Improving Safety-Critical Systems by Visual Analysis

Yi Yang[1], Patric Keller[1], Yarden Livnat[2], and Peter Liggesmeyer[1]

1 **Software Eningeering: Dependability Group**
University of Kaiserslautern, Germany
{yang,pkeller,liggesmeyer}@cs.uni-kl.de

2 **Scientific Computing and Imaging Institute**
University of Utah, USA
yarden@sci.utah.edu

Abstract

The importance analysis provides a means of analyzing the contribution of potential low-level system failures to identify and assess vulnerabilities of safety-critical systems. Common approaches attempt to enhance the system safety by addressing vulnerabilities using an iterative analysis process, while considering relevant constraints, e.g., cost, for optimizing the improvements. Typically, data regarding the analysis process is presented across several views with few interactive associations among them. Consequently, this hampers the identification of meaningful information supporting the decision making process. In this paper, we propose a visualization system that visually supports engineers in identifying proper solutions. The visualization integrates a decision tree with a plot representing the cause-effect relationship between the improvement ideas of vulnerabilities and the resulting risk reduction of system. Associating a component fault tree view with the plot allows to maintain helpful context information. The introduced visualization approach enables system and safety engineers to identify and analyze optimal solutions facilitating the improvement of the overall system safety.

1998 ACM Subject Classification B.4.5 Reliability, Testing, and Fault-Tolerance, I.3.8 Applications, D.2.4 Software/Program Verification

Keywords and phrases fault tree analysis, importance and sensitivity analysis, information visualization, decision tree, safety analysis

Digital Object Identifier 10.4230/OASIcs.VLUDS.2011.43

1 Introduction

Fault tree analysis is a widely used technique for the identification of vulnerabilities of safety-critical systems. This analysis uses a graphical model called fault tree to logically relate undesired failures at the system level (called top event) with failures at the component level (named basic events). A component fault tree is an advanced modularization concept supporting the fault tree analysis of complex systems. This allows to extend the regular fault tree model by decomposing it according to the architecture of the system under investigation into a hierarchical representation where each component is represented by an extended fault tree. The fault tree analysis provides a basis of the importance analysis and sensitivity analysis of those failure relations. It mainly focuses on the risk contributions of individual basic events to a top event. The important basic events represent the critical vulnerabilities of a system. Sensitivity analysis is applied to investigate relations between changes of basic events and the resulting impacts on a top event.

Proceedings of IRTG 1131 – Visualization of Large and Unstructured Data Sets Workshop 2011.
Editors: Christoph Garth, Ariane Middel, Hans Hagen; pp. 43–58
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
IRTG 1131

In order to improve the system safety, engineers usually carry out an iterative risk reduction approach consolidating importance and sensitivity analysis. As a result of the approach, engineers may identify an improvement solution consisting of a group of modifications with respect to system design. By a solution, the failure probability of top event is reduced to an acceptable level. In many cases, engineers may identify multiple possible solutions by various alternative design modifications in the analysis process. Thus, the safety improvement process consists of the determination of modifications and the review of solutions by taking the essential questions into account:

- Aspects of modifications:
  - What are the most important basic events contributing to a system failure?
  - What are possible modifications of the system design?
  - What are the impacts of the modifications regarding system safety?
  - Which modifications are optimal taking certain constraints into consideration?
- Aspects of solutions:
  - How good are the improvement solutions?
  - What is the best solution?

Usually, the data related to the questions is separated across individual views having various representation forms, e.g., fault trees, tables, histograms, plots, and decision trees. However, there are few interactive associations among the views. Mostly, engineers need to frequently switch views for accessing meaningful data during the analysis process. Additionally, there is no sufficient context information when engineers focus on a specific view. For example, modifications are organized using a decision tree and the detailed data of the queried modification is represented in a separate table. When focusing on the table of detail information, the context with respect to the overview of modifications may be lost. Furthermore, when analyzing the basic event corresponding to this modification, engineers need to manually locate the basic event in the fault tree view because the decision tree does not provide this information. Engineers spend more additional efforts for switching views and identifying significant information.

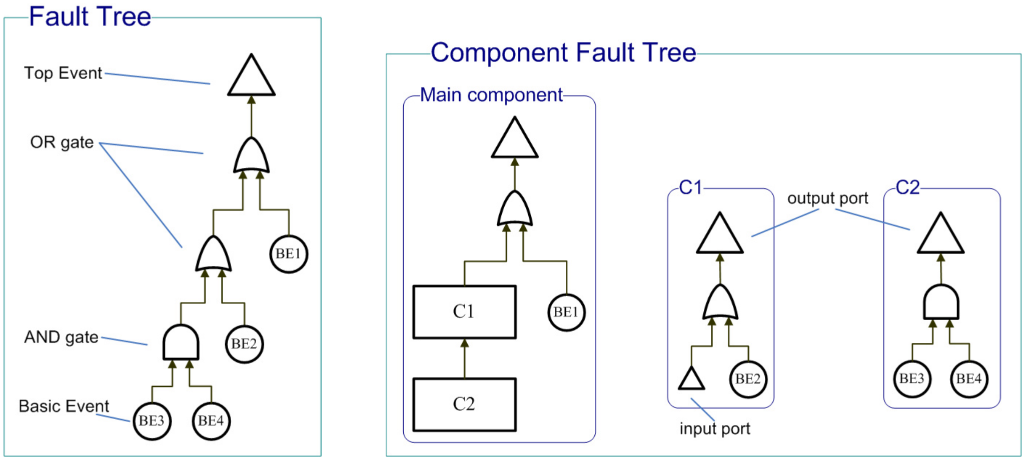
In this paper, we propose a visualization system that effectively integrates data which is essential for the analysis of the safety improvement process. To support the information access within different contexts, we additionally provide suitable interaction possibilities. The proposed visualization system facilitates to identify and analyze vulnerabilities of safety-critical systems, as well as determine the optimal/appropriate solution(s) by simulating system design modifications on an abstract level.

The remainder of the paper is organized as follows: Section 2 describes the basic principles of the (component) fault tree analysis, importance and sensitivity analysis, as well as the related representation concepts. We introduce our visualization system in Section 3. We review the proposed methods on the basis of a short application example in Section 4. The conclusion is subject to Section 5.

2 Background

2.1 Safety Analysis

The term *safety* often refers to a state of a system where the danger of a personal injury or property damage lies within an acceptable level [15, 13, 4, 24]. A *failure* is defined as an inconsistent behavior that deviates from the given specification of a system or a component



(a) Fault tree **(b)** Component fault tree

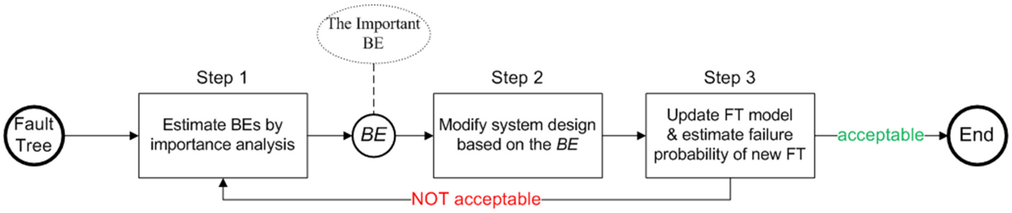
Figure 1 Fault tree and component fault tree. (a) A fault tree consisting of four basic events connected by an AND-gate and two OR-gates. (b) The component fault tree model based on the fault tree in (a), which contains a main component and sub component "C1" and "C2". "C2" inputs its failure to "C1" via ports.

[4, 24]. In this context a system is said to be *safety-critical* if the failure of the system could cause consequences that harm people [18, 13, 31, 24]. *Risk* is a combination of the frequency of a harmful failure and the severity of the harm caused by that failure [13, 24]. When talking about *safety analysis* we often refer to a process whose goal is to provide a reliable assessment and improvement of the risk of a safety-critical system [19, 20, 24]. To achieve this a variety of methods and techniques exist, e.g., fault tree analysis.

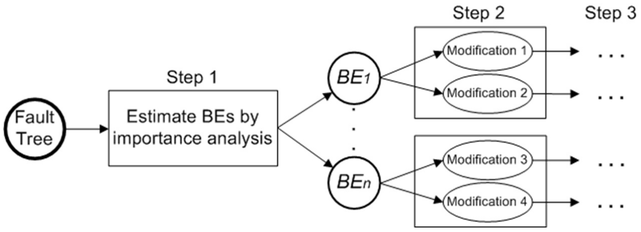
2.2 Fault Tree Analysis

Fault tree analysis (FTA) [13, 36, 35, 12] is a deductive method allowing to trace the causes of an undesired system state back to its roots. The method is based upon the usage of so called *fault trees (FTs).* A fault tree is a tree-like structure composed of different types of nodes. The root of the tree termed *top event* represents the undesired system state (e.g., system failure or outtake). The leaves are *basic events (BEs)* that represent the low-level failures which are connected by logic gates, such as "AND-gate"and "OR-gate". The way the leaf nodes are connected reflects how the low-level failures logically contribute to the undesired system state (see Figure 1(a)).

The ordinary modularization concept of fault tree allows to partition the independent sub-trees as modules. However, these modules are not be mapped to identify technical components of the system design. To solve this issue, Kaiser et al. [17] proposed an advanced modeling concept called *component fault trees (CFTs).* Technical components of a system are represented as the corresponding CFT components in the component fault tree model. The influences between technical components are transferred via in- and out-ports of CFT components. In this way, engineers may treat each CFT component as a black-box. Figure 1 (b) shows an example of the concept of component fault tree. The component fault tree modularizes the sub-trees as CFT components and replaces the sub-trees with rectangles in the main component. The detailed sub-trees are separately represented in individual views.


The Important BE
Step 1
Step 2
Step 3
Fault Tree
Estimate BEs by importance analysis
BE
Modify system design based on the BE
Update FT model & estimate failure probability of new FT
acceptable
End
NOT acceptable

(a) Procedure of single solution construction



(b) Multiple solutions construction

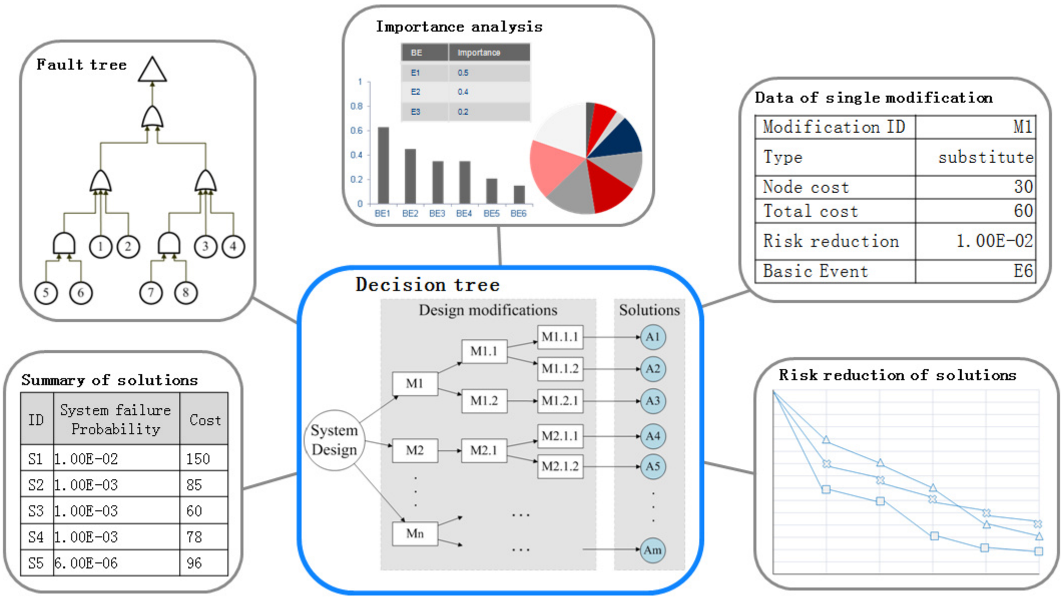
Figure 2 Construction of improvement solutions.

2.3 Importance and Sensitivity Analysis

Importance analysis and sensitivity analysis are quantitative approaches for evaluating (component) fault trees. Vesely et al. [35] suggested that, in general, more than 90% of the failure probability of a top event is due to less than 20% of the basic events. This implies that we only need to focus on a small subset of basic events having major contribution. To identify those, we determine the importance of each basic event with regard to the failure probability of the top event. Importance analysis considers both the failure probability and the logical relations of basic events. The Fussell-Vesely (FV) importance measure [11, 27] assigns each basic event an importance value between zero and one: the larger the value, the more important the basic event in terms of influence on the top event. The sum over the importance of all basic events of a system may be greater than one since, in some cases, simultaneous failures of multiple sub-systems may cause the system failure [10]. On the other hand, the sensitivity analysis investigates the resulting impact of changes applied to the basic events on the top event [14, 35, 23]. It is used for analyzing the accuracy of basic events as well as the effects of safety improvements [25, 7, 9].

2.4 Improvement of Safety-Critical Systems

The improvement of the system safety may necessitate design modification involving the replacement of critical parts of the system by elements having a better failure performance (substitution concept) or introduction of identical redundant parts (redundancy concept). Finding a satisfying solution in general is a non-trivial task underlying constraints and restriction for which formal methods are not always available. Generally, the procedure associated with this approach iteratively applies a set of alternative modifications until the complete solution is found, which reduces the risk of a system to an acceptable level. Taking the results obtained from the improvement analysis into account, it is possible to derive such solutions in a more guided fashion [7, 9, 8, 35, 25]. Each iteration consists of mainly three steps (see Figure 2 (a)):



BE	Importance
E1	0.5
E2	0.4
E3	0.2

Data of single modification

Modification ID	M1
Type	substitute
Node cost	30
Total cost	60
Risk reduction	1.00E-02
Basic Event	E6

Summary of solutions

ID	System failure Probability	Cost
S1	1.00E-02	150
S2	1.00E-03	85
S3	1.00E-03	60
S4	1.00E-03	78
S5	6.00E-06	96

Figure 3 Ordinary representations used by the analysis of safety improvement process. Improvement solutions are arranged by a decision tree [9]. The relevant data is distributed across several views [1, 7, 32, 9]. Commonly used representations are fault trees, charts or tables to show importance of basic events, and the summary of possible solutions, design modifications, and individual risk reductions.

1. Perform the importance analysis to identify the basic event having the largest contribution.
2. Find the hardware component related to that basic event. Modify the system design by replacing the component by another one featuring a better quality or by introducing identical ones in order to increase redundancy.
3. Update the (component) fault tree model and assess the modification with respect to the impact on the top event in terms of reduction of failure probability. If required, engineers may determine the optimal modification under consideration of additional constraints, e.g., the costs of the modifications. If a complete solution is found, i.e., the failure probability of the top event is reduced to the goal value, stop the process, otherwise start next iteration from step 1.

Constructing a solution necessitates to choose the proper basic event, and to decide for a suitable design alternative. In many cases, multiple improvement solutions exist because of multiple important basic events or/and various alternative design modification ideas corresponding to the identical basic event (see Figure 2 (b)). After constructing solutions, engineers may additionally identify the optimal one. An important assumption of the safety improvement process is that basic events are stochastically independent so that the change of a basic event does not influence other basic events.

2.5 Related Representation Concepts

Fault tree analysis tools provide the view on fault trees using standardized graphical symbols (see Figure 1 (a)). The data of the fault tree, e.g., failure probability of the top events and the basic events are represented by text or data-aggregated forms. Most fault tree analysis tools [16, 28, 1, 7, 32, 9] summarize the importance of basic events using a data table. Faulttree+ [16] shows importance values in a table associated with table presenting the properties of events, e.g., failure probability. Relex Architect [28] provides a table in which users may filter and show the importance of basic events belonging to a specific sub tree. RAMCommander [1] additionally provides charts for the importance values, e.g., histogram, pie-chart, and 2D/3D scatter plot. BlockSim [29] assigns colors to the histogram according to the failure probability of basic events. Additionally, BlockSim proposed a variant of pie-chart called “square pie-chart” that anti-clockwise arranges the basic events in descending order with respect to the importance values. Project CISA [7, 9] arranges data of design modifications in separate views and logically links them to a decision tree that represents the summary of improvement solutions (Figure 3).

A decision tree is a tree-based predictive model that is widely used for facilitating the decision-making in many domains. It partitions a data set into subsets according specific rules. The root represents the original data, the edges represent the partitioning rules, and the non-root nodes represent the outcomes of different rules. To construct a decision tree, the users need to quickly identify the nodes to be partitioned by navigating through the tree. Decision tree is a good way to provide overviews about complex decision-making process. Ankerst et al. [2, 3] applied an indentation diagram to represent a decision tree for arranging the partitioning steps in data mining. The work [34] integrated decision trees and data visualizations of attributes of each node for purposes at data classification. Pham et al. [26] presented a decision tree using the sunburst layout to visualize machine-learning algorithms. The decision trees represented by an icicle diagram were provided by the work [21, 3, 6]. The icicle concept represented tree hierarchies without wasting display space. Project PaintingClass [33] integrated parallel coordinates and star coordinates with a decision tree for exploring classified multi-dimensional data. Barlow et al. [6] proposed a visualization system that linked views of various decision tree layouts to represent the decision data of data mining process.

3 Visualization for Safety Improvement

3.1 Requirements of Analysis Process

The safety improvement process of a system concentrates on two phases: construction of solutions and review of solutions. Requirements of the analysis (in short “R”) are summarized as follows:

- Construction of solutions: A solution comprising a sequence of design modifications. It is constructed by performing an iterative analysis procedure (see Figure 2). The steps are:
  - Step 1: identify the the important basic events (R1).
  - Step 2: apply and test the risk reduction hypothesis by different modifications.
    - R2: identify the type of modification: substitution or redundancy.
    - R3: identify the value of modification: change of failure probability of the initial basic event.
    - R4: identify the cost of modification.
  - Step 3: evaluate the results of risk reduction:
    - R5: evaluate the update of the component fault tree model.
    - R6: evaluate the impact of top event by design modification.
    - R7: evaluate the cost-effectiveness of modification.
    - R8: evaluate the gap between updated failure probability of top event and the goal value.
- Review of solutions: Reviewing the constructed solutions may facilitate the understanding of solutions and determining the optimal ones resulting from the following requirement.

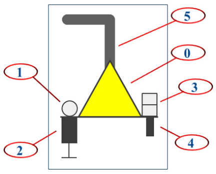


Figure 4 Risk-state node for a design modification. (0) Risk state of top event. Color indicates the level of failure probability of the top event. (1) Type of the modification. Circle indicates substitution concept, while small triangle indicates redundancy concept. (2) Change of failure probability of the corresponding important basic event. (3) Cost of the modification. (4) Cost-effectiveness of the modification. (5) An edge connecting the node with its predecessor node. The vertical part represents the resulting reduction of failure probability of the top event.

3.2 Visual Support for Construction of Solutions

3.2.1 Representing Design Modification

Along with performing the safety improvement process, the design modifications are sequentially connected as a decision tree to construct one or more solutions (see Figure 3). A branch of the decision tree is caused by either multiple important basic events or multiple modification ideas. We finally apply the node-link diagram for representing the decision tree by taking two points into consideration: readability and data integration. Barlow et al. [5] evaluated the readability of the treemap layout, the sunburst layout, the node-link diagram, and the icicle diagram. The authors conducted that the node-link diagram and the icicle diagram were the most favorable for representing the tree structure data. The node-link diagram has the sufficient space to integrate the visual attributes in nodes. However, the icicle diagram (as well as treemap and sunburst layout) is a compact layout in which the aspect ratio needs to be maintained for the semantic meaning. In this case, the nodes on the deeper hierarchy of the tree do not have sufficient space for representing the attributes. In sum, the node-link diagram is more appropriate for our decision tree than other layouts.

For understanding a modification, the cause (i.e., corresponding basic event) and the effect (i.e., resulting risk state of system) is the primary information. In order to represent the cause-effect relation, we place the modification nodes of the decision tree in a risk-reduction plot (see Figure 5 (2)) where x-axis represents ordinal basic events, while the achieved change in risk (in terms of failure probability) is projected along the y-axis that represents a range from the initial failure probability of a top event to the goal value in a top-down direction.

We then introduce the visualization properties of the node-link decision tree. In order to represent the associated significant data, for each modification, we propose a risk-state node that consists of a central triangle icon and four attached visual items representing data associated with the modification (see Figure 4).

- **Triangle icon**: shows the risk state of the system corresponding to the modification. This is the most significant data based on the updated component fault tree (with respect to R5). In many cases, the safety and system engineers intend to quickly and roughly estimate the change of risk of a system, e.g., by which step the failure probability is reduced from critical level to moderate level. The shape of triangle is applied because it is consistent with the shape of the top event of the component fault tree. Color is recommended for representing the ordinal data by the work at [22]. The color of the triangle depends on the level of failure probability described in Section 3.2.3. Using colors, one may quickly estimate the criticality of the top event, and decide whether the failure probability is acceptable or not. When the color becomes green, the risk reduction can

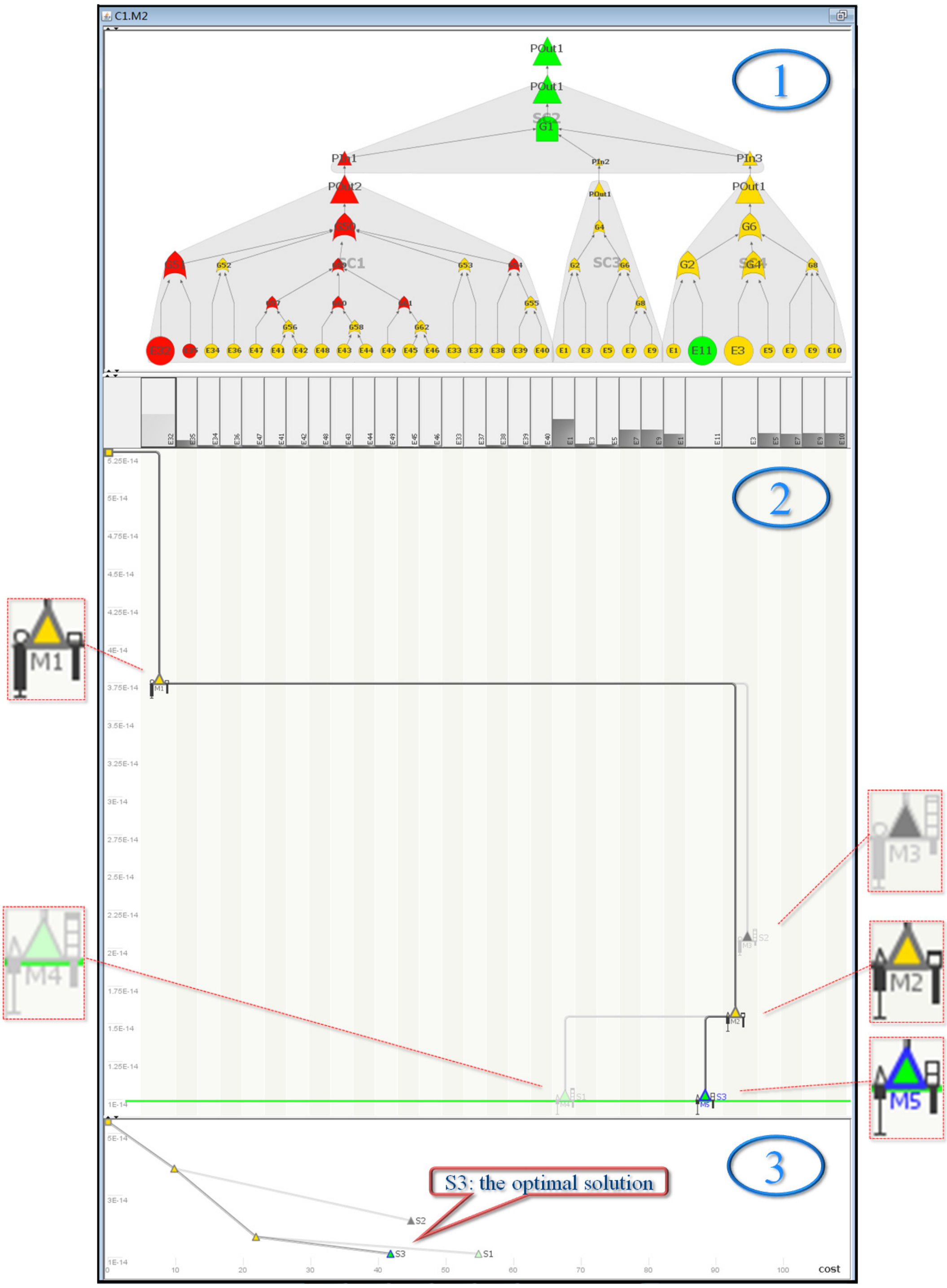


Figure 5 Visualization system for improving system quality with respect to safety. (1) The associated component fault tree view. (2) The risk-reduction plot. (3) The solution overview plot.

be finished and the corresponding solution is complete. Additionally, the label of the corresponding modification is presented below the icon.

- Item 1: the type of modification (with respect to R2). The types are nominal data that may be effectively represented by the graphical properties of position, color, texture, connection, density, and shape. The graphical properties of position, color and connection are already used in our visualization. Taking the size of the triangle icon into account, the graphical property of density is not suitable, too. Thus, we apply shape for representing the basic types of design modifications: a circle represents a component substitution whereas a small triangle represents the introduction of redundant components.
- Item 2: the reduction of the failure probability of the original basic event (with respect to R3). If engineers replace the initial hardware, the value of the new part becomes current. If engineers apply the redundancy concept, the new value is the failure probability of the new sub-tree of the redundant parts. The difference between the failure probability of the original basic event and the new basic event (or sub-tree) introduced represents the improvement of the vulnerability being addressed. To present this information in an intuitive way we have designed a bar graph using the graphical property of length that is recommended for representing the quantitative data [22]. The bottom line of the bar indicates the failure probability of the initial basic event. The filled part shows the new value. The item provides information about the context under which the modification has been applied. For example, following the substitution approach, it is possible to intuitively compare the existing with the new part in terms of failure probability.
- Item 3: the cost of modification (with respect to R4). In our work, the cost is an value representing a quantity consumed for the modification, e.g., money, time, and human-resources. The type of the cost needs to be defined at the beginning of the safety improvement process. It is an important information for evaluating design modifications (see Section 3.2.2) and solutions (see Section 3.3). We propose a scale bar to visualize the cost not only for the comparison of cost of modifications but also for the investigation of the absolute cost value. Engineers are allowed to define the scale of the bar, e.g., each box represents 10 dollars.
- Item 4: the cost-effectiveness ratio of a modification (with respect to R7). In cases where multiple design modifications exist it is important to choose those providing the proper balance between risk reduction and cost (see Section 3.2.2). We use the graphical property of length to represent the quantitative cost-effectiveness. Thus, a bar is introduced to represent the cost-effectiveness ratio for a given design modification. The larger the bar, the more cost-effective the modification.

There are two possible ways to composite the central icon and the visual items: the inside strategy and the outside strategy. When placing the visual items inside the central icon, the icon needs to be enlarged. In this case, the large icon cannot exactly indicate its position in the plot that represents significant semantic meaning of the analysis process. Thus, we apply the outside composition strategy. We place the four visual items closely around the central icon. The visualization properties corresponding to the method of the modification (items (1) and (2)) are placed at the left; the factors of evaluation of the modification are represented at the right (items (3) and (4)). This way, engineers may investigate the method and evaluation of modification in the corresponding side.

We connect a new risk-state node with its direct predecessor using a two-part orthogonal edge. A line between the predecessor node and the horizontal position of the new risk-state node represents the subsequent design modification. The vertical part of that line represents the reduction of the failure probability resulting from modification (with respect to R6).

This provides a reliable basis for guiding the analysis process. When there are alternative modifications for a basic event, multiple risk-state nodes are created. Between two nodes, there is an even distance dividing the width of the x-axis scale of the basic event (see Figure 5: "M2" and "M3"). This may address the overlapping issues of edges as well as of nodes.

To conveniently identify important basic event(s) in each iteration of the analysis procedure (with respect to R1), we present bars on a list of indicators of basic events along the x-axis on top of the risk-reduction plot (the more important a basic event, the longer the bar). Additionally, we provide a horizontal green line in the lower part of the plot for indicating the goal value. This enables us to assess the distance from the goal (with respect to R8).

3.2.2 Identifying optimal Modifications

The decision tree of the analysis process may exponentially grow because of its number of branches. Consequently, engineers might spend much efforts for analyzing a large set of modifications. In this case, engineers need to identify the optimal design modification(s) in each iteration of the process in order to effectively construct adequate solutions (with respect to the step 3 of the analysis procedure). The commonly used criterion is the maximal cost-effectiveness of the modification (referring to visual item (4)). Engineers may alternatively apply the criteria with respect to the largest reduction of failure probability of the top event (referring to the vertical position of risk-state node). The non-optimal modifications may be refused leading to the termination of the corresponding branches. We assign black color to fill up the risk-state node of the modification. This way, one can easily realize that the modification was considered and has been refused.

3.2.3 Adapting Component Fault Trees

Fault trees provide meaningful information for the safety improvement process. We apply the component fault tree in our visualization system instead of the ordinary fault tree because the component fault tree additionally provides the possibility to link failure mechanisms with the elements/components of the system design. According to the definition of the component fault tree, a CFT component reflects an architectural component of the system model in the design phase. This supports the identification of the vulnerable parts of the system design corresponding to the important basic events identified. Additionally, the structure of component fault tree supports the understanding of the effects of modifications along the way a failure propagates through the system when reviewing solutions.

We provide a visually enhanced component fault tree view for supporting the safety improvement process (see Figure 5 (1)). In order to associate the component fault tree view with the risk-reduction plot, we project the ordinal data of the x-axis of the plot (i.e., basic event list) according to their locations within the component fault tree view. This allows to link information from both views. We provide interaction mechanism on the component fault tree view in order to dynamically show the sub-trees of the desired CFT components. Each sub-tree is arranged inside a gray blob that indicates the scope of the CFT component. Our system automatically updates the component fault tree model in the background during the analysis process. In order to quickly assess the updated failure probabilities of nodes of the modified component fault trees, we propose a qualitative estimation method to classify failure probabilities into three levels and assign them colors: critical level (red), moderate level (yellow), and acceptable level (green).

In order to preserve the overview about the vulnerable basic events addressed in a solution, we maintain the initial structure of the component fault tree during the safety improvement

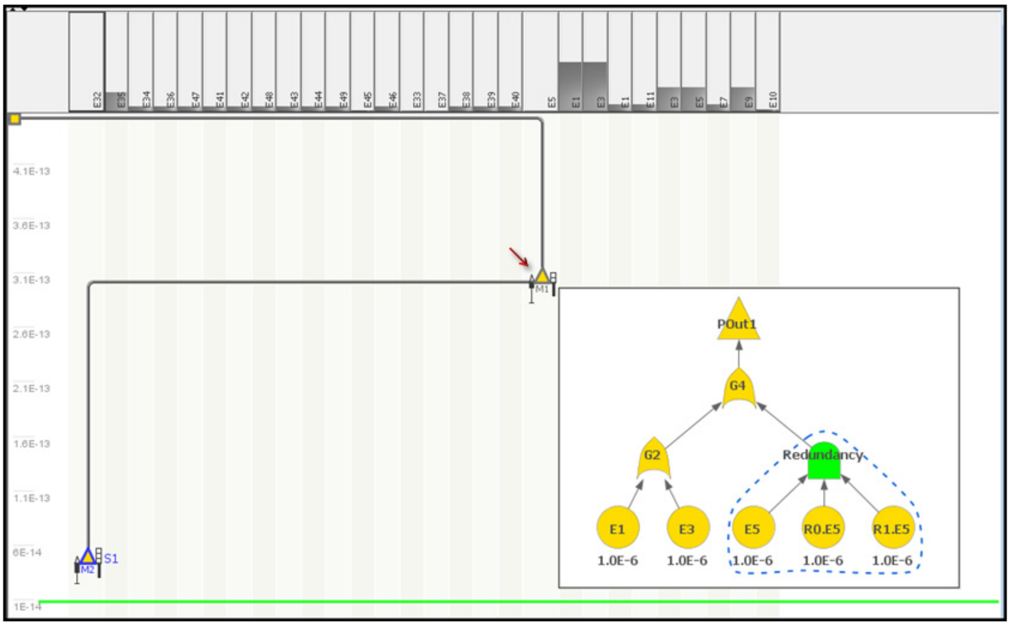


Figure 6 Pop-up window shows the updated logical structure of the CFT component with respect to the modification "M1". The new created sub-tree is arranged in a scope having a dotted border. Two basic events were added and connected with the initial basic event using an AND-gate.

process. That means, by modification performed according to the redundancy concept, the identified important basic events are not directly replaced by sub-trees. Instead, we adapt the color of the initial basic event node with respect to the failure probability of either the substitutional part or the new sub-tree of redundant parts. This can avoid disturbances caused by subsequently updating the component fault tree.

In case engineers intend to review the modified structure of the component fault tree of the specific design modification, our visualization system allows them to show a pop-up view representing the updated logical structure by a right-click on a risk-state node. Instead of displaying the whole component fault tree, the view only presents the structure of the CFT component that contains the basic event related to the design modification. The adapted part of the component fault tree is arranged in a scope indicated by a dashed border. This enables us to intuitively and flexibly view the adapted structures of the component fault tree. This is particularly useful for reviewing design modifications utilizing the concept of redundancy. For example, Figure 6 shows the adaption of a CFT component by a modification. A parallel redundancy is applied by adding two new homogeneous parts and connecting with the initial basic event by an AND-gate.

3.3 Visual Support for the Review of Solutions

While the risk-reduction plot supports the construction of improvement solutions, the overview of the solutions is not intuitive for analyzing the cost-related patterns of the proposed solutions. Such as the trend of risk reduction and of cost increase. It is not suitable for identifying the optimal solutions having the minimum total cost. Thus, we provide a simple and effective plot to present an overview about these quantities (Figure 5 (3)). The x-axis and the y-axis respectively represent the cost of modifications and the failure probability of the top event. We present a triangular node on the overview plot for each modification. We provide a brushing-and-linking interaction between the risk-reduction plot and the overview plot in order to simultaneously highlight the information associated with the same design modification. Engineers may obtain an intuitive summary of solutions and identify the optimal ones.

We focus on the reduction up to the goal value rather than the exhaustive risk reduction. The overloaded reduction may lead to a large improvement, however, simultaneously also take large costs. In our work, we assume that all complete solutions reduce the initial risk to the same goal value. In this case, for estimating a solution, we consider the total costs of a solution instead of the total cost-effectiveness because this has the identical risk reduction effects to other complete solutions.

4 Application Example

We provide an example intended to illustrate the use of our system with respect to two important aspects: construction of solutions and the review of existing solutions. The applied data originates from a component fault tree of a safety-critical sub-system of an autonomous mobile robot [30]. This model contains 30 basic events and 4 CFT components. The goal is to identify the most cost-effective solution. The initial failure probability of the top event amounts “1.2e-13”, the specified acceptable value is “1e-14”.

4.1 Construction Process

The construction process of the improvement solutions consists of three iterations that are illustrated in Figure 5 and described as follows:

- Iteration 1:
  - Step 1: Identifying of the important basic event(s). We identify the important basic event by examining the bars on the indicators of the risk-reduction plot. The basic event “E32” proves to be more important than others.
  - Step 2: Applying design modifications. By viewing the labels of the blobs in the component fault tree view, we know that the basic event belongs to the CFT component “SC1”. According to this, we may easily identify the corresponding hardware component of the system. Based on experience, we decide to replace the identified hardware component with a new part. The cost of this modification amounts to 10 units (see Figure 5: modification “M1”). The structure of the component fault tree is automatically updated according to the modifications performed and a new risk-state node appears on the risk-reduction plot. A solution “S1” is being constructed starting from this modification. By having a closer look at the solution, we come to the conclusion that the overall failure probability is not acceptable yet because the color of the node is not green. Thus, we start the next iteration.
- Iteration 2:
  - Step 1: Identifying of the important basic event(s). The basic event “E3” is identified as the important one.
  - Step 2: Applying design modifications. There are two possible ways to modify the system design for addressing this basic event. One is to add an homogeneous redundant component causing the costs of 11 units (see Figure 5: “M2”). Another one is to use a substitute causing the costs of 34 units (see Figure 5: “M3”). Because of the branches of the modification ideas, a new solution “S2” appears for the branch of “M3”.
  - Step 3: Evaluating the modifications. We compare the cost-effectiveness bars of both created risk-state nodes (referring to item (4)). The modification “M2” is obviously

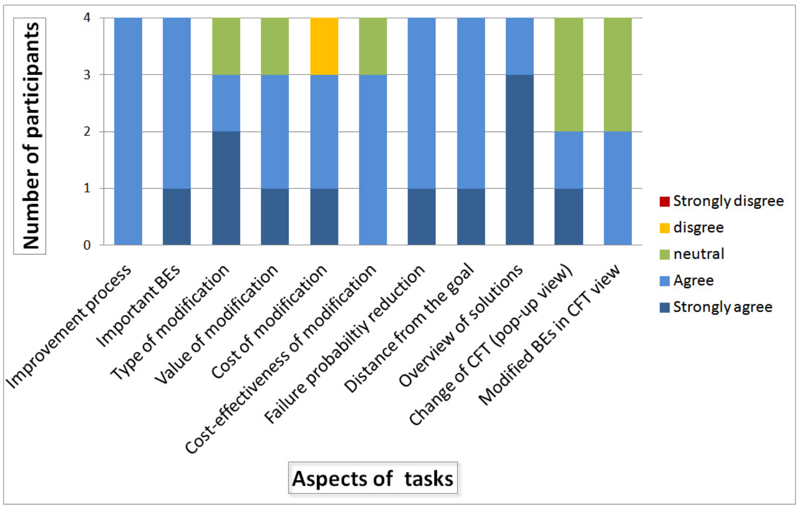


Figure 7 Qualitative evaluation.

more cost-effective. Hence, we abandon "M3" and terminate the corresponding solution "S2" (the last modification step of the solution "M3" is filled with black). The failure probability resulting from the updated component fault tree is not acceptable yet. Thus, we still need to perform the next iteration of risk reduction.

- Iteration 3:
  - Step 1: Identifying of the important basic event(s). We identify two important basic events having similar values.
  - Step 2: Applying design modifications. We apply redundancy-related modifications ("M4" and "M5") for the both basic events. A new solution "S3" appears for the branch generated by "M5".
  - Step 3: Evaluating the modifications. We decide to approve both modifications because the bars of the cost-effectiveness have similar length. The colors of both of the newly created risk-state nodes are now green. This indicates that the risk of the component fault tree is reduced to an acceptable level by applying either "S1" (ending in "M4") or "S3" (ending in "M5"). Because all the possible solutions are identified, we stop the construction process at this iteration.

4.2 Review Process

In this section, we review the solutions in the overview plot (see Figure 5 (3)) for identifying the optimal one. The fact that the total costs of solution "S3" is less than those of "S1" yields that "S3" is the more optimal way to improve the system safety.

5 Evaluation

We have performed an informal evaluation for our visualization approach. We invited four experts of the safety domain from the University of Kaiserslautern, all having profound proficiencies in the field of (component) fault tree analysis. We first introduced our approach to the participants, and then they were allowed to personally experience the visualization functionalities. Tasks with respect to the safety improvement process were provided for the experience. Finally, the participants filled a Likert scale questionnaire for a qualitative evaluation.

The results (see Figure 7) showed that the feedback was mostly positive. The risk-reduction plot was preferred because this visually provided a sequence of modifications, while intuitively presenting the important data of each modification in the same view. When comparing modifications or analyzing patterns, using the plot was more intuitive than investigating data in separate views. The bars for the importance of BEs also had good reviews because they were easy to understand and dynamically linked to the visualization of the modifications.

The risk-state node visualizing the modification data had got a little different opinions. Most complaints concentrated on the small size of the node. The graphic properties attached to the node was too small to be effectively used, particularly the comparison of the cost-effectiveness bars. A suggestion was to apply an interactive fish-eye zoom for the interesting node. A participant commented that a small risk-state node with all graphic properties looked crowded. For example, although the attributes of modifications (i.e., the modification cost, modification type, and modification value) provided significant information for analysis of the existing modifications, the graphical representations of the data did not play an important role when identifying a modification. He suggested to dynamically represent the data: show specific graphic properties only when requested.

The representations for the effects of modifications had good comments. Participants could clearly understand how much the risk reduced by a modification is and how much the actual risk still needed to be reduced. Considering the different points of the analysis view, participants also positively commented the overview plots of the solutions. For the adaptation of the CFT structure, participants commonly thought that the views for showing CFT structure was relative small, whether for the pop-up view or for the main CFT view. The suggestions included a size adjustable pop-up view, and space-efficient alignment between the main CFT view and the risk-reduction plot.

In sum, the invited domain experts preferred our approach because they believed that the proposed visualization methods and interactions could effectively facilitate the identification and analysis of the improvement solutions.

6 Conclusion

A safety-critical system may be improved by a set of design modifications developed by using a component fault tree-based safety improvement process. In case, where multiple design solutions exist the proposed method allows to identify appropriate solutions by taking the actual costs into account. Traditional representation methods separate the information generated in the safety improvement process across individual views which hampers the identification of solutions. We propose a visualization system that integrates all information that is relevant in a risk-reduction plot associated together with a component fault tree view. This allows to quickly identify and review individual design modification steps in the context of different solutions, while considering the optimization of solutions with respect to the cost of modifications. An assumption of our approach is that design modifications do not introduce new critical failures. Otherwise, we would need to apply additional modification steps for the newly introduced important basic events. We also assume that engineers address a vulnerability by only one design modification in a solution. In general, our visualization system supports engineers to identify a series of design modifications leading to an significant improvement of the overall system safety.

7 Acknowledgment

This work was supported by the International Research Training Group (IRTG 1131) of the DFG and the German Ministry of Education and Research (BMBF) in the context of the ViERforES (Virtuelle und Erweiterte Realität für höchste Sicherheit und Zuverlässigkeit von Eingebetteten Systemen) project. We thank our colleagues from the Robotics Research Lab of the University of Kaiserslautern for providing us the data as well as our colleagues of the University of Kaiserslautern and the University of Utah for their support.

References

1 Aldservice. RAMCommander. `http://www.aldservice.com`, accessed 15-May-2012.

2 M. Ankerst, C. Elsen, M. Ester, and H.P. Kriegel. Visual classification: An interactive approach to decision tree construction. *In Proc. 5th Int. Conf. on Knowledge Discovery and Data Mining (KDD '99)*, pages 392-396, 1999.

3 M. Ankerst, M. Ester, and H.P. Kriegel. Towards an effective cooperation of the user and the computer for classification. *In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '00)*, pages 179-188, 2000.

4 A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1):11-33, 2004.

5 T. Barlow and P. Neville. A Comparison of 2-D Visualizations of Hierarchies. *In Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS '01)*, pages 131-138. IEEE Computer Society, 2001.

6 T. Barlow and P. Neville. Case Study: Visualization for Decision Tree Analysis in Data Mining. *In Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS '01)*, pages 149-152. IEEE Computer Society, 2001.

7 S. Contini, L. Fabbri, and V. Matuzas. Concurrent Importance and Sensitivity Analysis Applied to Multiple Fault Trees. *JRC IPSC report, EUR 23825 EN, Ispra.*, 2009.

8 S. Contini, L. Fabbri, and V. Matuzas. A novel method to apply Importance and Sensitivity Analysis to multiple Fault-trees. *Journal of Loss Prevention in the Process Industries*, 3, 2010.

9 S. Contini, S. Scheer, and M. Wilikens. Sensitivity Analysis for System Design Improvement. *In Proceedings of the 2000 International Conference on Dependable Systems and Networks (DSN '00)*, pages 243-248. IEEE Computer Society, 2000.

10 R.B. Cross and J.E. Ballesio. An Integrated Quantitative Risk Assessment of an Oil Carrier. *In Safety and reliability: proceedings of ESREL 2003, European Safety and Reliability Conference 2003, Maastricht, The Netherlands*, 2003.

11 J. Fussell. How to hand calculate system reliability characteristics. R-24:169-174, 1975.

12 A.F. Hixenbaugh and The Boeing Company. Fault Tree for Safety. *D6-53604*, 1968.

13 Functional safety of electrical/electronic/programmable electronic safety-related systems. *International Standard IEC 61508*, 2000.

14 R.L. Iman. A matrix-based approach to uncertainty and sensitivity analysis for fault trees. *Risk Analysis*, 7(1), 1987.

15 ISO. Quality management and quality assurance – Vocabulary. *DIN EN ISO 8402*, 1994.

16 IsographSoftware. FaultTree+. `http://www.isograph-software.com`, accessed 15-May-2012.

17 B. Kaiser, P. Liggesmeyer, and O. Maeckel. A new component concept for fault trees. *In Proceedings of the 8th Australian Workshop on Safety Critical Systems and Software (SCS '03), Adelaide*, pages 37-46, 2003.

18 J. Knight. Safety critical systems: challenges and directions. *In Proceedings of the 24th International Conference on Software Engineering (ICSE '02)*, pages 547-550. ACM, 2002.

19 H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1st edition, 1997.

20 John C. Lee and Norman J. McCormick. *Risk and Safety Analysis of Nuclear Systems*. Wiley, 2011.

21 Y. Liu and G. Salvendy. Interactive visual decision tree classification. *In Proceedings of the 12th international conference on Human-computer interaction: interaction platforms and techniques (HCI '07)*, pages 92-105. Springer-Verlag, 2007.

22 J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5:110-141, 1986.

23 K. B. Misra. *Handbook of Performability Engineering*. Springer-Verlag, 2008.

24 Arbeitsgruppe Software Engineering: Dependability of University of Kaiserslautern. Lecture of Safety and Reliability of Embedded Systems, 2011.

25 Y. Ou and J.B. Dugan. Sensitivity Analysis of Modular Dynamic Fault Trees. *In Proceedings of the 4th International Computer Performance and Dependability Symposium*, page 35. IEEE Computer Society, 2000.

26 N-K Pham, T-N Do, F. Poulet, and A. Morin. Interactive Exploration of Decision Tree Results. *Applied Stochastic Model and Data Analysis International Conference (ASMDA '07)*, pages 152-160, 2007.

27 J. Rausand and A. Hoylany. System reliability theory: models, statistical methods, and applications, pages 183-206, Wiley Inter-Science, 2 edition, 2003.

28 RelexSoftware. RelexArchitect. `http://www.relexsoftware.co.uk`, accessed 15-May-2012.

29 ReliaSoft. BlockSim. `http://www.reliasoft.com/BlockSim`, accessed 15-May-2012.

30 Robotics Research Lab. The Robotics Research Lab of the University of Kaiserslautern. `http://agrosy.informatik.uni-kl.de`, accessed 20-may-2012.

31 Smith, D. J., and Simpson, K. G. L., and ScienceDirect (Online service). *Safety critical systems handbook: A Straightforward Guide to Functional Safety, IEC 61508 (2010 Edition) and Related Standards*. Butterworth-Heinemann, 2010.

32 SyncopationSoftware. DPL-faulttrees. `http://www.syncopationsoftware.com/faulttree.html`, accessed 15-May-2012.

33 S.T. Teoh and K-L Ma. PaintingClass: Interactive Construction, Visualization and Exploration of Decision Trees. *In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03)*, 2003.

34 Stef van den Elzen and J.J. van Wijk. BaobabView: Interactive construction and analysis of decision trees. *IEEE VAST*, pages 151-160, 2011.

35 W.E. Vesely, J. Dugan, J. Fragola, J. MinarickIII, J. Railsback, and M. Stamatelatos. Fault Tree Handbook with Aerospace Applications. *NASA*, 2002.

36 W.E. Vesely, F.F. Goldberg, N.H. Roberts, and D.F. Haasl. Fault Tree Handbook. *U.S.Nuclear Regulatory Commission*, 1981.