ViSSaAn: Visual Support for Safety Analysis

Yi Yang[1], Dirk Zeckzer[2], Peter Liggesmeyer[3], and Hans Hagen[4]

1 **University of Kaiserslautern, Germany**
`yang@informatik.uni-kl.de`
2 **University of Kaiserslautern, Germany**
`zeckzer@informatik.uni-kl.de`
3 **University of Kaiserslautern, Germany**
`liggesmeyer@informatik.uni-kl.de`
4 **University of Kaiserslautern, Germany**
`hagen@informatik.uni-kl.de`

Abstract

Safety of technical systems are becoming more and more important nowadays. Fault trees and minimal cut sets are usually used to attack the problems of assessing safety-critical systems. A visualization system named *ViSSaAn*, consisting of a matrix view, is proposed that supports an efficient safety analysis based on the information from these techniques. Interactions such as zooming and grouping are provided to support the task of finding the safety problems from the analysis information. An example based on real data shows the usefulness of ViSSaAn.

1998 ACM Subject Classification I.3.8 Applications

Keywords and phrases Safety Analysis, Fault Tree Analysis, Minimal Cut Sets, Safety Visualization, Information Visualization

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.378

1 Introduction

Fault Tree Analysis (FTA) [7,13,14,25,40,41] is an analysis technique that is based on the graphical model named *fault tree*. It is widely used for analyzing the safety of technical systems. In order to handle large and complicated fault tree models for complex systems, the Component Fault Tree (CFT) was proposed in [22,23]. The Minimal Cut Sets method (MCSs) [9,12,20,24,25,28,44] is a useful technique for analyzing fault trees. Improving MCS analysis is a good way to improve the safety analysis. For this objective, the following two aspects must be considered: obtaining the MCS analysis information; understanding the obtained information and find safety problems from the information. There are some approaches to obtain the MCS analysis information [9,12,20,24,28,44]. Our research is focusing on the second aspect. In this aspect, representation methods of MCSs are more often considered. We try to find a representation to conveniently, quickly, and correctly understand the MCS information and find problems behind the information, in order to improve the safety of the system. Another challenge is the representation of large-scale data sets. For large systems, users have to face thousands of MCSs and all the applicable data associated to them. Finding the relevant information in a large data set is not an easy task.

Information visualization is a suitable technique that can fulfill these requirements. The visualized data is easier and faster to understand. The problems behind the information can be easily found in the large-scale data sets via visualization. It supports users in making correct decisions as soon as possible. Thus, we focus on how to improve the representation of MCS information using information visualization techniques. The commonly used representation

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 378–395
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

methods for MCS information are textual form, tabular form, and highlighted fault tree paths. They show only the basic MCS information and do not scale to large data sets. A lot of hidden information is not visible. Users have to discover it by themselves. Therefore, a visualization system is needed, that can show hidden information from MCSs and fulfill the complex representation purposes, in order to assist finding the problems conveniently in a large-scale data set.

The visualization system ViSSaAn was implemented according to our research and is presented in this paper. It supports the safety analysis showing MCSs, Basic Events (BEs), and CFTs. A matrix-based representation is used to present the correlation between MCSs and BEs and the correlation between MCSs and CFTs. The safety level method provides a categorization for orders of MCSs, probabilities of MCSs, probabilities of BEs, and probabilities of CFTs. Colors are used to identify safety levels. ViSSaAn provides interactions to support the analysis process for different purposes, such as grouping functionalities that can sort and classify rows and columns of the matrix. Focus&context with semantic zooming is applied to integrate the fault tree structures and their attached data into the matrix view. Users can get detailed information from fault trees inside the matrix view. For the presentation problem of large-scale data sets, ViSSaAn provides DOI zooming and Table zooming, where they can effectively reduce the display space of rows and columns of the matrix view without losing important information.

This paper is structured as follows: In Section 2, we introduce the background of safety analysis, particularly, the Fault Tree Analysis, Component Fault Trees, Minimal Cut Sets, and the safety levels. Related work about representation of MCSs for safety analysis is discussed in Section 3. In Section 4, we introduce the visualization system ViSSaAn, while in Section 5 the interactions in ViSSaAn are presented. An example of ViSSaAn with real data is presented in Section 6. Finally, the discussion of ViSSaAn and conclusions are given in Section 7 and 8.

2 Technical Background

2.1 Basic Concepts

2.1.1 Safety Analysis

Safety is defined as a *State where the danger of a personal or property damage is reduced to an acceptable value* [17]. The safety analysis is a process ensuring that the actual risk is smaller than the acceptable value.

2.1.2 Failure and Fault

Failure and Fault are defined in [15]:

- Failure: *The inability of a system or component to perform its required functions within specified performance requirements.*
- Fault:
  1. A defect in a hardware device or component.
  2. An incorrect step, process, or data definition in a computer program.

2.1.3 Safety-Critical

If the failure of a system could lead to unacceptable consequences and we depend on it for our well-being, then the system is safety-critical [27].

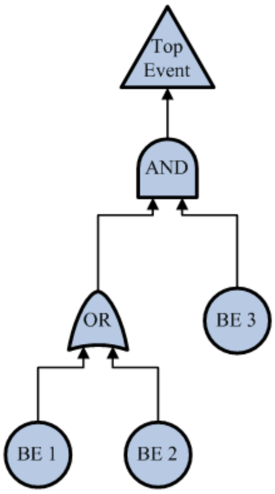


Figure 1 Fault Tree.

2.2 Fault Tree Analysis

Fault Tree Analysis (FTA) [7,13,14,25,40,41] is an effective safety analysis technique for technical systems, which is standardized in [14] and [7]. The kernel of FTA is the fault tree model.

A fault tree is a model that graphically and logically represents the various combinations of possible events, both faulty and normal, occurring in a system that lead to the top undesired event [25].

A fault tree is a graphical model that is presented using a tree structure (see Figure 1). It consists of three kinds of elements:

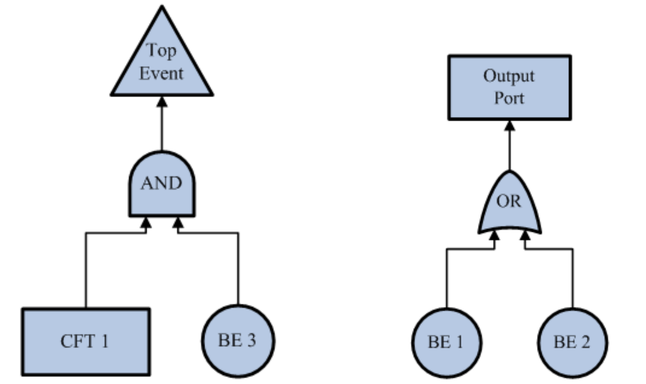
- *Top Event*: root of the tree. It is the top level undesired event.
- *Basic Events (BEs)*: leaves of the tree. They are possible causes of the Top Event. A BE will not be refined any more.
- *Gates*: inner nodes of the tree. They are logical connectives. There are different kinds of gates like the AND gate, the OR gate.

FTA provides some analysis methods in order to analyze fault tree models. Examples are minimal cut sets analysis (see Section 2.4), importance analysis, sensitivity analysis. Using FTA, the safety of systems can be analyzed. Systems can be improved according to the result of FTA.

2.3 Component Fault Trees

For complex technical systems, the fault tree model will be very large and complicated. In order to present the fault tree model more clearly and efficiently, the *Component Fault Tree (CFT)* was proposed in [22, 23]. With the component concept a traditional fault tree of a system can be divided into one or more independent components (see Figure 2). Each component is called a CFT. CFTs are connected amongst each other forming an overall CFT model for a system. The difficulty of FTA is reduced with CFTs.

We give a brief example of CFTs. The fault tree in Figure 1 can be transformed into the CFT depicted in Figure 2. It consists of a main model *CFT0* (see Figure 2 (a)) and a sub component *CFT1* (see Figure 2 (b)). CFT1 consists of the BE1 and the BE2 as well as an



(a) Main component: CFT0 **(b)** Sub component: CFT1

Figure 2 Component Fault Tree.

OR Gate. In CFT0, CFT1 is treated as a black box and used as a part. The CFT concept was implemented in an FTA tool *ESSaREL* (previous name: *UWG3*) [8,21].

2.4 Minimal Cut Sets

After constructing a fault tree, a method is needed to analyze this model. The *Minimal Cut Sets (MCSs)* method [9,12,20,24,25,28,44] is a useful technique for analyzing fault tree models.

A cut set for a fault tree is a set of basic events whose occurrence causes the Top Event to occur. A cut set is said to be a minimal cut set if, when any basic event is removed from the set, the remaining events collectively are no longer a cut set [25].

The number of different BEs in a minimal cut set is called order of the cut set [32]. Order is also called *size*, e.g., in [25].

For example, in Figure 1 and Figure 2 the MCSs of the fault tree are: MCSs = {MCS1, MCS2}, with MCS1 = {BE1, BE3}, MCS2 = {BE2, BE3}. The order of MCS1 is 2, because it contains two BEs.

With the MCS method users can perform both qualitative analysis and quantitative analysis for fault tree models. Qualitative analysis is used for finding BE combinations that are able to cause the occurrence of the top event. Quantitative analysis is used for calculating the probability of the top event from influencing probabilities. The MCSs method is suitable for CFT analysis as well.

2.5 Safety Levels

The safety level method can be used to estimate the safety states. It provides different quantitative levels to categorize safety. In this paper, we use a simple 3-level criterion for safety that can be extended to more complicated safety level methods. In this criterion, safety can be classified into three levels:

- critical level: It is dangerous and urgent to be solved.
- moderate level: It needs to be solved, but not urgently.
- acceptable level: It is below the acceptable value, i.e., it is not dangerous.

The safety level criterion can be applied to MCSs, BEs, and CFTs. The safety levels of MCSs depend on either the order of the MCS or the failure probability of the MCS. The order of a MCS is inversely proportional to its safety. Basically, a MCS with order 1 is critical for a system. In this situation, the top event will occur by only one BE. The trigger condition is easy to be achieved. The higher the probability of a MCS is, the more dangerous is the MCS. The safety levels of BEs depend on the failure probability of BEs. The higher the probability of a BE is, the more critical is the BE. The same is true for CFTs.

3 Related Work

Currently, most safety analysis tools provide the MCS analysis for FTA. The ordinary representation of MCSs is a textual/tabular form. Some ideas combine a textual/tabular form with highlighted paths on fault tree diagrams. The textual form only lists the MCSs and the contained BEs with simple information. The tabular form is a bit complicated. It provides a table to represent the MCSs and the related information. Some advanced functionalities, such as sort and filter, are possible in the tabular form. Associating to fault tree diagrams helps users to understand the MCS information more clearly.

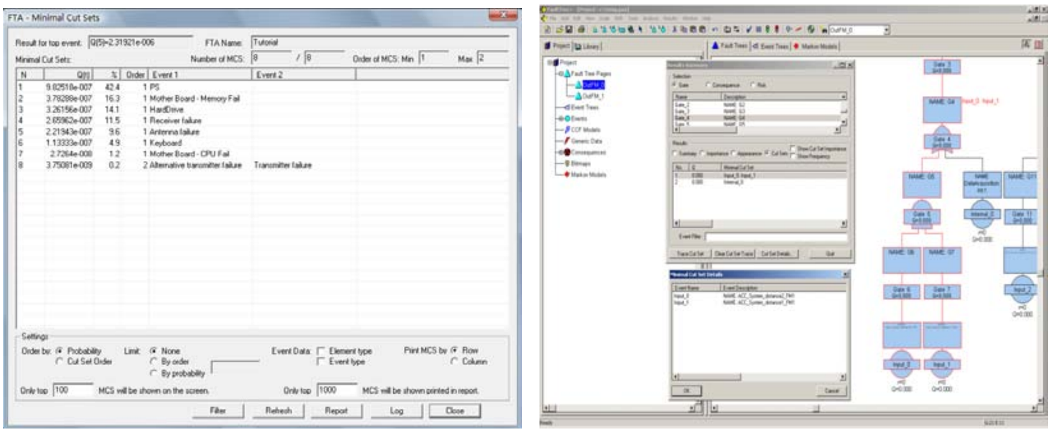
The tool *ESSaREL* [8] gives a textual MCS list. It shows each MCS with its ID and its BEs with BE IDs and BE labels. The MCSs are sorted by order. It also provides some general information, such as the ID of top event, generation date, path of data file, and count of MCSs. The tool *BlockSim* [34] shows the similar information. In additional, it shows probability of the MCSs. It provides filters for the order of MCSs and can sort MCSs by size, reliability, or unreliability. The tool *RAMCommander* [2] (see Figure 3(a)) provides a table for MCS information. Rows are MCSs. Besides the IDs of MCSs, the table also shows probability, contribution, and order of MCSs in the first four columns. The possibly used BEs are in subsequent columns. Sorting MCSs by ID, order, probability, or contribution can be performed in the table. Filtering by order is available to cut out the MCSs having an order outside the filter range. The "Element type" and "Event type" of BEs will be selectively shown in the table. Besides, it shows some common information, e.g., name of the FTA, probability of top event, count of MCSs, statistical information about the order of MCSs. The popular FTA tool *FaultTree+* [18] (see Figure 3(b)) also provides a tabular form for MCS information. In *FaultTree+*, the MCS table and the table for BEs use are separated. When selecting a MCS in the MCS table, the BEs used will be displayed in the BE table that can show a description and more detailed information of BEs. *FaultTree+* provides a filter to BEs. It also provides choices to show the importance and frequency of MCSs. It reflects the MCS information on the fault tree diagram using highlighting the paths that lead from the BEs of the MCS to the top event. It displays how the top event is reached from the BEs. Other tools like *Relex Architect* [33], *DPL faulttrees* [38], *FSAP/NuSMV-SA* [5], and *ITEM ToolKit* [19] provide similar representation methods for MCS information.

4 Visualization

4.1 Visualization Requirements

According to the introduction described in Section 1, the correlation between MCSs and BEs is essential and needs to be visualized. The MCSs can be evaluated using some properties, e.g., probability, and the result is then visualized. The component concept of FTA is also required to be represented, because it makes the system fault tree model easier to understand.

A suitable idea for large-scale data sets is needed. In consideration of the significance of the typical fault tree structure, it will be represented in the visualization system. Users can make sure how the MCSs effect the top event. More visual factors can be considered, if they are helpful for the representation of the MCS analysis information.



FTA - Minimal Cut Sets

Result for top event: Q(5)=2.31921e-006 FTA Name: Tutorial

Minimal Cut Sets: Number of MCS: 8 / 8 Order of MCS: Min 1 Max 2

N	Q(t)	%	Order	Event 1	Event 2
1	9.82518e-007	42.4	1	PS	
2	3.78288e-007	16.3	1	Mother Board - Memory Fail	
3	3.26156e-007	14.1	1	HardDrive	
4	2.65962e-007	11.5	1	Receiver failure	
5	2.21943e-007	9.6	1	Antenna failure	
6	1.13333e-007	4.9	1	Keyboard	
7	2.7264e-008	1.2	1	Mother Board - CPU Fail	
8	3.75081e-009	0.2	2	Alternative transmitter failure	Transmitter failure

Settings

Order by: ● Probability ○ Cut Set Order

Limit: ● None ○ By order ○ By probability

Event Data: □ Element type □ Event type

Print MCS by: ● Row ○ Column

Only top 100 MCS will be shown on the screen.

Only top 1000 MCS will be shown printed in report.

(a) Tabular form for MCSs (RAMCommander [2])

(b) Combination of tabular form and path highlighting (Faulttree+ [18])

Figure 3 MCSs representation methods of current tools.

4.2 ViSSaAn

Based on the requirements we developed a visualization system named *ViSSaAn (Visual Support for Safety Analysis)*. It was developed using Java [37] and the visualization library Prefuse [30]. There are two frames in ViSSan: the configuration frame *Main Control* and the analysis frame *MCS Matrix*.

4.3 Color Encoding

Colors can be used as a nominal code into classify objects to different categories. Color encoding for nominal information was presented, e.g., in [6, 26, 43]. We use colors in ViSSaAn according to the 3-level safety criterion described in 2.5: red (critical level), yellow (moderate level), and green (acceptable level).

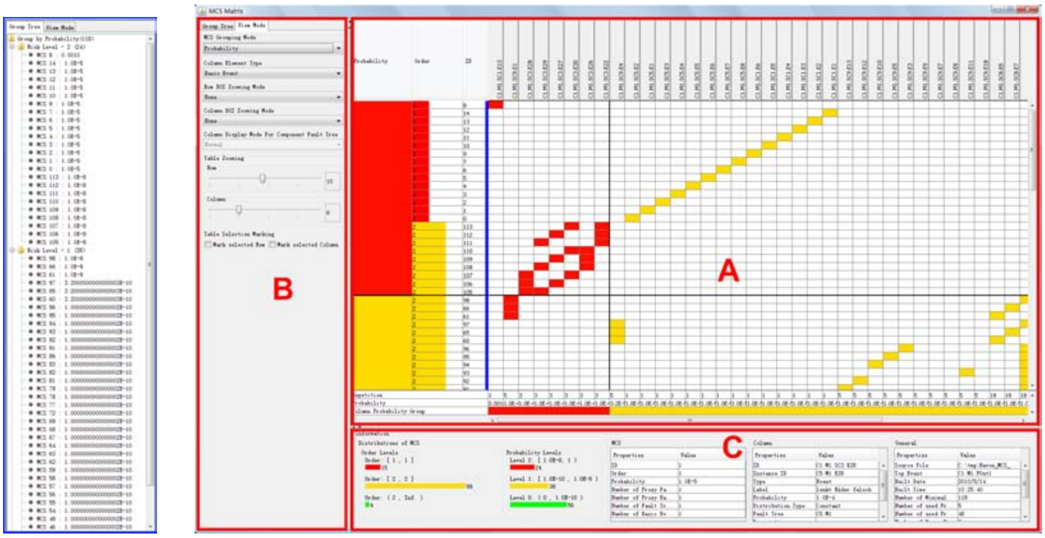
4.4 Main Control

Before starting the MCS analysis, the environment must be configured. *Main Control* is the configuration frame of ViSSaAn. With the help of the *Main Control* the data files can be loaded and users can specify the value ranges for the safety levels.

4.5 MCS Matrix

The tabular form is a useful method for MCS information. But it can not fulfill complex situations for MCS analysis, for example, the component concept of FTA, safety levels.

ViSSaAn provides an interactive matrix based representation to satisfy the advanced purposes. It is the main analysis view of the ViSSaAn. It is used to present correlation between MCSs and BEs/CFTs. The count of MCSs will be more than several thousands for a large fault tree model for a complex system. The matrix visualization is suitable for large-scale data. It is intuitive, clear, and good to represent large-scale data sets (see e.g., in [10] and [39]). After providing the settings in the *Main Control*, the analysis frame *MCS Matrix* will get started. *MCS Matrix* provides three view areas.



(a) Interaction panel (*Group Tree*) **(b)** part A: matrix view; part B: interaction panel (*View Mode*); part C: information panel

Figure 4 MCS Matrix.

- The first part is the matrix view (see part A of Figure 4 (b)). It is the central part of the *MCS Matrix*. It shows the correlation between MCSs and BEs or between MCSs and CFTs.
- The second part is an interaction panel (see part B of Figure 4 (b)). It provides interactions for the matrix view that are used for finding the most valuable information for particular purposes.
- The third part is an information panel (see part C of Figure 4 (b)). It provides instant and general information of the elements represented in the matrix view.

4.5.1 Matrix View

The matrix view is the central part of the *MCS Matrix*. Safety levels of MCSs, BEs, and CFTs are coded by colors. There are three areas in the matrix view (see Figure 5). Rows present the MCSs in the first and in the second area. The third area is at the bottom. It shows the probability and the number of repetitions of each BE or CFT (see area 3 of Figure 5). The first area consists of three columns: the MCS ID, the MCS Order, and the MCS Probability (see area 1 of Figure 5). In order to present the order of MCSs more intuitively, a bar graph is used to visualize the order of a MCS. The applications of bar graphs in a table view were introduced in [29, 31]. The larger the order, the longer the bar. Users can read the exact textual value in the bars. The second area starts at the 4th column (see area 2 of Figure 5). There are two types of columns: the *Basic Event* column and the *Component Fault Tree* column. Types can be freely switched for different purposes.

For the type *Basic Event* (see Figure 5), each column of the second area represents a BE. A colored cell indicates that a BE in the current column is used by the MCSs in the current row. The colors represent the safety levels of the BEs. The IDs of the BEs are printed on the column head. For the type *Component Fault Tree* (see Figure 6), each column of the second area represents a CFT. The IDs of CFTs are printed on the column head. Each

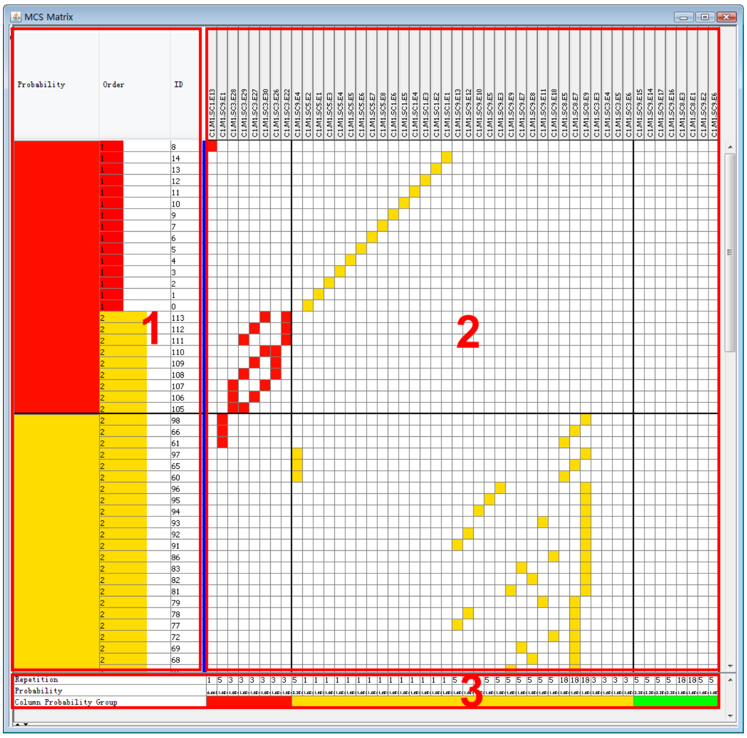


Figure 5 Matrix View – area 1: for MCS; area 2: for CFTs/BEs; area 3: for probability and number of repetitions. (Column Element Type: *Basic Event.*)

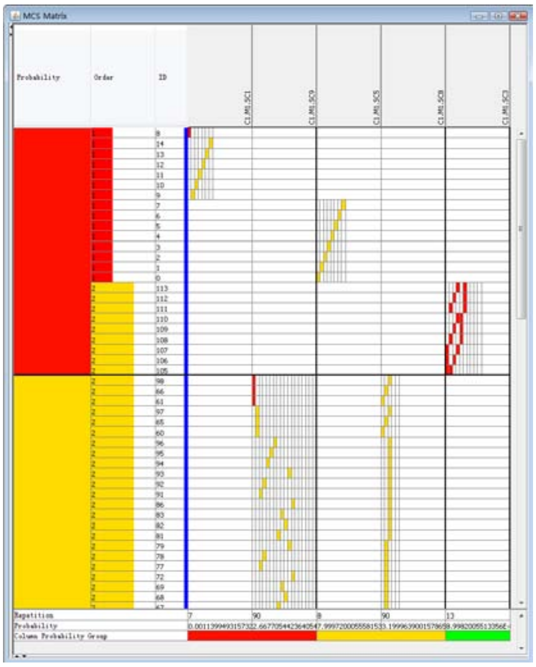


Figure 6 Column Element Type: *Component Fault Tree.*

non-empty cell contains at least one color filled sub-cell. The sub-cells indicate the contained BEs that are sorted by probability in descending order. A color filled sub-cell represents the correlation between inner BEs and the MCS in the current row. An empty cell indicates that no BE of the CFT in the current column is used by the MCSs in the current row. This type can be regarded as another kind of grouping for BEs. In this case, BEs are grouped to different CFTs and they are sorted by probability inside each CFT.

The safety levels are available for different properties, such as probability and order of MCSs. In order to analyze the safety levels of a specified property, ViSSaAn provides the grouping functionalities, where row grouping and column grouping are provided. The row grouping groups and sorts MCSs by the specified property and gathers the elements with the same level. Each group represents an aggregation of elements having the same safety level. There are two grouping modes for MCSs: *Order* and *Probability.* The related interaction will be introduced in 5.1. The column of the grouping property will always be set to the first column. Cells of the first column are merged into blocks according to the groups. The colored blocks can be treated as indicators of different safety levels as well. The column grouping groups and sorts the columns in the second area by probability of BEs or CFTs in descending order. The last row of the third area holds colored blocks as the indicators of the column groups.

Coloring associated to the grouping functionalities identifies the safety levels. Colors are more intuitive than text values, so that the safety levels of the elements are easily to be identified. In cooperation with the grouping functionality, the elements at the same level will be put together and sorted. It speeds up the process of finding the serious problems.

4.5.2 Interaction Panel

The interaction panel gathers the interactions of ViSSaAn that cannot be performed directly in the matrix view. There are two views in the interaction panel: *Group Tree* (see Figure 4 (a)) and *View Mode* (see part B of Figure 4 (b)). The *Group Tree* provides a tree structure for MCSs. The *View Mode* provides some view modes and zooming modes for the matrix view of *MCS Matrix.* The view modes "Column Element Type" have been introduced in Section 4.5.1. The interaction of *Group Tree* and other modes of *View Mode* will be described in Section 5.

4.5.3 Information Panel

The information panel presents the general information about the analyzed data set and the matrix view related data (see part C of Figure 4 (b)). There are four information areas:

- The first information area presents distributions of MCSs. This statistical information provides an overview of the safety state of the system. Users can have a rough idea of the safety of the system.
  - The first diagram visualizes the distribution of order levels. It displays how many MCSs are in each safety level for the order of MCSs.
  - The second diagram visualizes the distribution of probability safety levels.
- The second information area shows information about the current column, either for a CFT or for a BE. It supports to get instant information of BEs/CFTs without having to check fault tree diagrams.
- The third information area gives information about the selected row, i.e., a selected MCS. It gives detailed information about the current MCS.
- The last information area provides statistical information of the analyzed fault tree model. Such as count of BEs, count of MCSs.

5 Interaction with MCS Matrix

Interaction is an important aspect of a visualization system. It helps users to explore information. In this section, the interactions with the *MCS Matrix* is introduced. In the *MCS Matrix* the interaction "Semantic Zooming" is performed in the matrix view directly while others can be performed by using the interaction panel. Some view modes in the interaction panel have been described in Section 4.5.1. The rest of the interaction panel are "MCS Grouping Mode", "Column Display Mode", "Table Zooming", 'Table Selection Marking", and "Group Tree". These are introduced in this section.

5.1 MCS Grouping Mode

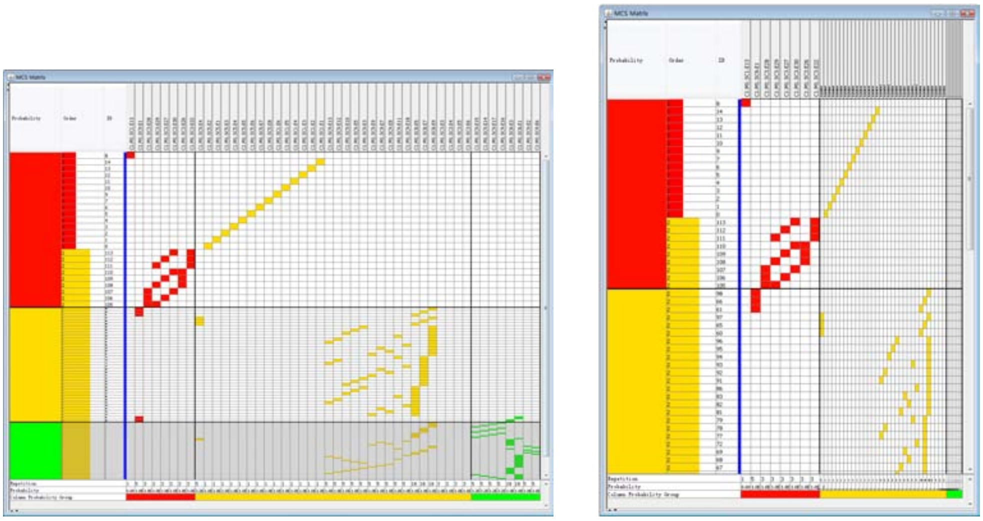
In 4.5.1 the grouping functionality has briefly been mentioned. In this paragraph, the detailed features of this interaction are introduced. "MCS Grouping Mode" is used to sort and group MCSs into different safety levels. There are three modes: *ID*, *Order*, and *Probability*. For *ID*, MCSs are displayed in an ascending sequence sorted by ID. All MCSs are classified in one group. When choosing *Order*, MCSs are grouped by order. The probability information will put on the second column. When choosing *Probability*, MCSs are grouped by probability according to the range defined for each safety level. MCSs are sorted in descending order in each group and the column *probability* will be put in the first place. The order information will be put in the second column. The MCS grouping function is useful to get the most important MCSs for different objectives.

5.2 DOI Zooming

There are two important zooming interactions in ViSSaAn, degree of interest (DOI) zooming described next and semantic zooming being described in Section 5.3.

When the count of MCSs and/or BEs is huge, the representation will be a challenge for MCS analysis. As much information should be shown in a limited amount of space as possible, particularly the interesting information. The DOI zooming is designed to meet this demand. DOI zooming is a zooming technique whose scale depends on the *degree of interest (DOI)*. The DOI distortion was proposed in [11] and used in Table Lens in [29, 31].

In ViSSaAn the DOI depends on the safety levels. The more critical the safety situation, the higher the interest and the larger the zoom scale. Each group has a zoom scale, because groups have one-to-one mappings to safety levels. Therefore, the display size of groups depends on the zoom scale that can be obtained with the help of the safety levels. Basically, the idea of DOI zooming is to compress the display space of uninteresting information, in order to guarantee the display of the overview of a large-scale data set. We assigned following scales to the zooming levels: the group for the critical level has zoom scale "1", the group for the moderate level has zoom scale "0.3", and the group for the acceptable level has the smallest zoom scale "0.15". The display size of a group for critical level remains unchanged. The remaining groups get smaller display space according to their levels. The DOI zooming consists of the "Row DOI Zooming Mode" and the "Column DOI Zooming Mode". By the "Row DOI Zooming Mode", different row heights are used for different MCS groups (see Figure 7 (a)). If a row is in the group for the critical level, it will have the original height. If it is in the group for the moderate level, it will have a smaller height. If it is in the group for the acceptable level, it will have the smallest height. The "Column DOI Zooming Mode" is very similar to the "Row DOI Zooming Mode". By applying the "Column DOI Zooming Mode" the grouped columns have different width because of the different zoom



(a) Row DOI Zooming - rows of different groups have different height

(b) Column DOI Zooming - columns for different safety levels have different width

Figure 7 DOI Zooming.

scales. By combining both DOI zooming methods, the display space is reduced both for rows and for columns. The column elements that are used by the critical MCSs and those that are classified in the critical level have the largest cells (see Figure 8).

In this way, there is enough space to show the important information, and the less important information is shown as context. Therefore, DOI zooming efficiently shows the large MCS information in a limited screen space. At the same time, DOI zooming highlights critical MCSs and critical column elements. The effect is shown in the Figure 8.

5.3 Focus&Context with Semantic Zooming

The MCS analysis information is often associated to fault tree diagrams. The fault tree structure contains a lot of useful information for safety analysis, such as the path from a BE to the top event or the logical connective between BEs. Users can learn how BEs effect the top event. Usually, users have to turn to fault tree diagram from the MCS analysis view to see the detailed information of the fault tree, e.g., the structure of the fault tree or the attached data. This is not convenient and it is easy to miss some context in the MCS information. In order to solve this problem, the fault tree structure should be integrated in the matrix view. The component concept is a powerful tool which needs to be reflected in ViSSaAn. Analyzing the CFTs is more efficient than checking the whole fault tree. Therefore, integrating CFT structure with corresponding data into the matrix view becomes an important task.

Focus&Context with semantic zooming is applied in ViSSaAn to accomplish this task. It was investigated in the projects *Pad, Pad++* [3], and *Jazz* [4], and was evaluated for program visualization in [36]. It was used for the matrix view in [1] as well. With Focus & Context more detailed content of the focused element can be shown dynamically without losing context information. In this paper the technique Focus&Context with semantic zooming is called semantic zooming for convenience.

ViSSaAn provides semantic zooming to access the structure of CFT. If a cell is double

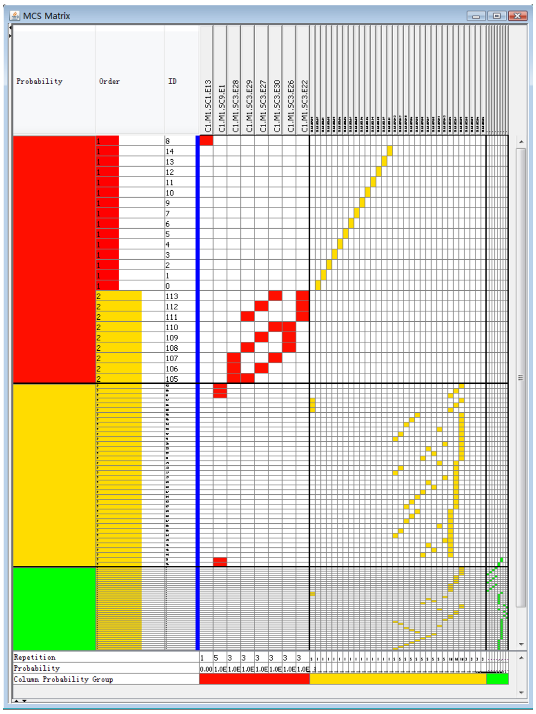
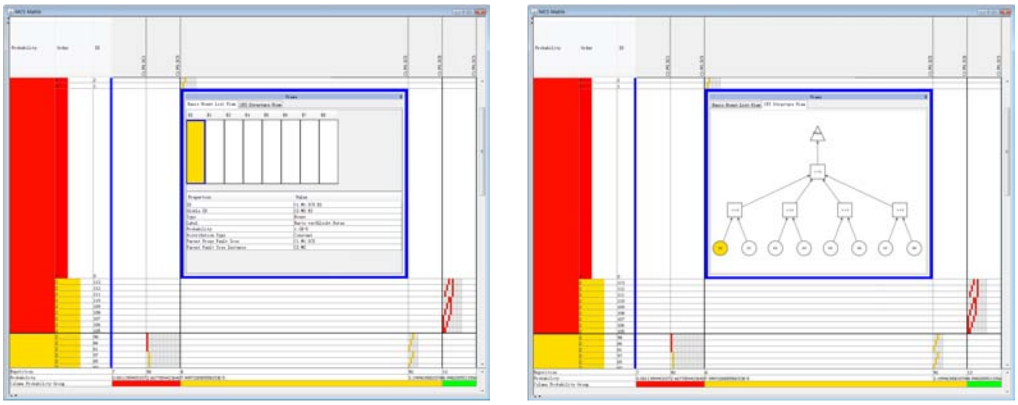


Figure 8 Row DOI Zooming + Column DOI Zooming - The most critical BEs are presented by the columns having the largest cells.

clicked, the cell will zoom out. There will be two views accessible using tabs in the zoomed cell. The first one is the *Basic Event List View*. It consists of a list of blocks and a data table (see Figure 9 (a)). Each block represents a BE in the selected CFT. The BEs represented by color filled blocks are used by the current MCS. The colors are according to the safety levels of the BEs. When a block is selected, the data table will display the data of the corresponding BE. This view gives the detailed data of the current CFT. The second one displays the *CFT Structure View* (see Figure 9 (b)). It shows a traditional structure of current CFT with colored BEs. Like in the *Basic Event List View*, the colors depend on the safety levels. In this view, only the BEs that are used by the MCS in current row are colored. With semantic zooming the detailed information of a CFT can be obtained without losing the context information of MCS analysis. The semantic zooming is similar for the "Column Element Type" *Basic Event*, but there is only the *CFT Structure View*. Pan, Zoom, and ZoomToFit are also provided as common interactions for the node-link diagrams in the *CFT Structure View*.

5.4 Column Display Mode

The "Column Display Mode" is available when using the *Component Fault Tree* mode. It provides an additional *Compact Mode* . By default, columns have the same width allowing the CFTs to be distinguished more clearly. Using *Compact Mode*, the width of a column depends on the number of the BEs contained in the CFT. Columns might have different width. As mentioned before, the column element type "Component Fault Tree" can be treated as the grouping of BEs according to CFTs. During *Compact Mode*, the BEs look more continuous, so that it is more reliable to check BEs grouped by CFTs. It reduces the column display space as well.



(a) Basic Event List View - a list of BEs is in the upper part; a data table is in the lower part

(b) CFT Structure View - *CFT Structure* integrated in the matrix view

Figure 9 Semantic Zooming.

5.5 Table Zooming

Usually, the space of rows and columns will be compressed in order to adapt to screen space. ViSSaAn provides "Row Zooming" and "Column Zooming" for obtaining a suitable overview of MCSs. These functionalities can zoom out / zoom in row height and column width. The zoom scale can be changed flexibly. This is a simple idea, but it is effective for representing the large-scale data set.

5.6 Table Selection Marking

The "Table Selection Marking" points out the selected row or /and column with a light-gray horizontal rectangle or / and vertical rectangle. There are two marking modes for table selection, the "Mark Selected Row" and the "Mark Selected Column". It helps to highlight the selected cell in a large matrix.

5.7 Group Tree

When performing zooming functionalities for representing a large number of MCSs in *MCS Matrix*, it will be hard to locate the desired MCS, because the rows are strongly compressed and the indicators of MCSs. In this case, *Group Tree* will show its usefulness. The *Group Tree* and the *matrix view* are coordinated views. They show the sorted and grouped MCSs in a tree structure. The *Group Tree* is associated with the MCS grouping modes, so that the information of the MCS on the tree is according to the grouping property. The *Group Tree* is helpful for positioning MCSs (see Figure 4 (a)).

6 Example

6.1 Data

We present how CFT models of real data can be analyzed using ViSSaAn. *RAVON (Robust Autonomous Vehicle for Offroad Navigation)* is a mobile robot made by the Robotics Research Lab of the University of Kaiserslautern [35]. It is used as an application example by the project *ViERforES* [42]. *RAVON* is a typical large and complex embedded system, and

Table 1 Settings of the example

	Acceptable	Moderate	Critical
BE Range	$(0, 1e^{-5})$	$[1e^{-5}, 1e^{-4})$	$[1e^{-4}, 1)$
BE Color	green	yellow	red
MCS Range	$(0, 1e^{-10})$	$[1e^{-10}, 1e^{-8})$	$[1e^{-8}, 1)$
MCS Color	green	yellow	red

safety is important for its survival. The safety of *RAVON* is analyzed with the CFT analysis using *ESSaREL*. These CFT models of *RAVON* are then explored using ViSSaAn. For our example the Top Event "C1.M1.POut1" of the CFT "Main, Sensoren, Aktoren" is selected in a CFT Model. There are 118 MCSs and 48 BEs contained in 5 CFTs for this top event.

6.2 Settings

After loading the data, the value ranges of probabilities of BEs, the range of probabilities of MCSs, the range of orders of MCSs, and the colors for the ranges are specified (see Table 1).

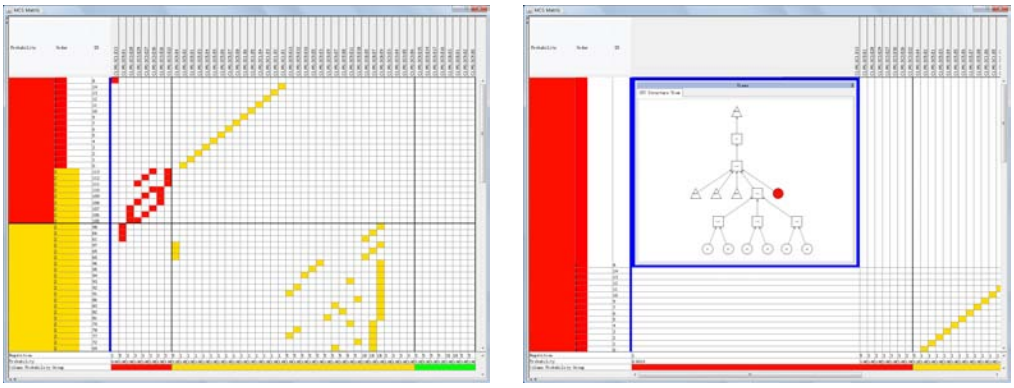
6.3 Scenario

This scenario concerns to explore the important safety problems from the MCS analysis information. The result of the scenario is presented in the Figure 10(a). In the *MCS Matrix*, the "MCS Grouping Mode" is set to *Probability*, in order to group MCSs according to different safety levels by probability. The group with red is the one with a "critical level". Next, we check the column for order of MCSs. There are 15 MCSs with order 1 and 9 MCSs with order 2 in the "red group". Because the MCSs with order 1 is more critical than ones with 2, we just focus on the MCSs having one BE in the group. Then, we check the sorted BEs, the BEs in red are critical. Now, we associate the consideration of critical MCSs found before, only the MCS with ID "8" is found to be the most critical MCS, because its BE "C2.M1.SC1.E13" is colored in red, i.e., this BE is critical. The rest critical MCSs found before have BEs in yellow. Thus, The BE "C2.M1.SC1.E13" is the declared important safety problem. Then, we can perform semantic zooming for the cell at the intersection between the row for the MCS with ID "8" and the column for the BE "C2.M1.SC1.E13" in order to check the fault tree diagram to find how this BE effects the Top Event (see Figure 10 (b)). The obtained information helps making a decision to solve the problem. We can also use DOI zooming to show as much information as possible on the screen (see Figure 8). The DOI zooming plays a role with highlighting of the important field as well.

7 Discussion

ViSSaAn is a multi-view visualization system for MCS analysis. It extends the common tabular form to an interactive matrix representation with suitable visual features. The visual features, e.g., coloring, represent the MCS information intuitively.

Generally, the correlation between MCSs and their BEs are the focus for qualitative analysis. In consideration of the contribution of the component concept for the FTA, ViSSaAn also visualizes the involved CFTs in the *MCS Matrix*. ViSSaAn combines the CFT structure and the matrix view by focus&context with semantic zooming interaction. The details of the



(a) The MCS in the first row is most critical. The BE inside is the critical problem

(b) Semantic zooming for the critical BE found

Figure 10 Results of the example.

related CFT can be check, without switching to another fault tree diagram. In this situation, the focus is not interrupted and the context is not missing. This interaction makes checking the CFT structure more convenient and efficient. ViSSaAn uses safety levels for classifying MCSs, BEs, and CFTs. A safety level is assigned to each element according to the specified property. It helps finding the urgent problems efficiently and precisely in a large data set. With the help of coloring, the safety levels are easier to identify. The grouping functionalities aggregate elements with the same safety level, so that users can easily find them. Sorting is also a characteristic of ViSSaAn. Each MCS has a sorted BEs sequence, when the order of the MCS is more than one. Therefore, it is easy to distinguish which is the most important one. Zooming functionalities help analyzing large-scale data sets. Row and Column Zooming can change the size of rows and columns. With DOI zooming, the important information has priority using the screen space. By these zooming techniques, as much information as possible can be shown in the limited screen space.

MCS Matrix is not a simple extension of the tabular form for MCSs information, but an integrated system. The statistical information can be used to understand the safety situation of systems. Combining it with a couple of data tables in the information panel, makes it convenient to get the information of MCSs, BEs, and CFTs in the *MCS Matrix*. ViSSaAn was designed to improve the MCS analysis from the angle of view of information representation. There is no optimization for algorithms and processes. It uses visualization techniques to improve the understandability of the MCS analysis and the readability of large-scale data sets. ViSSaAn provides an intuitive, convenient, and rapid visualization system to support the safety analysis using MCSs. It helps to understand the information analyzed to find the safety areas of a system, and can be used in support for decision making.

8 Conclusions

This paper introduced ViSSaAn, a visualization system for MCSs and CFTs. A matrix-based visualization called *MCS Matrix* is used to present MCS analysis information. Colors are used to encode different safety levels. Grouping functionalities are used to group the elements with the same safety levels. DOI zooming and Table zooming are used for large-scale data sets. Semantic zooming is used in the *MCS Matrix* in order to show details and the inner structure of CFTs. Users can focus on the detailed information while having the MCS information as context. With ViSSaAn users can better understand and analyze fault tree models. Finding safety problems from the MCS analysis information is convenient. Overall, ViSSaAn provides methods to visualize MCSs information improving the MCS analysis for fault trees of complex systems in representation aspect. In the future, ViSSaAn will be extended in several ways, including considering new visual metaphors and interactions. More safety level criteria will be considered.

Acknowledgements

This paper is supported by the International Research Training Group of DFG (IRTG 1131). We thank the Robotics Research Lab of the University of Kaiserslautern [35], the Fraunhofer Institute for Experimental Software Engineering (IESE) [16], and our colleagues for the data, their help, and their comments.

References

1 James Abello and Frank van Ham. Matrixzoom: A visual interface to semi-external graphs. *IEEE Symposium on Information Visualization (INFOVIS04)*, 2004.

2 ALD. RAM Commander. `http://www.aldservice.com`, Online; accessed 15-July-2010.

3 B.B. Bederson, J.D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas. A Zoomable Graphical Interface for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 1:3–31, 7 1996.

4 B.B. Bederson, J. Meyer, and L. Good. Jazz: an extensible zoomable user interface graphics toolkit in Java. *Proceedings of the 13th annual ACM symposium on User interface software and technology (UIST'2000), San Diego, CA*, pages 171–180, 2000.

5 Marco Bozzano and Adolfo Villafiorita. The FSAP/NuSMV-SA Safety Analysis Platform. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(1):5–24, 2007.

6 Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in information visualization: using vision to think.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

7 DIN 25424. Fehlerbaumanalyse (Fault Tree Analysis). *German Industry Standard (Part 1 & 2)*, 1981/1990. Beuth Verlag, Berlin.

8 ESSaREL. ESSaREL. `http://www.essarel.de`, Online; accessed 15-July-2010.

9 Nasser S. Fard. Determination of minimal cut sets of a complex fault tree. *Computers & Industrial Engineering*, 33(1-2):59–62, 1997. Proceedings of the 21st International Conference on Computers and Industrial Engineering.

10 J.-D. Fekete, N. Elmqvist, T.-N. Do, H. Goodell, and N. Henry. Navigating with the Zoomable Adjacency Matrix Explorer. *Technical report, INRIA Research Report (Paris)*, RR-6163, 1997.

11 George W. Furnas. Generalized fisheye views. *In Proceedings of the ACMSIGCHI Conference on Human Factors in Computing Systems*, pages 16–23, 04 1986.

12 J.B. Fussel and W.E. Vesely. A New Methodology For Obtaining Cut Sets From Fault Trees. *ANS Trans*, 15, 1972.

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

14 IEC61508. Functional safety of electrical/electronic/programmable electronic safety-related systems. *International Standard IEC 61508*, 2000.

15 IEEE. IEEE Std 610.12-1990. *IEEE Standard Glossary of Software Engineering Terminology*, 1990.

16 IESE. Fraunhofer Institute for Experimental Software Engineering (IESE). `http://www.iese.fraunhofer.de`, Online; accessed 15-July-2010.

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

18 ISOGRAPH. FaultTree+. `http://www.isograph-software.com`, Online; accessed 15-July-2010.

19 ITEM-Software. ITEM ToolKit. `http://www.itemtoolkit.com/`, Online; accessed 15-July-2010.

20 J.Vatn. Finding minimal cut sets in a fault tree. *Reliability Engineering & System Safety*, 36(1):59–62, 1992.

21 Bernhard Kaiser. Integration von Sicherheits- und Zuverlaessigkeitsmodellen in den Entwicklungsprozess eingebetteter System. *Softwaretechnik-Trends 22(4) Gesellschaft fuer Informatik (Hg.)*, 2002.

22 Bernhard Kaiser. A fault-tree semantics to model software-controlled systems. *Softwaretechnik-Trends 23(3) Gesellschaft fuer Informatik (Hg.)*, 2003.

23 Bernhard Kaiser, Peter Liggesmeyer, and Oliver Maekel. A New Component Concept for Fault Trees. *Proceedings of the 8th Australian workshop on safety critical systems and software (SCS' 03)*, 2003.

24 Chakib Kara-Zaitri. An improved minimal cut set algorithm. *International Journal of Quality & Reliability Management*, 13, 1996.

25 Dimitri Kececioglu. *Reliability Engineering Handbook*, volume 2. DEStech Publications, Inc, 1991.

26 Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors. *Information Visualization: Human-Centered Issues and Perspectives.* Springer-Verlag, Berlin, Heidelberg, 2008.

27 John Knight. *Safety critical systems: challenges and directions, ICSE '02: Proceedings of the 24th International Conference on Software Engineering.* ACM, New York, NY, USA, 2002.

28 Kyoichi Nakashima and Yoshio Hattori. An efficient bottom-up algorithm for enumerating minimal cut sets of fault trees. IEEE Trans. Reliab., R-28, (5) 353 (December 1979). *Microelectronics and Reliability*, 20(4):543–543, 1980.

29 Peter Pirolli and Ramana Rao. Table lens as a tool for making sense of data. *Proceedings of the workshop on Advanced visual interfaces*, pages 67–80, 1996.

30 Prefuse. Prefuse. `http://prefuse.org/`, Online; accessed 15-July-2010.

31 Ramana Rao and Stuart K. Card. The Table Lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI '94)*, 1994.

32 Marvin Rausand and Arnljot Hoyland. *System Reliability Theory: Models, Statistical Methods, and Applications, Second Edition.* Wiley-Interscience, 2003.

33 RELEXSOFTWARE. Relex Architect. `http://www.relexsoftware.co.uk`, Online; accessed 15-July-2010.

34 RELIASOFT. BlockSim. `http://www.reliasoft.com/BlockSim`, Online; accessed 15-July-2010.

35 Robotics Research Lab. The Robotics Research Lab of the University of Kaiserslautern. `http://agrosy.informatik.uni-kl.de`, Online; accessed 15-July-2010.

36 Kenneth L. Summers, Timothy E. Goldsmith, Steve Kubica, and Thomas P. Caudell. An experimental evaluation of continuous semantic zooming in program visualization. *Information Visualization, IEEE Symposium on*, 0:20, 2003.

37 SUN. JAVA. `http://www.java.com`, Online; accessed 15-July-2010.

38 SYNCOPATIONSOFTWARE. DPL-faulttrees. `http://www.syncopationsoftware.com/faulttree.html`, Online; accessed 15-July-2010.

39 Frank van Ham. Using multilevel call matrices in large software projects. *IEEE Symposium on Information Visualization*, 2003.

40 W.E. Vesely, F.F. Goldberg, N.H. Roberts, and D.F. Haasl. Fault Tree Handbook. *U.S. Nuclear Regulatory Commission*, 1981. NUREG-0492, Washington.

41 William Vesely, Joanne Dugan, Joseph Fragola, Joseph MinarickIII, Jan Railsback, and Michael Stamatelatos. Fault Tree Handbook with Aerospace Applications. *NASA*, 2002.

42 ViERforES. Virtuelle und Erweiterte Realitaet fuer hoechste Sicherheit und Zuverlaessigkeit von Eingebetteten Systemen (ViERforES). `http://www.vierfores.de`, Online; accessed 15-July-2010.

43 Colin Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann, 2000.

44 G. Zipf. Computation of minimal cut sets of fault trees: Experiences with three different methods. *Reliability Engineering*, 7(3):159–167, 1984.