Improvising the Security of Software Application by the Use of Fault Tree Analysis in Decision Making

SamithaKhaiyum
Senior Lecturer and Research Scholar, MCA
(VTU), DSCE, Bangalore-78

Y S Kumaraswamy,
Sr Prof and Head, MCA (VTU),
DSCE, Bangalore-78

ABSTRACT
Fault Tree Analysis (FTA) attempts to model and analyse failure processes of engineering and biological systems. FTA is basically composed of logic diagrams that display the state of the system and is constructed using graphical design techniques. Originally, engineers were responsible for the development of Fault Tree Analysis, as a deep knowledge of the system under analysis is required. Fault Tree Analysis usually involves events from hardware wear out, material failure or malfunctions or combinations of deterministic contributions to the event stemming from assigning a hardware/system failure rate to branches or cut sets. Typically failure rates are carefully derived from substantiated historical data such as mean time between failure of the components, unit, subsystem or function. Predictor data may be assigned. Assigning a software failure rate is elusive and not possible. Since software is a vital contributor and inclusive of the system operation it is assumed the software will function normally as intended. There is no such thing as a software fault tree unless considered in the system context. Software is an instruction set to the hardware or overall system for correct operation. Since basic software events do not fail in the physical sense, attempting to predict manifestation of software faults or coding errors with any reliability or accuracy is impossible, unless assumptions are made. Predicting and assigning human error rates is not the primary intent of a fault tree analysis, but may be attempted to gain some knowledge of what happens with improper human input or intervention at the wrong time.
FTA can be used as a valuable design tool, can identify potential accidents, and can eliminate costly design changes. It can also be used as a diagnostic tool, predicting the most likely system failure in a system breakdown. FTA is used in safety and reliability engineering and in all major fields of engineering.
This paper aims to provide an overview of some major uses of FTA and elaborates an appreciation of the breadth of applications of FTA in decision-making by considering an example of improvising the security of software application by the use of controlled access.

Keywords
Fault tree, risk assessment, faults, prioritisation, decision making

1. INTRODUCTION
FTA is nothing but an analytical technique, wherein an undesired state of the system is specified which generally refers to a state that is found critical from a safety or reliability point of view, and the system will then analyse the context of its environment along with its operation only to list all realistic ways for the occurrence of the desired event (top event). The fault tree is a graphic model of different parallel and sequential combinations of faults that lead to the occurrence of the predefined undesired event. The faults can be events that may be associated with component hardware failures, human errors, software errors, or any other pertinent events which can lead to the undesired event. A fault tree thus depicts the logical interrelationships of basic events that lead to the undesired event, the top event of the fault tree.
It is important to understand that a fault tree is not a model of all possible system failures or all possible causes for system failure. A fault tree is tailored to its top event that corresponds to some particular system failure mode, and the fault tree thus includes only those faults that contribute to this top event. Moreover, these faults are not exhaustive—they cover only the faults that are assessed to be realistic by the analyst.
It can also be noted that a fault tree is not in itself a quantitative model. It is a qualitative model that can be evaluated quantitatively. This qualitative aspect is true of virtually all possible varieties of system models. Since the fault tree is a particularly convenient model to quantify, it does not change the qualitative nature of the model itself.
Intrinsic to a fault tree is the concept that an outcome is a binary event i.e., to either success or failure. A fault tree is a complex composed of entities known as “gates” that permit or inhibit the passage of fault logic up the tree. The gates show the relationships of events needed for the occurrence of a “higher” event. The “higher” event is the output of the gate; the “lower” events are the “inputs” to the gate. The gate symbol denotes the type of relationship of the input events required for the output event.

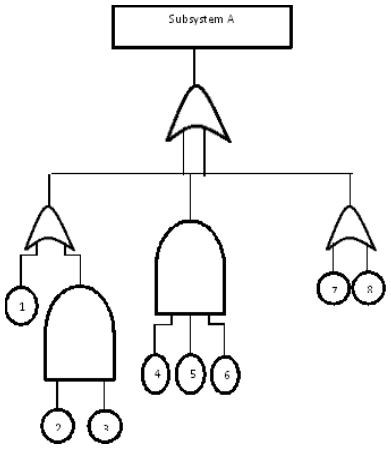

Figure 1 A simple fault tree

2. FTA ANALYSIS INVOLVES FIVE STEPS:

2.1.Definition of the undesired event to be studied

It is very hard to define the undesired event, even though some of the events are very easy and obvious to observe. It's only an engineer with a vast knowledge of the design of the system or any system analyst with an engineering background is the best person who can help define and number of the undesired events. Undesired events are used then to make the FTA, one event for one FTA; no two events will be used to make one FTA.

2.2.Understanding of the system

Once selection of the undesired event is done, all causes with probabilities of affecting the undesired event of 0 or more are studied and analysed. Getting exact numbers for the probabilities leading to the event is usually impossible for the reason that it may be very costly and time consuming to do so. Computer software is then used to study probabilities; this generally leads to a less costly system analysis.

System analysts can help with understanding the overall system. System designers have full knowledge of the system and this knowledge is very important for not missing any cause affecting the undesired event. For the selected event all causes are then numbered and sequenced in the order of occurrence and then are used for the next step which is drawing or constructing the fault tree.

2.3.Construction of the fault tree

On selection of the undesired event and after analysis of the system we know all the causing effects and possibly with their probabilities, a fault tree can now be constructed. Fault tree is based on AND and OR gates which define the major characteristics of the fault tree.

2.4.Evaluation of the fault tree

After the fault tree has been assembled for a specific undesired event, it is evaluated and analysed for any possible improvement. We then study the risk management and find ways for system improvement. This step is as an introduction for the final step which will be to control the hazards identified. In this step we basically identify all possible hazards affecting in a direct or indirect way the system.

2.5. Control the hazards identified

This step is very specific and differs largely from one system to another, but the most important point will always be that after identifying the hazards all possible methods are pursued to decrease the probability of occurrence.

3. ROLE OF FTA IN DECISION MAKING:

A variety of information is provided by FTA to assist decision-making. Some of the major uses of FTA are given here to draw an appreciation of the breadth of applications of FTA in decision-making.

3.1. Use of FTA to understand of the logic leading to the top event.

FTA provides a visual, logic model of the basic causes and intermediate events leading to the top event. Typically, fault trees are not limited to a single system, but cross system boundaries. Because of this, they have shown great benefit in identifying system interactions that impact redundancy. The combination of failures and events that propagate through a system are clearly shown. The minimal cut sets can be organized and prioritized according to the number of events involved and their nature. For example, if there are minimal cut sets that contain only one component failure then this shows that single component failures can cause failure of the system. A failure path of only human errors shows that human errors alone can cause system failure.

3.2. Use of FTA to prioritize the contributors leading to the top event. One of the most important types of information from FTA is the prioritization of the contributors to the top event. If a FT is quantified, the failures and basic events that are the causes of the top events can be prioritized according to their importance. In addition, the intermediate faults and events leading to the top event can also be prioritized. Different prioritizations and different importance measures are produced for different applications. One of the valuable conclusions from FTAs is that generally only a few contributors are important to the top event. Often only 10% to 20% of the basic events contribute significantly to the top event probability. Moreover, the contributors often cluster in distinct groupings whose importance differs by orders of magnitude.

The prioritizations obtained from FTA can provide an important basis for prioritizing resources and costs. Significant reductions in resource expenditures can be achieved with no impact to the system failure probability. For a given resource expenditure, the system failure probability can be minimized by allocating resources to be consistent with contributor importance. The importance measures obtained from a FTA are as important as the top event probability or the ranked cut set lists obtained from the analysis.

3.3. Use of FTA as a proactive tool to prevent the top event.

FTA is often used to identify vulnerable areas in a system. These vulnerable areas can be corrected or improved before the top event occurs. Upgrades to the system can be objectively evaluated for their benefits in reducing the probability of the top event. The evaluation of upgrades is an important use of the FTA. Advocates of different corrective measures and upgrades will often claim that what they are proposing provides the most benefit and they may be correct from their local perspective. However, FTA is a unique tool that provides a global perspective through a systematic and objective measure of the impact of a benefit on the top event. The probability of the top event can be used to determine the criticality of carrying out the upgrades. The probability of the to prevent can be compared to acceptability criteria or can be used in cost benefit evaluations. Advances in cost benefit methodology allow uncertainties and risk aversion to be incorporated as well as the probabilities. Furthermore, success paths provided from FTA can be used to identify specific measures that will prevent the to prevent. The proactive use of FTA has been shown to be one of its most beneficial uses.

3.4. Use of FTA to monitor the performance of the system.

The use of the FT as a monitoring tool is a specific proactive use that has been identified because of its special features. When monitoring performance with regard to the top event,

FTA can account for updates in the basic event data as well as for trending and time dependent behaviours, including aging effects. Using systematic updating techniques, the fault tree can be re-evaluated with new information that can include information on defects and near failures. Actions can then be identified to maintain or replace necessary equipment to control the failure probability and risk. This use of FTA as a monitoring tool is common in the nuclear industry.

3.5. Use of FTA to minimize and optimize resources.

This particular use of FTA is sometimes overlooked but it is one of the most important uses. Through its various importance measures, a FTA identifies not only what is important but also what is unimportant. For those contributors that are unimportant and have negligible impact on the top event, resources can be relaxed with negligible impact on the top event probability. In fact, using formal allocation approaches, resources can be re-allocated to result in the same system failure probability while reducing overall resource expenditures by significant amounts. In various applications, FTA has been used to reduce resource burdens by as much as 40% without impacting the occurrence probability of the top event. Software has been developed to help carry out these resource re-allocations for large systems.

3.6. Use of FTA to assist in designing a system.

When designing a system, FTA can be used to evaluate design alternatives and to establish performance-based design requirements. In using FTA to establish design requirements, performance requirements are defined and the FTA is used to determine the design alternatives that satisfy the performance requirements. Even though system specific data are not available, generic or heritage data can be used to bracket performance.

3.7. Use of FTA as a diagnostic tool to identify and correct causes of the top event.

This use of FTA as a diagnostic tool is different from the proactive and preventative uses described above. FTA can be used as a diagnostic tool when the top event or an intermediate event in the fault tree has occurred. When not obvious, the likely cause or causes of the top event can be determined more efficiently using the FTA power to prioritize contributors. The chain of events leading to the top event is identified in the fault tree, providing valuable information on what may have failed and the areas in which improved mitigation could be incorporated. When alternative corrective measures are identified, FTA can be used to objectively evaluate their impacts on the top event re-occurrence. FTA can also be an important aid to contingency analysis by identifying the most effective actions to be taken to reduce the impact of a fault or failure. In this case, components are set to a failed condition in the fault tree and actions are identified to minimize the impact of the failures. This contingency analysis application is often used to identify how to reconfigure a system to minimize the impact of the component failures. Allowed downtimes and repair times can also be determined to control the risk incurred from a component failure.

As can be seen from the above, FTA has a wide variety of uses and roles it can play in decision-making. FTA can be used throughout the life cycle of the system from design through system implementation and improvement. As the system proceeds to the end of life, its performance can be monitored to identify trends before failure occurs. When consciously used to assist decision-making, the payoffs from FTA generally far outweigh the resources expended performing the analysis.

4. ADVANTAGES OF APPLYING FTA

Let us see how FTA can be advantageous to software projects. We can summarise three possible points:

Value addition: FTA is generally used to exploit its potential to serve as a defect-prevention tool. Valuable information on application failures and their mechanisms can be obtained if FTA is performed before base lining the design. This information can be utilized to improve the design by preventing the potential defects or even by introducing fault-tolerating abilities. FTA is proved to be most effective for more complex functions but generally is found not to add much value when applied to simple functions in any software applications.

Simplicity: Minimum training is required to prepare FTA as it is very simple .Its graphical presentation improves readability and makes it easy to maintain any event of changes.

Traceability: Traceability could be added to FTA by appropriately identifying the individual scenario as some of the conventional test case tools provide a unique identification to individual test cases.

5. AN FTA CASE STUDY

Let us consider an example of improvising the security of software application by the use of controlled access. Choosing the login name or password of our choice may result in weaker application security. Below given figure, figure2 illustrates how this is represented.

The user ID and the password are considered here to see what could lead to a defect, i.e., in case of poor security. The short length, non-use of digits or special characters, and validity not bounded by time, etc., could be reasons to have a weak password. Same factors can apply ot login names as well.

Each scenario is identified with a unique number to establish traceability. Such traceability helps test cases to be related to other project phases like requirements, design or program specifications. The valid and invalid conditions for these scenarios are given as a reference during testing.

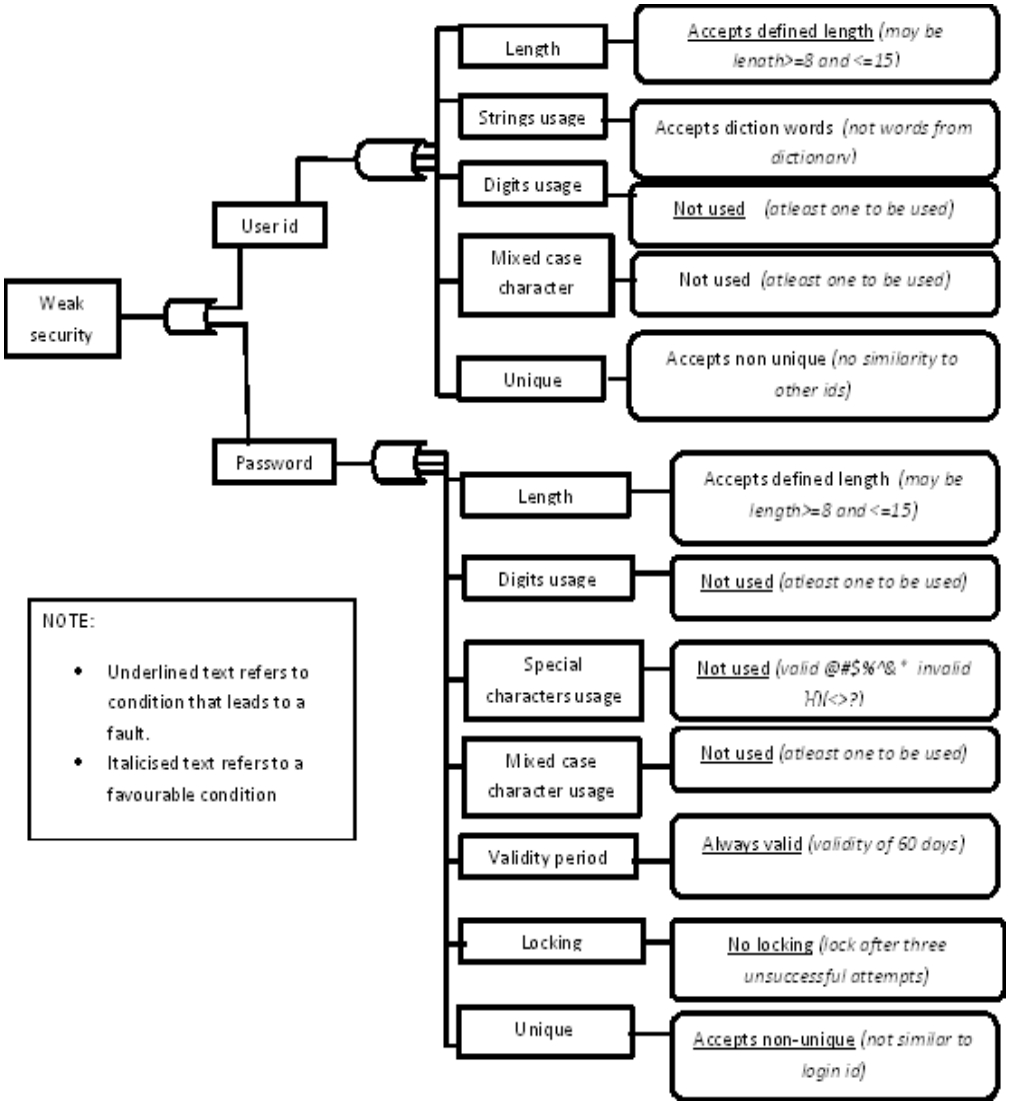

Figure 2: FTA to improvise the security of software application by the use of controlled access

6. CONCLUSION

FTA mainly has its applications in Root cause analysis where it identifies the relevant events and conditions that generally lead to an undesired event; it also determines parallel and sequential event combinations and also models diverse as well as complex event interrelations that are involved.

FTA also focuses on risk assessment to calculate the probability of an undesired event where it helps to identify the level of risk. It identifies the safety critical components, functional and phases in the system and also helps to measure the effect of various changes in the design.

FTA also helps to design safety assessment by demonstrating the compliance with its requirement; it shows where the safety requirements are needed. It helps to identify and evaluate the potential design defects and also helps to determine the common mode failures.

7. REFERENCES

[1] Fault Tree Analysis for Software Design Massood Towhidnejad, Dolores R Wallace, Albert M Gallo, Nasa Goddard, Space Flight in Engineering (2003)

[2] Fault Tree Handbook with Aerospace ApplicationsNasa Office, Mission Assurance, Nasa Headquarters in Director (2002)

[3] Fault Tree Analysis, Mark W Averettin Risk Analysis (1988)

[4] A Fault-Tree Semantics to model Software-Controlled Systems Bernhard KaiserHasso-Plattner-Institute for Software Systems Engineering at the University of Potsdam,Dept. for Software Engineering and Quality Management, Prof.-Dr.-Helmert-Str. 2-3, Potsdam

[5] Fault Trees by NikolaosLimnios, *University of Technology of Compiègne*, France ISBN: 9781905209309 pages 49-63

[6] Fault Tree Handbook by William E. Vesely, N. H. Roberts

[7] Fault tree handbook by NUREG-0492,1981, N H Roberts, W E Vesely, D F Haasl & FF Goldberg 1981

[8] System Reliability Theory (2nd ed), Wiley, 2004 ,Marvin Rausand

[9] Condition-based fault tree analysis (CBFTA): A new method for improved fault tree analysis (FTA), reliability

and safety calculations [An article from: Reliability Engineering and System Safety]

[10] Reliability & Risk assessment, Longman Scientific & Technical 1993, J D Andrews & T R Moss

[11] hq.nasa.gov/office/codeq/doctree/fthb.pdf

[12] weibull.com/basics/fault-tree/index.htm

[13] Fault tree analysis by Colin S. Howat

[14] fault-tree.net/papers/clemens-fta-tutorial.pdf

[15] fault-tree.net/papers/andrews-fta-tutor.pdf

[16] fault-tree.net/papers/ericson-fta-tutorial.pdf

[17] isograph-software.com/ftpover.htm