Interactive information zoom on Component Fault Trees

Santiago Velasco[1], Jan Reich[2] and Maxime Tchangou[3]

Abstract: The visualization approach for Component Fault Trees (CFTs) realized in this work was implemented as an extension of the safeTbox modeling tool (safeTbox.iese.fraunhofer.de). Its goal is to enhance the understandability of compositional and hierarchical models while facilitating reviewing purposes. Safety Analysts makes use of CFTs to perform fault analyses at system level. However, such analyses are hindered by the traditional approach of hiding the realization information of components behind the specification views. The approach presented here overcomes this problem thanks to an information-based zoom. Through it, it is possible to gradually present at the specification level information extracted out of the internal realization views.

Keywords: Component Fault Trees, Information Zoom, Visual Zoom, Black-box, specification view, realization view, hierarchical abstraction.

Fault Tree Analysis (FTA) is a deductive technique for the identification of faults in a system. It is very well-known in the safety-critical domain of embedded systems, since its use is demanded or recommended by several standards (e.g. IEC 61508 or ISO 26262) depending on the criticality of the system to be designed. As its name indicates, this technique leads to the construction of a tree of undesired events (see Fig1 left): on its top, a top event (event under analysis) will be found, on the leaf, the so called "Basic Events" (being the root causes), and in between intermediate events (depict the logical combinations of basic or other intermediate events).

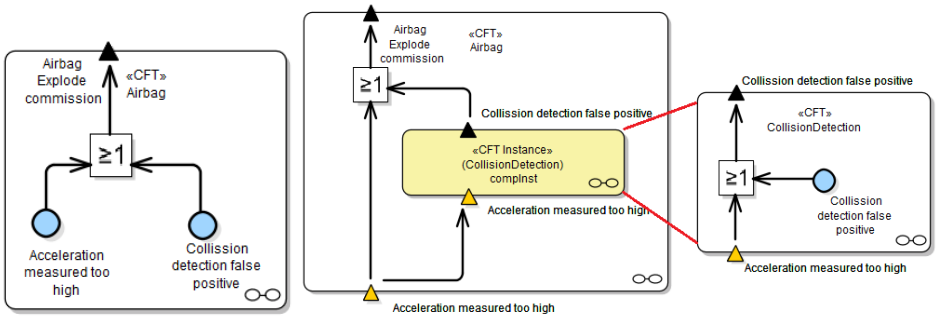

Fig. 1: Fault Tree (Left), Component Fault Tree (Right)

[1] Fraunhofer IESE, Embedded Systems Quality, 67663 Kaiserslautern, santiago.velasco@iese.fruanhofer.de
[2] Fraunhofer IESE, Embedded Systems Quality, 67663 Kaiserslautern, jan.reich@iese.fruanhofer.de
[3] TU-Kaiserslautern, 67663 Kaiserslautern, maximetchangou2011@yahoo.fr

Kaiser and Liggesmeyer [BPO00] extended the FTA approach with modularization concepts to encapsulate parts of a typically monolithic representation into blocks (see Fig1 right). The resulting Component Fault Tree (CFT) concept introduces new modeling elements to support a compositional approach, namely CFT instances and input events. The former allow to reuse existing modularized fault trees within other fault trees, and the later (Yellow triangles in the figure) represent failure modes propagating into the modularized fault trees.

Domis [D05] enhanced the CFT approach to get a formal traceability to system and architectural design models by introducing traces between CFT artifacts and architecture artifacts. For this reason this approach has the advantage that fault models are easier to maintain with respect to changes in system design models. In Adler [ASH17], the integrated approach has been further adapted to support cause analyses in models that describe control schemes. However, CFTs modeled following these approaches have the disadvantage that they have to adopt the hierarchical abstraction of the architecture models they have been integrated to.

In the embedded systems domain, hierarchical abstraction is typically used as means to handle complexity. This is frequently used in the modeling of large systems when building component networks in which it is easy to distinguish between the black-box and white-box views. Meaning the specification and the realization by means of composition of other components. A black-box representation can be very useful to get the big picture in terms of composition, but it can be difficult to work with, when model details are required for reasoning. This is the case in the area of fault analysis for instance, where it is quite important to have a better understanding of the behavior of the system parts. This will be required to identify faults that might not only occur as a pure fault of a component but due to their intricate interaction. Traditionally, compositional models have been supported by most modeling tools by enabling the navigation from the black-box (i.e. specification) to the white-box (i.e. realization) representation of a component and by supporting different visual zoom levels. The navigation from the white to the black representation is generally not well-supported, since the navigation is not unique (e.g. in the case of reuse). For this reason an analyst trying to review the system model has a hard time, since he has to switch continuously between the specification and realization views. Only in this way he will be able to get a big picture of the system, while being able to understand the details of each component at the same time. Such context switching is quite demanding, since it assumes that the analyst is able to retain the realization view of one or more components in mind while analyzing others.

In order to facilitate the review process of complex models, particularly in the context of Component Fault Tree analysis, we have implemented a prototype of a visualization feature as part of the safeTbox™ modeling tool (safetbox.iese.fraunhofer.de). It allows performing an information zoom, in which details of the realization view of a component are brought up to the level of the specification representation. The goal of this feature is to eliminate the need of switching context while obtaining more details within a

hierarchical and compositional model. With the start of the visualization feature, the model will be displayed in the visualization feature as displayed in the modeling tool (see Fig 2 - Entry level). From this point on an analyst has the possibility to control the amount of information being displayed by means of a visual and information zoom. The deeper the selected information level the more details of the realization views of the instances will be displayed. This will occur recursively until no refinement is possible, i.e. the user sees the full realization of the deep most component in the hierarchy.

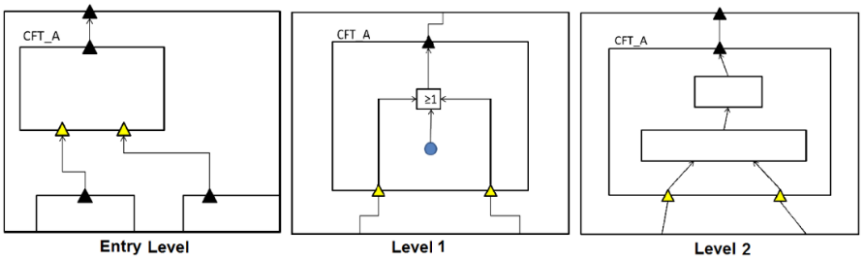

Fig. 2: Solution concept

One of the major differences between the regular visualization approach and ours is that in the black/white box alternative the modeler has a big jump with respect to the amount of information being displayed whenever he switches from the specification (only interfaces being displayed) to the realization (i.e. all information). In our approach, we have introduced several intermediate levels of details which are computed out of the model information and which smoothly increments the amount of information being displayed (see level 1 and 2 in figure 2). From the entry point, i.e. the typical black-box view, the user can zoom into Level 1, which does not present the realization of CFT_A as it would be expected, but a reduced fault tree which has been computed out of the results of the Minimal Cut Sets (MCS) analysis. The MCS Analysis is used to compute the minimal set of events required to trigger the occurrence of a top event. This view abstracts from the underlying fault model with a small set of modeling elements, but is still showing the most important information, namely the faults of all internal sub components of CFT_A. If the user needs to know where these faults originate he can zoom in further into zoom level 2, where he will start seeing an abstract version of the realization of the component. In this level, the focus rests on depicting the fault relationship among sub components. The information with respect to the instance interfaces is removed by abstracting the fault relationships between each pair of sub components. In summary, the feature will display the same information as in the entry level, but for the sub components of CFT_A.

As shown in the figures 3, while zooming in purely through the information zoom, the feature will add more and more information to the model. For this reason, a visual pan and zoom functionality is still required to allow the user focus temporary in certain aspects.

In summary, our visualization approach provides an alternative way in displaying information for component-oriented fault trees. It shall enhance the understandability of hierarchical models while facilitating reviewing purposes by avoiding context switching. In the near feature this feature will be extended with other visualization functionalities to support other analysis activities, e.g. Common Cause Failures (CCF) and Boolean cycle visualization in CFTs.

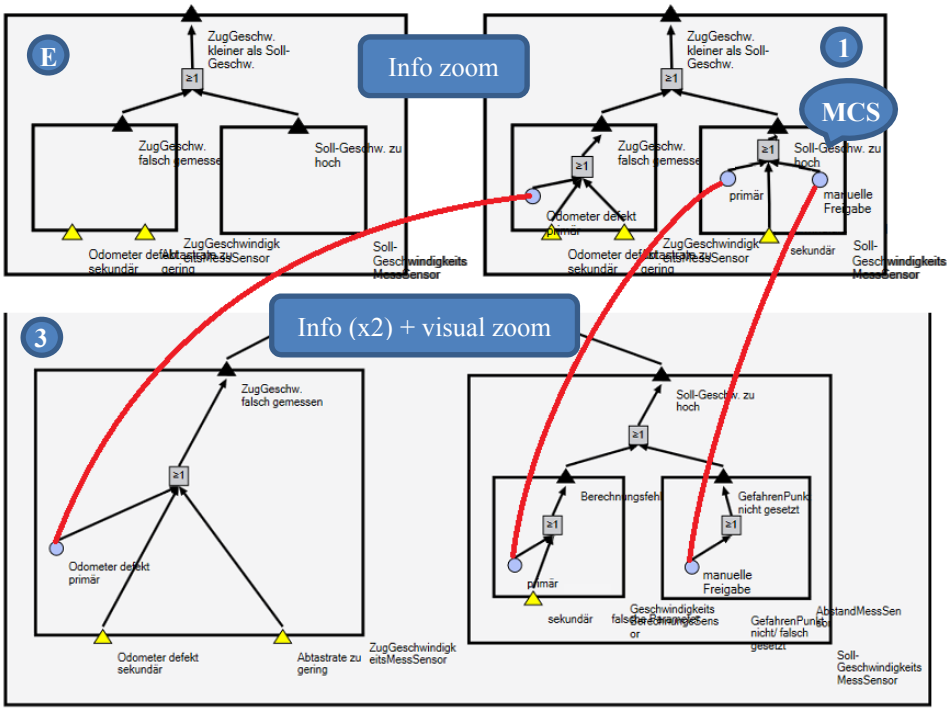

Fig. 3: Visualization feature levels example

[ASH17] Adler,R., Schneider D., Höfig K., ”Evolution of fault trees from hardware safety analysis to integrated analysis of software-intensive control systems”, In proceedings of ESREL 2017 (Portoroz, Slovenia, 18-22 June, 2017)

[D05] Domis, D.: "Integrating fault Tree Analysis and Component-Oriented Model-Based Design of Embedded Systems", PHD Thesis, Technische Universitaet Kaiserslautern, Fachbereich Informatik, 2005

[BPO03] Bernhard Kaiser , Peter Liggesmeyer , Oliver Mäckel, A new component concept for fault trees, Proceedings of the 8th Australian workshop on Safety critical systems and software, pp. 37-46, October 01, Canberra, Australia, 2003