A Compositional Reliability and Availability Evaluation Tool*

Hichem Boudali[1], Pepijn Crouzen[2,**], and Mariëlle Stoelinga[1]

[1] Department of Computer Science, University of Twente,
P.O.Box 217, 7500AE Enschede, The Netherlands.
[2] Saarland University, Department of Computer Science,
D-66123 Saarbrücken, Germany.
{hboudali@cs,p.crouzen@alumnus,marielle@cs}.utwente.nl

Reliability and availability measures, such as system failure probability during a given mission time and system mean-time-between-failures, are often important measures to assess in embedded systems design. There exist several techniques and formalisms for reliability/availability assessment. One such formalism is dynamic fault trees (DFT). DFTs are a graphical, high-level and versatile formalism to analyze the reliability of computer-based systems. A DFT describes the failure of a system in terms of the failure of its components and is comprised of basic events (modeling the failure of physical components) and gates (modeling how component failures induce system failures). DFTs extend standard (or static) fault trees (FT) by allowing the modeling of complex system components' behaviors and interactions. Typically, a DFT is analyzed by first converting it into a continuous-time Markov chain (CTMC) and by then computing the reliability measures from this Markov chain. For over a decade now, DFTs have been experiencing a growing success among reliability engineers.

Unfortunately, a number of issues remain when using DFTs, most notably: (1) the DFT semantics is rather imprecise and the lack of formality has, in some cases, led to undefined behavior and misinterpretation of the DFT model. (2) DFTs lack modular analysis. That is, even if stochastically-independent sub-modules exist in a DFT module, these sub-modules can not always be solved separately. Consequently, DFT become vulnerable to the well-known state-space explosion problem; that is the size of the underlying Markov Chain grows exponentially with the number of basic events in the DFT. (3) DFTs also lack modular model-building, i.e. there are some rather severe restrictions on the type of allowed inputs to certain gates which greatly diminish the modeling flexibility and power of DFTs.

We have developed a formal semantics of DFTs in terms of input/output interactive Markov chains (I/O-IMCs), which extend continuous-time Markov chains with discrete input, output and internal actions [3]. This semantics addresses issue (1) mentioned above and provides a rigorous basis for the interpretation and analysis of DFTs. Our semantics is fully compositional, that is, the semantics of a DFT is expressed in terms of the semantics of its elements (i.e. basic events and gates). This enables an efficient analysis of DFTs through compositional aggregation, which helps to alleviate the state-space explosion problem by incrementally building the DFT state space. Our techniques is completely modular, which allows us to overcome issue (2). We have also tackled issue (3) and lifted some previously enforced restrictions on DFTs.

We have implemented our methodology by developing a prototype tool based on the CADP tool set [5]. We have compared our approach to the existing

* This research has been partially funded by the Netherlands Organisation for Scientific Research (NWO) under FOCUS/BRICKS grant number 642.000.505 (MOQS); the EU under grant number IST-004527 (ARTIST2); and by the DFG/NWO bilateral cooperation programme under project number DN 62-600 (VOSS2).

** The majority of this work was done while the author was at the University of Twente.

analysis tool Galileo [1] through several case studies, and showed the merit of our approach and its effectiveness in reducing the state space to be analyzed [3].

The prototype tool takes as input a DFT in Galileo's textual format and a composition script, which describes the order in which the I/O-IMC models must be composed in a simple textual format. The tool proceeds in three steps:

1. **Translation:** The DFT is translated into a group of I/O-IMC models. In particular, each DFT element is translated into a corresponding elementary (with few states and few transitions) I/O-IMC model.
2. **Compositional aggregation:** Using the composition script the I/O-IMC models are iteratively composed, abstracted and aggregated until one I/O-IMC model remains.
3. **Analysis:** In most cases the resulting I/O-IMC model can be easily transformed into a continuous-time Markov chain. Transient analysis (using the CADP tool set) can then be applied to find the unreliability of the DFT.

The compositional semantics also allows the DFT formalism to be easily extended or modified. In [2] we show how several of these extensions (for instance, repairable components [7]) could be realized in our framework. Such extensions only impact the translation to the corresponding I/O-IMC models of the modified or added DFT elements. Thus only the translation step (i.e. step 1) of the tool is affected.

At the present time, the prototype tool is not fully automatic: The user must supply the order in which the I/O-IMC models are composed (as a composition script). The focus of the future work will be to fully automate the tool. To do this an algorithm to find good (i.e. computationally efficient) composition orders is needed. Other possible topics for future research include the investigation of improvements to our compositional aggregation process such as using context constraints [4] or interface specifications [6]. We are also currently looking at other reliability/availability formalisms and architectural design languages (such as the architecture analysis and design language (AADL) standard and its error model annex) and trying to map their constructs into I/O-IMC models. Lastly, we are planning to improve the overall usability of the tool to make it available to a wider audience.

References

1. Galileo DFT analysis tool. http://www.cs.virginia.edu/~ftree.
2. H. Boudali, P. Crouzen, and M.I.A. Stoelinga. Dynamic fault tree analysis using input/output interactive markov chains. *Accepted to Dependable Systems and Networks 2007 conference.*
3. H. Boudali, P. Crouzen, and M.I.A. Stoelinga. Compositional analysis of dynamic fault trees. Technical report, University of Twente, to appear.
4. S.C. Cheung and J. Kramer. Context constraints for compositional reachability analysis. *ACM Transactions on Software Engineering and Methodology*, 5(4), October 1996.
5. Construction and Analysis of Distributed Processes (CADP) software tool. http://www.inrialpes.fr/vasy/cadp/.
6. S. Graf, B. Steffen, and G. Lüttgen. Compositional minimisation of finite state systems using interface specifications. *Formal Aspects of Computing*, 8(5), September 1996.
7. D.C. Raiteri, M. Iacono, G. Franceschinis, and V. Vittorini. Repairable fault tree for the automatic evaluation of repair policies. In *Internation Conference on Dependable Systems and Networks*, 2004.