Automating the Failure Modes and Effects Analysis of Safety Critical Systems

Yiannis Papadopoulos & David Parker
Department of Computer Science,
University of Hull U.K.
{y.i.papadopoulos, d.j.parker}@hull.ac.uk

Christian Grante
Volvo Cars Corporation
Sweden
cgrante@volvocars.com

Failure Modes and Effects Analysis (FMEA) is a classical system safety analysis technique which is currently widely used in the automotive, aerospace and other safety critical industries. In the process of an FMEA, analysts compile lists of component failure modes and try to infer the effects of those failure modes on the system. System models, typically simple engineering diagrams, assist analysts in understanding how the local effects of component failures propagate through complex architectures and ultimately cause hazardous effects at system level.

Although there is software available that assists engineers in performing clerical tasks, such as forming tables and filling in data, the intelligent part of an FMEA process remains a manual and laborious process. Thus, one of the main criticisms of FMEA is that the time taken to perform the analysis can often exceed the period of the design and development phases and therefore the analysis *de facto* becomes a mere deliverable to the customer and not a useful tool capable of improving the design. Difficulties naturally become more acute as systems grow in scale and complexity.

To address those difficulties, a body of work is looking into the automation and simplification of FMEA [1-3]. To mechanically infer the effects of component failures in a system, several approaches have been proposed which use domain specific qualitative or quantitative fault simulation. These approaches are restricted to particular application domains such as the design of electrical or electronic circuits. Limitations in scope but also difficulties with the efficiency and scalability of algorithms seem to have so far limited the industrial take-up of this automated FMEA technology which is still under development.

In this paper we propose a new approach to the automatic synthesis of FMEAs which builds upon recent work towards automating fault tree analysis [4]. In this approach, FMEAs are built from engineering diagrams that have been augmented with information about component failures. The proposed approach is generic, i.e. not restricted to an application domain, and potentially applicable to a range of widely used engineering models. The models that provide the basis for the analysis identify the topology of the system, i.e. the system components and the material energy and data transactions among those components. Models can also be hierarchically structured and record in different layers the decomposition of subsystems into more basic components. We should note that this type of structural models include piping and instrumentation diagrams, data flow diagrams and other models commonly used in many areas of engineering design.

The first step in the analysis of such models is the establishment of the local failure behaviour of components in the model as a set of failure expressions which show how output failures of each component can be caused by internal malfunctions and deviations of the component inputs. Once this local analysis has been completed for all components, the structure of the model is then used to automatically determine how local failures propagate through connections in the model and cause functional failures at the outputs of the system. This global view of failure is initially captured in a set of fault trees which are automatically constructed by traversing the model of the system backward moving from the final elements of the design, i.e. the actuators, towards system inputs and by evaluating the failure expressions of the components encountered during this traversal.

The fault trees synthesized using this approach show how functional failures or malfunctions at the outputs of the system are caused by logical combinations of component failures. These fault trees may share branches and basic events in which case they record common causes of failure, i.e. component failures that contribute to more than one system failures.

Thus, in general, the result of the fault tree synthesis process is *a network of interconnected fault trees* which record logical relationships between component and system failures as this is illustrated in figure 1.

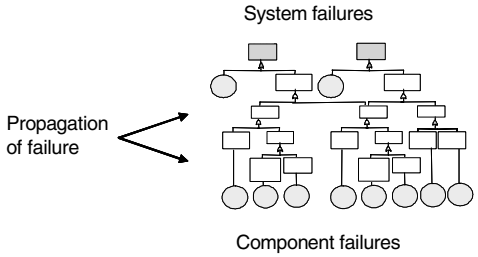

Figure 1. A network of automatically created fault trees

Proceedings of the Eighth IEEE International Symposium on High Assurance Systems Engineering (HASE'04)
1530-2059/04 $20.00 © 2004 IEEE
IEEE COMPUTER SOCIETY

The top events of these fault trees represent system failures. Leaf nodes represent component failure modes while the body of intermediate events (and intervening logic) records the propagation of failure in the system and the progressive transformation of component malfunctions to system failures.

In the final step of the process, this complex body of fault propagation logic is removed from the analysis by an automated algorithm which translates the network of interconnected fault trees into a simple table of *direct relationships between component and system failures*. In a similar way to a classical FMEA, this table determines for each component in the system and for each failure mode of that component, the effect of that failure mode on the system, i.e. *whether*, and *how*, the failure mode contributes to one or more system failures and malfunctions (i.e. the top events of fault trees).

Note that in a classical manual FMEA only the effects of single failures are typically assessed. Thus, one advantage of generating an FMEA from fault trees is that fault trees record the effects of combinations of component failures and this useful information can also be transferred into the FMEA. To accommodate this additional information, the resultant FMEA tables are split into two, one containing the direct effects on the system, i.e. those effects caused by single component failures, and the other containing further effects, i.e. those effects caused by two or more component failure modes. This allows separate, easy access to the most critical information, the single points of failure. Perhaps more importantly, the FMEA shows all functional effects that a particular component failure mode causes. This is useful as a failure mode that contributes to multiple system failures is potentially more significant than those that only cause a single top event.

The FMEA can, in practice, help analysts not only to locate problems in the design, but also to determine the level of fault tolerance in the system, i.e. determine whether the system can tolerate any single or any combination of two, three or more component failures.

To enable the practical and useful application of the above concept in engineering design, we have developed a tool that generates fault trees and FMEAs from models developed in Matlab Simulink, a popular modeling and simulation tool. The proposed method and tool are currently being evaluated by *Volvo cars* in a case study of medium complexity performed on a Matlab-Simulink model of an advanced steer-by-wire prototype system for cars.

This work is still at early stages and we have not had a chance yet to perform a rigorous performance evaluation of the proposed algorithms. First applications indicate though that this approach can lead to fast and efficient ways of generating useful safety analyses from system design representations. The process is largely automated and can make use of design information from the early stages of the design thus minimising the effort required to examine system safety and, perhaps more importantly, to study the effect of design modifications on safety.

An indication of the present performance of the system is that it is taking a little more than a minute in an average personal computer to generate an FMEA from a model of a steer-by-wire system for cars that contains more than a hundred components and results in over seven thousand cut sets. This result refers to an FMEA that records the effects of up to four simultaneously occurring component failures modes. When this limit is set at two, the time dramatically decreases, obtaining timings in the order of a few seconds. To the best of our knowledge, these speeds compare favorably with other results reported in the literature of automated FMEA, where systems have been reported to take hours even when considering only the effects of single component failures. Direct comparisons, however, are not possible because the proposed approach leads only to semi-automatic synthesis of FMEAs, while most other work aims to fully automate the process.

To further improve the speed of the synthesis, we currently consider using a recently proposed minimal cut-set calculation algorithm [5] for the conversion of the network of fault trees into an FMEA. This algorithm preprocesses fault trees, converting them into Binary Decision Diagrams, using ordering rules to determine the position of failure modes in the hierarchy of the tree. We hope that the improvements in efficiency that could be achieved by using this algorithm will further improve the scalability of the proposed techniques and ultimately enable their application in problems of industrial scale.

References

[1] Renovell M., Cambon G. and Auvergne D., "FSPICE: a tool for fault modelling in MOS circuits", *VLSI Journal,* 1985, 3:245-255.

[2] Lehtela M., "Computer-Aided FMEA of Electronic Circuits", *Microelectronics and Reliability*, 1990, 30(4):761-773.

[3] Price C. J., Taylor N., "Automated multiple failure FMEA", *Reliability Engineering and System Safety*, 2002, 76:1-10.

[4] Papadopoulos Y., McDermid J. A., Sasse R. Heiner G., "Analysis and synthesis of the behaviour of complex programmable systems in conditions of failure", *Reliability Engineering and System Safety*, 2001, 71:229-247.

[5] Sinammon R. M., Andrews J. D., "New approaches to evaluating fault trees", *Reliability Engineering and System Safety*, 1997, 58:89-96.

1530-2059/04 $20.00 © 2004 **IEEE**

IEEE COMPUTER SOCIETY