19. Test coverage of manufacturing test; final verification and validation 20. Burn-in test

표 4. Common Cause Failure List (Check List, IEC-61508)

5.2. Use Diverse(Unlike) Redundancy when Possible

CCF를 감소시키기 위한 방법 중 하나로, 서로 다른 계통(방식)으로 중복 구성을 사용하는 것이다. 원자로 보호 시스템을 예로 들면 다음과 같다.

- 원자로 보호 시스템의 안전 기능 요구사항 설계는 서로 다른 개발팀에 의하여 개발되어야한다.
- 원자로 보호 시스템은 전기적으로, 물리적으로 분리되어야 한다.
- 원자로 보호 시스템은 서로 다른 센서로부터 입력 값을 받아 수행되어야한다.
- 원자로 보호 시스템의 신호는 서로 다른 경로를 통해 전달되어야하며, 서로 다른 Logic Solver에 의해 처리되어야 한다.
- 원자로 보호 시스템의 구동기는 서로 다른 제조 환경에서 제조된 것들을 사용한다.
- 원자로를 차단하기 위한 방법으로 서로 다른 물리적 원리를 사용한다.

5.3. Perform a Fault Tree Analysis (FTA)

CCF의 Common Cause를 발견하고 감소시키기 위한 방법 중 하나로, 그림 17과 같이 Common Failure 간의 관계와 경로를 파악하고 정의한다.

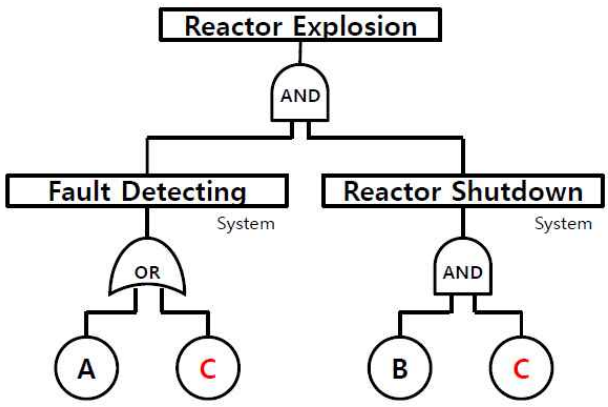

그림 17. Fault Tree Analysis (FTA)

그림 17에서 Fault Detecting 시스템은 A 또는 C 중 하나가 고장나면 고장나

게 되며, Reactor Shutdown 시스템은 B와 C 모두 고장 났을 경우 고장나게 된다. 그리고 Reactor Explosion은 Fault Detecting 시스템과 Reactor Shutdown 시스템 모두 고장 났을 경우 발생되는 최악의 상황이다. 만약 B와 C가 고장나는 상황이 발생한다면 Fault Detecting 시스템과 Reactor Shutdown 시스템 모두 고장을 일으키며, 이는 Reactor Explosion이라는 최악의 상황을 초래하게 된다. 이 때 Common Cause는 C가 되며, 이로 인한 CCF를 감소시키기 위한 방법을 찾아야 한다.

5.4. The β-Factor Model, The C-Factor Model, Others

β-Factor Model은 보편적으로 가장 많이 쓰이는 CCF Modeling의 Implicit Model 중 하나이다. β-Factor Model은 전체 고장 비율에서 특정 %가 CCF에 의한 고장이라고 가정을 하며, 전체 고장률을 λ, 독립 고장률을 λ_I, CCF에 의한 고장률을 λ_C라고 할 때, 다음과 같이 나타낼 수 있다.

$$\lambda = \lambda_I + \lambda_C$$

β-Factor Model은 전체 고장률에서 CCF에 의한 고장률의 비율로써 정의할 수 있으며 다음과 같다.

$$\beta = \frac{\lambda_C}{\lambda}, \quad \beta = P(CCF|Failure)$$

*m*개의 유사한 *Item*을 갖는 시스템에서 각각의 *Item*은 서로 다른 *Item*에게 영향을 끼치지 않는 독립적인 원인에 의하여 고장이 나거나, 다른 *Item*에게도 영향을 끼치는 원인에 의하여 고장이 날 수 있다. 이 때 β-Factor Model은 다른 *Item*에게도 영향을 끼치는 원인에 의한 고장일 경우, *m*개의 *Item*들이 모두 고장나거나 모두 고장나지 않는 경우만 고려할 수 있다. 즉 CCF Event는 1이거나 *m*일 뿐이며, CCF Event는 Intermediate Multiplicity를 가질 수 없다. 예를 들어 *m*개의 *Item*을 갖는 시스템에서 1개의 *Item*이 고장났을 경우는 다음과 같이 나타낼 수 있다. (k = 2, 3, ..., m-1)

$$f_{1,m} = 1 - \beta$$
$$f_{k,m} = 0$$

$$f_{m,m} = \beta$$

β-Factor Model은 β라는 파라미터 하나만을 이용하고 파라미터의 의미를 이해하기 쉬워 간단하고 사용하기 쉽다는 장점이 있다.

하지만, CCF를 줄이기 위한 노력은 β를 감소시키겠지만, 전체 고장률 λ는 변하지 않는 상수이기 때문에 이는 다음과 같이 독립 고장의 비율 λ_I를 증가시키게 된다.

$$\lambda_I = (1-\beta) \bullet \lambda$$

C-Factor Model은 β-Factor Model의 위와 같은 단점을 보완한 것으로, CCF에 의한 고장률 λ_C를 다음과 같이 독립 고장률 λ_I의 fraction(C)로 정의한다.

$$\lambda = \lambda_I + C \bullet \lambda_I$$

이는 β-Factor Model에서 CCF를 줄이기 위한 노력이 독립 고장의 비율 λ_I을 증가시킨다는 문제점을 고쳐, CCF를 줄이기 위한 노력이 전체 고장률 λ를 감소시킬 수 있다는 것을 의미한다.

β-Factor Model과 C-Factor Model 이외의 다른 CCF Modeling 종류는 다음과 같다.

- Basic Parameter Model
- Alpha-Factor Model
- Shock Models
  - The Multinomial Failure Rate Model
  - The Random Probability Shock Model
  - The Random Probability Shock Model
- Markov Analysis
  - The Matrix Multiplication Method
  - The Differential Equations Method

References

[1] Jim Thomson, “Common-Mode Failure Considerations in High-Integrity C&I Systems”, Safety in Engineering, 2012.

[2] Torbjorn Lilleheier, “Analysis of common cause failures in complex safety instrumented systems”, 2008.

[3] Kimberly A. Ford, Glenn Raney, “Estimation and evaluation of common cause failures in SIS”, Loss Prevention Symposium, 1999.

[4] Dana L. Kelly, Dale M. Rasmuson, “Common-Cause Failure Analysis in Event Assessment”, ANS PSA 2008 Topical Meeting, 2008.

[5] James E. Stott, Paul T. Britton, Robert W. Ring, Frank Hark, G. Spencer Hatfield, “Common Cause Failure Modeling Aerospace vs. Nuclear”, NASA Marshall Space Flight Center.

[6] “Common Cause Failure Analysis”, IAEA Workshop.

[7] Jon Wetherholt, Timothy J. Heimann, “Common Cause Failure Modes”, NASA Marshall Space Flight Center.

[8] Jeffrey Voas, Anup Ghosh, Frank Charron, Lora Kassab, “Reducing Uncertainty About Common-Mode Failures”, 1997.

[9] W.Kroger, “Inclusions of common cause failures and geographically distributed events (seismic hazard analysis)”, 2009.

[10] Marvin Rausand, “ Risk Assessment, Common Cause Failures”, RAMS Group.

[11] Fault Tree Analysis (결함수분석법)

[12] Mary Ann Lundteigen, Marvin Rausand, “Common cause failures in safety instrumented systems on oil and gas installations: Implementing defense measures through function testing”, Journal of Loss Prevention in the Process Industries, 2007.

[13] 서순근, “공통원인고장을 고려한 안전제어시스템의 신뢰성 평가척도에 관한 고찰 : IEC 61508을 중심으로”, KIIE, 2012.

[14] 김명희, 박만곤, “소프트웨어 안전성 평가를 위한 소프트웨어 고장 유형과 영향 분석에 관한 연구”, Journal of Korea Multimedia Society, 2012.

[15] 공명복, 이상용, “디지털 원자로 보호시스템의 공통원인고장 분석에 관한 사례연구”, KIIE, 2012.

[16] 손현일, 권기량, 김진오, “종속고장을 고려한 전력시스템의 신뢰도 평가”, Journal of the

Korean Institute of Illuminating and Electrical Installation Engineers, 2011.
[17] 권영민, 송진호, 박종균, “원자로 보호계통의 공통원인고장시 사고해석 방법론에 대한 고찰”, 한국원자력학회, 1996.
[18] “Safety Instrumented Systems, The Logic of Single Loop Logic Solvers”, Moore Industries, 2009.
[19] Bill Mostia Jr., PE, “The safety instrumented function”, PutmanMedia, 2003.
[20] http://elec4.co.kr/article/articleView.asp?idx=10176
[21] http://www.weibull.com/hotwire/issue54/relbasics54.htm
[22] https://iso26262fs.wordpress.com/tag/ccf/
[23] http://www.processoperations.com/SafeInstrSy/SS_Chp05.htm
[24] http://kimegoo.blog.me/120120086002