Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 783–790

ISBN 978-83-60810-22-4

Enhanced CakES representing Safety Analysis results of Embedded Systems

Yasmin I. Al-Zokari, Daniel Schneider, Dirk Zeckzer, Liliana Guzman, Yarden Livnat, Hans Hagen
Kaiserslautern University
Computer Graphics and HCI, Software Engineering: Processes and Measurement
Kaiserslautern, Germany
Email: (alzokari,zeckzer,guzman,hagen)@informatik.uni-kl.de,danielschneider84@gmail.com,yarden@sci.utah.edu

***Abstract*—Nowadays, embedded systems are widely used. It is extremely difficult to analyze safety issues in embedded systems, to relate the safety analysis results to the actual parts, and to identify these parts in the system. Further, it is very challenging to compare the system's safety development and the different safety metrics to find their most critical combinations. Due to these fundamental problems, a large amount of time and effort is spent for analyzing the data and for searching for important information. Until now, there is a lack of visualization metaphors supporting the efficient analysis of safety issues in embedded systems. Therefore we present "Enhanced CakES", a system that combines and links the existing knowledge of the safety analysis and the engineering domain and improves the communication between engineers of these domains. The engineers can directly explore the most safety critical parts, retaining an overview of all critical aspects in the actual model. A formal empirical evaluation was performed and showed the increase of accuracy from ESSaRel 28.7% to 83% for CakES .**

***Index Terms*—Safety analysis, fault tree analysis, minimal cutset, embedded systems, visualization.**

I. Introduction

THE COMPLEXITY of embedded systems is currently a major problem. Cars, trains, airplanes, etc. contain an increasing number of these systems. Their safety is one very important aspect. Interactive graphical representations of the data can significantly help to ease the analysis, exploration, and fast comprehension of this complex information. We present "Enhanced CakES" (Enhanced Cake metaphor for safety analysis of Embedded Systems), a visualization system to solve these problems. It provides a new research direction combining the system engineering and the safety analysis domains. This approach comprises different types of data that are aggregated, visualized, and interacted with. We extract and visualize the most important features of the fault tree analysis of an embedded system and display them together with the parts of the physical model. For our approach, a multi-application framework was developed that enables us to combine various applications and their views for different environments (standard monitors and tiled walls [15]).

We performed an empirical evaluation comparing Enhanced CakES, our new safety analysis tool, to ESSaRel (Embedded Systems Safety and Reliability Analyser) [10], the standard tool for safety analysis using component fault tree analysis. Further, we used questionnaires and asked the users to think aloud to obtain additional information. The evaluation was performed on real data from the embedded systems domain (robotics). We found that our method had a huge positive impact on the participants. The accuracy increased significantly from 28.7% for ESSaRel to 83.1% for Enhanced CakES, while the time for searching and exploring the data increased only slightly from 24.6 minutes for ESSaRel to 29.8 minutes for CakES (not statistically significant $p = 0.4875$). Additionally, we compared the "standard monitors" and the "tiled wall" environments and found that the participants preferred the standard monitors for their work.

The paper is structured as follows: Section II provides the problem statement, including definitions, taks, measurements, and related work. Our approach, the visual metaphors and the interaction, is presented in Section III, where we illustrate its usage using a real world data set. The empirical evaluation of our approach is presented in Section IV. The future work is presented in Section V. We close this paper with conclusions in Section VI.

II. Problem Statement and Related Work

a) Definitions: Fault Tree Analysis (FTA) is a widely used analytical technique in all fields of safety [5], p. 9, 54. According to [16], "The *Fault Tree* (FT) itself is a graphic model of the various parallel and sequential combinations of faults that will result in the occurrence of the predefined undesired event". An event is any proposition that is true with a certain failure probability (FP) [12]. *Basic Events (BEs)* are the lowest-level influence factors in the FT and they are represented as the leaves. One of the methods to support the system design, is to provide all possible smallest combinations (*Minimal CutSet* (MCS)) of failures (BE) that lead to a *top level event* (TLE, the undesirable event) [5], p. 58. The authors of [20], [17] point to the importance of MCSs. Almost all FTA tools and methods can be used to generate the MCSs of the system being analyzed.

b) Tasks: There are some tasks to be fulfilled by the analysts and the engineers (in our case: the robotics engineers) to improve the safety of any system for a specified cost (e.g., time, effort, money). These tasks–regardless to the tool/method–are as follows: 1-Build FTs, to analyze the safety

of the system (performed by both analysts and engineers). 2-From the FTs, compute the MCSs (safety analysts). 3-Find the most critical MCSs, to start the analysis while spending as little time and effort as possible (safety analysts). 4-Find the BEs of these MCSs (safety analysts). 5-Therefore, there will be a need to have the relations between the safety data (e.g., BEs' ID) and the engineering data (e.g., actual parts in the model). 6-Improve/replace the BEs or add redundancy (decisions made by analysts and engineers). 7-Update the FT accordingly (This is done by exchanging a considerable amount of knowledge between the analysts and the engineers).

c) Measurements: The analysts consider one or more measurements for the MCS criticality. Most commonly considered are: the MCSs with the highest *Failure Probability* (FP) or with the highest *Failure Rate* (FR) and the MCSs with the smallest order. The order is also called size, which is the smallest number of BEs in it. For example, single points of failures are very critical.

In addition to the MCSs measurements there are some BEs measurements that needs to be considered, e.g., its id, name, location, shape, FP, and/or the number of occurrence of BEs in the system.

d) Problem Statement and Related Work: Visualization of safety data in embedded systems is an emerging research topic. With the growth of embedded systems and fault tree analysis data, visualization is becoming an important method to ease and speedup the developer's analysis. From the IBM & Industry Studies, Customer Interviews it was found that 30% of people's time is spent for searching for relevant information [23]. Moreover, as this exploration depends on humans, this leads to an increasingly high amount of human errors, specially when the data being analyzed is presented in a simple way (e.g., text) and this increases when the amount of the data. When the number of BEs increases, the effort and time needed to analyze the FT information increases significantly, especially when the FT is complex.

In [2], tools and methods related to FTA were examined to assess their support of the tasks outlined above. It was found that almost all tools and methods provide the MCSs' information–if provided–in textual formats. This textual representation is usually not complete. To obtain these information, the analysts have to deal with some difficulties. Most safety analysis tools support the analysts powerfully in the tasks 1 and 2. However, while tasks 3-7 are considered to be difficult and time consuming tasks for the analysts, they are not supported very well. For more details please refer to [2].

ESSaRel [10], [13], [11] is the standard tool for component fault tree generation used by, e.g., Siemens, that visualizes component fault trees, models, gates, BEs, and others in a 2D representation. We choose it as model for the tools reviewed in [2], because it is a freely available standard tool and because we had experts whom we could consult. Fig. 1 shows a screen shot of the ESSaRel tool.

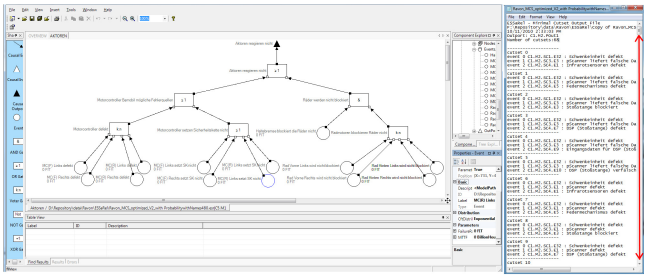

Fig. 1. ESSaRel: FT representation and textual results of the FTA (68 MCSs).

III. Our Approach

In this work, we extend [1], where we focus on the tasks that supports the analysts in a basic way (3-7). This is done by representing the results of the analysis in a different way to ease tasks 3 and 4 without navigating through the FTs or through the text file. Next, we link the data from both domains (safety analysis and engineering) to ease tasks 5-7. Additionally, we added some extra functionalities, such as: -Providing information about the FP distribution over the system. -Ability to trace the temporal safety development of a system. -Providing information about the parts: their shape in 3D and their actual location in the system. -Ability to compare between BEs by their number of occurrence and the quality of the MCSs that contains it. -Supporting users who have color vision deficiency by providing color schemes. -Providing one slider that performs filtering for different levels of MCSs' FP and which adapts its coloring to the selected color scheme. -The system can be applied on different environments such as standard monitors for daily usage and tiled-wall for demonstrations and discussions [15].

The system was developed in close cooperation with the domain experts to reduce the gap of knowledge between safety analysts and embedded systems engineers. This paper presents the enhanced visualization of [1] based on the results and on the feedback of an informal evaluation. Further, it provides many additional utilities:

- Enhancements in the menu:
  - multi-selection of MCSs, BE selection
  - multi range slider
  - added another type of color vision deficiency
  - number of BEs' occurrence in the visualized MCSs
- Enhancements in the MCS view:
  - anti aliasing
  - arranging the MCSs by FP from center
  - visualization of the order of the MCS
  - included the ghost and rotation of the BEs in the MCS when selected to give a 3D non-occluded view
  - automatic zooming towards the selected MCS
  - Three different saturation levels not two, for more distinction
- Enhancements in the BE view:
  - shows the most important BE by FP of the selected MCS

- Enhancements in the interaction:
  - faster interaction, added two different speeds
- performed the evaluations

A. Real World Scenario and Setup

We envisioned four safety analysis scenarios based on fault tree analysis data that were created to assess the safety of the embedded system RAVON:

1) An engineer wants to find which are the most critical parts in the system to improve them.
2) Both analyst and engineer want to discuss which critical parts should be improved by reducing their criticality.
3) An engineer and an analyst present the system to managers. They show the improvements by comparing the cake before improving the critical parts with the cake after enhancing the system.
4) A safety expert wants to point out the most critical parts to an engineer, i.e., a specialist from the robotics group by showing the most critical BE by both FP and number of occurrence.

Before we introduce our system, we describe RAVON, the safety analysis tool ESSaRel, and the safety data obtained from the analysis.

RAVON is a Robust Autonomous Vehicle for Off-road Navigation [18], [25]. The original model was converted from ProE to open inventor with a size of 162MB. The model is hierarchical and therefore each part can be accessed individually.

The safety of this complex embedded system was analyzed using fault tree analysis. ESSaRel (Section II, Fig. 1) was used to perform the fault tree analysis. We used the results of this analysis, a textual description of the MCSs and the BEs in the FT, as input to our system. This FT contains 540 MCSs and 29 unique BEs. FPs are associated with each BE and each part of RAVON is linked to a BE of the FTA data obtained from this analysis.

B. Overview of the System

Our system provides four views: the *MCS View* displays the collection of MCSs and their BEs (safety related data), the *Menu View* displays a menu and additional safety related data, the *Model View* displays the model, and the *BE View* displays the most important BE in the MCS that is selected by the user.

These different views allow the user to have different levels of focus and context at the same time finding the most critical MCSs. The user is directly involved in the data analysis process without the need to navigate through the system's fault trees. We use the first scenario described above to explain our system for single selection and the fourth for multi-selection.

1) MCS View Fig. 2: The first step for finding the critical parts is to get an overview over the MCSs (the safety of the system). Therefore, the analyst starts with the MCS view.

In Enhanced CakES, the MCSs are visualized using the cake metaphor. A cake consists of three different levels. Each visible level holds a number of MCSs, which are represented as cylinders. Each MCS contains a certain number of BEs and has a specific FP. Fig. 2 (right) shows the MCSs FP and size in the MCS view.

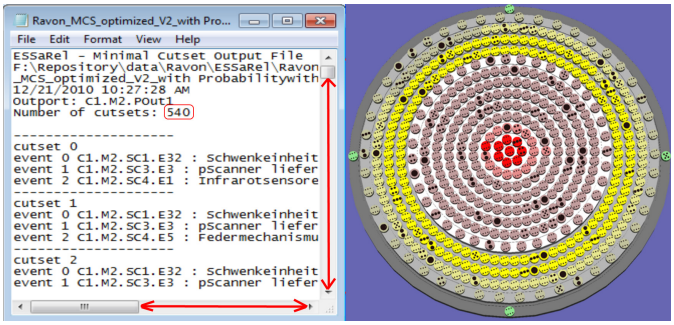

Fig. 2. The MCSs' information. (left) ESSaRel. (right) CakES showing the failure probability distribution of the whole system, the MCSs' failure probability (cylinders color), and their size (number of dots on the cylinders), in 2D. This system is unsafe because it has mostly red cylinders (high failure probability).

Each *level* (also called *holder*) is represented by a cylinder. The MCSs that are placed on each level are determined according to their FP. The FP of an MCS is computed as the product of the FPs of its BEs. However, any other function could be also used to calculate the FP of MCS. There are four FP values that influence the placement of the MCSs and the number of levels displayed: minimum, border between lowest and middle risk acceptance range (lower border), border between middle and highest (upper border), and maximum. The user can change them using the multi-thumb slider in the menu (Fig. 3 B).

The innermost cylinder (the first level) corresponds to the highest range of FPs, those between the upper border and the maximum. It is the upper part of the cake and includes the most important MCSs. The middle cylinder (the second level) corresponds to the middle range between lower and upper border. The outermost cylinder (the third level) corresponds to the range between the minimum and the lower border.

Three different saturation levels are available. Therefore nine different levels of FP are provided, which can be seen in one blink. A high saturation is assigned to MCSs having the highest FP, a medium saturation level is assigned to MCSs having a high FP, and a low saturation level is assigned to MCSs having a low FP in each specific holder. Fig. 2 (right) shows the MCSs of the first and the second holder (red, yellow MCSs) having three different saturation values, whereas the MCSs in the third holder (green MCSs) all have the same saturation. The order of each MCS is presented as dots [3], importance is reflected by the size of the dots. The most important MCSs are the ones with smallest order (e.g., single BE). Usually, most analysts examine MCSs with order in the range of [1-6], because the larger the number of BEs in a MCS the less critical it is. Therefore, we represented six BEs as a maximum of any MCS containment. The dots visibility can be set or unset by the user. The parts are sized according to the space available inside the MCS.

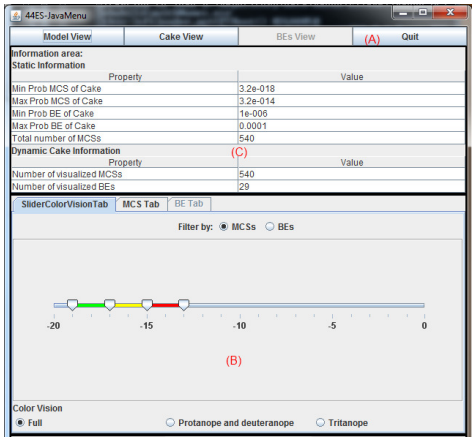

Fig. 3. Information and interaction (FP range slider and color vision deficiency radio buttons) area in the menu.

As 10% of males and 1% of females have color vision deficiency [22], [24], we included two more coloring schemes for different types of color vision deficiency (Protanopia, Deuteranopia, and Tritanopia) in addition to the normal coloring scheme [14]. We used the "Vischeck" tool [9], [21] for simulating the color vision deficiency types and assessing the quality of our choice. When the color schemes are changed, the colors of the multi-thumb slider in the menu also change.

2) Menu View (Fig. 3): The menu view is primarily used for interaction. There are three interaction areas in the menu.

The first area consists of four buttons (Fig. 3 A). Three of them can be used to select the views in focus that the user would like to interact with (MCS, BE, and Model view), one, if pressed, terminates the application.

The second area consists of a multi-thumb slider for setting the FP ranges of each levels (Fig. 3 B, top) and three radio buttons (Fig. 3 B, bottom) for choosing the color scheme. The multi-thumb slider is also used as a filter determining the minimum, lower border, upper border, and maximum FPs of the holders (Section III-B1). We adapted it from [19].

MCSs are assigned to different holders when the border FPs are changed. A logarithmic scale is used for the slider. The default values of the minimum and maximum thumbs are provided directly when loading the data, by taking the minimum and maximum FPs of the MCSs in the data. The second function of the menu view is to display additional information (Fig. 3 C). On the one hand, static quantitative information for the data set is provided (Fig. 3 C, top). The static quantitative information includes the total number of MCSs and the minimum and maximum FPs of both the MCSs and the BEs of the system, while the dynamic information shows the visualized number of MCSs and BEs in the chosen FP ranges. This information changes whenever the user manipulates the thumbs of the slider. On the other hand, dynamic quantitative and qualitative information about selected elements is displayed in the MCS and the BE tab of the menu (Fig. 4 and Fig. 5).

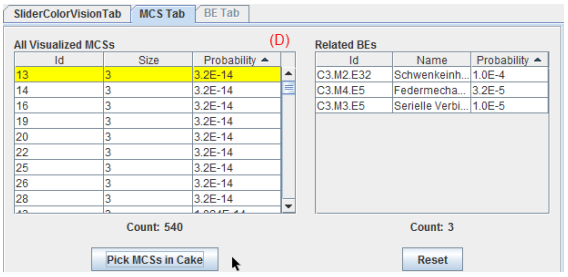

Fig. 4. MCS area in the menu, view after picking an MCS.

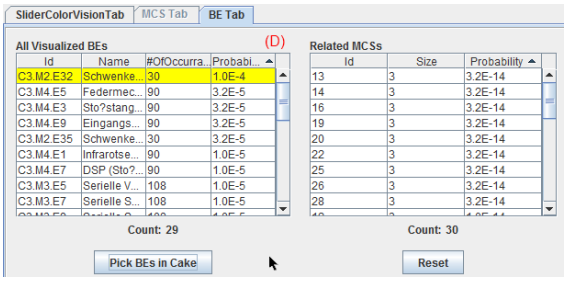

Fig. 5. BE area in the menu, view after selecting a BE which leads to multi-selection of MCSs.

C. Single Selection

Now, lets get back to our first scenario. The user can identify the most critical MCSs in the MCS view of CakES. Those are the ones in the innermost holder having the highest saturation. If the user suffers from one of the color vision deficiency types, he can change the color scheme to one adapted to his color vision. From the overview, he also sees the distribution of the MCSs criticality. So, he directly gets an insight about the criticality of the system and can compare it with the previous state of the system if available. If he wants to further investigate an MCS, he selects it either in the MCS view or from the menu view. Then, the entry of the selected MCS (its ID, its size, and its FP) is highlighted in the MCS area of the menu (Fig. 4) and information (ID, Name, FP) about its BEs are shown.

Further, this leads to the following automatic changes in the views. First, in the MCS view, the selected MCS becomes translucent and its BEs become visible inside it. These BEs rotate to show the user the 3D shapes of the physical parts related to its BEs. Fig. 6 (left) shows the effect. Until now, the BEs of an MCS are only related to hardware parts of the system. The physical parts related to the BEs of the fault tree analysis are visualized inside each MCS to give the user the relationship between the MCS view, the BE view, and the model view. The positioning of the BEs in the MCSs depends on the number of the BEs in an MCS.

Second, the system displayed in the model view changes to a translucent model with the physical parts related to the BEs of the selected MCS being opaque. This shows the user the parts, the shape of the parts, and additionally their location in the system as shown in Fig. 6 (right). Third, the

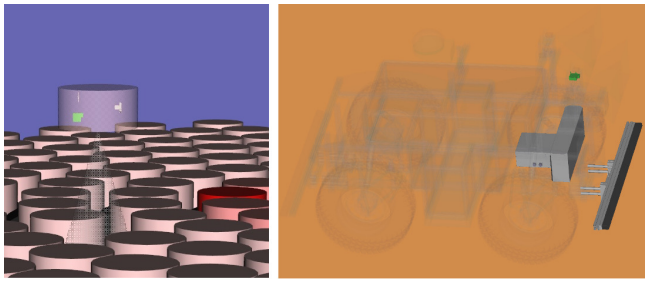

Fig. 6. (left) The MCS view, MCS raising (ghost effect). (right) Rotated view of the model view in 3D.

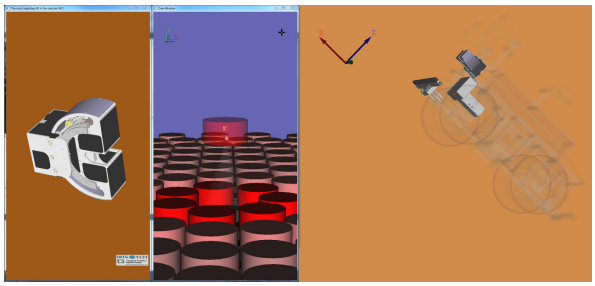

Fig. 7. After selecting an MCS.

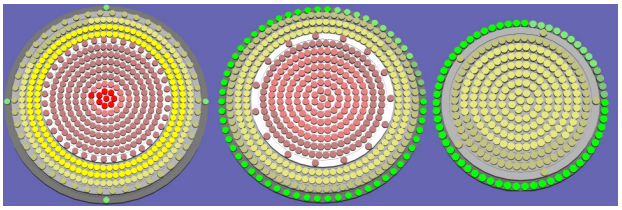

Fig. 8. The safety of the system is increasing during development (MCS view).

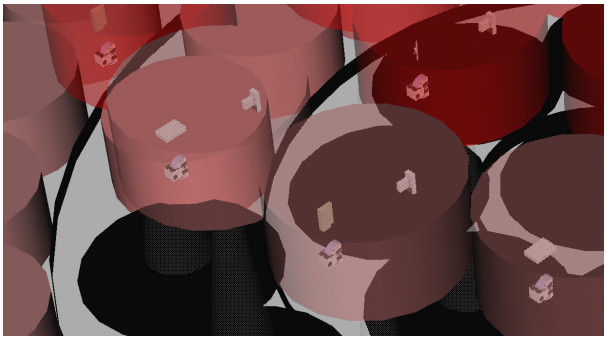

Fig. 9. Multiple MCSs show their BEs.

BE view visualizes the most important BE in the selected MCS (Fig. 7, left) and facilitates the detailed examination of each BE.

Now, the engineer explores and understands the relation between the BEs of the MCS in the model. Thus, he can identify the parts that are most critical in the system and that should be improved with respect to safety.

The MCS view shows the system's criticality and additionally provides the ability for temporal comparison of the system's evolution. Fig. 8 shows how the system is developing after enhancements.

D. Multi Selection

For our fourth scenario, the analyst explores the BEs in the BE area of the menu. The BE's ID, name, FP, and number of occurrence (the number of the MCSs influenced by this BE) are shown. When he selects an interesting BE (Fig. 5), the BE entry is highlighted and all visualized MCSs containing it are listed together with their information (ID, size, FP). At the same time, these MCSs are highlighted in the MCS view making the selected MCSs distinguishable (Fig. 9 and Fig. 10). In this view, he can immediately see the distribution of the MCSs over the different FP levels (critical, tolerable, negligible) without searching the list and thus an overview over the system's criticality is provided that eases comparison between different BEs as shown in Fig. 10. Assume, that there are two different BEs having the same number of occurrence, namely 30. The result of selecting these BEs is shown in Fig. 10: the set of MCSs in the left image is more critical than the set of MCSs in the right image, which means that the first BE (left image) influences the safety of the system more negatively than the second (right image) and should be considered for improvement. Fig. 11 shows the BE view, the MCS view, and the model view after selecting a BE.

E. Visualization Environments

We used two visualization environments: a two monitor system, where one monitor could display the model in mono or stereo view, and a tiled wall. In both environments, we had multiple coordinated views. More details are described in [15], [1].

IV. Empirical Evaluation

We performed a preliminary empirical evaluation for assessing the impact of CakES on the performance of safety analysis for analyzing a given system based on a fault tree model. Since an evaluation requires a reference for comparison, we compare the results between CakES and ESSaRel for tasks that are supported by both tools. Consequently, we define the statistical hypotheses as follows:

- HT_1: $\mu_{T,CakES} \neq \mu_{T,ESSaRel}$
- HT_0: $\mu_{T,CakES} = \mu_{T,ESSaRel}$

with μ_T = mean time required for performing a set of tasks

- $HAcc_1$: $Acc_{CakES} > Acc_{ESSaRel}$
- $HAcc_0$: $Acc_{CakES} \leq Acc_{ESSaRel}$

with Acc = accuracy

Additionally, we evaluated the usability and the usefulness of both tools for supporting the safety analysis of embedded system. In this context, usability means "degree to which a person believes that using a particular system would be free of effort" and usefulness means "the degree to which a person believes that using a particular system would enhance his or her job performance" [7]. We measured usability and

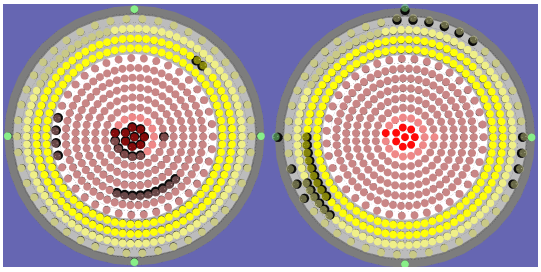

Fig. 10. Selecting a BE which causes selecting multiple MCSs.

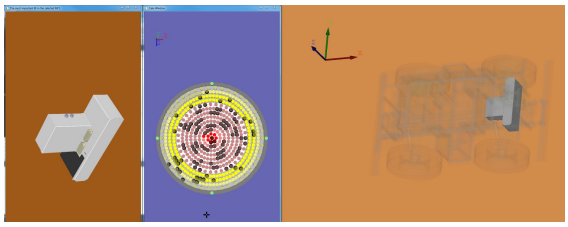

Fig. 11. After selecting a BE (multi-selection of MCSs).

usefulness using a questionnaire with closed questions and statements using a 5-point likert-scale.

A. Design

We conducted an experiment with non-probability sampling and one control group according to the recommendations of [6]. We used a convenient sample including software engineers from the Software Research Groups Dependability and Process and Measurement of the University of Kaiserslautern and from the Fraunhofer Institute for Experimental Software Engineering. All subjects were assigned randomly to the experimental group (CakES) and the control group (ESSaRel).

The experimental treatment was applied separately to each subject in the presence of two researchers, one moderator and one observer. During the experiment, the observer registered all questions and comments of the subject. The experimental treatment includes the following steps:

a) Training: First, the moderator briefly introduced the purpose of the experiment and the confidentiality and the anonymity of the responses. Then, a structured questionnaire was used for eliciting demographic information including subject age, gender, profession, and experience in safety analysis. Additionally, we conducted a color deficiency test according to [9], [14], [8], [4], [21].

Finally, each subject received the corresponding tutorial of either CakES or ESSaRel. The tutorials were prepared by an expert of each tool. Additionally, the moderator provided the corresponding tool and a data set example to give the subject the opportunity to explore the corresponding tool. The moderator instructed the subject to comment any doubt, uncertainty, or difficulty she or he had about the use of the corresponding tool. All questions of the subject were resolved. We registered the time that each subject required for the training.

b) Safety analysis: After the training, the moderator loaded the data set used for the evaluation. For that purpose, we selected a real problem in the robotics domain. The corresponding system includes 540 MCSs and 29 distinct BEs. The moderator also gave the subject a list of tasks to be accomplished using the corresponding tool without time limit. The tasks to be performed by the subject are based on the task list presented in Section II (3-5): 1. Determine how many MCSs should be improved. 2. Provide the identification of the MCSs you want to explore. 3. Give the failure probabilities of these MCSs. 4. Provide the identification, failure probability, and name of the BEs that could cause a failure to the system of each MCS of the previous question 5. If the subject used CakES: describe the location of the objects associated to the BEs in the model. 6. Judge, if the system you analyzed is safe or critical and if it is worth analyzing and spending time and effort on.

The accuracy of the tasks described above was measured against the baseline defined by a safety expert. For the first and second task, we considered the rate of correctly identified MCSs. Defining M_C as the set of correct MCSs according to the expert judgment and M_F the set of identified MCSs by the user, we determined the accuracy as $\frac{|M_F \cap M_C|}{|M_C|}$. For the third task, accuracy took a binary value (1: correct; 0: incorrect) considering the value computed by 3 safety experts. For the fourth and fifth task, we considered the rate of correctly identified BEs. Specifying B_C as the set of correct BEs $B_C = \{BE | BE \in MCS, MCS \in M_C\}$ according to the safety expert and B_F as the set of identified BEs by the user (B_F), the accuracy was determined as: $\frac{|B_F \cap B_C|}{|B_C|}$. For the last tasks we also used a binary value.

c) Evaluation of usability and usefulness: A structured questionnaire with closed questions and statements using a 5-point Likert-scale was used for eliciting the impressions of the subjects regarding the usability and usefulness of the corresponding tool. According to [7], we refined usability into: easy to understand, easy to learn, and the aesthetic value of the tool. We also decomposed usefulness into: allows to work faster, increase productivity, and the subjects confidence in his or her results.

It is important to remark that we performed 3 pilot tests for reviewing the experimental instruments (i.e., training material, problem definition, instructions, templates, and questionnaires) and the experimental treatments. The goals of the pilot tests were to identify on time possible confounding variables and important omissions and to prevent misunderstandings and mistakes regarding the experimental instruments. The pilot tests were conducted with one software engineer, one safety expert, and one robotic engineering expert.

B. Results

d) Sample: The experiment took place during August 2010. As we choose people working on a high level in safety related areas, we had only 12 participants, who were split up into the experimental and the control group, i.e., CakES and ESSaRel respectively. The subjects of the CakES group were

between 22 and 33 years old. Considering a seven point Likert-scale (1: extremely low and 7: extremely high), they have on average rather low experience in safety analysis and neutral experience in visualization. Out of six subjects, two had used FTA tools before. The subjects of the ESSaRel group were between 26 and 36 years old, they have on average rather low experience in safety analysis and neutral experience in visualization. Out of six subjects, 3 had used a FTA tools before.

e) Training: The training for CakES took on average 34 minutes ($\sigma = 8.1$) and the training for ESSaRel 13.6 minutes ($\sigma = 9.5$). The difference is explained because 3 subjects in the ESSaRel group had worked before with the tool. Therefore, they neither read the tutorial nor used the tool during the training. The training was conducted according to the experimental plan.

f) Performance in safety analysis: Whereas all the subjects in the CaKES group completed the assigned tasks and spent on average 35 minutes ($\sigma = 0.8$) on solving those tasks, only 5 subjects of the ESSaRel group completed the assigned tasks and they spent on average 32 minutes ($\sigma = 5.3$). The subject who did not finish the tasks claimed that he or she did not want to explore all views and to compute manually something that should be supported by the tool.

Considering the size of the sample, Lilliefors test shows that the time is normally distributed for both groups. Consequently, we used ANOVA to test HT_0 (i.e., $\mu_{T,ESSaRel} = \mu_{T,CakES}$, with μ_T being the mean time to complete all tasks) with a significance level of 0.05. It shows that we can not reject the null hypothesis ($p = 0.4875$). This means that there is no significant difference between the time that the subjects spent in solving the assigned tasks in both tools.

g) Accuracy: The results indicate that subjects using CakES achieved more accurate results than using ESSaRel. For the accuracy variables of task 1 to 4, Lilliefors test shows that they are not normally distributed for both groups (significance level $= 0.05$; $p < 0.001$). Therefore we conducted a Wilcoxon rank sum test for testing the null hypotheses HAcc_0 (i.e., $\text{Acc}_{CakES} = \text{Acc}_{ESSaRel}$, with Acc = accuracy) for tasks 1 to 4. The corresponding results show that HAcc_0 can be rejected with significance level 0.1 for tasks 1 ($p = 0.06$) and 2 ($p = 0.08$) and with significance level 0.05 for tasks 3 ($p = 0.03$) and 4 ($p = 0.03$). So, subjects using CakES were significantly more accurate.

Consequently, the results of this empirical evaluation show that the participants achieved better performance using CakES than ESSaRel. But considering the sample size and composition, these results can not be interpreted as being conclusive. More empirical studies with larger samples, including safety analysts and engineers from several domains are necessary to obtain more reliable conclusions regarding the performance of CakES. Table I shows the descriptive statistics for the accuracy of each task for both tools. $*^1$ One participant did not know where is the front or the back of the robot is, so he/she said that the BEs are in the back of the robot, but pointed the correct location by hand. $*^2$ One participant did not understand the question, he thought the system means the visualization system

TABLE I
ACCURACY. TS:TOTAL NUMBER OF CORRECT ANSWERS FROM THE PARTICIPANTS

Task id	Accuracy Measure Statistics	ESSaRel	CakES
1	$\frac{\lvert M_F \cap M_C \rvert}{\lvert M_C \rvert}$, mean, std	0.38, 3.5, 4.03	0.94, 8.5, 1.11
2	$\frac{\lvert M_F \cap M_C \rvert}{\lvert M_C \rvert}$, mean, std	0.27, 2.5, 3.09	0.79, 7.16, 2.85
3	TS,$TS/\lvert Mc \rvert$, mean, std	0.20, 1.83, 3.23	0.88, 8, 1.15
4	$\frac{\lvert B_F \cap B_C \rvert}{\lvert B_C \rvert}$, mean, std	0.36, 9.83, 9.02	0.92, 25, 4.47
5	$\frac{\lvert B_F \cap B_C \rvert}{\lvert B_C \rvert}$, mean, std	Not applicable	0.83,0.83,0.37 $*^1$
6	1: correct; 0 : incorrect, number of correct occurrences	2 out of 6	5 out of 6 $*^2$

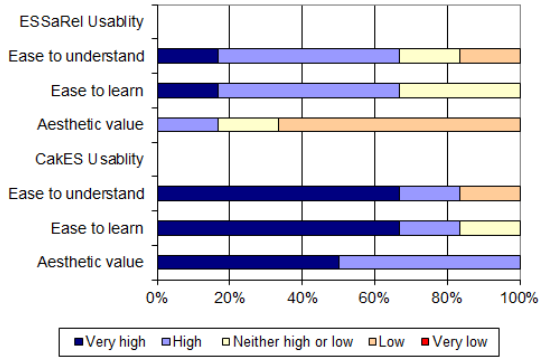

Fig. 12. The usability of ESSaRel and CakES.

(CakES), not the system being analyzed. So, he/she provided his/her positive opinion.

h) Perception on usability and usefulness: The preliminary results shows that the participants tend to perceive CakES as being more easy to understand, easy to learn, and with a greater aesthetic value than ESSaRel. Participants tend also to consider CakES more suitable for working faster than ESSarel. They believe that they increased their productivity and confidence in the produced results more by using CakES. Even though the results provide positive feedback, since we measured usability and usefulness only based on participants perception, it is important to conduct more empirical studies including more objective measures related to usability and usefulness in order to get more reliable results. Fig. 12 and Fig.13 show the usability and the usefulness of both tools.

V. FUTURE WORK

In the future, we want to test our approach using other examples from other domains. Additionally, we would like to get larger data to assess the scalability of our approach. As mentioned is the evaluation section, more empirical studies should be performed, layering the results separately with respect to the peoples' safety-analysis-experience level. Further, we will allow selecting parts in the model view highlighting all related MCSs in the MCS view and in the Menu. Adding software will be the next major task to achieve. Finally, adding labels on the parts shown in the model view would be nice to have.

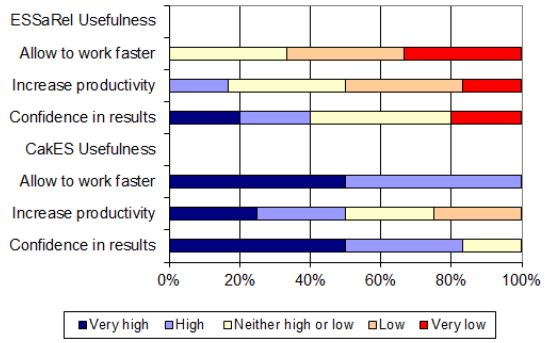

Fig. 13. The usefulness of both ESSaRel and CakES.

VI. Conclusion

CakES is an easy to learn and intuitive to use visual environment providing the most critical factors that are required in the safety analysis domain. It facilitates the exploration of the data and alleviates the comparison and the understanding of the system's safety. It combines safety analysis results and the models of embedded systems enabling the user to directly relate safety issues with the corresponding parts of the embedded system. As it is applicable on different screen configurations, the user can choose the most suitable one for his or her needs.

First of all, Enhanced CakES provides the minimal cutsets, their FPs and size, their basic events, and the overall FP distribution of the system. Further, it relates the basic events to the actual parts (shape) of the system (in 3D) and its location in the system, and provides its information: id, name, and FP. It shows, which is the most critical basic event of a minimal cutset. It provides different color schemes and it works in different visualization environments. It is interactive and provides 3D stereoscopic views of the model and is also applicable on tiled-wall environments, which could be used for users representations and discussions [15]. In addition to visualizing the results of fault tree analysis, Enhanced CakES also allows to easily compare the criticality of different systems or different versions of the same system. Finally, after gaining experience in either domain the expert can work efficiently alone exploring and judging how to choose the parts to improve and the parts to replace depending on their safety and importance. Even though we performed the evaluation only on the single selection mechanism, the CakES performed significantly better than the standard tool.

VII. Acknowledgments

We would like to thank our colleagues at the TU Kaiserslautern, Martin Proetsch Lisa Kiekbusch, Zhensheng Guo, and all the participants in our evaluation. This project was partially supported by the DAAD, the BMBF project ViERforES, and the IRTG 1131.

References

[1] Y. Al-Zokari, T. Khan, D. Schneider, D. Zeckzer, and H. Hagen. CakES: Cake Metaphor for Analyzing Safety Issues of Embedded Systems. In H. Hagen, editor, *Scientific Visualization: Advanced Concepts*, volume 2 of *Dagstuhl Follow-Ups*, Wadern, Germany, 2010. Schloss Dagstuhl–Leibniz Center for Informatics.

[2] Y. I. Al-Zokari, Y. Yang, D. Zeckzer, P. Dannenmann, and H. Hagen. Towards Advanced Visualization and Interaction Techniques for Fault Tree Analyses, Comparing existing methods and tools, 2011. to be submitted to proceedings of the IRTG 2011.

[3] Y. I. Al-Zokari, D. Zeckzer, and H. Hagen. Safety-Domino Representing Criticality of Embedded Systems. In *EuroVis 2011, Bergen, Norway,Eurographics /IEEE Symposium on Visualization 2011, poster proceedings*, page 21, 2011.

[4] Archimedes' Lab. Color Blindness or Color Vision Deficiency, 2010. http://www.archimedes-lab.org/colorblindnesstest.html; Online; accessed 31-March-2011.

[5] M. Bozzano and A. Villafiorita. *Design and Safety Assessment of Critical Systems*. CRC Press (Taylor and Francis), an Auerbach Book, 2010.

[6] J. Creswell. *Research design: qualitative, quantitative, and mixed methods approaches*. Sage Publications, 2009.

[7] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340, 1989.

[8] S. Deeb and A. Motulsky. Red-Green Color Vision Defects, 2005. http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=gene&part=rgcb; Online; accessed 31-Mar-2011.

[9] B. Dougherty and A. Wade. Vischeck simulates colorblind vision, 2008. http://www.vischeck.com/; Online; accessed 31-March-2011.

[10] ESSaRel. Background information — ESSaRel, 2002. http://www.essarel.de/index.php?site=backgroundtext; Online; accessed 31-March-2011.

[11] B. Kaiser, C. Gramlich, and M. Förster. *Computer Safety, Reliability, and Security*, volume 3219/2004, chapter State-Event-Fault-Trees - A Safety Analysis Model for Software Controlled Systems, pages 195–209. Springer Berlin, Heidelberg, 2004. http://www.springerlink.com/content/j886uwajl8tnu9y6.

[12] B. Kaiser, C. Gramlich, and M. Förster. State/event fault trees - A Safety Analysis model for software-controlled systems. *Reliability engineering & systems safety*, 92:1521–1537, 2007.

[13] B. Kaiser, P. Liggesmeyer, and O. Mäckel. 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.

[14] M. Kalloniatis and C. Luu. Psychophysics of Vision: The Perception of Color, 2007.

[15] T. Khan, D. Schneider, Y. Al-Zokari, D. Zeckzer, and H. Hagen. Framework for Comprehensive Size and Resolution Utilization of Arbitrary Displays. In H. Hagen, editor, *Scientific Visualization: Advanced Concepts*, Dagstuhl Follow-Ups, Wadern, Germany, 2010. Schloss Dagstuhl–Leibniz Center for Informatics.

[16] NASA Office of Safety and Mission Assurance. Fault Tree Handbook with Aerospace Applications. Technical report, NASA Headquarters, Washington, DC, USA, August 2002.

[17] F. Ortmeier, W. Reif, and G. Schellhorn. Formal safety analysis of a radiobased railroad crossing using deductive cause-consequence analysis (DCCA). In *Proceedings of 5th European Dependable Computing Conference EDCC, volume 3463 of LNCS*. Springer, 2005.

[18] RAVON. AG Robotersysteme: Ravon, 2009. http://agrosy.informatik.uni-kl.de/en/robots/ravon/; Online; accessed 31-Mar-2011.

[19] Swing. JSlider Examples 1. http://en.pudn.com/downloads3/sourcecode/java/detail6037_en.html; Online; accessed 31-Mar-2011.

[20] A. Thums and G. Schellhorn. Formal safety analysis in transportation control. In *Proceedings of the Workshop on Software Specification of Safety Relevant Transportation Control Tasks, VDI Verlag GmbH*, 2002.

[21] B. Wandell, B. Dougherty, and A. Wade. Try Vischeck on Your Image Files. http://vischeck.com/vischeck/vischeckImage.php; Online; accessed 31-Mar-2011.

[22] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., 2004.

[23] M. Weber. A survey of Semantic Annotations for Knowledge Management, 2008. http://www.mendeley.com/profiles/markus-weber/; Online; accessed 31-Mar-2011.

[24] Wikipedia. Color blindness. http://en.wikipedia.org/wiki/Color_blindness; Online; accessed 31-Mar-2011.

[25] Wikipedia. Ravon — Wikipedia, The Free Encyclopedia, 2009. http://en.wikipedia.org/wiki/Ravon; Online; accessed 31-Mar-2011.