Controlled Experiments Comparing Fault-Tree-based Safety Analysis Techniques

Adrien Mouaffo, Davide Taibi, Kavyashree Jamboti

University of Kaiserslautern
Gottlieb-Daimler-Straße 47
67663 Kaiserslautern, Germany
{ adrien.mouaffo | taibi | jamboti } @ cs.uni-kl.de

ABSTRACT

The capability to model dynamic aspects of safety-critical systems, such as sequence or stochastic dependence of events, is one important requirement for safety analysis techniques. State Event Fault Tree Analysis, Dynamic Fault Tree Analyis, and Fault Tree Analysis combined with Markov Chains Analysis have been developed to fulfill these requirements, but they are still not widely accepted and used in practice. In order to investigate the reasons behind this low usage, we conducted two controlled experiments. The goal of the experiments was to analyze and compare applicability and efficiency in State Event Fault Tree analysis versus Dynamic Fault Tree Analyis and Fault Tree Analysis combined with Markov Chains Analysis. The results of both experiments show that, notwithstanding the power of State Event Fault Tree Analysis, Dynamic Fault Tree Analyis is rated by participants as more applicable and is more efficient compared to State Event Fault Tree Analysis, which, in turn, is rated as more applicable but is less efficient than Fault Tree Analysis combined with Markov Chains Analysis. Two of the reasons investigated are the complexity of the notations used and the lack of tool support. Based on these results, we suggest strategies for enhancing State Event Fault Tree Analysis to overcome its weaknesses and increase its applicability and efficiency in modeling dynamic aspects of safety-critical systems.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: D.2.4 Software/Program Verification

General Terms

Measurement, Experimentation.

Keywords

Fault Tree Analysis; Markov Chain; State Event Fault Tree; Dynamic Fault Tree; Safety-Critical Systems; Controlled Experiment; Safety-analysis.

1. INTRODUCTION

More and more safety-critical functionalities in the automotive, avionics, and healthcare domains are implemented by means of embedded systems [3]. These systems are often used in dynamically changing environments, and have to adapt and optimize their behavior to maintain good quality of service when changes are detected. For example, they are used for controlling safety-critical features such as health monitoring [14], adaptive cruise control [10], and traction control [13], which have strict safety requirements. Moreover, in some domains such as avionics, any component has to be certified before it can be used, and safety analysis helps to create evidences for certification.

The goal of safety analysis is to show that the risk associated with a system failure, which might endanger human lives or damage property, is below an acceptable level. There exist safety analysis techniques such as Model Checking for safety properties, Failure Mode Effect and Criticality Analysis, and Fault Tree Analysis. In this paper we only consider Fault Tree Analysis techniques. In Fault Tree Analysis, a system failure occurs as a result of the combination of different events. It is performed mainly in two steps: (1) modeling a fault tree that represents the safety analysis perspective of the system; (2) analyzing the model to obtain qualitative (combination of events that leads to the system failure) and quantitative (probability of occurrence of the system failure) results.

In our work, we focus only on the modeling step.

Various techniques have been developed to model fault trees. Fault Tree Analysis [24] gained popularity because of its ability to provide both quantitative and qualitative analysis results, and to capture combinations of events that lead to system failures even when individual components are working as required. One of the main disadvantages of fault tree models is low maintainability, which is due to the lack of direct mapping between the system and the fault tree structure [2]. Moreover, fault tree models cannot capture dynamic properties of a system such as priorities, sequences, and stochastic dependencies of events. In order to capture this dynamism, two new techniques have been defined: Dynamic Fault Tree Analysis (DFT) [1] and Fault Tree Analysis combined with Markov Chains Analysis (FT+MC)[15].

To increase the maintainability of fault tree models, Component Fault Tree Analysis (CFT) [2] was developed. CFT defines a direct mapping between the system and the fault tree structure but, like Fault Tree Analysis, it cannot capture the dynamic aspects of components. So in order to capture this dynamism and keep up a certain level of maintainability, State Event Fault Tree analysis (SEFT) [12] was developed as an extension of CFT. Fault tree models built during SEFT have the inherent advantages of fault tree models built with CFT, which have a strong coupling between the system architecture and the failure model, allowing to maintain traceability between the fault tree model and the system architecture. Moreover, SEFT allow modeling many dynamic aspects of the system such as timing behaviors, stochastic dependence, and sequence of events.

Notwithstanding the power of SEFT, in our experience, it is not broadly used in practice, whereas DFT and FT+MC are commonly used to deal with dynamic embedded systems [27] [6] [18]. We

EASE'14, May 13–14, 2014, London, UK.

believe that the problem is not related to the power of SEFT but rather to the applicability and efficiency of the technique.

In this research, we conducted two controlled experiments to analyze SEFT, DFT, and FT+MC for the purpose of evaluating and comparing their applicability and efficiency with regard to modeling dynamic aspects of safety-critical systems. We expected SEFT to be more applicable and efficient than DFT or FT+MC in the context of safety analysis.

During the first controlled experiment, SEFT was compared to DFT. The participants had to apply only one of the techniques to model safety aspects of an Ambient Assisted Living System in three tasks. For the second controlled experiment, where SEFT was compared to FT+MC, the participants had to apply both techniques in one comprehensive task. The change of design between the first and the second experiment was imposed by the lessons learned from the first controlled experiment.

Unexpectedly, our results show that DFT and FT+MC are more efficient than SEFT and that DFT is perceived as being more applicable than SEFT. We further investigated the reasons behind the results we had obtained and found that the complexity and variety of the notations used for SEFT make it difficult for users to know which notation to use and to understand the semantics behind the notation.

The remainder of this paper is structured as follows. After this introductory section, we briefly introduce background knowledge in Section 2 and related works in Section 3. In Section 4, we present the experimental approach describing our study design, our goal, and our hypotheses. In Section 5, we present the results of the experiments and discuss them in Section 6. Finally, in Section 7 we draw conclusions and give an outlook on future work.

2. BACKGROUND KNOWLEDGE

In this section, we describe fault-tree-based safety analysis techniques.

2.1.1 Fault Tree Analysis

Fault trees [24] are constructed using a backward searching technique starting with a top event. The causes identified are combined using Boolean gates. After its construction, a fault tree can provide quantitative results such as the top event probability or qualitative results in the form of Minimal Cut Sets. A Minimal Cut Set signifies a set of events where the non-occurrence of even one event prevents the top event from occurring. Minimal Cut Sets can be ranked according to the number of events comprising them and for those with fewer events, it needs to be ensured that their occurrence probabilities are reduced or eliminated. In cases where the Minimal Cut Set consists of just one event called a single point failure, special attention must be paid in order to ensure that it does not occur or its chances are minimized. Fault trees can be enhanced with Markov chains to model aspects such as stochastic dependencies or sequences of basic events.

2.1.2 Dynamic Fault Tree Analysis (DFT)

As the name suggests, this technique [1] allows analyzing top events for safety-critical systems where the notion of spare components is predominant. DFT enables the modeling of stochastically dependent events (failures of spare parts and triggered events) and sequencing by using new types of gates, which are listed below:

- PAND (Priority-And)
- SEQ (Sequence-Enforcing)
- FDEP (Functional Dependency)
- CSP (Cold Spare), WSP (Warm Spare) or HSP (Hot Spare)

It is important to note that fault tree models produced during DFT are analyzed by analyzing the underlying Markov chains that capture the sequencing and stochastic dependencies of events.

2.1.3 State Event Fault Tree Analysis (SEFT)

SEFT [12] build on CFT, which is an approach for building fault trees based on failures of components of a system. CFT overcomes the drawbacks of Fault Tree Analysis, which only conveys how a failure can occur, but does not specify which components influence each other in such a manner that the failure will occur. Fault tree models produced during CFT can be easily reused as they have clear decomposition semantics based on the system architecture. Though CFT overcomes some of the drawbacks of Fault Tree Analysis, it is incapable of handling some other issues such as dependencies, sequence, and timing of events. CFT cannot handle stochastic dependence and cannot be integrated with state-based design models showing the behavior of the system. SEFT has been developed to overcome this problem. It allows modeling the failures of a component showing the internal safety-relevant state changes. Unlike CFT, it makes a clear distinction between a state and an event. In the context of SEFT, a state is defined as the collectivity of the variable properties of a component that are relevant to its behavior and its reaction to external events, and an event is defined as a sudden phenomenon without temporal expansion in the context of discrete event systems. A state or event occurrence in one component can trigger state changes in another component. SEFT enables the use of a wide range of gates, which need not be only Boolean operators. Gates in SEFT can be made up of Boolean operators and state-based models that allow modeling the order and timing of the occurrence of states and events in a fault tree model. Some of the gates used during SEFT are:

- AND (with n state inputs)
- AND (with n state inputs and one event input)
- OR (with n state inputs)
- OR (with n event inputs)
- History-AND
- Priority-AND

Fault tree models produced during SEFT are quantitatively analyzed by translation to Petri nets. The top event probability can be calculated by calculating the probability for the corresponding place in the Petri net.

3. RELATED WORK

Various empirical studies have been performed to compare constructive and analytical safety techniques. Constructive techniques, such as multi-version programming, standards, design patterns, and others are used when developing a product to ensure that the required quality will be fulfilled. Safety analysis techniques such as testing, reviews, or audits, which are used when a product or part of it is available to ensure that the required quality has been fulfilled, have also been evaluated in empirical studies.

Some empirical studies on constructive safety techniques and on safety standards [26] have been published in various domains, such as ISO 61508 [8], ISO 26262 [9], or ARP4761 [19].

Analytical safety techniques have also been empirically evaluated, but controlled experiments are rarely reported.

Hansen [7] performed a survey where he describes safety techniques and methodologies, such as fault trees, sneak circuit

analysis, or petri nets, with regard to their utility and applicability based on system complexity.

Cullyer et al. [5] conducted a study where they elicited the benefits and shortcomings of tools and techniques for testing safety-critical software. They found that most tools have safety-relevant features like testing automation, document generation, and test planning support. Nevertheless, they still lack important features such as support for testing in the host and target environments or the consideration of real-time constraints, or they have not yet been assessed for their safety integrity level.

Stålhane [20] reports on a controlled experiment comparing two analytical safety techniques. He investigated Failure Mode and Effect Analysis versus Misuse Cases in terms of perceived ease of use and perceived usefulness by means of the Technology Acceptance Model [23]. The result was that Misuse Cases are better than Failure Mode and Effect Analysis in analyzing user interaction failures and Misuse Cases are generally perceived as being easier to use than Failure Mode and Effect Analysis.

In a subsequent study, Stålhane compared Misuse Cases on use-case diagrams with textual use cases. He found that textual use cases are easier to use due to the more detailed information provided [21]. He then compared Misuse Cases with System Sequence Diagrams with respect to finding and documenting hazards [22] with the result that Misuse Cases are better for identifying hazards related to system operations, while System Sequence Diagrams are better for identifying hazards regarding the whole system.

The effectiveness of Misuse Cases has also been compared to Attack Trees by Opdahl and Sindre [16] by means of the Technology Acceptance Model [23]. The result was that Attack Trees are more effective in finding threats.

Finally, a replicated controlled experiment was published by Jung et al. [11], comparing the capabilities of Fault Trees Analysis and those of Component Fault Trees Analysis. The experiment was first run with seven researchers in a controlled university environment and then replicated in a company environment with eleven participants. The experiment showed that Component Fault Trees Analysis can be beneficial for employees with little or no experience in fault tree analysis.

To the best of our knowledge, no study has been published on the comparison of SEFT with DFT or FT+MC.

4. EXPERIMENTAL APPROACH

The objective of our research is to understand how SEFT behaves in comparison to DFT and FT+MC when modeling the safety analysis perspective of safety -critical systems. For this purpose, we conducted two controlled experiments. The first controlled experiment was conducted to analyze SEFT and DFT, while the second experiment was conducted to analyze SEFT and FT+MC.

In this section, we present the goal, questions, and metrics for both controlled experiments. We also define hypotheses for each metric. We then describe the designs used for the two experiments, the measurement instruments, and the analysis results.

4.1 Study Goal and Hypotheses

SEFT uses the architectural model of the system and combines concepts for modeling dynamic aspects of a system with component fault trees, such as state, event, and gates; we therefore assume that building and maintaining fault tree models using SEFT can be done efficiently and the notations used are applicable for modeling dynamic aspects of safety-critical systems. Efficiency is defined here as the time needed for modeling fault trees and Applicability is defined as the capability of natation used to capture the semantic behind dynamic aspects of safety critical systems. Moreover, we also wanted to determine whether fault tree modeling using SEFT is more efficient and more applicable than fault tree modeling using.

Accordingly to our expectation we formulate the goal of both controlled experiments as followed:

analyze SEFT, DFT, and FT+MC
for the purpose of evaluating and comparing
with respect to applicability and efficiency
in the context of modeling dynamic aspects of safety-critical systems.

With respects to the quality attributes efficiency and applicability, the following questions and metrics are derived:

- Applicability
  - Q1: Is SEFT more applicable than DFT / FT+MC for modeling dynamic aspects of safety-critical systems?
  - Q2: Do the participants need less time to perform their tasks when using SEFT than when using DFT or FT+MC?
- Efficiency
  - Q3: What do the participants think about the applicability of SEFT?

Each question derived is further explained in the following sub sections.

4.1.1 Q1: Is SEFT more applicable than DFT / FT+MC for modeling dynamic aspects of safety-critical systems?

This question is related to the applicability of the notations used during SEFT. Safety-critical systems exhibit characteristics such as stochastic dependency and timing or sequencing of events. Such characteristics have to be taken into account when modeling the safety perspective of a system. Safety analysis techniques have specifics notations that can be used to model those characteristics.

To answer this question, we collected the opinions of the participants on the following metrics, after they had applied the technique to a concrete system:

- **Completeness:** measures the degree to which the participant believes that by using the given notations he was able to completely capture the semantics behind dynamic aspects of safety-critical systems.
- **Understandability:** measures the effort needed by the subject to understand the models built with the given technique. If the notations are appropriate, the participants will need to exert little effort to understand the relationship with the system concepts.
- **Easiness:** measures the degree to which the subject believes that he or she was able to model aspects of the system with little effort by using the notations provided.

With regard to the metrics, we derived the following three hypotheses:

- **H_{11}:** The participants will perceive the completeness of SEFT notations differently than the completeness of DFT or FT+MC notations.
  - **H_{111}:** $\mu SEFT \neq \mu DFT$

- **$H0_{111}$:** μSEFT = μDFT
- **H_{112}:** μSEFT ≠ μFT+MC
- **$H0_{112}$:** μSEFT = μFT+MC

- **H_{12}:** The participants will perceive the understandability of SEFT notations differently than the understandability of DFT or FT+MC notations.
  - **H_{121}:** μSEFT ≠ μDFT
  - **$H0_{121}$:** μSEFT = μDFT
  - **H_{122}:** μSEFT ≠ μFT+MC
  - **$H0_{122}$:** μSEFT = μFT+MC
- **H_{13}:** The participants will perceive the ease of use of SEFT notations differently than the ease of use of DFT or FT+MC notations.
  - **H_{131}:** μSEFT ≠ μDFT
  - **$H0_{131}$:** μSEFT = μDFT
  - **H_{132}:** μSEFT ≠ μFT+MC
  - **$H0_{132}$:** μSEFT = μFT+MC

4.1.2 Q2: Do the participants need less time to perform their tasks when using SEFT than when using DFT or FT+MC?

This question is related to the efficiency of fault tree modeling using the technique. To answer the question, we collected the time needed by the participants to perform each task and thus derived the following hypothesis:

- **H_2:** Subjects using SEFT will need a different amount of time for fault tree modeling than subjects using DFT or FT+MC.
  - **H_{21}:** μSEFT ≠ μDFT
  - **$H0_{21}$:** μSEFT = μDFT
  - **H_{22}:** μSEFT ≠ μFT+MC
  - **$H0_{22}$:** μSEFT = μFT+MC

4.1.3 Q3: What do the participants think about the applicability of SEFT?

It was important for us to collect some subjective information about how the participants felt when using SEFT. This information will be used later to improve the technique. For this purpose, we used the following metrics:

- **Attitude towards using SEFT:** measures the overall affective reaction to using the notations of the technique.
- **Self-efficacy by applying SEFT:** measures the degree to which the subject believes that he or she will perform better if given some help.

We assumed that feedback on using SEFT would not be negative or would at least be better than an average value α and therefore derived the following three hypotheses:

- **H_{31}:** The subjects' feedback regarding their attitude toward using SEFT will be different than the average value α.
  - **H_{31}:** μSEFT ≠ α
  - **$H0_{31}$:** μSEFT = α
- **H_{32}:** The subjects' feedback regarding self-efficacy will be different than the average value α.
  - **H_{32}:** μSEFT ≠ α
  - **$H0_{32}$:** μSEFT = α

4.2 Study Design

The study was designed as two controlled experiments conducted as part of two lectures, with students and researchers as participants. The first controlled experiment (SEFT vs. DFT) was conducted in the context of the lecture Empirical Model Building and Methods (EMBM) during the summer term 2012, with researchers of the Software Engineering department and students enrolled in the lecture. We were also able to get researchers of the Computer Science department of the University of Kaiserslautern to participate in the controlled experiment. The second controlled experiment (SEFT vs. FT+MC) was conducted in the context of the lecture SRES during the winter term 2012-2013.

In this section, we describe the design used for each controlled experiment.

4.2.1 Controlled experiment 1: SEFT vs. DFT

The first study was conducted with fourteen subjects: eight students from the EMBM lecture and six researchers from the Computer Science department. The study was designed to take place within a 90-minute time frame normally used for exercise classes. Hypotheses H_{111}, H_{121}, H_{131}, H_{21}, H_{31} and H_{32} were investigated during the first study. To minimize possible threats to validity, several considerations were taken into account for the study.

4.2.1.1 Study preparation

All participants had good knowledge of fault tree analysis; the selected students had learned about it extensively during the lecture SRES offered during the winter term 2011-2012 and the researchers had used it for their research work.

We used an experimental design randomly distributing participants into two groups with equal numbers of students or researchers in both groups.

The participants from each group had to apply only one technique: Group 1 used SEFT and Group 2 used DFT. To make sure that they had a minimal level of knowledge regarding application of these techniques and to allow them to provide valuable feedback, they were trained by an expert on the technique they had to apply.

An Ambient Assisted Living (AAL) system was used for the first study. AAL systems have been developed for monitoring the health condition of elderly people by collecting and analyzing continuous sensor data and performing some routine check-ups (e.g., following medical treatment or surgery) [14]. AAL systems consist of various sensors, actuators, and software components integrated into everyday items or worn/used by patients. They are used for emergency treatment (core functionality), autonomy enhancement, and comfort [14]. An AAL living lab had been established at the Fraunhofer Institute for Experimental Software Engineering IESE and we reused an excerpt from the living lab system model in this study.

Based on the defined system model, an expert in safety analysis from the Software Engineering Research Group: Dependability of the University of Kaiserslautern developed corresponding fault tree models following DFT and SEFT.

In total, three tasks were defined with the help of the expert in safety analysis. During the definition of the tasks, we considered following criteria:

- Limited time: The participants would have only a limited amount of time to solve each task

- Familiarity: The participants were familiar with certain notations and we used these for describing the system model.

To reduce the learning effect, the tasks were to be solved independently and the participants had to randomly choose in which order they performed all tasks. For each task, changes were made to the behavior of the system and the participants had to change the system model to include new information. For the first task, a new sensor was added to the original system and its behavior was modified. For the second task, the original system model was upgraded by adding a new software component and the behavior of the system was changed accordingly. For the third task, more details were added to the original system model, which made building the fault tree model more complex.

Developing a fault tree model can be done using tools, but to avoid the impact the quality of the tool would have had on the study results, the participants had to draw the fault tree models manually. For each task, there was a description of how the participants should proceed. They were provided with an original fault tree model and had to draw the missing parts in accordance with the task. Additionally, they received blank sheets in case they needed more space to draw additional information.

Due to the internationality of the participants, instructions, tutorial, task descriptions, system model, and questionnaires were provided in English.

4.2.1.2 Study procedure

The procedure followed during the study is described below:

(1) The experimenter introduced the study to the study participants. This included informing the participants about the goal of the study and describing the procedure to be followed.

(2) A pre-questionnaire was filled out by the participants in which they provided information about their background.

(3) The participants were then randomly assigned to each group (Group 1 used SEFT and Group 2 used DFT).

(4) The training on techniques was done thereafter. Group 1 participants were trained on SEFT and Group 2 participants were trained on DFT.

(5) The participants were provided with experiment materials and had to perform each task. After each task, they had to fill in a questionnaire regarding the task.

(6) At the end of the experiment, each participant had to fill in a post-questionnaire concerning their subjective assessment of the technique applied. They also had to answer open-ended questions.

4.2.2 Controlled experiment 2: SEFT vs. FT+MC

Initially the designs of both experiments were supposed to be similar. However, based on lessons learned from the first experiment, we improved the design of the second experiment.

The second study was conducted with twenty-seven subjects from the Safety and Reliability Engineering of Embedded Systems (SRES) lecture offered during the winter term 2012-2013. The study took place during two exercise class time slots of ninety minutes each. Hypotheses H_{112}, H_{122}, H_{132}, H_{22}, H_{31} and H_{32} were investigated during the second study.

4.2.2.1 Study preparation

Following a pre-analysis of the data collected in the first study, where we had low statistical significance, we decided to have all participants apply both techniques to improve the statistical significance of the results. All participants had good knowledge of fault tree analysis, which had been covered during their class in one chapter and several exercises. During the lecture, the participants also got an introduction to Markov chains.

Because SEFT was not part of the lecture, we used one of the exercise class time slots to provide training on SEFT to make sure that the level of knowledge on both techniques was similar for all participants. The training was conducted by an expert on SEFT.

For the second study, an Adaptive Cruise Control system (ACC) was used as a sample system. ACC is an automotive feature that allows a vehicle's cruise control system to adapt the vehicle's speed to the traffic environment [10]. Sensors placed in front and behind the vehicle are used to detect the speed of cars in front and behind the vehicle. In combination with vehicle speed and driving activities (braking or throttling), this information is used by the ACC to adapt the vehicle speed and thereby avoid a collision. We reused the ACC model from the concept car developed at Fraunhofer IESE .

Based on the defined system model, an expert in safety analysis from the Software Engineering Research Group: Dependability of the University of Kaiserslautern developed corresponding fault tree models following SEFT and FT+MC.

Based on our experience from the first study, we decided to define only one task for the second study. For this task we defined a set of modifications that had to be considered while extending the fault tree models. To reduce the learning effect from performing the same task, the participants had to randomly choose the order in which they wanted to perform the two techniques (SEFT and FT+MC).

As in the case of the first study, the participants were told how to proceed. They were also provided with an original fault tree model and had to draw missing parts in accordance with the task. Additionally, they received blank sheets in case they needed more space to draw additional information.

Similarly to the first experiment, instructions, tutorial, task descriptions, system model, and questionnaires were provided in English.

4.2.2.2 Study procedure

The procedure followed during the study is described below:

(1) The participants were trained during the first time slot on performing SEFT and also get a review of FT+MC. This session was conducted as a normal lecture or exercise class.

(2) The second time slot started with an introduction of the study provided to the participants. This included informing the participant about the goal of the study and describing the procedure to be followed.

(3) A pre-questionnaire was filled out by the participants in which they provided information about their background.

(4) The participants were provided with experiment materials and had to perform each task. After each task, they had to fill in a questionnaire regarding the task.

(5) At the end of the experiment, each participant had to fill in a post-questionnaire concerning their subjective assessment of the technique applied. They also had to answer open-ended questions.

statement (Table 1), and each of these statements had to be rated on

Table 1. Questionnaire related to Q1

Perceived Completeness	Perceived Understandability	Perceived Ease of Use
1-1) I am sure that I was able to transfer the description from the system model completely to the SEFT/DFT/FT+MC model.	2-1) Because of the graphical representation of SEFT/DFT/FT+MC, it was easy for me to keep the overview of the failure logic.	3-1) It was easy for me to transfer the descriptions of the system model to the SEFT/DFT/FT+MC.
1-2) I was able to identify appropriate gates for describing all failure logics.	2-2) The relationship between the SEFT/DFT/FT+MC and the system is easy for me to comprehend.	3-2) The SEFT/DFT/FT+MC supported me during the accomplishment of the tasks.
1-3) I am sure that I was able to identify all changes that have to be made from the original SEFT/DFT/FT+MC.	2-3) The SEFT/DFT/FT+MC methodology helped me to keep the overview of the failure logic.	3-3) It was easy for me to implement the modifications.
		3-4) I was able to make the modifications with minor effort.
		3-5) I was able to reuse a lot from the existing model during the modifications.
5-point ordinal Likert scale: 1=strongly disagree, 2=disagree, 3=neither agree nor disagree, 4=agree, 5=strongly agree		

Table 2. Questionnaire related to Q3

Attitude towards using SEFT	Self-efficacy by applying SEFT
1-1) I am sure that I was able to transfer the description from the system model completely to the SEFT/DFT/FT+MC model.	2-1) Because of the graphical representation of SEFT/DFT/FT+MC, it was easy for me to keep the overview of the failure logic.
1-2) I was able to identify appropriate gates for describing all failure logics.	2-2) The relationship between the SEFT/DFT/FT+MC and the system is easy for me to comprehend.
1-3) I am sure that I was able to identify all changes that have to be made from the original SEFT/DFT/FT+MC.	2-3) The SEFT/DFT/FT+MC methodology helped me to keep the overview of the failure logic.
5-point ordinal Likert scale: 1=strongly disagree, 2=disagree, 3=neither agree nor disagree, 4=agree, 5=strongly agree	

Table 3. Cronbach's α measure for controlled experiment 1

Attribute	Completeness		Understandability		Ease of Use		Attitude toward using SEFT	Self-efficacy by using SEFT
Method	SEFT	DFT	SEFT	DFT	SEFT	DFT	SEFT	SEFT
Cronbach's α measure	0,92	0,88	0,91	0,96	0,93	0,92	0,81	0,15

Table 4. Cronbach's α measure for controlled experiment 2

Attribute	Completeness		Understandability		Ease of Use		Attitude toward using SEFT	Self-efficacy by using SEFT
Method	SEFT	FT+MC	SEFT	FT+MC	SEFT	FT+MC	SEFT	SEFT
Cronbach's α measure	0,91	0,83	0,63	0,86	0,95	0,91	0,92	0,81

4.3 Measurement Instruments

To answer each question, we collected the resulting models from each task and also collected the participants' opinion using various questionnaires.

4.3.1 Q1: Is SEFT more applicable than DFT / FT+MC for modeling dynamic aspects of safety-critical systems?

For this question, we defined a questionnaire that had to be answered at the end of each task. Each metric was refined into a an ordinal 5-point Likert sclale.

4.3.2 Q2: Do the participants need less time to perform their tasks when using SEFT than when using DFT or FT+MC?

To answer this question, the participants had to write down the starting and ending times for each task they performed. When collecting the data, we therefore calculated the time needed by each participant to perform each task in minutes.

4.3.3 Q3: What do the participants think about the applicability of SEFT?

With this question we wanted to get the general impression the users had when they were using SEFT. Therefore, each participant had to fill in a post-questionnaire. The post-questionnaire contains statements refined from the metrics related to Q3 (Table 2). Just like for Q1, each statement had to be rated on a 5-point Likert scale.

4.4 Data Analysis

In this section, we describe the procedure used for analyzing the collected data and then report on the experimental results.

Q1 and Q2 were analyzed by performing the following steps:

(1) We performed a descriptive analysis of the collected data.

(2) We tested the data for normality using a Shapiro-Wilk test.

(3) If the data were normally distributed, we performed an Independent Sample T-test for independent samples (controlled experiment 1) or a Paired-Sample T-test for dependent samples (controlled experiment 2) with a confidence interval of 95% to test our hypotheses.

(4) If the data were not normally distributed, we performed a Median test for independent samples (controlled experiment 1) or a Wilcoxon Signed-Rank test for dependent samples (controlled experiment 2) with a significance level of 0.05 to test our hypotheses.

Q3 was analyzed by performing the following steps:

(1) We performed a descriptive analysis of the collected data.

(2) We performed a One-Sample Wilcoxon Signed-Rank test for comparing the obtained medians to the hypothesized median (α=3).

To investigate Q1 and Q3, we collected the opinions of the participants regarding the metrics by using a set of statements.

For Q1, the participants provided their opinions on the techniques applied (SEFT, DFT or FT+MC) with regard to the metrics **Completeness, Understandability** and **Ease of use**. For Q3, they had to provide their opinions regarding the metrics **Attitude toward using SEFT** and **Self-efficacy by using SEFT**. As shown in Table 1 and Table 2, each of these metrics was related to several statements.

To make sure that the statements on the given scale were measuring the same underlying assumption, we performed a reliability test by calculating the Cronbach's α reliability measure. Cronbach's α is a common measure of internal consistency. The Cronbach's α reliability measure varies between 0 and 1 (values close to 1 indicate high internal consistency while values close to 0 indicate low internal consistency). For controlled experiment 1, good to very good internal consistency was obtained for all metrics except for Self-efficacy. For controlled experiment 2, we obtained good to very good internal consistency for all metrics except for Understandability regarding SEFT, which has fairly good internal consistency.

Despite the internal consistency obtained for Self-efficacy in controlled experiment 1 and for Understandability regarding SEFT in controlled experiment 2, we proceeded by aggregating the results obtained for each statement. Regarding Q1, we first performed the aggregation (by calculating the average) at the task level and aggregated (also by calculating the average) the results obtained at the task level into the final result. For Q3, we aggregated the result by calculating the average obtained for each statement. By doing this, we were able to compare the results obtained for each metric.

For Q2, the participants had to report when they started and when they ended each task. Based on that, we calculated the time needed for performing each task. The metric time value was obtained by averaging the times needed for each task.

5. RESULTS

Here we report on the results obtained after analyzing the collected data.

5.1 Controlled Experiment 1: SEFT vs. DFT

Fourteen subjects participated in controlled experiment 1. Eight of the fourteen participants were Master students from the lecture EMBM. Seven of the eight students took a lecture on safety analysis where they learned about fault tree analysis. One subject did not have enough knowledge for performing the experiment and her data were removed from the collected data. The students were randomly assigned to two groups. Group1 (using SEFT) included three students and Group 2 (using DFT) included four students. In addition to the students, we also had six researchers participating in the experiment. They all have solid knowledge in fault tree analysis, which is part of their research work. All six researchers were also randomly assigned to the two groups.

To summarize: Group 1 had six subjects (three students and three researchers) and Group 2 had seven subjects (four students and three researchers).

Regarding Q1, we calculated the mean and the median values of each metric for each technique. Since the data collected for each metric were distributed normally as assessed by the Shapiro-Wilk test, we compared the obtained means (H_{111}, H_{121} and H_{131}) by performing a Sample T-test. The results are shown in Table 5. After applying the techniques, the participants perceived the Completeness, Understandability, and Ease of Use of DFT to be higher than the Completeness, Understandability, and Ease of Use of SEFT. Our results were not statistically significant ($p>0.05$) and $H0_{111}$, $H0_{121}$ and $H0_{131}$ were retained.

For Q2, we calculated the mean and median values for the efficiency of the metric. The results are shown in Table 5. As assessed by the Shapiro-Wilk test, the collected data were distributed normally; therefore, we performed an Independent Sample T-test to compare the means (H_{21}). The results show that the participants applying SEFT needed eight minutes more on average than the participants applying DFT. This result was statistically significant with $p<0.05$; therefore, $H0_{21}$ was rejected.

To investigate Q3, we assessed the mean and median values and the results are shown in Table 8. Although our assumption was met for all three metrics (i.e., result > 3), we performed a One-Sample Wilcoxon test to test if the result obtained was statistically different from 3 (H_{31} and H_{32}).

The difference was statistically significant ($p<0.05$) only for the metric Self-efficacy. $H0_{31}$ was thus retained while $H0_{32}$ was rejected.

Table 5. Controlled Experiment 1: Results for Completeness, Understandability, Ease of Use, and Time

	Completeness		Understandability		Ease of Use		Time	
Method	DFT	SEFT	DFT	SEFT	DFT	SEFT	DFT	SEFT
Mean	4,04	3,88	4,28	3,94	3,99	3,91	11,66	20,38
Median	4,22	3,94	4,55	4,11	3,93	3,90	11,66	21,16
Standard Deviation	0,77	0,48	0,68	0,60	0,76	0,34	6,14	6,91
Independent Sample T-Test Mean Difference (DFT-SEFT)	0,15		0,34		0,08		-8,72	
Significance	0,67		0,36		0,80		0,03	

Table 6. Controlled Experiment 2: Results for Completeness, Ease of Use, and Time

	Completeness		Ease of Use		Time	
Method	SEFT	FT+MC	SEFT	FT+MC	SEFT	FT+MC
Mean	3,66	3,43	3,54	3,63	22,36	15,46
Median	4	3,66	3,8	3,8	22	15
Standard Deviation	0,96	0,90	1,00	0,77	8,93	5,82
Paired Sample T-Test Mean Difference (SEFT-FT+MC)	0,15		-0,19		6,09	
Significance	0,41		0,39		0,06	

Table 7. Controlled Experiment 2: Results for Understandability

	Understandability	
Method	SEFT	FT+MC
Mean	3,69	3,62
Median	4	3,66
Standard Deviation	0,78	0,87
Wilcoxon Signed-Rank test statistic	0,56	
Significance	0,56	

Table 8. Results for Understandability Attitude toward using SEFT and Self-Efficacy

	Controlled Experiment 1		Controlled Experiment 2	
	Attitude toward Using SEFT	Self-Efficacy	Attitude toward Using SEFT	Self-Efficacy
Mean	3,66	3,625	3,52	3,43
Median	3,75	3,62	3,62	3,5
Standard Deviation	0,60	0,34	0,85	0,72
One-Sample Wilcoxon Signed-Rank test statistic	1,89	2,21	2,67	2,29
Significance	0,058	0,027	0,008	0,02

5.2 Controlled Experiment 2: SEFT vs. FT+MC

The second controlled experiment took place in the context of the SRES lecture. In total, 25 students participated in the experiment. During the lecture, the participants learned about fault tree analysis and Markov chains. In addition, we used a 90-minute exercise class to train the participants in SEFT. Based on informal feedback we received, we were confident that the participants were capable of applying both techniques.

To investigate Q1, we calculated the mean and the median values obtained for each metric (see Table 6 and Table 7). Except for the attribute Ease of Use, the results obtained for SEFT were always better than the results obtained for FT+MC. For the attribute Ease of Use, the results obtained for both techniques were similar. We tested the normality of the collected data with a Shapiro-Wilk test. Only for the attribute Understandability were the collected data not distributed normally. We therefore performed Related-Samples Wilcoxon Signed-Rank test for H_{122}. The difference between the Understandability of SEFT and the Understandability of FT+MC was not statistically significant ($p>0.05$), hence $H0_{122}$ was retained. For the attributes Completeness and Ease of Use, the collected data were distributed normally. We therefore performed a Paired Samples test for H_{112} and H_{132}. The difference between SEFT and FT+MC regarding Completeness and Ease of Use was not statistically significant ($p>0.05$); hence $H0_{122}$ and $H0_{132}$ were retained.

For Q2, we calculated the mean and median values for the metric Time. The results are shown in Table 6. As assessed by the Shapiro-Wilk test, the collected data were distributed normally; therefore we performed an Independent Sample T-test to compare the means (H_{21}). The results show that the participants applying SEFT needed six minutes more on average than the participants applying FT+MC. This result was statistically significant with $p<0.05$; therefore, $H0_{21}$ was rejected.

To investigate Q3, we assessed the mean and median values and the results are shown in Table 8. Although our assumption was met for all three metrics (i.e., result > 3), we performed a One-Sample Wilcoxon test to test if the results obtained were statistically different from 3 (H_{31} and H_{32}). The difference was statistically significant ($p<0.05$) for all metrics. $H0_{31}$ and $H0_{32}$ were thus rejected.

6. DISCUSSION

6.1 Interpretation and Usage of Results

SEFT were developed to take adavantage of component fault tree for modeling the safety of dynamic embedded systems. Both experiments were designed to get feedback on the newly constructed method (SEFT) and forward this feedback to the development team.

Compared to DFT, the participants believe that they obtained lower levels of Completeness, Understandability, and Ease of Use when using SEFT. More subjectively, they also needed less time for building fault tree models when using DFT. Based on the informal feedback provided by the participants, we were able to investigate the reason behind these results. First, the notion of states and events introduced with SEFT adds complexity to the models obtained. This negatively impacts the understandability of SEFT models. Second, the number of gates make it difficult for the participants to know which gates are appropriate for a given situation, and the participants therefore needed too much time to choose the gate and were unable to build complete models. Third, the representation used for SEFT was not familiar to the participants. Nevertheless, the participants at least agreed on all of our statements about SEFT regarding Effort Expectancy, Attitude toward Using the Technology, and Self-Efficacy.

Compared to FT+MC, the subjects believed that Ease of Use was lower when using SEFT. They also needed less time for building models using FT+MC. Based on the feedback collected , the reason for this result was the amount of information they had to process in order to build SEFT models. Nevertheless, they believe that the SEFT models were more understandable and complete than the FT+MC models. They argued that the process for constructing SEFT models was clearer and the mapping to the system models makes it more understandable. Moreover, they agreed with all of our statements regarding SEFT with regard to Effort Expectancy, Attitude toward Using SEFT, and Self-Efficacy.

Based on our analysis, we recommended to the team developing SEFT to focus on simplifying the modeling process. One solution could be to provide a tool that would help users to semi-automatically generate SEFT models from the system model. They would then not need to look at all possible gates in order to choose the appropriate gates. The search space would be reduced.

6.2 Threats to Validity

Concerning internal validity, the subjects were trained in the use of the techniques before experimentation started. Moreover, they were graduate students and were sufficiently motivated because the experiments were also used as practical exercises of the knowledge acquired during their lecture. A standardized process was used for both experiments, which assured that experimenter expectancies did not influence the participants.

As for external validity, the systems used during the experiment were part of the AAL lab and the concept car developed at Fraunhofer IESE (which are close enough to real systems).

Regarding conclusion validity, the questionnaires were checked by an expert on empirical studies and it was ensured that the subjects of both groups had similar backgrounds and knowledge regarding safety analysis. Although some of our results were not statistically significant, they indicate trends that can be validated in future experiments.

Regarding construct validity, we followed the Goal Question Metric (GQM) methodology [4] to define the goal, the questions, and the metrics. The goal was refined into clearly defined metrics to avoid misunderstandings, and the tasks were designed to avoid threats due to mono-operation bias. Each question was reviewed by an expert in empirical software engineering. The subjects were not aware of the hypotheses to be tested or the measures to be taken.

7. CONCLUSION AND FUTURE WORK

In this paper, we reported on two controlled experiments that were carried out to compare the applicability and efficiency of SEFT versus DFT and FT+MC.

We first provided an overview on the few existing studies in safety engineering. Then we explained how Fault Trees, DFT, SET and FT+MC are related to each other.

We tested the applicability and efficiency of these techniques in two controlled experiments. In the first experiment we analyzed SEFT and DFT with fourteen subjects while the second one was applied by twenty-five subjects. Subjects were graduated students with a certain amount of experience in safety analysis, trained on the safety techniques during lectures and researchers in safety domain.

The results of our studies show that the subjects found DFT more applicable than SEFT and SEFT more applicable than FT+MC and needed less time to perform DFT or FT+MC than to perform SEFT.

We also investigated the reasons behind these results by discussing with subjects and safety engineers. We found that the notion of states and events, which are not used in DFT and FT+MC, adds complexity to SEFT. Moreover, the large number and high complexity of the gates used during SEFT made it difficult for the participants to use them appropriately. Based on those results, we concluded that the following measures could help to increase the usage of SEFT in practice:

- Simplification of the notations
- Comprehensive tool support
- A knowledge database providing guidance on how to model frequently used architectural patterns
- Sufficient training regarding the technique

Although we designed both studies to minimize threats to validity as much as possible, it was difficult for us to obtain good statistical significance with the collected data. We are planning to replicate the experiment in an industrial environment to improve its statistical significance.

8. ACKNOWLEDGMENTS

This work was funded by the German Federal Ministry of Education and Research (BMBF) in the context of the projects

ViERforES II (01|M10002D) and ARAMIS (01|511035W). We are thankful to Liliana Guzman from the University of Kaiserslautern, who provided a great help during the design of the study. Special thanks go to our research supervisor Prof. Dr. Dr. h.c. Dieter Rombach for his helpful comments and improvement suggestions.

9. REFERENCES

[1] Bechta Dugan, J., Bavuso, S. J., and Boyd, M. A. 1992. Dynamic fault-tree models for fault-tolerant computer systems. Reliability, IEEE Transactions on 41, 3, 363–377.

[2] Bernhard Kaiser, Peter Liggesmeyer, and Oliver Mĭckel. 2003. A new component concept for fault trees. In Proceedings of the 8th Australian Workshop on Safety Critical Systems and Software (SCSS03), 37–46.

[3] Broy, M., Kruger, I. H., Pretschner, A., and Salzmann, C. 2007. Engineering Automotive Software. Proceedings of the IEEE 95, 2, 356–373.

[4] Caldiera, Victor R Basili1 Gianluigi and Rombach, H. D. 1994. The goal question metric approach. Encyclopedia of software engineering 2, 1994, 528–532.

[5] Cullyer, W. J. and Storey, N. 1994. Tools and techniques for the testing of safety-critical software. Computing Control Engineering Journal 5, 5, 239–244.

[6] Dong Seong Kim, Machida, F., and Trivedi, K. S. 2009. Availability Modeling and Analysis of a Virtualized System. In Dependable Computing, 2009. PRDC '09. 15th IEEE Pacific Rim International Symposium on, 365–371.

[7] Hansen, M. D. 1989. Survey of available software-safety analysis techniques. In Reliability and Maintainability Symposium, 1989. Proceedings, Annual, 46–49.

[8] International Electrotechnical Commission (IEC). 2010. IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems.

[9] International Organization for Standardization (ISO). 2011. ISO 26262: Road vehicles – Functional safety.

[10] Jonas Mitschang. Evaluation of a Model-Based Development Process for Automotive Embedded Systems.

[11] Jung, J., Hoefig, K., Domis, D., Jedlitschka, A., and Hiller, M. 2013. Experimental Comparison of Two Safety Analysis Methods and Its Replication. In Empirical Software Engineering and Measurement, 2013 ACM / IEEE International Symposium on, 223–232.

[12] Kaiser, B. and Gramlich, C. 2004. State-Event-Fault-Trees – A Safety Analysis Model for Software Controlled Systems. In Computer Safety, Reliability, and Security, M. Heisel, P. Liggesmeyer and S. Wittmann, Eds. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 195–209.

[13] Michael Jung. Application, Enhancement and Evaluation of a Modeling Methodology for Adaptive Embedded Systems.

[14] Nehmer, J., Becker, M., Karshmer, A., and Lamm, R. 2006. Living assistance systems: an ambient intelligence approach. In Proceedings of the 28th international conference on Software engineering. ICSE '06. ACM, New York, NY, USA, 43–50.

[15] Neuman, C. P. and Bonhomme, N. M. 1975. Evaluation of Maintenance Policies using Markov Chains and Fault Tree Analysis. Reliability, IEEE Transactions on R-24, 1, 37–44.

[16] Opdahl, A. L. and Sindre, G. 2009. Experimental Comparison of Attack Trees and Misuse Cases for Security Threat Identification. Inf. Softw. Technol. 51, 5, 916–932.

[17] Patterson-Hine, F. A. and Bechta Dugan, J. 1992. Modular techniques for dynamic fault tree-analysis. In Reliability and Maintainability Symposium, 1992. Proceedings, Annual, 363–369.

[18] Qichuan Geng, Haibin Duan, and Shuangtian Li. 2011. Dynamic fault tree analysis approach to Safety Analysis of Civil Aircraft. In Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on, 1443–1448.

[19] SAE International. 1996. ARP4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment.

[20] Stålhane, T. and Sindre, G. 2007. A Comparison of Two Approaches to Safety Analysis Based on Use Cases. In Conceptual Modeling - ER 2007, C. Parent, K.-D. Schewe, V. Storey and B. Thalheim, Eds. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 423–437.

[21] Stålhane, T. and Sindre, G. 2008. Safety Hazard Identification by Misuse Cases: Experimental Comparison of Text and Diagrams. In Model Driven Engineering Languages and Systems, K. Czarnecki, I. Ober, J.-M. Bruel, A. Uhl and M. Völter, Eds. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 721–735.

[22] Stålhane, T., Sindre, G., and Bousquet, L. 2010. Comparing Safety Analysis Based on Sequence Diagrams and Textual Use Cases. In Advanced Information Systems Engineering, B. Pernici, Ed. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 165–179.

[23] Venkatesh, V., Morris, M. G., Davis, G. B., and Davis, F. D. 2003. User acceptance of information technology: Toward a unified view. MIS quarterly, 425–478.

[24] Vesely, W., Dugan, J., Fragola, J., Minarick, and Railsback, J. 2002. Fault Tree Handbook with Aerospace Applications. National Aeronautics and Space Administration, Washington, DC.

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

[26] Wallace, D. R., Kuhn, D. R., and Ippolito, L. M. 1992. An analysis of selected software safety standards. Aerospace and Electronic Systems Magazine, IEEE 7, 8, 3–14.

[27] Yao Yiping, Yang Xiaojun, and Li Peiqiong. 1996. Dynamic fault tree analysis for digital fly-by-wire flight control system. In Digital Avionics Systems Conference, 1996., 15th AIAA/IEEE, 479–4