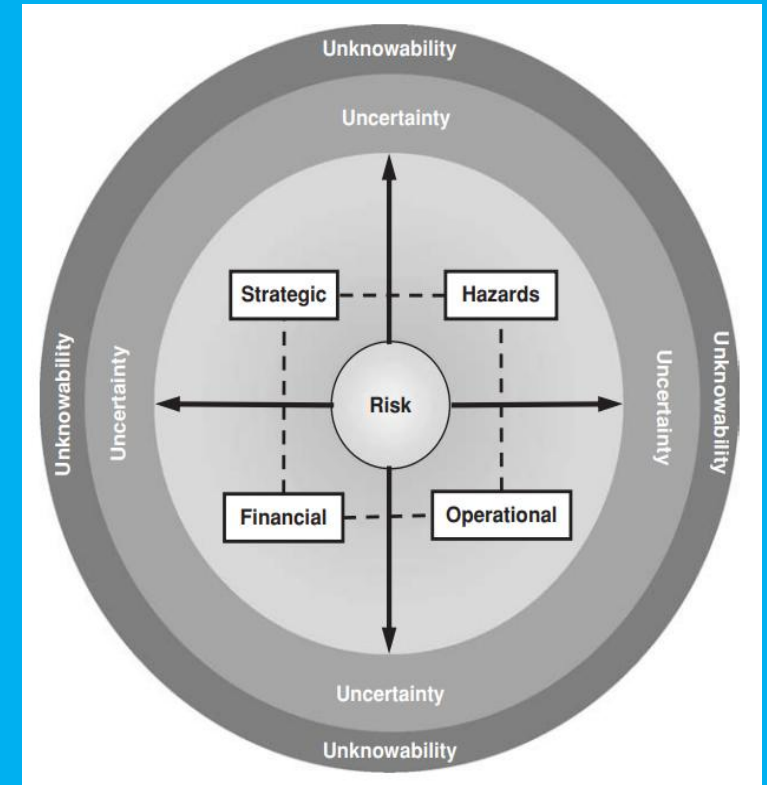


Strategic And Operational Risk Management

PMG630

Prepared by

Dr. Ayman Hassan



Lecture .6 “Tools and Techniques” fault Tree analysis,... FTA

Chapter 3

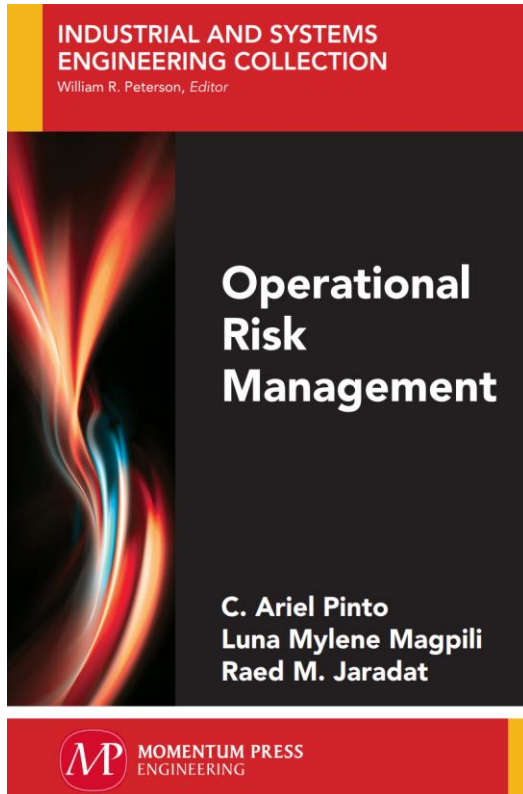
Tools and Techniques

fault Tree analysis

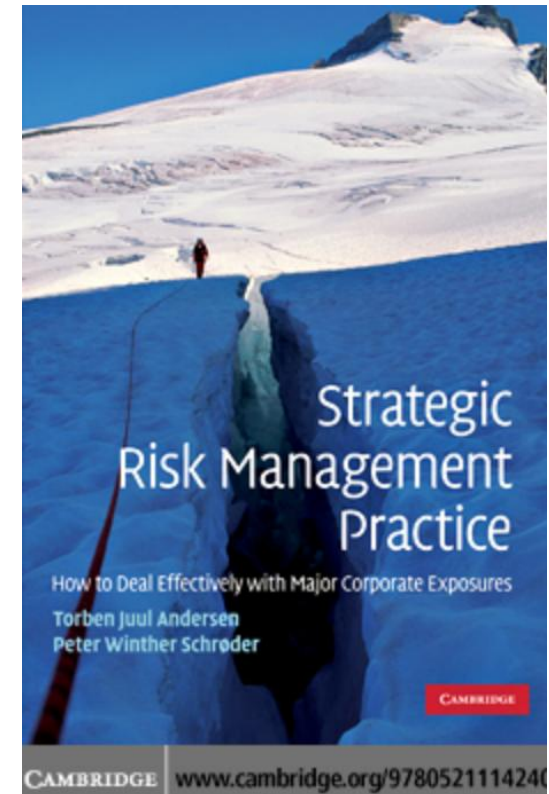
FTA

Prepared by

Dr. Ayman Hassan



REF, 1



REF,2

- 1. Summary of the previous lecture**
- 2. Introduction**
- 3. What is Fault Tree Analysis**
- 4. Where and when to use FTA**
- 5. FTA formate**
- 6. Standard Symbols in FTA**
- 7. Generalized Steps for FTA**
- 8. Example**
- 9. Summery**
- 10. 10. References & Resources**

What is fault tree analysis model

- It is a deductive top-down model technique used to analysis how a failure can occur.
- It maps out causal chain of events by connecting hazards and events that can bring about a total or partial system failure.
- Unlike other risk tools that examine single events or single hazards at a time.



What is Fault tree analysis ?

Fault Tree Analysis

- Fault tree analysis is able to analyze the relationships between more than one event together and identify combined effects.
- Hence, fault tree analysis is widely used in reliability and safety engineering to understand how systems fail.

There are 2 Types of FTA

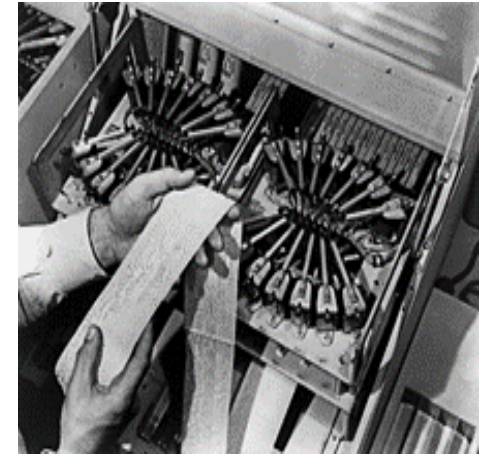
Proactive FTA

- FTA during system design development
- Improve design by mitigating weak links in the design
- Prevent undesired events and mishaps

Reactive FTA

- FTA during system operation
- Find root causes of a mishap/accident
- Modify the design to prevent future similar accidents

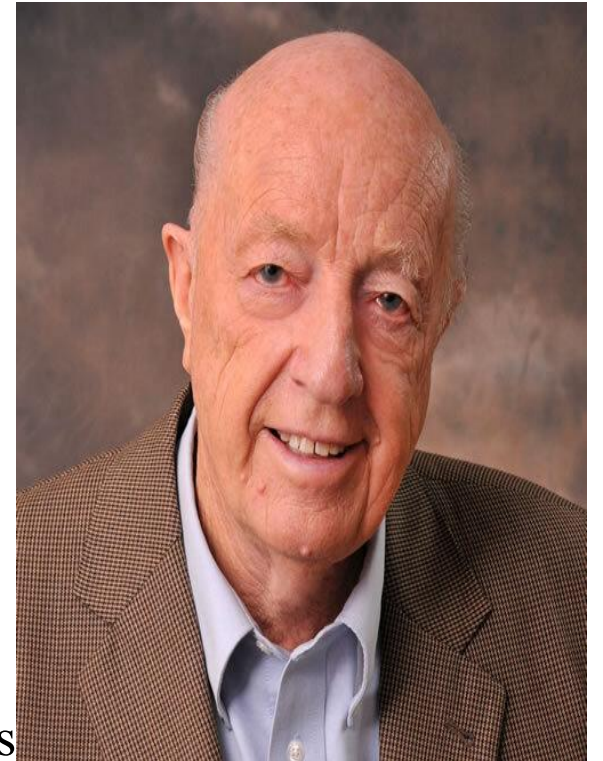
- Fault tree analysis was first used by Bell Telephone Laboratories in connection with the safety analysis of the Minuteman missile launch control system in 1962
- Technique improved by Boeing Company
- Extensively used and extended during the Reactor safety study **(WASH 1400)**
- The National Aeronautics and Space Administration (NASA) adopted fault tree technology in the 1960s



While the fault tree approach was becoming popular, John Garrick completed his Ph.D. dissertation at the University of California Los Angeles based on his work for the AEC contractor, Holmes & Narver.

He developed a fairly sophisticated fault tree approach specifically for a nuclear reactor.

Dr. Garrick's fault tree is an example of the state of the art beginning to move forward with the development of methodologies that GE did not have in the 1950s. An example of Garrick's fault trees (on page 18) shows what Dr. Wellock called "a fairly sophisticated fault tree approach for that time."



- Fault tree analysis is useful in understanding and managing larger, more complex systems.
- It can be used to demonstrate compliance with system safety and reliability requirements or support the design of a more robust system.
- It is useful for developing system or product specifications and for showing important contributors and paths to system failure.
- Fault tree analysis can extend the analysis of other risk tools such as FMEA, HAZOP and CCA.

Fault trees show graphically the interaction of failures and other events in a system. Basic events are depicted at the bottom of the fault tree and are linked via logic symbols (known as gates) to one or more of the top (TOP) events.

These TOP events represent hazards or system failure modes for which predicted reliability or availability data is required. Typical TOP events might be:

- Total loss of production
- Explosion
- Toxic emission
- Safety system unavailable

- As indicated, the fault tree begins at the end, so to speak.
- This top-down approach starts by supposing that an accident takes place.
- It then considers the possible direct causes that could lead to this accident.
 - Next it looks for the origins of these causes.
 - Finally, it looks for ways to avoid these origins and causes.
- The resulting diagram resembles a tree, thus the name.
- Fault trees can also be used to model success paths as well.
 - In this regard they are modelled with the success at the top, and the basic events are the entry-level success that put the system on the path to success.

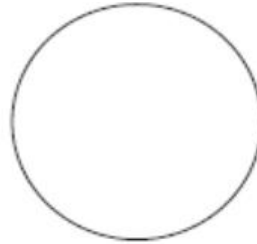
- The goal of fault tree construction is to model the system conditions that can result in the undesired event.
- The analyst must acquire a thorough understanding of the system before beginning the analysis.
- A system description should be part of the analysis documentation. The analysis must be bounded, both spatially and temporally, in order to define a beginning and endpoint for the analysis.
- The fault tree is a model that depicts graphically and logically represents the various combinations of possible events, both fault and normal, occurring in a system, leading to the TOP event.
- The term “event” denotes a dynamic change of state that occurs to a system element. System elements include hardware, software, human, and environmental factors.

- FTA uses graphical methods or diagrams to represent failure events that can occur and show connections that define pathways مسارات to system failure.
- Failure events in FTA like FMEA are described as risk events.
- Risk events can be conceived by asking the question What can go wrong?
- The top event represents the state of the system taken to be the top risk event.
- Often it refers to a state of total system failure.
- Intermediate events represent events that may lead up to the top event.
- It can refer to subsystem failures or risk events that can cause the top event to occur.

Standard Symbols in FTA

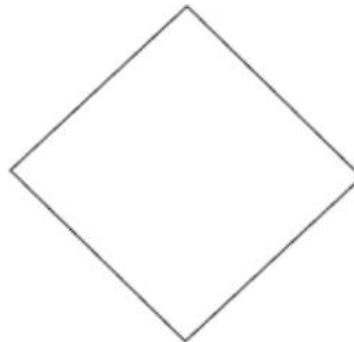
Common Fault Tree Symbols

Basic event



A basic initiating fault (or failure event)

Undeveloped event



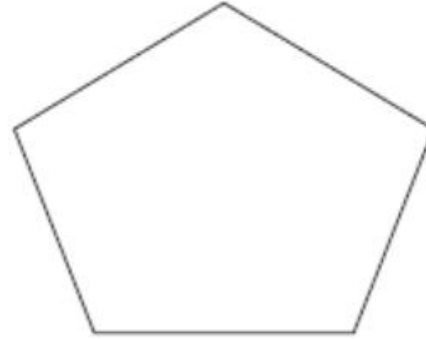
An event that is not further developed. It is a basic event that does not need further resolution

Output event



An event that is dependent on the logic of the input events

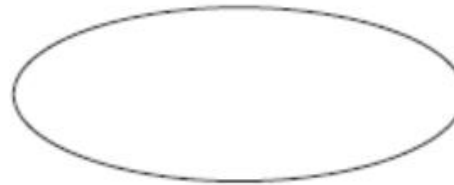
External event
(house event)



An event that is normally expected to occur

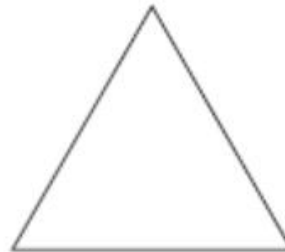
In general, these events can be set to occur or not occur, i.e. they have a fixed probability of 0 or 1

Conditioning event



A specific condition or restriction that can apply to any gate

Transfer



Indicates a transfer to a subtree or continuation to another location

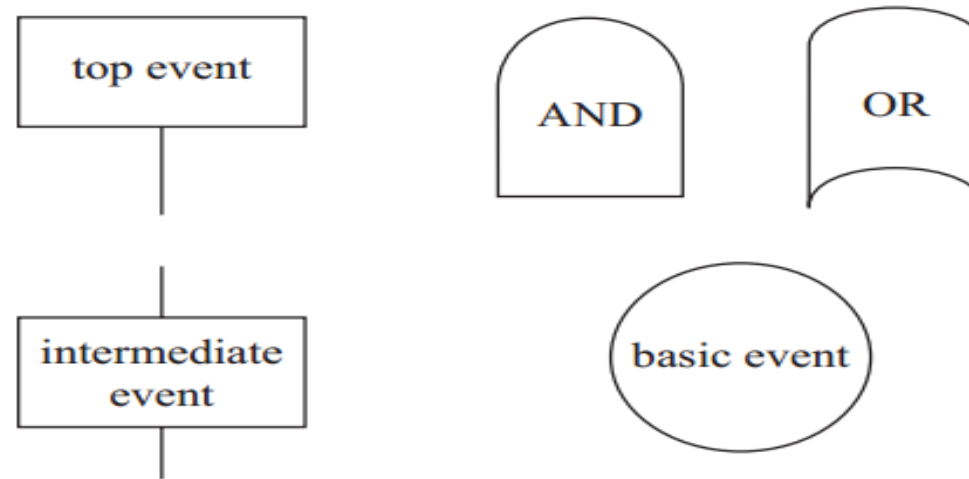
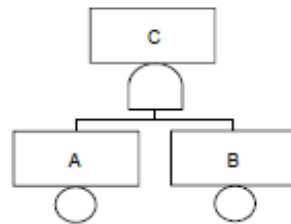


Figure 3.7. Standard symbols used in FTA.

AND Gate



Fault Tree

A	B	C
0	0	0
1	0	0
0	1	0
1	1	1

Truth Table

- Both A and B are necessary to cause C
- A and B must occur simultaneously
- Example: No power available because Primary power fails AND Secondary power fails

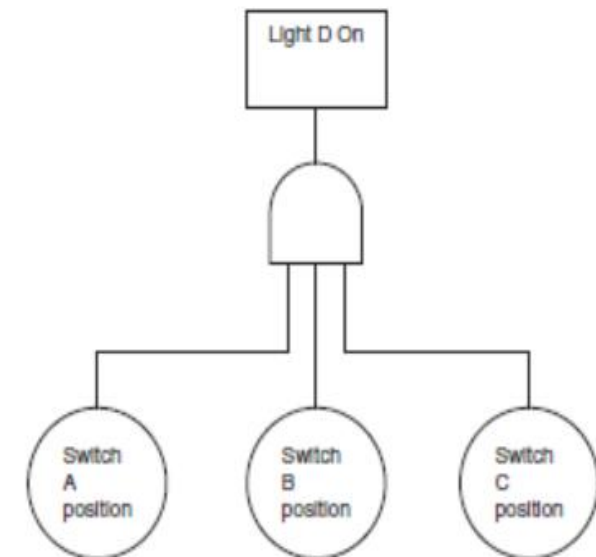
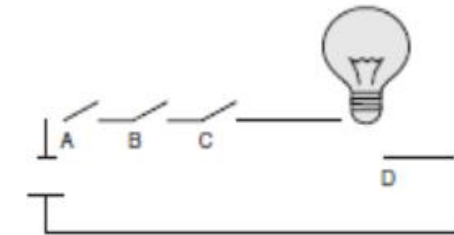
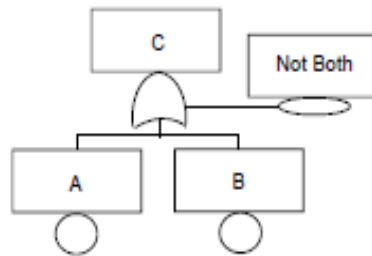


FIGURE 14.1 Switches representing AND gate.

Exclusive OR Gate



Fault Tree

A	B	C
0	0	0
1	0	1
0	1	1
1	1	0

Truth Table

- Either A or B is necessary and sufficient to cause C
- But, both A and B cannot occur together (at same time)
- Only allow two inputs (cascade down for more ExOR inputs)
- Example: Relay is energized OR Relay is de-energized, but not both

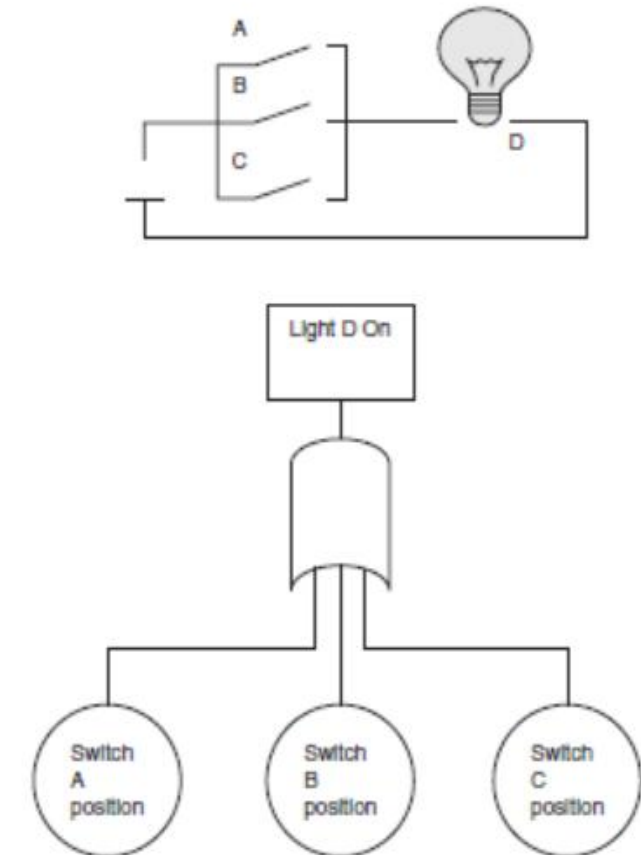


FIGURE 14.2 Switches representing OR gate.

1. Define the system to be analyzed.
2. Describe the top-level risk event (e.g., system failure).
3. Describe intermediate events (e.g., subsystems failure) that contribute to the top-level event and connect them with logic gates as appropriate.
4. Identify sublevel events that contribute to intermediate events and connect same level events with logic gates as appropriate.
5. Repeat (4) until the underlying contributory events are regarded as root causes (basic events).
6. Identify critical events and pathways to failure.

Step 1: Defining the Problem

A top event and boundary conditions must be determined when defining the problem.

Boundary conditions include:

- System physical boundaries
- Level of resolution
- Initial conditions
- Not allowed events
- Existing conditions
- Other assumptions

Top events should be precisely defined for the system being evaluated. A poorly defined top event can lead to an inefficient analysis.

Construction begins at the top event and continues, level by level, until all fault events have been broken into their basic events. Several basic rules have been developed to promote consistency and completeness in the fault tree construction process.

These rules, as listed in Table 14.4, are used to ensure systematic fault tree construction (excerpted from Guidelines for Hazard Evaluation Procedures, Center for Chemical Process Safety of the American Institute of Chemical Engineers).

1. Fault tree statements

Write the statements that are entered in the event boxes and circles as malfunctions.

State precisely a description of the component and the failure mode of the component.

The “where” and “what” portions specify the equipment and its relevant failed state.

The “why” condition describes the state of the system with respect to the equipment, thus telling why the equipment state is considered a fault.

Resist the temptation to abbreviate during construction

2. Fault event evaluation

When evaluating a fault event, ask the question “Can this fault consist of an equipment failure?”

If the answer is yes, classify the fault event as a “state-of-equipment” fault.

If the answer is no, classify the fault event as a “state-of-system” fault.

This classification aids in the continued development of the fault event

3- No miracles

If the normal functioning of equipment propagates a fault sequence, assume that the equipment functions normally. Never assume that the miraculous and totally unexpected failure of some equipment interrupts or prevents an accident from occurring

4- Complete each gate

All inputs to a particular gate should be completely defined before further analysis of any other gate. For simple models, the fault tree should be completed in levels, and each level should be completed before beginning the next level. This rule may be unwieldy when constructing a large fault tree

5- No gate to gate

Gate inputs should be properly defined fault events; that is, gates should never be directly connected to other gates.

Shortcutting the fault tree development leads to confusion because the outputs of the gate are not specified.

Many times, it is difficult to identify all of the possible combinations of failures that may lead to an accident by directly looking at the fault tree.

One method for determining these failure paths is the development of “minimal cut sets.”

Minimal cut sets are all of the combinations of failures that can result in the top event.

The cut sets are useful for ranking the ways the accident may occur and are useful for quantifying the events, if the data is available.

Large fault trees require computer analysis to derive the minimal cut sets, but some basic steps can be applied for simpler fault trees:

- Uniquely identify all gates and events in the fault tree.
- If a basic event appears more than once, it must be labeled with the same identifier each time.
- Resolve all gates into basic events. Gates are resolved by placing them in a matrix with their events.
- Remove duplicate events within each set of basic events identified.
- Delete all supersets that appear in the sets of basic events.
- By evaluating the minimal cut sets, an analyst may efficiently evaluate areas for improved system safety.

The analyst should provide a description of the system.

There should be a discussion of the problem definition, a list of the assumptions, the fault tree model(s), lists of minimal cut sets, and an evaluation of the significance of the minimal cut sets.

Any recommendations should also be presented.



The following example will show the fundamentals of FTA.

We will start with analyzing a simple cooling system flushing procedure.

we will use FTA at this point.

The procedure reads as follows:

Warning – Cooling system must be below 100 °F prior to draining

1. Begin with the engine cold and ignition off.
2. Remove the radiator pressure cap.

Example of Fault Tree Analysis.2

Open the petcock at the bottom of the radiator and drain the coolant into a bucket.

4. Close the petcock and fill the radiator with water.

5. Start the engine and turn the heater control to hot. Add cooling system cleaner and idle the engine for 30 minutes (or as per the instructions on container).

Warning – Cooling system must be below 100 °F prior to draining.

6. Stop the engine and allow it to cool for five minutes. Drain the system.

7. Close the petcock, fill the radiator with water, and let the engine idle for five minutes.

8. Open petcock and drain the water.

9. Repeat Step 6–8.

10. Close the petcock.

11. Fill cooling system with 50/50 mixture of water and nontoxic antifreeze/ coolant.

The first step will be to determine the credible top events.

In this case it will be: Mechanic is Burned - Cooling Flushing Failed

In fact, the “Mechanic is Burned” top event can be grouped under the “Cooling Flushing Failed” top event because if the mechanic were burned, then the task would fail in a sense. From the procedure we can identify several basic events. These are shown in Table 14.5, along with their credibility.

Example of Fault Tree Analysis.4

TABLE 14.5
Car radiator Failures

Failure	Description	Credible failure
Engine not below 100 °F before beginning flushing procedure	In this failure the mechanic begins the coolant draining process without ensuring the engine is cool enough	This is a credible error. It happens all the time to professional as well as amateur mechanics. Since the system is under pressure, a severe burn can occur
Failure to remove radiator cap	As it says, the mechanic fails to remove the radiator cap	This is not a credible error, unless we are modeling the fact that the mechanic does not do the task at all
Failure to drain radiator	The mechanic fails to drain the radiator	This is not a credible error, unless, again, we are modeling the fact that the mechanic does not do the task at all
Failure to close petcock valve	This failure involves the mechanic not closing or incorrectly closing the petcock valve. This error can occur at least four times in the procedure	This is a credible error and can lead to an environmental spill

Example of Fault Tree Analysis.5

Failure to add flushing agent	^{procedure} The mechanic fails to add the flushing agent	This is a credible error because the mechanic can get busy and forget where they are in the process
Failure to remove flushing agent	The mechanic fails to remove the flushing agent	This is a credible error because, once again, the mechanic can get busy and forget where they are in the process. OR a shift change occurs or job change over and the second mechanic does not know where in process they are. This happens in the airline industry every day
Failure to rinse engine	The mechanic fails to rinse the remaining flushing agent from the engine	This might not be a catastrophic error, once the engine is drained of the flushing agent. It depends on how corrosive the flushing agent is
Failure to fill engine with 50/50 nontoxic antifreeze mix	The mechanic fails to fill the cooling system with the proper mixture, the right amount of coolant, or coolant at all	This again is a credible error

Once done assessing the credibility of the failures, we next construct the fault tree.

We will not include errors that were deemed to be credible. Our top event will be the Cooling Flushing Failed.

All the basic events will be entry points into the tree.

The tree is shown in Figure 14.3. as following

construct the fault tree

لاحظ أننا لم نتمكن من بناء شجرة الأخطاء إلا باستخدام بوابة OR والأحداث الأساسية. وذلك لأن حالة واحدة فقط من حالات الفشل الموثوقة يمكن أن تؤدي إلى الفشل من المهمة.

- ❑ Notice that we were able to build the fault tree only using an OR gate and the basic events.
- ❖ That is because only one of the credible failures can lead to the failure of the task.

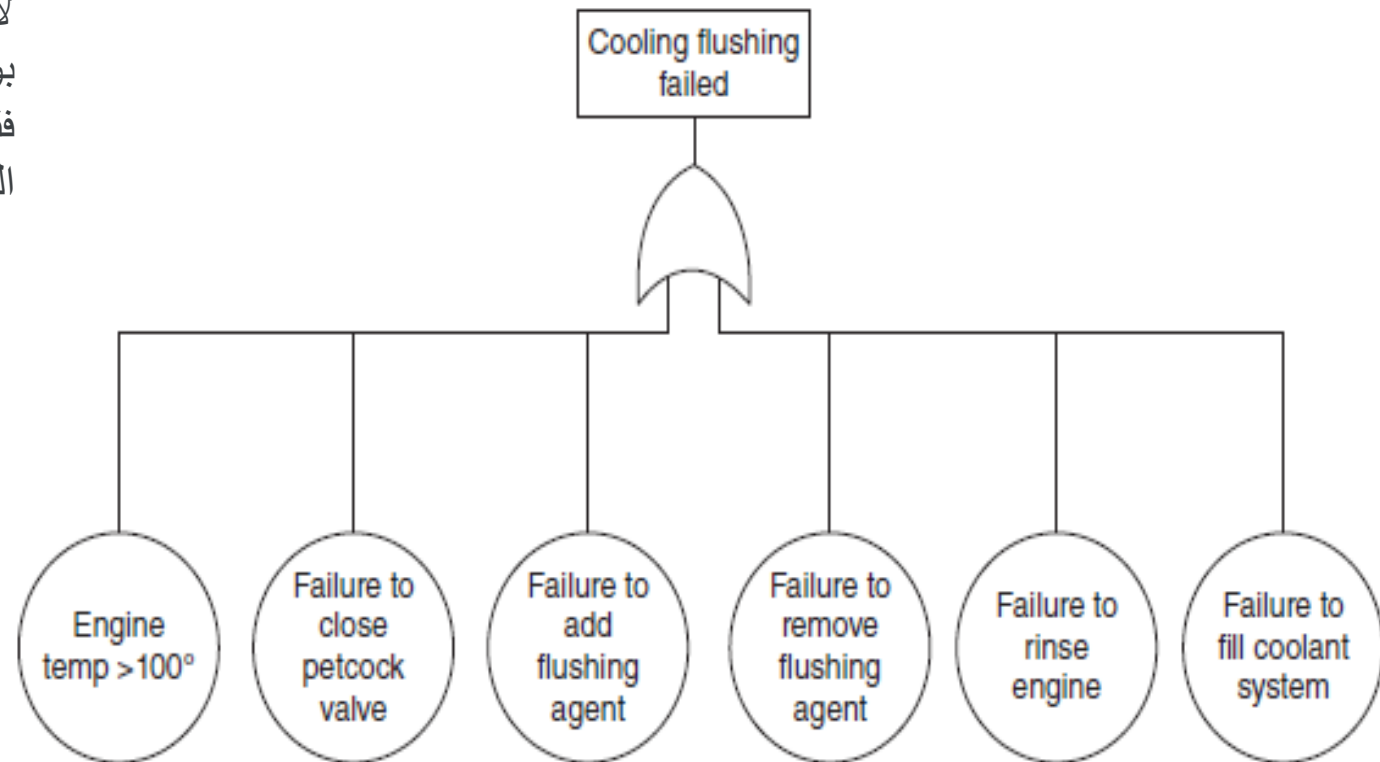


FIGURE 14.3 Fault tree analysis of coolant flushing task.

Hardware

- ☐ System level
- ☐ Subsystem level
- ☐ Component level
- ☐ Environmental effects

- ☐ Software
- ☐ System level control
- ☐ Hardware/software interface

- ☐ System Events
- ☐ Failures Events
- ☐ Normal Events
- ☐ Environmental Events

- ☐ Human Interaction
- ☐ Human error
- ☐ Human performance
- ☐ Organizational structures

- ☐ Procedures
- ☐ Operation, maintenance, assembly

- ✓ Visual model -- cause/effect relationships
- ✓ Easy to learn, do and follow
- ✓ Models complex system relationships in an understandable manner
- ✓ Follows paths across system boundaries
- ✓ Combines hardware, software, environment and human interaction
- ✓ Interface analysis - contractors, subsystems
- ✓ Probability model
- ✓ Scientifically sound
- ✓ Boolean Algebra, Logic, Probability, Reliability
- ✓ Physics, Chemistry and Engineering
- ✓ Commercial software is available
- ✓ FT's can provide value despite incomplete information
- ✓ Proven Technique

All failures are faults, but not all faults are failures

كل فشل خطأ ، لكن ليس كل خطأ فشل

Failure **فشل**

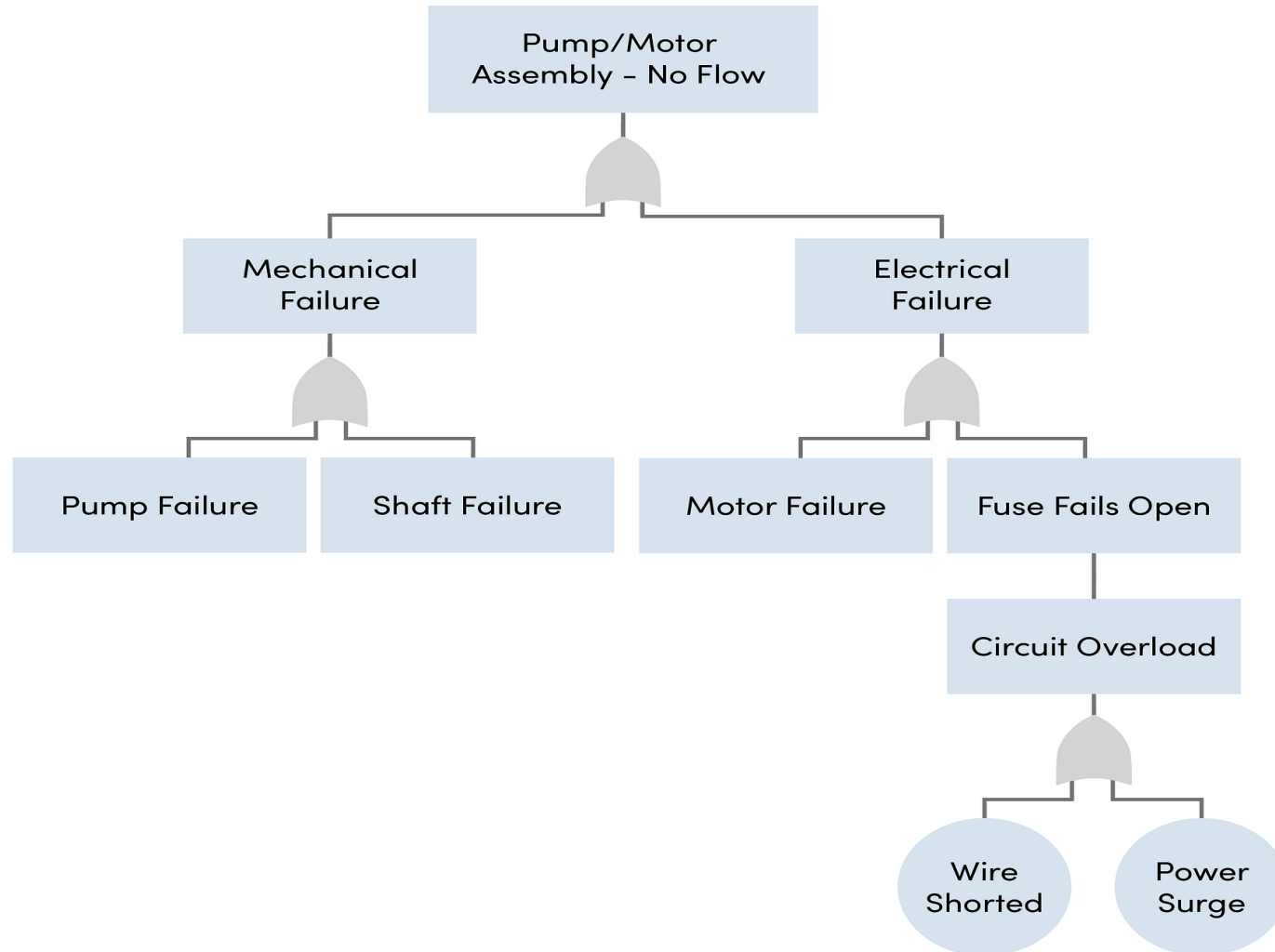
- The occurrence of a basic component failure.
- The result of an internal inherent failure mechanism, thereby requiring no further breakdown.
- Example - Resistor R77 Fails in the Open Circuit Mode.

Fault **خطأ**

- The occurrence or existence of an undesired state for a component, subsystem or system.
- The result of a failure or chain of faults/failures; can be further broken down.
- The component operates correctly, except at the wrong time, because it was commanded to do so.
- Example – The light is failed off because the switch failed open, thereby removing power.

--- FT Construction Process ---

- Tree is developed in:
 - Layers
 - Levels
 - Branches
- Tree Levels:
 - Top Level
 - ◆ Defines the top in terms of discrete system functions that can cause the top UE
 - ◆ Shapes the overall structure of the tree
 - Intermediate Level
 - ◆ Defines the logical relationships between system functions and component behavior
 - ◆ Function – systems – subsystems – modules - components
 - Bottom Level
 - ◆ Consists of the Basic Events or component failure modes

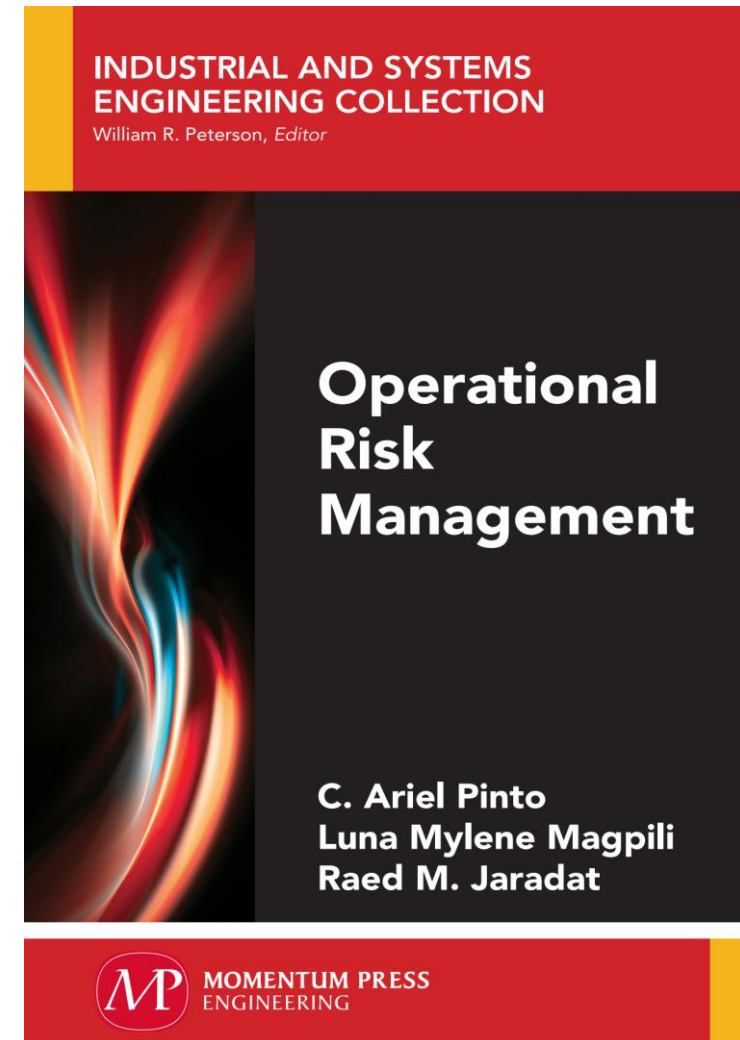


- The FTA technique has proven to be a very rigorous and valuable tool for analyzing complex systems.
- Strengths include an ability to analyze down to a great level of detail, a simple presentation, and a systematic method for analysis.
- Fault trees are used in a variety of disciplines and can model many types of systems.
- FTA is used in conjunction with other techniques to analyze complex systems.

❑ [Causal Chain - Tool/Concept/Definition \(thwink.org\)](http://thwink.org)

- Clifton A. Ericson II
- Design Safety Solutions LLC
- cliftonericson@cs.com

- Risk Assessment: Tools, Techniques, and Their Applications, Second Edition.
Lee T. Ostrom and Cheryl A. Wilhelmsen. © 2019 John Wiley & Sons, Inc.
Published 2019 by John Wiley & Sons, Inc.
- Companion website: www.wiley.com/go/Ostrom/RiskAssessment_2e





كل التوفيق لكم

My Best Wishes ...
* Dr. Ayman Hassan *

