Middle-East Journal of Scientific Research
24 (Special Issue on Innovations in Information, Embedded and Communication Systems): 90-97, 2016
ISSN 1990-9233; © IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.IIECS.23145

A Survey on Different Software Safety Hazard Analysis and Techniques in Safety Critical Systems

[1]S. Ravikumar and [2]Chandrasekaran Subramaniam

[1]Department of InformationTechnology,
Valliammai Engineering College, Anna University, Chennai, India
[2]Department of Computer Science and Engineering,
Sri Ranganathar Institute of Engineering and Technology, Anna University, Chennai, India

Abstract: Software safety plays a significant role in safety critical system in various domains like aircraft flight control, nuclear system, medical systems and driver vehicle assistant in automobile. The software for safety critical system must deal with hazard analysis to make the software risk free and fail safe. Hazard is a state or a situation that causes threat to life which will leads to an accident. Hazard analysis identifies the hazards in the system life cycle to make the system safe. Safety is a subset of reliability or a subset of security, safety and security are closely related. The important difference between safety and security is that security focuses on malicious action, safety concerned with well-intended action. This paper presents a review of various software safety hazard analysis like fault tree analysis, event tree analysis, cause-consequence analysis, hazards and operability analysis, failure mode effect analysis and fault hazard analysis in safety critical systems.

Key words: Hazard Analysis · Safety Critical Systems · Software Safety · Hazard Evaluation · Software Faults

INTRODUCTION

The software plays an important role in many safety critical applications like automotive, avionics, power system, medical systems and sensor network [1]. The safety critical system has the potential to cause hazard that leads to an accident and it cause injury or loss to life of human being, so the software in the safety critical system should be hazard and fault free [2]. Software is hazardous and the hazard one software component will affect other software components in integrated embedded environment that leads to a catastrophic effect of failure of system components. In order to avoid loss or injury safety critical system requires utmost care in their requirement specification, design, implementation, operation and maintenance [3]. The system safety program shall provide a methodically approach to control and evaluate the safety aspects. The safety aspects are to identify hazard and prescribe corrective action in a timely and cost effective manner. Risk is the hazard level combined with the likelihood of the hazard leading to an accident and hazard exposure or duration. The system safety program provides a formal plan and it ensures the objective like safety, consistent with mission requirements, is designed in to the system. Hazards are associated with each system, subsystem and equipment's are identified and evaluated and eliminated or controlled to an acceptable level. Control over hazards that cannot be eliminated is established to protect personnel, equipment and property. Minimum risk is involved in the acceptance and use of new material and new production and testing techniques [4]. The basic concepts of system safety assure that system safety emphasis building in safety, not adding it on to a completed design. Safety systems deal with system as a whole rather than with subsystems or components. System safety emphasizes analysis rather than past experience and standards and it focuses on qualitative rather than quantitative approaches [5]. Hazard analysis for system safety involves four stages like preliminary hazard analysis, system hazard analysis, subsystem hazard analysis and operating and support hazard analysis to investigate factor related to accident. Once the hazard has been identified priority should be assigned to

Corresponding Author: S. Ravikumar, Department of InformationTechnology, Valliammai Engineering College, Anna University, Chennai, India.

eliminate them and it involves hazard elimination, hazard reduction, hazard control, damage reduction. If a hazard cannot be completely eliminated, the next best choice is to prevent or minimize its occurrence by designing in hazard reduction measures, such as lockouts, lockins, interlocks to prevent or minimize condition that could lead to the hazard. In this paper various hazard analysis models and techniques like fault tree analysis, event tree analysis, cause-consequence analysis, fault hazard analysis, state machine hazard analysis and their limitations were discussed [6]. The software hazard and requirement analysis, design for safety, verification of safety of the critical system are also reviewed in this paper.

Terminology: To begin for the purpose of this paper, we define the terms Failure and Error, Accident and Incident, Hazard, Risk, Safety, Safety and Security according to the definition found in the literature by Nancy G. Leveson defines safe as:

Definition 1: Failure is the non-performance or inability of the system or component to perform its intended function for a specified time under specified environmental conditions.

Definition 2: Error is a design flaw or deviation from a desired or intended state.

Definition 3: Accident is an undesired and unplanned event that results in a specified level of loss.

Definition 4: Incident is an event that involves no loss but with the potential for loss under different circumstances.

Definition 5: Hazard is a state or set of conditions of a system that, together with other conditions in the environment of the system, which will inevitably leads to an accident.

Definition 6: Risk is the hazard level combined with the likelihood of the hazard leading to an accident and hazard exposure or duration.

Definition 7: Safety is freedom from accident or losses.

Definition 8: Safety and Security-Safety is a subset of reliability or a subset of security.

Process Hazard Analysis (PHA): Hazard analysis process is both continual and iterative it consists of the following steps:

Step 1: Hazard identification starts at the early stage of the project and it is often called preliminary hazard analysis to determine what hazard might exists during operation of the system. Developing guidelines, specification and criteria to be followed in the system design [7]. Actions should be initiated to control the hazard. Identifying management and technical responsibilities for risk acceptance and an effective control is assured over hazard. Complexity of the safety problem is determined.

Step 2: Evaluating Hazard evaluates the hazard identified in hazard analysis the hazard category or level is identified by likelihood and severity then the hazard is prioritized and managed. Based on the hazard severity it is categorized in to Category 1: Catastrophic may cause death or system loss, Category 2: Critical may cause severe injury or illness, Category 3: Marginal may cause minor injury, Category 4: Negligible will not result in injury. Based on the hazard level it is commonly divided in to frequent, probable, occasional, remote, improbable and physically impossible.

Step 3: Control Measure identifies the causes and effects associated with each hazardous condition. Each hazard can have several potential causes and each causes can have several potential consequences or effects. The control measures of hazard analysis should reduce the severity of the hazard based on the priority of the hazard. The control measure should focus on system hazard analysis, subsystem hazard analysis and software hazard analysis.

Forward and Backward Searches: Forward and backward searches are useful when the underlying structure is temporal a forward search traces the safety application in forward time. Tracing an event in forward can produce large number of states and the problem of determining all reachable states from an initial state may be unsolvable using a reasonable set of resource.

Limitation: Forward analysis is limited to only a small set of temporarily ordered events.

Top-Down and Bottom-up Searches: A basic event, set, task, or system may be broken down in to more basic events, conditions, tasks, or sub systems in top-down approach [8]. This approach examines the effect of each individual component on the system the result of the bottom up search is not same as that of forward search. The effect of each individual component failures on the overall behaviour of the system and it determines the effect of a component failure at the system level using bottom up approach.

Limitation: It is difficult for complex systems.

Fault Tree Analysis (FTA): Fault Tree Analysis technique is widely used in application like aerospace, electronics and nuclear industries. It is primarily used to analyse the causes of hazards, not identifying hazards. Each level in the tree lists the more basic events that are necessary and sufficient to cause problem. It has four basic steps system definition, fault tree construction, qualitative analysis, quantitative analysis [9].

System Definition: It requires in determining the top events, initial conditions, existing events and the top level events are called crucial. The initial state of the event is analysed for the occurrence of top events for example the collision between the two automobiles will depend upon traffic speed and density.

Fault Tree Construction: In this step system definition, top event, casual events that are related to top events and the logical relationships between them are identified [10]. Most frequently used gates are AND, OR gates whereas the output of the AND gate exists only when two input exist, the output of the OR gate exists only atleast for a single input.Monte Carlo simulation can be used to construct the tree and to determine the function.

Qualitative Analysis: The purpose of the qualitative analysis is to reduce the tree to a logically equivalent form showing the specific intersection of basic events sufficient to cause the top events. In this intermediate false events are removed and only the relation between the top events and the primary events are described these are called cut set [11]. The main goal of the analysis is to form a minimal cut set and it provides the information helps to identify weakness in the system.

Quantitative Analysis: It uses the minimal cut sets to calculate the probability of occurrence of the top events from the probability of occurrence of the basic events. The probability of the top events will be the sum of the probabilities of all the cut sets if they are all statically independent [12]. If there is any replication of events in any cut sets, independence is compromised and the replication must be taken in to account in any quantitative analysis.

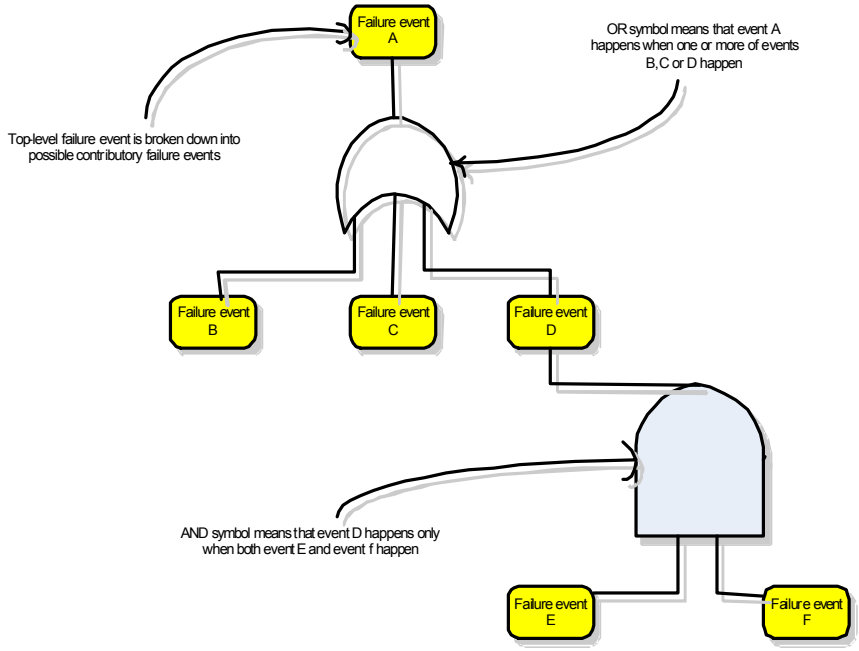

Fig. 1: Logical AND, OR in Fault Tree Analysis

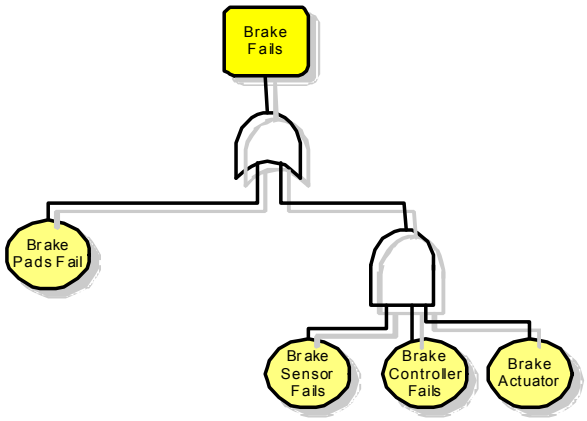

Fig. 2: Example of a simple fault tree for a brake system

Limitation: A common mistake in quantifying fault tree is multiplying two or more frequencies yields meaningless result. FTA becomes very difficult to apply in complicated systems.

Figure 1 describes the Logical AND,OR gates in Fault Tree Analysis whereas the failure event A occurs when one or more of events B,C or D happens and it is termed as the Top level failure event. Failure event D happens only when both the events E and F occurs. Figure 2 represents the example of a simple fault tree for a brake system by using the logical AND OR gates.

Event Tree Analysis: The Event Tree Analysis technique uses the forward search to identify the various possible outcomes of a given initiating event [13]. The initiating event might be a failure of a system component or some event external to the system. The event tree is drawn from left to right with branches the two alternatives are successful performance of the protection system forms the upper branch and failure of the protection system forms the lower branch. The probability of occurrence of the event in each path is determined ant the probability of failure is assumed to be small, the probability of success is always close to 1. The main goal of event tree analysis is to reduce the size of the tree by eliminating the meaningless relationships and to eliminate the zero conditional probability.

Limitation: Timing issues can cause problem in event tree construction.

Figure 3 describes Event Tree for a brake system whereas it predicts the probability of occurrences of the event and the causes and effects of the event when a brake sensor fails, brake controller fails or an actuator fails then the results are obtained based on the probability and severity of the causes and effects of the events.

Cause Consequence Analysis: This analysis starts with a critical event and determines the cause of the event by using top-down or backward search approach and it shows the both the time dependency and casual relationship among events [14]. First the procedure stars with a selection of critical events and it determines the causes and effects of the event. The logical symbol includes gates to describe the relation between cause and events and vertices to describe the relation between consequences AND, OR gates forms the logic gates and

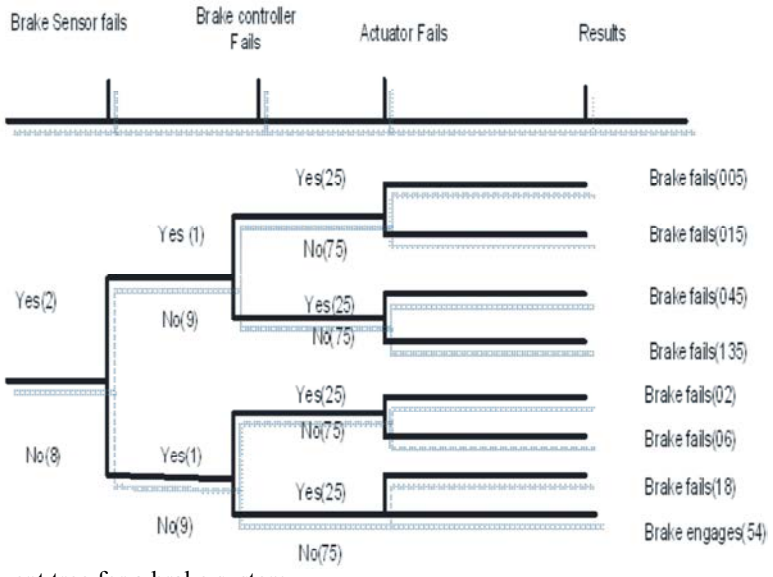

Fig. 3: Example of an event tree for a brake system

vertices. The blocks are described by arithmetic or transfer function whereas condition is a predicate that restricts the possible state of the system, an event is described by a pair of pre-and post-conditions. Cause Consequence has the advantage over Event Tree by allowing the representation of time delay, alternative consequence paths and combination of events.

Limitation: Separate diagrams are needed for each initiating event and the outcomes are only related to the cause being analysed, although they could be caused by other initiating events.

Hazards and Operability Analysis (HAZOP): HAZOP is based on a systems theory model of accidents that assumes accidents are caused by deviations from the design or operating intension of safety critical systems. It is a qualitative technique its purpose is to identify all possible deviation from the design expected operation and all hazards associated with the deviations. HAZOP is able to elicitate hazard from the new design and from the existing design of the critical applications [15]. The strength of the system is its simplicity and ease of application and in the early identification of design problems. This analysis does not concentrate only on failures, but it has the potential to find more complex types of hazard events and their causes.

Limitation: The drawback of this technique are time and effort required, it is labour dependent, it lies heavily on the judgement of the engineers performing the assessment.

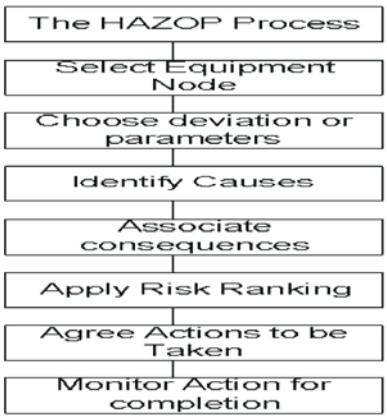

Fig. 4: Steps in HAZOP

Figure 4 describes the steps in HAZOP hazard analysis whereas the first step is to select equipment node second step is to choose the deviation from the expected, third step determines the cause identification then the consequence if the causes are identified. The risk ranking is applied based on the severity of the causes then the appropriate actions are to be taken to mitigate and to manage the risk.

Interface Analysis: This analysis uses structured walk-through to examine the communication between the components to determine whether the connection provides a path for failure propagation. The partial failure of the component degrades the performance of the system and it causes unstable output [16]. The specialized version of interface analysis considers the potential for common mode failures to affect redundant hardware components. The hardware and software common mode failure analysis examines the connection between the redundant components to determine whether the connection provides a path for failure propagation. Its limitation is same as that of Hazard and Operability Analysis.

Failure Modes and Effect Analysis: This analysis was developed to predict the reliability of the equipment in safety critical applications. The specified goal of this model is to evaluate the probability of the occurrence of failure of the system in specified time to calculate the mean time between failures [17]. The first step of this analysis is to identify and list all components and failure models with their possible operating modes. The probability of occurrence of events and the severity of the events are calculated. The results are documented in a table with column heading such as component, failure probability of the component, failure operating mode, percentage of failure in that mode and effects are categorized. The strength of this technique is completeness.

Limitation: It is effective for analysing single units or single failures, it is time consuming and can become tedious and costly if applied to all parts of a complex design.

State Machine Hazard Analysis: This model consists of set of states and the transition between them the arrow represents transition between the states. State Machine

Hazard approach involves forward search that starts from the initial state of the system, generates all possible paths from the state and determines whether any of the state is hazardous. It is used to identify the software related hazards in the early stage of the software development life cycle [18]. This algorithm was demonstrated by using petri-net model to analyse a design for safety and fault tolerance, to determine software safety requirements directly from the system design, to identify safety critical software function and to help in the design of failure detection and recovery procedures and fail-safe requirements.

Limitation: It is difficult and time consuming; it is impractical for large complex systems. The analysis of SMHA is performed on the model not on the system. It is very hard to learn and use without an advanced degree in mathematics.

Systems Theoretic Accident Model and Process (STAMP): An undesired or unplanned event that results in a loss including loss of human life or human injury, property damage, environmental pollution and mission loss. All hazards related to human injury or damage to be eliminated or mitigated by the system design. A reasonable effort must be made to eliminate or to mitigate the hazard. The safety requirements and constraints on the physical system design acts as input to the standard system engineering process and must be incorporated in to the physical system design and safety control structure.

STAMP model of accident causation is built on three basic concepts like safety constraints, a hierarchical safety control structure and process models along with basic systems theory concepts. Accidents can be understood, using STAMP, by identifying the safety constraint that were violated and determining why the controls were inadequate in enforcing them. Component failures may results from inadequate constraints on the manufacturing process; inadequate engineering design such as missing or incorrectly implemented fault tolerance. Component failures may be prevented by increasing the integrity or resistance of the component to internal or external influences or by building safety margins or safety factors of the critical applications. STAMP models are more complete than most accident reports and other models [19]. It is useful not only in analysing accidents that have occurred but in developing new and potentially more effective system engineering methodologies to prevent accidents before it occur. Compared with other traditional hazard analysis techniques such as Fault Tree Analysis and various types of failure analysis STAMP works well for complex systems, for software errors and system design errors. Control process operates between levels of severity to control the process at lower levels in the hierarchy. These control processes enforce the safety constraints for which the control process is responsible. Accidents occur when these processes provide inadequate control and the safety constraints are violated in the behaviour of the low-level components. At each level of severity, inadequate control may results from missing constraints, inadequate safety control commands, commands that were not executed correctly at a lower level, or inadequately communicated or processed feedback about constraint enforcement. The concept used in STAMP along with safety constraints and hierarchal safety control structure is process model. process models are an important part of control theory.

System Theoretic Process Analysis (STPA): This analysis can be used at any stage of the system life cycle. STPA has two main steps first step is to identify the potential for inadequate control of the system that could lead to a hazardous state. Hazardous states results from inadequate control or enforcement of the safety constraint. It is to determine how each potentially hazardous control action is identified for each unsafe control action examines the parts of the control loop to see the cause. The steps in STPA involves

Step 1: Identify hazards

Step 2: Identify unsafe control actions

Step 3: Identifying casual factors.

In the first step hazard are identified then draw the control structure to identify the major component and controller causing hazards then label the control using feedback arrows. In the second step unsafe control actions are identified then a table is constructed to store the information related to the hazardous components and wrong timing actions then safety constraints are regulated for those components. In the third step casual factors are identified and the flow control and feedback pat flows are identified [20]. The scenarios are identified using STPA included those caused by potential component failure as expected scenarios were identified that involve unsafe interaction among the components. Most modern system

involves interaction among the components in order to avoid the hardware, software and human error. The fault or failure of one components relays and penetrate to other components and causes catastrophic failure of system this can be avoided by communication between the components so the modern system must address the component failure and a worst case analysis should be done not the best case analysis for the hazardous component.. Most modern failure analysis technique incorporates reliability technique to analyse the safety related factors as a part of safety analysis. STPA is to analysis the safety issue related to design error, software flaws, component interaction accident and human decision making errors.

Limitation: It is very difficult to

- Identify basic risk for new components.
- Defining control structure.
- Limiting the domains taken in to account.

Software Safety Assessment related to Context: The assessment process involves safety program plan in this the overall assessment of the system is done. This plan identifies the safety related requirements and their standards to be applied in order to maintain the completeness and correctness of the system. Functional Hazard Assessment (FHA) is developed to access the functionality and conceptual design of the system and it is documented. The effects of failure are. accessed against the standard regulations to identify how much the failure component is deviated from the standards. System Functional Hazard Assessment (SFHA) is similar to FHA whereas it analyses the system architecture to identify and classify the failure conditions and combinations of failure conditions. Three separate analyses are typically performed to evaluate the causes they are particular risk analysis which evaluates the events and their influences which are outside the system, a zonal safety analysis is to analysis the different zones and the context, common mode analysis is to analyse the common effect during development, implementation, test, manufacturing, installation, maintenance and failure mode operation of the system.

CONCLUSION AND FUTURE WORK

Software safety emphasis early in the requirement stage and in conceptual design process of the early stage of the software development lifecycle of the system. It is directly related to critical design aspects and safety attributes in software and system functionalities. In this paper various hazard analysis and their limitations are discussed briefly whereas safety encompasses functional hazards, functional paths, domains and boundaries to ensure correct system functionality and to detect malfunctions of software and hardware, failures, faults and the procedures to mitigate the hazards based on the safety standards. The future work of this paper is to propose a design for safety model in automotive software safety critical systems focusing the context awareness features, user actions and unexpected reaction from the environment

REFERENCES

1. Leanna Rierson, 2013. Developing Safety Critical Software, A practical guide for aviation software and DO-178C Compliance CRC Press 2013.
2. Nancy G. Leveson, 2011. Engineering a Safer World, System Thinking Applied to Safety, MIT press Cambridge 2011.
3. Debra S. Herrmann, 1999. Software Safety and Reliability, Techniques, Approaches and Standards of key Industrial Sectors, IEEE Computer Society, California 1999.
4. Carlo Cacciabue, P., 2007. Modelling Driver Behaviour in Automotive Environments, Critical Issues in Driver Interactions with Intelligent Transport Systems, Springer-Verlag London ltd 2007.
5. Nancy G. Leveson, 1995. Safeware System Safety and Computers, A Guide to Preventing Accidents and Losses Caused by Technology, Addision-Wesley Publishing Company, Inc 1995.
6. Gerad J. Holzmann, 2006. The Power of 10: Rules for Developing Safety Critical Code, NASA Software Engineering Laboratory at NASA, June 2006.
7. Komal Bashir, Faria Kanwal, *et al.,* 2013. Software Quality Assurance for Safety Critical Systems, The International Journal of Soft Computind and Software Engineering, 3(3), March 2013.
8. Ben Swarup, M. and P. Seetha Ramaiah, 2009. A Software Safety Model for Safety Critical Applications, International Journal of Software Engineering and its Applications, 3(4), October 2009.
9. By Ben Swarup Medikonda, P. Seetha Ramaiah and Anu A. Gokhale, 2011. FMEA and Fault Tree based Software Safety Analysis of a Railroad Crossing Critical Syatem, GLOBAL Journal of Computer Science and Technology, 11(8) Version 1.0, May 2011.

10. Phani Kumar, S., Dr. P. Seethe Ramiah and V. Khanaa, 2009. A Methodology for Modeling Software Safety in Safety Critical Computing Systems, IJCSNS International Journal of Computer Science and Network Security, 9(7), July 2009.
11. Avizienis, A., J.C. Laprie, B. Randell and C. Landwehr, 2004. Ba-sic concepts and taxonomy of dependable and secure computing.IEEE Transactions on Dependable and Secure Computing, 01(1): 11-33.
12. Chalé Góngora, H.G., O. Taofifenua and T. Gaudré, 2010. A Process and Data Model for Automotive Safety-Critical Systems Design. In Proceedings of the 20th annual International Symposium of the INCOSE (Chicago, IL). Seattle: INCOSE.
13. Jayasri, K. and P. Seetha Ramaiah, 2014. Hazard Analysis for Software Safety in Road Traffic Control System, International Journal of Advanced Research in Computer Science and Software Engineering, 4(10), October 2014.
14. Marvin Zelkowithz and Ioana Rus, 2001. Understanding IV and V in a safety critical and complex evolutionary Environment: the NASA Space Shuttle Program, Proceedings of the 23rd International Conference on Software Engineering (ICSE), Toronto, Ontario, Canada.
15. Jesty, P.H., D.D. Ward and R.S. Rivett, 2007. Hazard Analysis for Programmable Automotive System, International Conference on Institution of Engineering and Technology System Safety.
16. John Gould, Michael Glossop and Agamemnon Loannides, 2000. Review of Hazard identification Techniques, Report on Health and Safety Laboratory, HSL/2005/58, 2000.
17. Klim, Z.H., 2004. Preliminary Hazard Analysis for Design Alternatives based on Fuzzy Methodology, IEEE Fuzzy Information Conference.
18. Aftab Ali Haider and Aamer Nadeem, 2013. A Survey of Safety Analysis Techniques for Safety Critical Systems, International Journal of Future Computer and Communication, 2(2), April 2013.
19. Vesley, W., J. Dugan, J. Fragole, J. Minarik and J. Railsback, 2002. Fault Tree Handbook with Aerospace pplications, NASA Office of Safety and Mission Assurance, NASA Headquarters, Washington DC 20546, August 2002.
20. Bruns, G. and S. Anderson, 1993. Validating safety models with fault trees, in Proc. of 12th International Conference on Computer Safety, Reliability and Security, pp: 21-30, Springer-Verlag, 1993.